

Stochastic Domain Decomposition Based on Variable-Separation Method *

Liang Chen[†] Yaru Chen[‡] Qiuqi Li[§] Zhiwen Zhang[¶]

October 11, 2023

Abstract

This work proposes a stochastic domain decomposition method for solving steady-state partial differential equations (PDEs) with random inputs. Specifically, based on the efficiency of the variable-separation (VS) method in simulating stochastic partial differential equations (SPDEs), we extend it to stochastic algebraic systems and employ it in the context of stochastic domain decomposition. The resulting method, termed stochastic domain decomposition based on variable-separation (SDD-VS) alleviates the challenge commonly known as the “curse of dimensionality” notably by leveraging explicit representations of stochastic functions derived from physical systems. The primary objective of the proposed SDD-VS method is to obtain a separate representation of the solution for the stochastic interface problem. To enhance computational efficiency, we introduce a two-phase approach consisting of offline and online computation. In the offline phase, we establish an affine representation of stochastic algebraic systems by systematically applying the VS method. During the online phase, we estimate the interface unknowns of SPDEs using a quasi-optimal separated representation, facilitating the construction of efficient surrogate models for subproblems. We substantiate the effectiveness of our proposed approach through numerical experiments involving three specific instances, demonstrating its capability to provide accurate solutions.

keywords: stochastic partial differential equation; stochastic domain decomposition; stochastic interface problem; variable-separation method; uncertainty quantification

MSC: 65N99, 65N55, 60H35, 35R60

1 Introduction

In the realm of computational science and engineering, many problems such as modeling water flow and solute transport in heterogeneous soil and aquifer formations involve uncertainties due to inadequate knowledge about physical properties and measurement noise. To provide accurate and reliable predictions, these uncertainties are often represented by random variables, and their impact on the system is explored through the lens of stochastic partial differential equations (SPDEs). A few numerical methods have been proposed for simulating SPDEs, including Galerkin projections [1, 2, 3], stochastic interpolation [4, 5, 6, 7] and the methods based on deep learning [8, 9]. This work focuses on the domain decomposition method (DDM), a promising approach for addressing stochastic problems. We aim to apply DDM techniques for solving SPDEs efficiently and enabling robust predictions in the presence of uncertainties.

Domain decomposition methods have a long history of successful application in solving deterministic problems [10, 11, 12, 13, 14]. These methods are to partition a large computational domain into several subdomains so that the subproblems on each subdomain can be solved independently [15, 16]. Under this setting, DDM has primarily focused on developing parallel solvers for deterministic partial differential equations (PDEs) to improve computational efficiency [17, 18, 19]. These methods are generally categorized into overlapping (Schwarz iteration) methods and non-overlapping (Schur complement) methods

*The authors are grateful to the anonymous referees and the handling editor for their helpful suggestions. The research of this work was supported by the National Key R&D Program of China (No. 2021YFA1001300), the National Natural Science Foundation of China (Nos. 12271150, 12101216, 12171406), the Natural Science Foundation of Hunan Province (Nos. 2023JJ10001, 2022RC1190, 2022JJ40030), and the Hong Kong RGC grant projects (Nos. 17300318, 17307921).

[†]School of Mathematical, Hunan University, Changsha 410082, China (ch1@hnu.edu.cn).

[‡]School of Mathematical, Hunan University, Changsha 410082, China (yrchen@hnu.edu.cn).

[§]School of Mathematical, Hunan University, Changsha 410082, China (qli28@hnu.edu.cn).

[¶]Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong Special Administrative Region of China (zhangzw@hku.hk).

[20]. Our proposed stochastic domain decomposition (SDD) method is based on the latter, specifically the Schur complement methods [21, 22, 23, 24]. In this approach, the computational domain is divided into non-overlapping subdomains, and the degrees of freedom on each subdomain are separated into interior and interface parts. The global system can be reduced to a Schur complement system by applying block Gaussian elimination. The unknown variables on the interface are then determined by solving the Schur complement system.

The application of domain decomposition methods in stochastic simulations has emerged as a promising research area. Notably, Sarkar et al. [25] proposed a domain decomposition approach with Schur complement in the physical space and functional decomposition in the probability space for solving stochastic partial differential equations. Subsequently, Subber et al. [26, 27] extended this method by incorporating a typical preconditioner to determine the coefficients of the interface solution. In a separate study by Chen et al. [28], the authors employed a domain decomposition approach to solve stochastic elliptic PDEs. They approximated the local solution in each subdomain within a low-dimensional parametric space. In Hadigol et al. [29], a stochastic model reduction approach based on low-rank separated representations for the stochastic space was studied, where the coefficients of the interface solution are done by finite element tearing and interconnecting method. Liao and Willcox [30] proposed an offline-online approach that combines DDM with importance sampling. This enables the use of different strategies in local systems, and the interface solution is generated by weighting precomputed PDE solutions. To handle elliptic PDEs in high-contrast random media, Hou et al. [31] incorporated multiscale finite element methods into the DDM framework. Incorporating a Schwarz-type iterative algorithm, Zhang et al. [32] introduced a moment-minimizing interface condition to match the stochastic interface solution. Lastly, Mu and Zhang [33] recently combined model reduction methods with sparse polynomial approximation. They developed a stochastic domain decomposition method comprising an offline procedure and an online procedure for linear steady-state convection-diffusion equations with random coefficients.

In this paper, we focus on integrating Schur complement methods with the variable-separation (VS) method for stochastic simulations. The VS method is one of the most effective model order reduction methods and has been successfully used to solve SPDEs in a low-dimensional manifold [34, 35, 36]. Building upon the extension of the VS method to stochastic algebraic systems, we propose a novel model reduction approach for solving the global stochastic interface problem of SPDEs, termed the stochastic domain decomposition based on the variable-separation method (SDD-VS). To enhance simulation efficiency, an offline-online computational decomposition is employed for the stochastic interface problem. The offline phase consists of three stages. Firstly, the original domain is partitioned into non-overlapping subdomains, and the Schur complement system is established, including local Schur complement matrices and corresponding right-hand side vectors with random inputs. Secondly, a reduced stochastic algebraic system is generated using low-rank representations of the global stochastic Schur complement matrices and right-hand side vectors. Although the reduced model requires less computational effort compared to the original stochastic Schur complement system, it still involves the discrete degrees of the full model and may not be considered small-scale. To address this, the third stage introduces a functional decomposition expression for the stochastic solution on the interface, achieved through the extended VS method. In the online phase, each realization of the stochastic interface problem is recovered using the outputs from the offline phase. Finally, efficient surrogate models for the stochastic subproblems can be obtained using the VS method or other model reduction techniques [37, 38, 39].

The SDD-VS method proposed in this paper shares the advantages of both the variable separation and the domain decomposition method. Specifically, the SDD-VS method can alleviate the ‘‘curse of dimensionality’’ in an effective way and this is achieved by the explicit representation of the stochastic functions deduced from the physical system instead of using a suitable set of basis functions (e.g., polynomial chaos basis and radial basis functions) of the random variables. Meanwhile, the proposed method is much easier than other stochastic domain decomposition methods when implemented, thanks to the applications of the VS method. Furthermore, the whole computation of the SDD-VS method consists of an offline stage and an online stage. The online stage has high computational efficiency and its computational cost is completely independent of spatial discretization. Moreover, the SDD-VS method can reduce the dimension of the random variable for local problems in subdomains, it leads to a more efficient surrogate model for each subproblem than that obtained on the entire domain. Finally and most importantly, the proposed method maintains the same merits as DDM for deterministic PDEs, such as the ability to solve subproblems independently and in parallel. In summary, the SDD-VS method offers an efficient and reliable approximation for SPDEs, which is particularly valuable in many-query contexts such as optimization, control design, and inverse analysis.

This paper is organized as follows. Section 2 provides the necessary notation and preliminaries, and introduces the VS method for stochastic algebraic systems. It also provides a brief overview of the domain decomposition method for deterministic PDEs. In Section 3, we present the SDD-VS method proposed in this paper, detailing its key steps and procedures, and we discuss the corresponding numerical methods of the subproblems in Section 3.2. In Section 4, three numerical examples are presented to demonstrate the performance and computational advantages of the proposed SDD-VS method. Finally, we draw conclusions and offer some final remarks on the method and its potential applications.

2 Preliminaries and notations

In this section, we present some preliminaries and specify the notation of this paper. Let (Ω, \mathcal{B}, P) be a finite dimensional probability space, where Ω is the event space, \mathcal{B} is a σ -algebra on Ω , and P is the probability measure on \mathcal{B} . Let D be a given convex and bounded physical domain with Lipschitz continuous boundary ∂D . We use \mathcal{V} to denote a Hilbert space defined on D with an inner product defined by $(\cdot, \cdot)_{\mathcal{V}}$, and the induced norm is given by $\|\cdot\|_{\mathcal{V}} = \sqrt{(\cdot, \cdot)_{\mathcal{V}}}$. We consider the following SPDE defined on D

$$\begin{cases} \mathcal{L}(x, \boldsymbol{\xi}; u(x, \boldsymbol{\xi})) = f(x, \boldsymbol{\xi}), \quad \forall x \in D, \boldsymbol{\xi} \in \Omega, \\ \mathcal{B}(x, \boldsymbol{\xi}; u(x, \boldsymbol{\xi})) = g(x, \boldsymbol{\xi}), \quad \forall x \in \partial D, \boldsymbol{\xi} \in \Omega, \end{cases} \quad (1)$$

where $\boldsymbol{\xi} \in \Omega$ is a set of real-valued random variables, \mathcal{L} is a stochastic differential operator, \mathcal{B} is the boundary condition operator, f is the source term and g is the boundary term, $u(x, \boldsymbol{\xi})$ is the solution to this stochastic PDE. A practical instance of (1) is a stochastic diffusion equation given by

$$\begin{cases} -\nabla \cdot (c(x; \boldsymbol{\xi}) \nabla u(x; \boldsymbol{\xi})) = f(x; \boldsymbol{\xi}), \quad \forall x \in D, \boldsymbol{\xi} \in \Omega, \\ u(x; \boldsymbol{\xi}) = 0, \quad \forall x \in \partial D, \boldsymbol{\xi} \in \Omega, \end{cases}$$

where $c(x; \boldsymbol{\xi})$ is the diffusion coefficient.

The weak formulation of the problem (1) reads as follows: find $u \in \mathcal{V}$ such that

$$a(u(\boldsymbol{\xi}), v; \boldsymbol{\xi}) = b(v; \boldsymbol{\xi}), \quad \forall v \in \mathcal{V}, \quad (2)$$

where $a(\cdot, \cdot; \boldsymbol{\xi})$ and $b(\cdot; \boldsymbol{\xi})$ are a bilinear form and a linear form on \mathcal{V} , respectively. We assume that $a(\cdot, \cdot; \boldsymbol{\xi})$ and $b(\cdot; \boldsymbol{\xi})$ are affine with respect to $\boldsymbol{\xi}$, i.e.,

$$\begin{cases} a(w, v; \boldsymbol{\xi}) = \sum_{k=1}^{m_a} p^k(\boldsymbol{\xi}) a^k(w, v), \quad \forall w, v \in \mathcal{V}, \forall \boldsymbol{\xi} \in \Omega, \\ b(v; \boldsymbol{\xi}) = \sum_{k=1}^{m_b} q^k(\boldsymbol{\xi}) b^k(v), \quad \forall v \in \mathcal{V}, \forall \boldsymbol{\xi} \in \Omega, \end{cases} \quad (3)$$

where $p^k(\boldsymbol{\xi}) : \Omega \rightarrow R$ is a $\boldsymbol{\xi}$ -dependent stochastic function and $a^k : \mathcal{V} \times \mathcal{V} \rightarrow R$ is a bilinear form independent of $\boldsymbol{\xi}$, for each $k = 1, \dots, m_a$. Each $q^k(\boldsymbol{\xi}) : \Omega \rightarrow R$ is a $\boldsymbol{\xi}$ -dependent stochastic function and each $b^k : \mathcal{V} \rightarrow R$ is a linear form independent of $\boldsymbol{\xi}$, for $k = 1, \dots, m_b$. When $a(\cdot, \cdot; \boldsymbol{\xi})$ and $b(\cdot; \boldsymbol{\xi})$ are not affine with respect to $\boldsymbol{\xi}$, we can use the novel VS method for multivariate function [34] to obtain such an affine expansion approximation for them.

In particular, we consider the finite element (FE) approximation of problem (1) in an n -dimensional subspace $\mathcal{V}_h \subset \mathcal{V}$. Let $\{\psi_i\}_{i=1}^n$ be the set of basis functions of the FE space \mathcal{V}_h , the solution $u_h(\boldsymbol{\xi})$ can be represented by

$$u_h(\boldsymbol{\xi}) = \sum_{i=1}^n u_i(\boldsymbol{\xi}) \psi_i.$$

With the assumption (3) of affine decomposition, we have the matrix form of equation (2) in an FE space \mathcal{V}_h as follows

$$\left(\sum_{k=1}^{m_a} p^k(\boldsymbol{\xi}) A^k \right) \mathbf{u}(\boldsymbol{\xi}) = \sum_{k=1}^{m_b} q^k(\boldsymbol{\xi}) F^k, \quad (4)$$

where

$$(A^k)_{ij} = a^k(\psi_i, \psi_j), \quad (\mathbf{u}(\boldsymbol{\xi}))_j = u_j(\boldsymbol{\xi}), \quad (F^k)_j = b^k(\psi_j), \quad 1 \leq i, j \leq n.$$

Now we consider equation (4) as an example and introduce the VS method for stochastic algebraic systems. This is the key technology for the stochastic domain decomposition method we will propose in Section 3.

2.1 VS method for stochastic algebraic systems

We attempt to achieve an approximation of the stochastic algebraic system (4) in the form

$$\mathbf{u}(\boldsymbol{\xi}) \approx \mathbf{u}_N(\boldsymbol{\xi}) := \sum_{i=1}^N \zeta_i(\boldsymbol{\xi}) \mathbf{c}_i, \quad (5)$$

where $\zeta_i(\boldsymbol{\xi})$ are stochastic functions and \mathbf{c}_i are deterministic vectors, and N is the number of the separated terms. The VS method was first proposed in [34] for linear stochastic problems, and has seen extensions to stochastic saddle point problems in [35], and to nonlinear parameterized PDEs in [36]. In this contribution, we develop the strategy of VS for stochastic algebraic systems. The VS method employs an offline-online computational decomposition to enhance efficiency. In offline stage, to generate the reduced basis functions $\{\mathbf{c}_i\}_{i=1}^N$ and $\{\zeta_i(\boldsymbol{\xi})\}_{i=1}^N$, we need to compute a set of snapshots, which are the solutions of the stochastic problem (4) corresponding to a set of optimal parameter samples. In the online stage, the output is computed by the quasi-optimal separated representations for many instances of parameters, and the influence of the uncertainty is estimated.

Here we describe the detail of the VS algorithm to obtain $\{\mathbf{c}_i\}_{i=1}^N$ and $\{\zeta_i(\boldsymbol{\xi})\}_{i=1}^N$ in (5). To this end, we define the residual for the VS method by

$$e(\boldsymbol{\xi}) := \mathbf{u}(\boldsymbol{\xi}) - \mathbf{u}_{k-1}(\boldsymbol{\xi}). \quad (6)$$

By (4), we have that

$$\left(\sum_{j=1}^{m_a} p^j(\boldsymbol{\xi}) A^j \right) e(\boldsymbol{\xi}) = \sum_{j=1}^{m_b} q^j(\boldsymbol{\xi}) F^j - \left(\sum_{j=1}^{m_a} p^j(\boldsymbol{\xi}) A^j \right) \mathbf{u}_{k-1}(\boldsymbol{\xi}). \quad (7)$$

Let $r_k(\boldsymbol{\xi})$ be the residual of equation (4) when using $\mathbf{u}_{k-1}(\boldsymbol{\xi})$ to approximate $\mathbf{u}(\boldsymbol{\xi})$, that is,

$$r_k(\boldsymbol{\xi}) := \begin{cases} \sum_{j=1}^{m_b} q^j(\boldsymbol{\xi}) F^j, & k = 1, \\ \sum_{j=1}^{m_b} q^j(\boldsymbol{\xi}) F^j - \left(\sum_{j=1}^{m_a} p^j(\boldsymbol{\xi}) A^j \right) \mathbf{u}_{k-1}(\boldsymbol{\xi}), & k \geq 2. \end{cases} \quad (8)$$

Combining the representation of residual equation (8) with equation (7), we get the following error residual equation as

$$\sum_{j=1}^{m_a} p^j(\boldsymbol{\xi}) A^j e(\boldsymbol{\xi}) = r_k(\boldsymbol{\xi}). \quad (9)$$

At step k , we choose $\boldsymbol{\xi}_k$ as follows

$$\boldsymbol{\xi}_k := \begin{cases} \text{chosen randomly in } \Omega, & k = 1, \\ \operatorname{argmax}_{\boldsymbol{\xi} \in \Xi} \|r_k(\boldsymbol{\xi})\|_2, & k \geq 2, \end{cases}$$

where Ξ is a collection of a finite number of samples in Ω . Let \mathbf{e}_h be the solution of (9) with $\boldsymbol{\xi} = \boldsymbol{\xi}_k$, then we obtain the k -th deterministic column vector $\mathbf{c}_k = \mathbf{e}_h$ in (5). Given $e(\boldsymbol{\xi}) := \mathbf{c}_k e_{\boldsymbol{\xi}}(\boldsymbol{\xi})$, and $\mathbf{u}_{k-1}(\boldsymbol{\xi}) := \sum_{i=1}^{k-1} \zeta_i(\boldsymbol{\xi}) \mathbf{c}_i$ in equation (7), it follows that

$$\sum_{j=1}^{m_a} p^j(\boldsymbol{\xi}) A^j \mathbf{c}_k e_{\boldsymbol{\xi}}(\boldsymbol{\xi}) = \sum_{j=1}^{m_b} q^j(\boldsymbol{\xi}) F^j - \sum_{j=1}^{m_a} \sum_{i=1}^{k-1} p^j(\boldsymbol{\xi}) \zeta_i(\boldsymbol{\xi}) A^j \mathbf{c}_i. \quad (10)$$

Both sides of the equation (10) are taken dot product with \mathbf{c}_k . Then, we have

$$\sum_{j=1}^{m_a} p^j(\boldsymbol{\xi}) (\mathcal{A})_{kj} e_{\boldsymbol{\xi}}(\boldsymbol{\xi}) = \sum_{j=1}^{m_b} q^j(\boldsymbol{\xi}) (\mathcal{F})_j - \sum_{j=1}^{m_a} \sum_{i=1}^{k-1} p^j(\boldsymbol{\xi}) \zeta_i(\boldsymbol{\xi}) (\mathcal{A})_{ij},$$

where the matrix \mathcal{A} is defined by $(\mathcal{A})_{ij} = (\mathbf{c}_k)^T A^j \mathbf{c}_i$, for $1 \leq i \leq k$, $1 \leq j \leq m_a$, and the vector \mathcal{F} is defined by $(\mathcal{F})_j = (\mathbf{c}_k)^T F^j$, for $1 \leq j \leq m_b$. This gives rise to

$$e_{\boldsymbol{\xi}}(\boldsymbol{\xi}) = \frac{\sum_{j=1}^{m_b} q^j(\boldsymbol{\xi})(\mathcal{F})_j - \sum_{j=1}^{m_a} \sum_{i=1}^{k-1} p^j(\boldsymbol{\xi}) \zeta_i(\boldsymbol{\xi})(\mathcal{A})_{ij}}{\sum_{j=1}^{m_a} p^j(\boldsymbol{\xi})(\mathcal{A})_{kj}}. \quad (11)$$

Consequently, the k -th stochastic function in (5) can be obtained by taking $\zeta_k(\boldsymbol{\xi}) = e_{\boldsymbol{\xi}}(\boldsymbol{\xi})$. When $\|r_k(\boldsymbol{\xi}_k)\|_2 := \sqrt{(r_k(\boldsymbol{\xi}_k))^T r_k(\boldsymbol{\xi}_k)}$ is small enough, we can stop the iteration procedure. Algorithm 1 summarizes the above procedure of the VS method for stochastic algebraic systems.

Algorithm 1 The VS method for stochastic algebraic systems

Input: The stochastic algebraic system in (4), a set of samples $\Xi \in \Omega$, and the error tolerance ε .

Output: The separated representation $\mathbf{u}_N(\boldsymbol{\xi}) := \sum_{i=1}^N \zeta_i(\boldsymbol{\xi}) \mathbf{c}_i$.

- 1: Initialize the residual $r(\boldsymbol{\xi}) := \sum_{j=1}^{m_b} q^j(\boldsymbol{\xi}) F^j$, a random $\boldsymbol{\xi}_1 \in \Xi$, the iteration counter $k = 1$;
 - 2: Calculate $\mathbf{c}_k = \mathbf{e}_k$ by solving (9) with $\boldsymbol{\xi} = \boldsymbol{\xi}_k$, $\zeta_k(\boldsymbol{\xi}) = e_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ by (11);
 - 3: Update $\Xi = \Xi \setminus \boldsymbol{\xi}_k$, and take the approximation $\mathbf{u}_k(\boldsymbol{\xi}) := \sum_{i=1}^k \zeta_i(\boldsymbol{\xi}) \mathbf{c}_i$;
 - 4: $k \rightarrow k + 1$;
 - 5: Take the residual $r(\boldsymbol{\xi}) := \sum_{j=1}^{m_b} q^j(\boldsymbol{\xi}) F^j - \sum_{j=1}^{m_a} p^j(\boldsymbol{\xi}) A^j \mathbf{u}_{k-1}(\boldsymbol{\xi})$, and choose $\boldsymbol{\xi}_k = \operatorname{argmax}_{\boldsymbol{\xi} \in \Xi} \|r(\boldsymbol{\xi})\|_2$;
 - 6: Return to Step 2 if $\|r(\boldsymbol{\xi})\|_2 \geq \varepsilon$, otherwise **terminate**.
 - 7: $N = k$.
-

Note that the matrix \mathcal{A} and the vector \mathcal{F} are independent of random variables $\boldsymbol{\xi}$, and their computation is once in the offline phase. The online computation is to calculate (5) for any $\boldsymbol{\xi} \in \Omega$, which is efficient because the online computation only involves the separated representation (5).

2.2 Domain decomposition method for deterministic problem

In this section, we briefly review the non-overlapping domain decomposition method for deterministic PDEs (see, e.g., [27, 33] for more details), and take the equation (1) with Dirichlet boundary condition for a realization of random variable $\boldsymbol{\xi} \in \Omega$ as an example. In this case, we have the following weak formulation which is independent of random variables

$$a(u, v) = b(v), \quad \forall v \in \mathcal{V}.$$

Assumed that D is partitioned into N_s non-overlapping subdomains, denoted by $D_i, i = 1, 2, \dots, N_s$, such that

$$D = \bigcup_{i=1}^{N_s} \overline{D_i} \quad \text{and} \quad D_i \cap D_j = \emptyset, \quad \text{if } i \neq j,$$

and we define the interface of two adjoint subdomains D_i and D_j by Γ_{ij} , a notional example for two subdomains is shown in Figure 1. Moreover, we denote the Hilbert space \mathcal{V} restricted on subdomain D_i by \mathcal{V}_i .

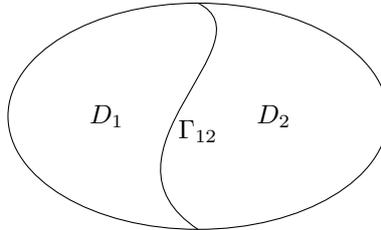


Figure 1: Illustration of a domain partitioned into two subdomains.

Then the weak formulation in a typical subdomain D_i can be expressed as $a(u, v)_i = b(v)_i, \forall v \in \mathcal{V}_i$. The finite element approximation of the above equation leads to a local linear system as follows

$$A_i \mathbf{u}_i = \mathbf{f}_i, \quad (12)$$

where A_i and \mathbf{f}_i are the local stiffness matrix and local load vector, respectively, and \mathbf{u}_i is the vector of local nodal values in the subdomain D_i . Due to the lack of boundary conditions, the local system (12) is singular. Then we consider dividing \mathbf{u}_i into two parts: the nodal shared by two or more adjacent subdomains, i.e. the interface part \mathbf{u}_Γ^i and the interior part \mathbf{u}_I^i that belong to subdomain D_i . Consequently, the system (12) can be rewritten as

$$\begin{bmatrix} A_{II}^i & A_{I\Gamma}^i \\ A_{\Gamma I}^i & A_{\Gamma\Gamma}^i \end{bmatrix} \begin{Bmatrix} \mathbf{u}_I^i \\ \mathbf{u}_\Gamma^i \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_I^i \\ \mathbf{f}_\Gamma^i \end{Bmatrix}.$$

Performing the Gaussian elimination technique, we know that once the interface unknowns \mathbf{u}_Γ^i are obtained, the interior unknowns \mathbf{u}_I^i can be obtained by solving the interior problem on each subdomain D_i as follows

$$\mathbf{u}_I^i = (A_{II}^i)^{-1}(\mathbf{f}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma^i). \quad (13)$$

Next, we discuss how to get the interface unknowns \mathbf{u}_Γ^i . First, we introduce the restriction matrix R_i (consisting of zeros and ones) represents a scatter operator, which relates the global interface unknowns \mathbf{u}_Γ and the local interface unknowns \mathbf{u}_Γ^i as

$$R_i \mathbf{u}_\Gamma = \mathbf{u}_\Gamma^i.$$

Define the local Schur complement matrix S_i and the corresponding right-hand side vector F_i as

$$S_i = A_{\Gamma\Gamma}^i - A_{\Gamma I}^i (A_{II}^i)^{-1} A_{I\Gamma}^i, \quad F_i = \mathbf{f}_\Gamma^i - A_{\Gamma I}^i (A_{II}^i)^{-1} \mathbf{f}_I^i.$$

Then the global interface unknowns \mathbf{u}_Γ can be obtained by solving the global interface problem

$$S \mathbf{u}_\Gamma = F, \quad (14)$$

where

$$S := \sum_{i=1}^{N_s} R_i^T S_i R_i, \quad F := \sum_{i=1}^{N_s} R_i^T F_i,$$

and the interior unknowns in each subdomain D_i can be obtained through (13).

It is worth noting that the interface system described above exhibits a smaller but denser structure compared to the original global system. Furthermore, the condition number of the Schur complement matrix S is generally better than the original global stiffness matrix [11]. Notably, the computational cost associated with calculating S and F primarily arises from the inversion of A_{II}^i , for $1 \leq i \leq N_s$, particularly when dealing with a large number of degrees of freedom. In this study, our objective is to explore the application of the domain decomposition method to stochastic partial differential equations (PDEs). In this context, all the aforementioned matrices are influenced by random inputs, introducing significant challenges when attempting to solve the stochastic interface system.

3 Stochastic domain decomposition based on Variable-separation method

In this section, we will present an offline-online method for the interface problem of the SPDEs, which builds a relation between the random inputs and the stochastic interface solution. The purpose of the offline stage is to construct all the components that are needed in the online stage. Details about the offline stage are provided in Subsections 3.1-3.2. Assuming that the original domain D is divided into N_s non-overlapping subdomains, the interface problem of stochastic PDEs can be expressed as follows (referring to equation (14)):

$$S(\boldsymbol{\xi}) \mathbf{u}_\Gamma(\boldsymbol{\xi}) = F(\boldsymbol{\xi}), \quad (15)$$

where

$$S(\boldsymbol{\xi}) := \sum_{i=1}^{N_s} R_i^T S_i(\boldsymbol{\xi}) R_i, \quad F(\boldsymbol{\xi}) := \sum_{i=1}^{N_s} R_i^T F_i(\boldsymbol{\xi}),$$

and

$$S_i(\boldsymbol{\xi}) = A_{\Gamma\Gamma}^i(\boldsymbol{\xi}) - A_{\Gamma I}^i(\boldsymbol{\xi})(A_{II}^i(\boldsymbol{\xi}))^{-1}A_{I\Gamma}^i(\boldsymbol{\xi}), \quad (16)$$

$$F_i(\boldsymbol{\xi}) = \mathbf{f}_{\Gamma}^i(\boldsymbol{\xi}) - A_{\Gamma I}^i(\boldsymbol{\xi})(A_{II}^i(\boldsymbol{\xi}))^{-1}\mathbf{f}_I^i(\boldsymbol{\xi}). \quad (17)$$

For an elementary exposition of the methodology, we consider dividing D into two subdomains D_1 and D_2 , and redefine the interface of D_1 and D_2 by Γ instead of Γ_{12} , as shown in Figure 1. It should be clarified that all the discussions in two subdomains can be generalized to the case of multiple subdomains. Then we have the interface problem of stochastic PDEs as follows

$$(S_1(\boldsymbol{\xi}) + S_2(\boldsymbol{\xi}))\mathbf{u}_{\Gamma}(\boldsymbol{\xi}) = F_1(\boldsymbol{\xi}) + F_2(\boldsymbol{\xi}). \quad (18)$$

Note that, for each sample $\boldsymbol{\xi} \in \Omega$, the calculation of $S_1(\boldsymbol{\xi}), S_2(\boldsymbol{\xi}), F_1(\boldsymbol{\xi}), F_2(\boldsymbol{\xi})$ by equations (16-17) depends on the inversion of the full order matrices $A_{II}^1(\boldsymbol{\xi}), A_{II}^2(\boldsymbol{\xi})$, which will substantially impact on the computation efficiency of solving the interface problem (18). To improve computational efficiency, we aim to reconstruct equation (18) such that $S(\boldsymbol{\xi})$ and $F(\boldsymbol{\xi})$ are affine with respect to $\boldsymbol{\xi}$, i.e.

$$S(\boldsymbol{\xi}) = \sum_{j=1}^{m_S} \hat{\eta}^j(\boldsymbol{\xi})\hat{\mathcal{X}}^j, \quad (19)$$

$$F(\boldsymbol{\xi}) = \sum_{j=1}^{m_F} \hat{\gamma}^j(\boldsymbol{\xi})\hat{F}^j. \quad (20)$$

The details of the process and definitions for $\hat{\eta}^j(\boldsymbol{\xi}), \hat{\gamma}^j(\boldsymbol{\xi}), \hat{\mathcal{X}}^j$ and \hat{F}^j will be introduced in Subsection 3.1. By equations (19-20), we have the stochastic interface problem such as

$$\sum_{j=1}^{m_S} \hat{\eta}^j(\boldsymbol{\xi})\hat{\mathcal{X}}^j\mathbf{u}_{\Gamma}(\boldsymbol{\xi}) = \sum_{j=1}^{m_F} \hat{\gamma}^j(\boldsymbol{\xi})\hat{F}^j. \quad (21)$$

The matrices $\hat{\mathcal{X}}^j$ and \hat{F}^j are independent of the random variables $\boldsymbol{\xi}$, and their computation is a one-time operation. For any $\boldsymbol{\xi} \in \Omega$, we just need to solve equation (21) instead of equation (15) involving the inversion of the full order matrices.

3.1 Assemble strategies for S and F

Now we describe the detail of the strategy for constructing equations (19-20). With the assumption (3) of affine decomposition, the stochastic matrices in equations (16-17) can be written as follows

$$\begin{aligned} A_{II}^i(\boldsymbol{\xi}) &= \sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi})A_{II}^{ij}, & A_{I\Gamma}^i(\boldsymbol{\xi}) &= \sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi})A_{I\Gamma}^{ij}, \\ A_{\Gamma I}^i(\boldsymbol{\xi}) &= \sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi})A_{\Gamma I}^{ij}, & A_{\Gamma\Gamma}^i(\boldsymbol{\xi}) &= \sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi})A_{\Gamma\Gamma}^{ij}, \end{aligned} \quad (22)$$

where $A_{II}^{ij}, A_{I\Gamma}^{ij}, A_{\Gamma I}^{ij}, A_{\Gamma\Gamma}^{ij}$ are independent of random variables $\boldsymbol{\xi}$. Thus the first item of $S_i(\boldsymbol{\xi})$ in equation (16), i.e., $A_{\Gamma\Gamma}^i(\boldsymbol{\xi})$ is affine with respect to $\boldsymbol{\xi}$ naturally. To achieve affine expression for the second item of $S_i(\boldsymbol{\xi})$, i.e., $A_{\Gamma I}^i(\boldsymbol{\xi})(A_{II}^i(\boldsymbol{\xi}))^{-1}A_{I\Gamma}^i(\boldsymbol{\xi})$, we perform it as follows.

- *Step 1: Construct the low-rank approximation of $X(\boldsymbol{\xi}) = (A_{II}^i(\boldsymbol{\xi}))^{-1}A_{I\Gamma}^i(\boldsymbol{\xi})$ such as*

$$X_N(\boldsymbol{\xi}) := \sum_{j=1}^{N_{S_i}} \beta_j(\boldsymbol{\xi})X_j. \quad (23)$$

First, we rewrite $(A_{II}^i(\boldsymbol{\xi}))^{-1}A_{I\Gamma}^i(\boldsymbol{\xi})$ as the following stochastic algebraic system

$$A_{II}^i(\boldsymbol{\xi})X(\boldsymbol{\xi}) = A_{I\Gamma}^i(\boldsymbol{\xi}), \quad (24)$$

with assumptions of affine decomposition (22), we have

$$\sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi})A_{II}^{ij}X(\boldsymbol{\xi}) = \sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi})A_{I\Gamma}^{ij}.$$

Let n_Γ be the number of the interface unknowns, we rewrite the stochastic matrix as $X(\boldsymbol{\xi}) = [x_1^i(\boldsymbol{\xi}), x_2^i(\boldsymbol{\xi}), \dots, x_{n_\Gamma}^i(\boldsymbol{\xi})]$. It follows that to get the solution of equation (24) is equivalent to solving the following n_Γ equations

$$\sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi}) A_{II}^{ij} x_k^i(\boldsymbol{\xi}) = \sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi}) \alpha_k^{ij}, \quad 1 \leq k \leq n_\Gamma, \quad (25)$$

where α_k^{ij} represents the k -th column of the matrix A_{II}^{ij} for $1 \leq k \leq n_\Gamma$, and $1 \leq j \leq m_a$. For each equation of (25), we adopt Algorithm 1 proposed in Subsection 2.1 to get the low-rank approximation of $x_k^i(\boldsymbol{\xi})$ such as (5). Consequently, the low-rank approximation of $X(\boldsymbol{\xi})$ in the form of (23) can be given by rearranging the low-rank approximations of $x_k^i(\boldsymbol{\xi})$ for $1 \leq k \leq n_\Gamma$.

• *Step 2: Assemble affine expression for*

$$\mathcal{X}^i(\boldsymbol{\xi}) = -A_{\Gamma I}^i(\boldsymbol{\xi})(A_{II}^i(\boldsymbol{\xi}))^{-1}A_{II}^i(\boldsymbol{\xi}). \quad (26)$$

Based on the low-rank representation of $X(\boldsymbol{\xi})$ and equation (23), we have

$$\mathcal{X}^i(\boldsymbol{\xi}) \approx -A_{\Gamma I}^i(\boldsymbol{\xi}) \sum_{k=1}^{N_{S_i}} \beta_k(\boldsymbol{\xi}) X_k = - \sum_{j=1}^{m_{a_i}} \sum_{k=1}^{N_{S_i}} p^{ij}(\boldsymbol{\xi}) \beta_k(\boldsymbol{\xi}) A_{\Gamma I}^{ij} X_k. \quad (27)$$

The affine decomposition of $A_{\Gamma I}^i(\boldsymbol{\xi})$ in equation (22) yields the second equality.

To simplify notation, the affine expression for $\mathcal{X}^i(\boldsymbol{\xi})$ can be obtained by using the single-index notation as follows

$$\mathcal{X}^i(\boldsymbol{\xi}) = \sum_{j \in J} \eta_{ij}(\boldsymbol{\xi}) \mathcal{X}^{ij}, \quad (28)$$

where $J = \{1, 2, \dots, m_{a_i}\} \times \{1, 2, \dots, N_{S_i}\}$, $\eta_{ij}(\boldsymbol{\xi}) = p^{ij_1}(\boldsymbol{\xi}) \beta_{j_2}(\boldsymbol{\xi})$, and $\mathcal{X}^{ij} = -A_{\Gamma I}^{ij_1} X_{j_2}$ is matrix of $n_\Gamma \times n_\Gamma$ for arbitrary $1 \leq j \leq m_{a_i} N_{S_i}$. Subsequently, we have the low-rank representation for $S_i(\boldsymbol{\xi})$ based on equation (16) as follows

$$S_i(\boldsymbol{\xi}) = A_{\Gamma \Gamma}^i(\boldsymbol{\xi}) + \mathcal{X}^i(\boldsymbol{\xi}) \approx \sum_{j=1}^{m_{a_i}} p^{ij}(\boldsymbol{\xi}) A_{\Gamma \Gamma}^{ij} + \sum_{j=1}^{m_{a_i} N_{S_i}} \eta_{ij}(\boldsymbol{\xi}) \mathcal{X}^{ij},$$

where the first equality follows from the definition of $\mathcal{X}^i(\boldsymbol{\xi})$ in equation (26), the second equality follows from the affine decomposition of $A_{\Gamma \Gamma}^i(\boldsymbol{\xi})$ and equation (28). Consequently, we have the low-rank representation for $S(\boldsymbol{\xi})$ as follows

$$\begin{aligned} S(\boldsymbol{\xi}) &= S_1(\boldsymbol{\xi}) + S_2(\boldsymbol{\xi}) \\ &\approx \sum_{j=1}^{m_{a_1}} p^{1j}(\boldsymbol{\xi}) A_{\Gamma \Gamma}^{1j} + \sum_{j=1}^{m_{a_2}} p^{2j}(\boldsymbol{\xi}) A_{\Gamma \Gamma}^{2j} + \sum_{j=1}^{m_{a_1} N_{S_1}} \eta_{1j}(\boldsymbol{\xi}) \mathcal{X}^{1j} + \sum_{j=1}^{m_{a_2} N_{S_2}} \eta_{2j}(\boldsymbol{\xi}) \mathcal{X}^{2j} \\ &= \sum_{j=1}^{m_{a_1}(N_{S_1}+1)+m_{a_2}(N_{S_2}+1)} \hat{\eta}^j(\boldsymbol{\xi}) \hat{\mathcal{X}}^j, \end{aligned} \quad (29)$$

where the last equality comes from stacking the variables and sorting the corresponding indices via a single one. Algorithm 2 outlines the assembling process of $S(\boldsymbol{\xi})$.

Algorithm 2 The Assemble Process of $S(\boldsymbol{\xi})$

Input: The stochastic matrices $A_{II}^i(\boldsymbol{\xi}), A_{\Gamma I}^i(\boldsymbol{\xi}), A_{\Gamma \Gamma}^i(\boldsymbol{\xi}), A_{II}^i(\boldsymbol{\xi}), i = 1, 2$.

Output: The low-rank representation $S(\boldsymbol{\xi}) = \sum_{j=1}^{m_{a_1}(N_{S_1}+1)+m_{a_2}(N_{S_2}+1)} \hat{\eta}^j(\boldsymbol{\xi}) \hat{\mathcal{X}}^j$.

- 1: Get the approximation of $X(\boldsymbol{\xi}) = (A_{II}^i(\boldsymbol{\xi}))^{-1}A_{II}^i(\boldsymbol{\xi})$ by solving equations (25) use Algorithm 1;
 - 2: Assemble the affine expression of $-A_{\Gamma I}^i(\boldsymbol{\xi})(A_{II}^i(\boldsymbol{\xi}))^{-1}A_{II}^i(\boldsymbol{\xi})$ by (27);
 - 3: Assemble $S_i(\boldsymbol{\xi})$ based on (22) and the expression derived in step 2;
 - 4: Assemble the low-rank representation of $S(\boldsymbol{\xi})$ by (29).
-

Similarly, with the assumption (3) of affine decomposition

$$\mathbf{f}_I^i(\boldsymbol{\xi}) = \sum_{j=1}^{m_{b_i}} q^{ij}(\boldsymbol{\xi}) \mathbf{f}_I^{ij}, \quad \mathbf{f}_\Gamma^i(\boldsymbol{\xi}) = \sum_{j=1}^{m_{b_i}} q^{ij}(\boldsymbol{\xi}) \mathbf{f}_\Gamma^{ij}, \quad (30)$$

where $\mathbf{f}_I^{ij}, \mathbf{f}_\Gamma^{ij}$ are independent of $\boldsymbol{\xi}$. Then following the assemble strategy of $S(\boldsymbol{\xi})$, the low-rank representation for $F(\boldsymbol{\xi})$ can be constructed as follows

$$\begin{aligned} F(\boldsymbol{\xi}) &= F_1(\boldsymbol{\xi}) + F_2(\boldsymbol{\xi}) \\ &\approx \sum_{j=1}^{m_{b_1}} q^{1j}(\boldsymbol{\xi}) \mathbf{f}_\Gamma^{1j} + \sum_{j=1}^{m_{b_2}} q^{2j}(\boldsymbol{\xi}) \mathbf{f}_\Gamma^{2j} + \sum_{j=1}^{m_{a_1} N_{F_1}} \gamma_{1j}(\boldsymbol{\xi}) F^{1j} + \sum_{j=1}^{m_{a_2} N_{F_2}} \gamma_{2j}(\boldsymbol{\xi}) F^{2j} \\ &= \sum_{j=1}^{m_{b_1} + m_{b_2} + m_{a_1} N_{F_1} + m_{a_2} N_{F_2}} \hat{\gamma}^j(\boldsymbol{\xi}) \hat{F}^j. \end{aligned} \quad (31)$$

We present the detail of the process of assembling $F(\boldsymbol{\xi})$ in Algorithm 3.

Algorithm 3 The Assemble Process of $F(\boldsymbol{\xi})$

Input: The stochastic matrices $A_{II}^i(\boldsymbol{\xi}), A_{\Gamma I}^i(\boldsymbol{\xi})$ and vectors $\mathbf{f}_I^i(\boldsymbol{\xi}), \mathbf{f}_\Gamma^i(\boldsymbol{\xi}), i = 1, 2$.

Output: The low-rank representation $F(\boldsymbol{\xi}) = \sum_{j=1}^{m_{b_1} + m_{b_2} + m_{a_1} N_{F_1} + m_{a_2} N_{F_2}} \hat{\gamma}^j(\boldsymbol{\xi}) \hat{F}^j$.

- 1: Get the approximation of $X(\boldsymbol{\xi}) = (A_{II}^i(\boldsymbol{\xi}))^{-1} \mathbf{f}_I^i(\boldsymbol{\xi})$ by Algorithm 1;
 - 2: Assemble the affine expression of $-A_{\Gamma I}^i(\boldsymbol{\xi})(A_{II}^i(\boldsymbol{\xi}))^{-1} \mathbf{f}_I^i(\boldsymbol{\xi})$;
 - 3: Assemble $F_i(\boldsymbol{\xi})$ based on (30) and the expression derived in step 2;
 - 4: Assemble the low-rank representation of $F(\boldsymbol{\xi})$ by (31).
-

Remark 3.1 Note that $m_{a_1}(N_{S_1} + 1) + m_{a_2}(N_{S_2} + 1)$ is the maximum number of the separated terms of $S(\boldsymbol{\xi})$. For practical problem, the number of the separated terms of $S(\boldsymbol{\xi})$ is smaller than $m_{a_1}(N_{S_1} + 1) + m_{a_2}(N_{S_2} + 1)$ when there exist $p^{1j}(\boldsymbol{\xi}) = p^{2k}(\boldsymbol{\xi}), j = 1, 2, \dots, m_{a_1}, k = 1, 2, \dots, m_{a_2}$, the same is true for $F(\boldsymbol{\xi})$.

3.2 Reduced model representation for the stochastic interface problem

As we mentioned before, once the separation approximations of $S(\boldsymbol{\xi})$ and $F(\boldsymbol{\xi})$ are available, the stochastic interface problem (18) can be given by

$$\sum_{j=1}^{m_S} \hat{\eta}^j(\boldsymbol{\xi}) \hat{\chi}^j \mathbf{u}_\Gamma(\boldsymbol{\xi}) = \sum_{j=1}^{m_F} \hat{\gamma}^j(\boldsymbol{\xi}) \hat{F}^j, \quad \forall \boldsymbol{\xi} \in \Omega, \quad (32)$$

where $m_S = m_{a_1}(N_{S_1} + 1) + m_{a_2}(N_{S_2} + 1)$ and $m_F = m_{b_1} + m_{b_2} + m_{a_1} N_{F_1} + m_{a_2} N_{F_2}$. The amended model defined in (32) for the interface problem is a linear algebraic system with n_Γ unknowns. Although the amended model needs much less computation effort compared with the original model (18), it may be not a very small-scale problem because the amended model defined in (32) involves the discrete degree of the original full model. In order to significantly improve the computation efficiency, we want to get the reduced model representation of the stochastic interface problem, i.e.,

$$\mathbf{u}_\Gamma(\boldsymbol{\xi}) \approx \sum_{i=1}^M \zeta_i(\boldsymbol{\xi}) \mathbf{c}_i, \quad (33)$$

where M is the number of the separated terms of $\mathbf{u}_\Gamma(\boldsymbol{\xi})$, each $\mathbf{c}_i, i = 1, \dots, M$ is a vector of n_Γ -dimension. Then the functional decomposition expression of the stochastic solution for the interface problem can be written as follows

$$u_\Gamma(\boldsymbol{\xi}) \approx \sum_{i=1}^M \zeta_i(\boldsymbol{\xi}) c_i(x), \quad (34)$$

with $c_i(x) = \sum_{j=1}^{n_\Gamma} (c_i)_j \psi_j(x)$, $i = 1, \dots, M$ and $\{\psi_j\}_{j=1}^{n_\Gamma}$ being the corresponding basis functions of the FE space \mathcal{V}_h on the interface Γ . We call this reduced model representation, which is more applicable to solving the subproblems on subdomains.

We employ the VS method for stochastic algebraic systems presented in Subsection 2.1 to derive the reduced model representation (33), which is beneficial to construct the efficient surrogate model of the subproblems.

The online stage of the SDD-VS method is to use the output of the offline stage to recover the solution to the stochastic interface problem for a large number of new samples. The online stage is efficient thanks to the reduced model representation (33) and (34) for the stochastic interface solution.

3.3 The numerical method for the subproblems

In the above subsections, we have a detailed description of the SDD-VS method for the stochastic interface problem. Once we get the stochastic solution $u_\Gamma^i(\boldsymbol{\xi})$ of the interface, the interior solution $u_I^i(\boldsymbol{\xi})$ in the subdomain D_i can be obtained by solving the following stochastic problem

$$\begin{cases} \mathcal{L}(x, \boldsymbol{\xi}; u_I^i(\boldsymbol{\xi})) = f(x, \boldsymbol{\xi}), & \forall x \in D_i, \boldsymbol{\xi} \in \Omega, \\ \mathcal{B}(x, \boldsymbol{\xi}; u_I^i(\boldsymbol{\xi})) = g(x, \boldsymbol{\xi}), & \forall x \in \partial D_i \setminus (\partial D_i \cap \Gamma), \boldsymbol{\xi} \in \Omega, \\ \mathcal{B}(x, \boldsymbol{\xi}; u_I^i(\boldsymbol{\xi})) = h_i(x, \boldsymbol{\xi}), & \forall x \in \partial D_i \cap \Gamma, \boldsymbol{\xi} \in \Omega, \end{cases} \quad (35)$$

where $h_i(x, \boldsymbol{\xi})$ is defined by the interface solution $u_\Gamma^i(\boldsymbol{\xi})$ such as equation (34) on the interface boundary, i.e., $\partial D_i \cap \Gamma$. To address the computational complexity associated with solving the stochastic partial differential equation (35), we attempt to apply model reduction methods to construct an efficient surrogate model. Model reduction methods have been proposed to reduce the computation complexity especially when the full model are expensive to perform numerical simulations. These methods construct an approximate model with lower dimensionality but still describe important aspects of the full model. The reduced basis method is one of the model order reduction methods and usually provides an efficient and reliable approximation of the input-output relationship [40, 41, 42, 43, 44, 45]. Another class of model reduction methods is based on the variable-separation (VS) method. As an example, the proper generalized decomposition (PGD) method has been used in solving stochastic partial differential equations (SPDEs) [47]. The PGD method constructs optimal reduced basis from a double orthogonality criterium [48, 49], and it requires the solutions of a few uncoupled deterministic problems solved by classical deterministic solution techniques and the solutions of stochastic algebraic are solved by classical spectral stochastic methods. We note that PGD requires many iterations with the arbitrary initial guess to compute each term in the separated expansion at each enrichment step. This will deteriorate the simulation efficiency.

Here, we would like to adopt the VS method we proposed in [34] for the stochastic problem (35) to get a separated representation for the solution without iterations at each enrichment step. Moreover, the proposed VS method can alleviate the ‘‘curse of dimensionality’’ when dealing with problems in high-dimensional stochastic spaces.

Finally, we summarize the SDD-VS method for stochastic PDEs in Figure 2.

4 Numerical experiments

In this section, we will present various numerical results to demonstrate the applicability and efficiency of the proposed SDD-VS method on several stochastic PDEs. For each problem, we seek a separate representation approximation to the interface problem induced by the model depending on random variables. All the numerical experiments in this paper were run in Python on a Dell desktop with Intel Core i7-4970 CPU @3.60GHz and 16GB of RAM. In Section 4.1, we consider the one-dimensional (1D) stochastic diffusion equation to illustrate the performance of the proposed SDD-VS method. In Section 4.2, we study the SDD-VS method for a two-dimensional (2D) stochastic diffusion equation with three subdomains. The 2D stochastic convection-diffusion equation with high-dimensional random inputs is considered in Section 4.3.

In order to quantify the accuracy of the proposed SDD-VS method, we use the relative mean error ϵ for the stochastic interface problem as follows

$$\epsilon = \frac{1}{N} \sum_{i=1}^N \frac{|u_\Gamma(\boldsymbol{\xi}_i) - \hat{u}_\Gamma(\boldsymbol{\xi}_i)|}{|u_\Gamma(\boldsymbol{\xi}_i)|}, \quad (36)$$

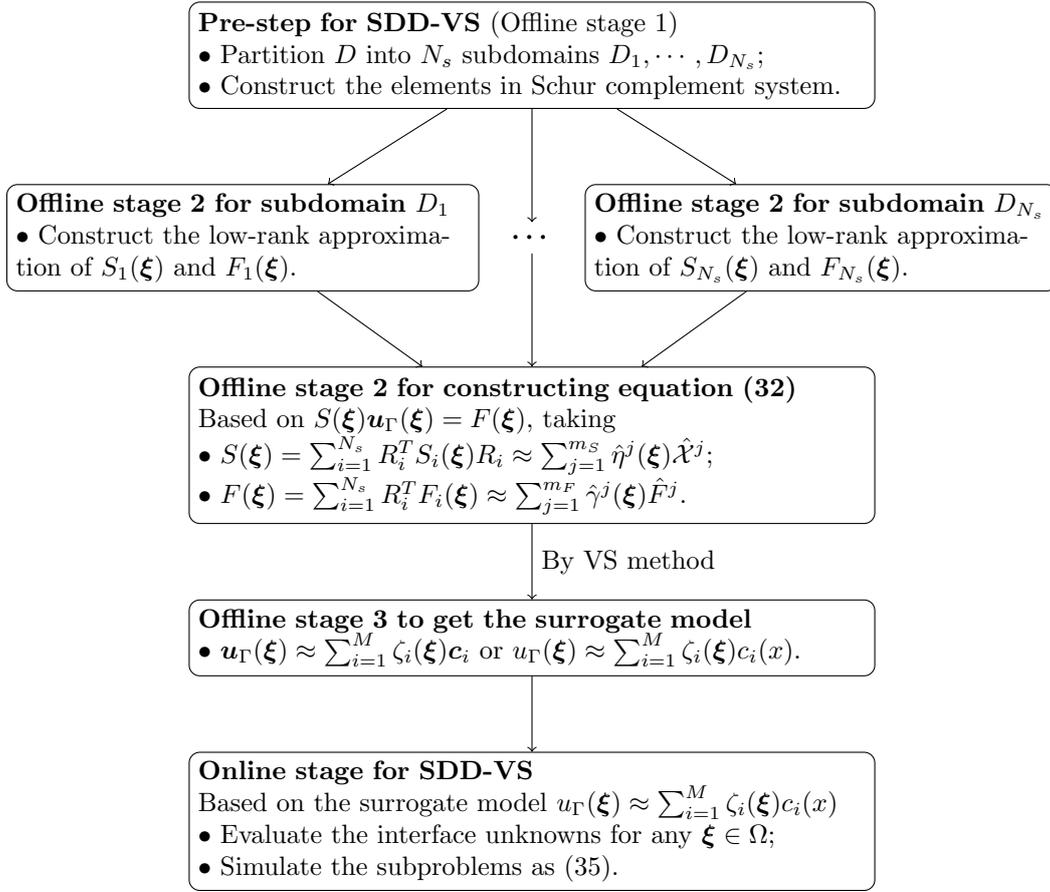


Figure 2: Flowchart of the SDD-VS method.

where N is the number of samples used to compute the mean error, $\hat{u}_\Gamma(\boldsymbol{\xi})$ is the approximation of the interface unknowns obtained by the SDD-VS method, $u_\Gamma(\boldsymbol{\xi})$ is the restriction of the reference solution, which is solved by the finite element method over the whole domain D , on the interface part Γ .

4.1 1D stochastic diffusion equation

We begin by considering a 1D stochastic diffusion equation with homogeneous Dirichlet boundary conditions

$$\begin{cases} -\frac{d}{dx}(c(x; \boldsymbol{\xi}) \frac{du(x; \boldsymbol{\xi})}{dx}) = f(x; \boldsymbol{\xi}), \quad \forall x \in D, \\ u(0; \boldsymbol{\xi}) = u(1; \boldsymbol{\xi}) = 0, \end{cases}$$

where the original domain $D = [0, 1]$ is divided into two subdomains $D_1 = [0, 0.5]$, $D_2 = (0.5, 1]$, and the random coefficient $c(x; \boldsymbol{\xi})$, the source function $f(x; \boldsymbol{\xi})$ are defined as follows

$$c(x; \boldsymbol{\xi}) = \begin{cases} \boldsymbol{\xi}x + 4, & \text{when } x \in D_1, \\ x + 1, & \text{when } x \in D_2, \end{cases}$$

$$f(x; \boldsymbol{\xi}) = \begin{cases} \cos(2\pi x), & \text{when } x \in D_1, \\ \boldsymbol{\xi}^2 x, & \text{when } x \in D_2. \end{cases}$$

In this example, we set the random variable $\boldsymbol{\xi}$ to be a truncated Gaussian distribution with mean 0, standard deviation 1, and range $[-3, 3]$. Here the reference solution is calculated by the finite element method with mesh size $h = 1/1000$. For the stochastic problems of one dimension, the reduced model (32)

$$\sum_{j=1}^{m_S} \hat{\eta}^j(\boldsymbol{\xi}) \hat{\mathcal{X}}^j \mathbf{u}_\Gamma(\boldsymbol{\xi}) = \sum_{j=1}^{m_F} \hat{\gamma}^j(\boldsymbol{\xi}) \hat{F}^j, \quad \forall \boldsymbol{\xi} \in \Omega,$$

is just a linear algebraic equation, where $\{\hat{\mathcal{X}}^j\}_{j=1}^{m_S}$ and $\{\hat{F}^j\}_{j=1}^{m_F}$ are constants. Thus we can obtain the simple analytic expression for the stochastic interface unknowns

$$\mathbf{u}_\Gamma(\boldsymbol{\xi}) = \frac{\sum_{j=1}^{m_F} \hat{\gamma}^j(\boldsymbol{\xi}) \hat{F}^j}{\sum_{j=1}^{m_S} \hat{\eta}^j(\boldsymbol{\xi}) \hat{\mathcal{X}}^j}, \quad \forall \boldsymbol{\xi} \in \Omega, \quad (37)$$

without further calculations. Therefore, most of the cost in the offline phase concentrates on the assemble process of $S_i(\boldsymbol{\xi})$, $F_i(\boldsymbol{\xi})$, $i = 1, 2$ in equations (16-17), i.e. the construction of their low-rank representation. To this end, we choose $|\Xi| = 20$ samples for training in Algorithm 1.

It should be noted that the relative mean error in all the assemble processes becomes smaller as the number of the separated terms increases. We take the construction of $S_1(\boldsymbol{\xi})$ as an example, and depict the relative mean error versus the number of the separated terms for the VS method in Figure 3, where the relative mean error ϵ is calculated by equation (36) with $N = 10^4$ test samples. This could provide evidence for the selection of $N_{S_1} = 4$ such that the relative mean error $\epsilon < 10^{-5}$ is small enough. Similarly, we take $N_{S_2} = 1, N_{F_1} = 4, N_{F_2} = 1$ for the construction of $S_2(\boldsymbol{\xi})$, $F_1(\boldsymbol{\xi})$ and $F_2(\boldsymbol{\xi})$ respectively. Thus we have $m_S = m_{a_1}(N_{S_1} + 1) + m_{a_2}(N_{S_2} + 1) = 12$ and $m_F = m_{b_1} + m_{b_2} + m_{a_1}N_{F_1} + m_{a_2}N_{F_2} = 11$ in the reduced model (32), where $m_{a_1} = 2, m_{a_2} = 1$ and $m_{b_1} = m_{b_2} = 1$.

Based on the reduced model (32), the interface solution by the SDD-VS method can be obtained by equation (37) directly. We implement it in two different conditions: (1) $N_{S_1} = 4, N_{S_2} = 1, N_{F_1} = 4, N_{F_2} = 1$; (2) $N_{S_1} = 6, N_{S_2} = 1, N_{F_1} = 4, N_{F_2} = 1$, and plot the relative errors in Figure 4 to visualize the individual relative error of the first 100 samples from $N = 10^4$ random samples. From the figure, we see that the method we proposed gives good approximation, and the relative error for each random sample becomes smaller when the number of the separated terms increases.

Figure 5 shows the mean of the solution generated by our proposed SDD-VS method with $N_{S_1} = 4, N_{S_2} = 1, N_{F_1} = 4, N_{F_2} = 1$ and the reference solution defined on each subdomain, we find that our method provides good approximations.

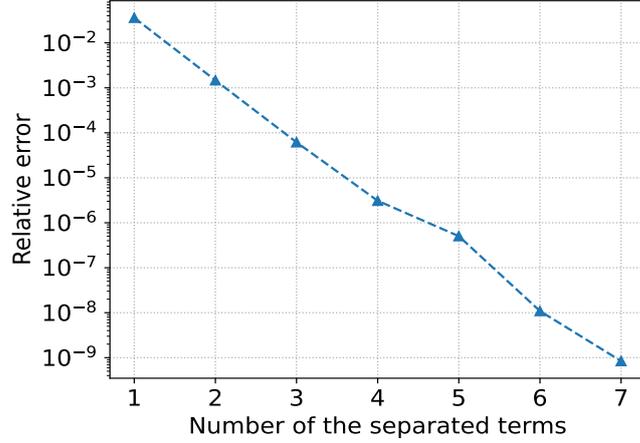


Figure 3: Comparison of the relative mean error corresponding to the different numbers of the separated terms N_{S_1} .

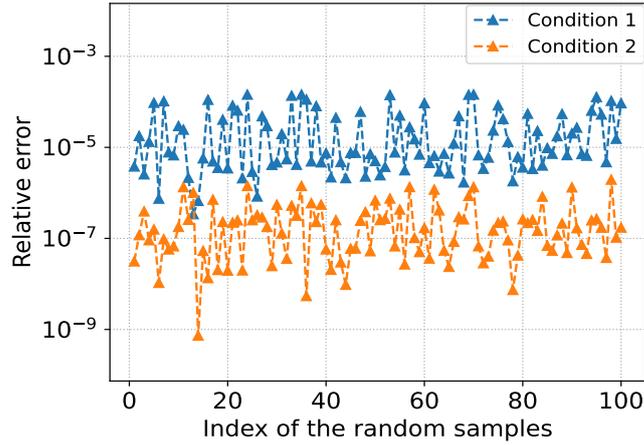


Figure 4: The relative error for 100 random samples by the SDD-VS method in two conditions: (1) $N_{S_1} = 4, N_{S_2} = 1, N_{F_1} = 4, N_{F_2} = 1$; (2) $N_{S_1} = 6, N_{S_2} = 1, N_{F_1} = 4, N_{F_2} = 1$.

4.2 2D stochastic diffusion equation with three subdomains

In this example, the stochastic diffusion equation is posed on domain $D = [0, 100] \times [0, 100]$ consists of three subdomains, which is defined by

$$\begin{cases} -\operatorname{div}(c(x; \xi) \nabla u(x; \xi)) = f(x; \xi), \quad \forall x \in D, \\ u(x; \xi) = g(x; \xi), \quad \forall x \in \partial D_1, \\ \frac{\partial u(x; \xi)}{\partial n} = h(x; \xi), \quad \forall x \in \partial D_2, \end{cases}$$

where $\partial D_1 = \{(0, 0), (100, 100)\}$, $\partial D_2 = \partial D \setminus \partial D_1$, the source function $f(x; \xi) = 0$, the Dirichlet boundary function $g(x; \xi)$ satisfied: $g((0, 0); \xi) = 20$, $g((100, 100); \xi) = 15$, the Neumann boundary function

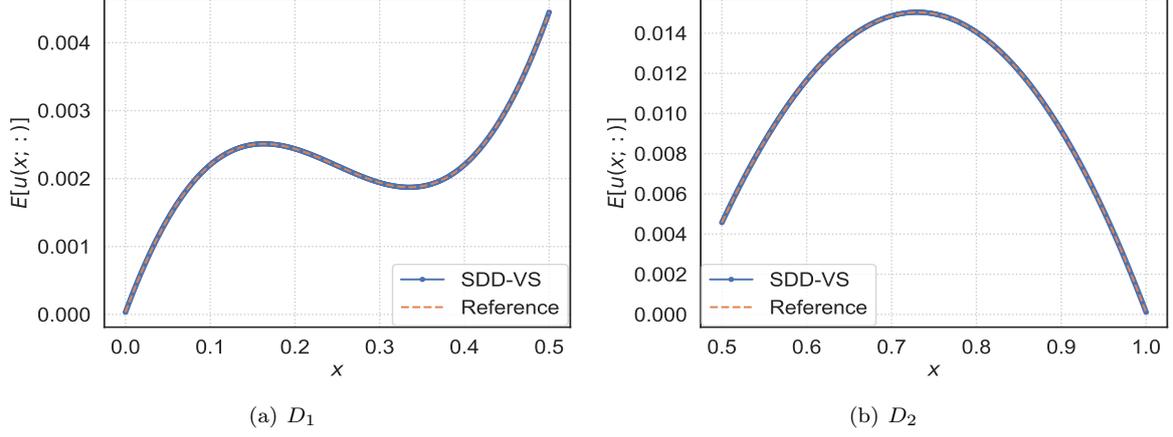


Figure 5: Comparison of the mean solution for reference and the SDD-VS method with $N_{S_1} = 4, N_{S_2} = 1, N_{F_1} = 4, N_{F_2} = 1$ in subdomains D_1 and D_2 .

$h(x; \xi) = 0$, and the random coefficient $c(x; \xi)$ is defined as

$$c(x; \xi) = \begin{cases} \frac{80}{\mu}, & \text{when } x \in D_1, \\ \frac{\xi}{\mu}, & \text{when } x \in D_2, \\ \frac{20}{\mu}, & \text{when } x \in D_3, \end{cases}$$

where $\mu = 0.02$, $D_1 = [0, 100] \times [0, 30]$, $D_2 = [0, 100] \times (30, 70]$, D_3 is the remainder part of the domain D . We set the random variable ξ to be uniformly distributed in the interval $[1, 4]$. Here, the finite element method calculates the reference solution with mesh size $h_x = h_y = 1/100$.

For this numerical example, the affine expansion approximation of the stochastic stiffness matrix and the load vector defined on each subdomain has only one term, i.e. $m_{a_i} = m_{b_i} = 1$, $i = 1, 2, 3$, and then $N_{S_1} = N_{S_2} = N_{S_3} = N_{F_1} = N_{F_2} = N_{F_3} = 1$. The affine expansion of $S(\xi)$ and $F(\xi)$ have a directly analytic expression with $m_S = 2$, $m_F = 1$ such as

$$\begin{aligned} S(\xi) &= \sum_{i=1}^{N_s} R_i^T S_i(\xi) R_i = \hat{\mathcal{X}}^1 + \hat{\mathcal{X}}^2 \xi, \\ F(\xi) &= \sum_{i=1}^{N_s} R_i^T F_i(\xi) = \hat{F}, \end{aligned} \tag{38}$$

where

$$\begin{aligned} \hat{\mathcal{X}}^1 &= \begin{bmatrix} A_{\Gamma_{12}\Gamma_{12}}^1 - A_{\Gamma_{12}I}^1 (A_{II}^1)^{-1} A_{I\Gamma_{12}}^1 & 0 \\ 0 & A_{\Gamma_{23}\Gamma_{23}}^3 - A_{\Gamma_{23}I}^3 (A_{II}^3)^{-1} A_{I\Gamma_{23}}^3 \end{bmatrix}, \\ \hat{\mathcal{X}}^2 &= \begin{bmatrix} A_{\Gamma_{12}\Gamma_{12}}^2 - A_{\Gamma_{12}I}^2 (A_{II}^2)^{-1} A_{I\Gamma_{12}}^2 & A_{\Gamma_{12}\Gamma_{23}}^2 - A_{\Gamma_{12}I}^2 (A_{II}^2)^{-1} A_{I\Gamma_{23}}^2 \\ A_{\Gamma_{23}\Gamma_{12}}^2 - A_{\Gamma_{23}I}^2 (A_{II}^2)^{-1} A_{I\Gamma_{12}}^2 & A_{\Gamma_{23}\Gamma_{23}}^2 - A_{\Gamma_{23}I}^2 (A_{II}^2)^{-1} A_{I\Gamma_{23}}^2 \end{bmatrix}, \end{aligned}$$

and

$$\hat{F} = \begin{Bmatrix} \mathbf{f}_{\Gamma_{12}}^1 - A_{\Gamma_{12}I}^1 (A_{II}^1)^{-1} \mathbf{f}_I^1 + \mathbf{f}_{\Gamma_{12}}^2 - A_{\Gamma_{12}I}^2 (A_{II}^2)^{-1} \mathbf{f}_I^2 \\ \mathbf{f}_{\Gamma_{23}}^2 - A_{\Gamma_{23}I}^2 (A_{II}^2)^{-1} \mathbf{f}_I^2 + \mathbf{f}_{\Gamma_{23}}^3 - A_{\Gamma_{23}I}^3 (A_{II}^3)^{-1} \mathbf{f}_I^3 \end{Bmatrix},$$

are independent of random variable ξ , and their computation is once. We choose $|\Xi| = 20$ samples to get the efficient surrogate model of $\mathbf{u}_\Gamma(\xi)$ by Algorithm 1 based on the following stochastic algebraic system

$$(\hat{\mathcal{X}}^1 + \hat{\mathcal{X}}^2 \xi) \mathbf{u}_\Gamma(\xi) = \hat{F}.$$

Firstly, we focus on the results of the stochastic interface problem solved by the SDD-VS method. In Figure 6, we depict the relative mean error of the stochastic interface problem by the SDD-VS method

versus a different number of the separated terms M , where the relative mean error ϵ is calculated with $N = 10^4$ test samples. According to the figure, we can see that as the number of the separated terms M increases, the approximation becomes more accurate.

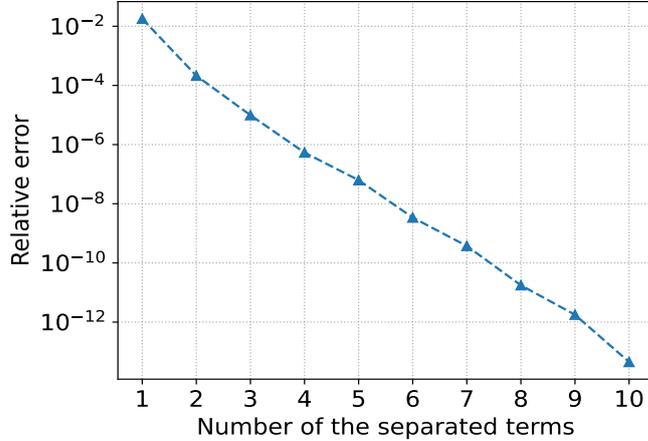


Figure 6: Comparison of the relative mean error corresponding to the different number of the separated terms M .

To visualize the individual relative error, we choose the first 100 samples from $N = 10^4$ random samples and plot the relative error with the number of the separated terms being $M = 2, M = 4, M = 6$ in Figure 7. This shows that the relative error for each individual sample becomes smaller when the number of the separated terms M increases.

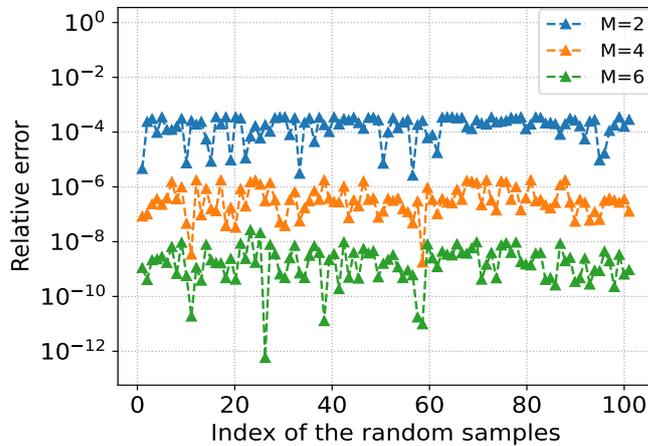


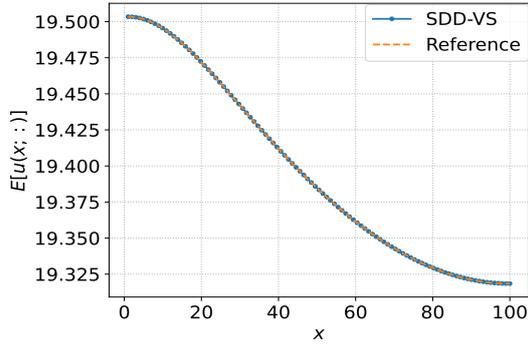
Figure 7: The relative error for 100 samples versus the number of the separated terms being $M = 2, M = 4, M = 6$.

Figure 8 shows the mean of the solution generated by the SDD-VS method and the reference solution in two interfaces $[0, 100] \times 30$ and $[0, 100] \times 70$ with the number of the separated terms $M = 3$ based on 10^4 random samples, as we can see, both interfaces yield an accurate mean solution.

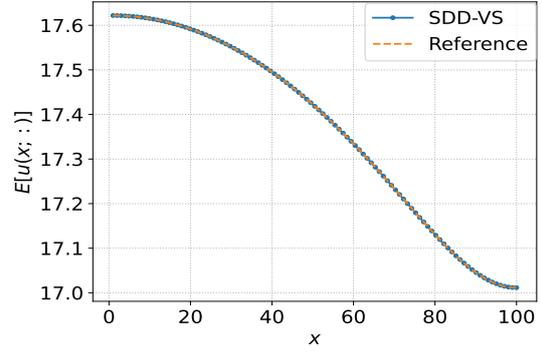
The probability density estimates of the reference and the SDD-VS method at a single measurement location in interfaces are shown in Figure 9. From the figure, we find that the SDD-VS method can give a good approximation for the reference probability density.

Following (13), the stochastic interior unknowns $\mathbf{u}_I^i(\boldsymbol{\xi})$ belong to each subdomain D_i can be obtained by solving the following stochastic linear equation

$$A_{II}^i(\boldsymbol{\xi})\mathbf{u}_I^i(\boldsymbol{\xi}) = \mathbf{f}_I^i(\boldsymbol{\xi}) - A_{I\Gamma}^i(\boldsymbol{\xi})\mathbf{u}_\Gamma^i(\boldsymbol{\xi}).$$

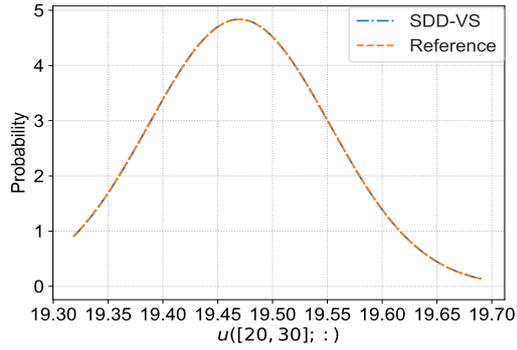


(a) Γ_1

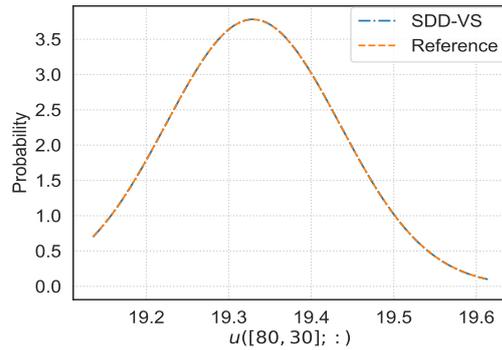


(b) Γ_2

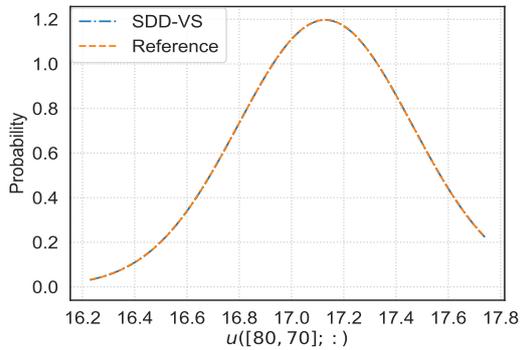
Figure 8: Comparison of the mean solution for the reference and the SDD-VS method in two interfaces with the number of the separated terms $M = 3$.



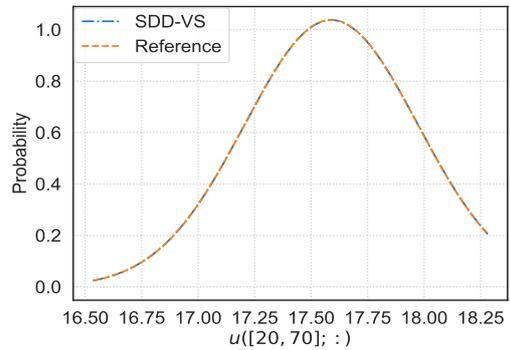
(a)



(b)



(c)



(d)

Figure 9: Comparison of the probability density of $u(x_0; \xi)$ for reference and the SDD-VS method with the number of the separated terms $M = 3$.

Indeed, for this type of SPDE, the subproblems can be solved more efficiently. In subdomain D_i , once we get the reduced model representation of the stochastic interface unknowns $\mathbf{u}_\Gamma^i(\boldsymbol{\xi})$, the reduced model representation of the interior unknowns $\mathbf{u}_I^i(\boldsymbol{\xi})$ can be obtained directly, which can be written as

$$\begin{aligned} \mathbf{u}_I^i(\boldsymbol{\xi}) &= (A_{II}^i)^{-1}(\mathbf{f}_I^i(\boldsymbol{\xi}) - A_{I\Gamma}^i \mathbf{u}_\Gamma^i(\boldsymbol{\xi})) \\ &\approx (A_{II}^i)^{-1} \sum_{j=1}^{m_{b_i}} q^{ij}(\boldsymbol{\xi}) \mathbf{f}_I^{ij} - (A_{II}^i)^{-1} A_{I\Gamma}^i \sum_{j=1}^M \zeta_j(\boldsymbol{\xi}) \mathbf{c}_j^i \\ &= \sum_{j=1}^{m_{b_i}} q^{ij}(\boldsymbol{\xi}) \hat{\mathbf{f}}_I^{ij} - \sum_{j=1}^M \zeta_j(\boldsymbol{\xi}) \hat{\mathbf{c}}_j^i, \quad i = 1, 3, \end{aligned} \quad (39)$$

$$\begin{aligned} \mathbf{u}_I^i(\boldsymbol{\xi}) &= (A_{II}^i)^{-1} \left(\frac{\mathbf{f}_I^i(\boldsymbol{\xi})}{\boldsymbol{\xi}} - A_{I\Gamma}^i \mathbf{u}_\Gamma^i(\boldsymbol{\xi}) \right) \\ &\approx (A_{II}^i)^{-1} \sum_{j=1}^{m_{b_i}} \frac{q^{ij}(\boldsymbol{\xi})}{\boldsymbol{\xi}} \mathbf{f}_I^{ij} - (A_{II}^i)^{-1} A_{I\Gamma}^i \sum_{j=1}^M \zeta_j(\boldsymbol{\xi}) \mathbf{c}_j^i \\ &= \sum_{j=1}^{m_{b_i}} \frac{q^{ij}(\boldsymbol{\xi})}{\boldsymbol{\xi}} \hat{\mathbf{f}}_I^{ij} - \sum_{j=1}^M \zeta_j(\boldsymbol{\xi}) \hat{\mathbf{c}}_j^i, \quad i = 2, \end{aligned} \quad (40)$$

where the second equation following from $A_{II}^i(\boldsymbol{\xi}) = A_{II}^i \boldsymbol{\xi}$, $A_{I\Gamma}^i(\boldsymbol{\xi}) = A_{I\Gamma}^i \boldsymbol{\xi}$, when $i = 2$, for each i , $\hat{\mathbf{c}}_j^i$ is the corresponding vector of \mathbf{c}_j restricted on the interface of D_i , $\hat{\mathbf{f}}_I^{ij} = (A_{II}^i)^{-1} \mathbf{f}_I^{ij}$, $\hat{\mathbf{c}}_j^i = (A_{II}^i)^{-1} A_{I\Gamma}^i \mathbf{c}_j^i$. In Figure (10), we plot the mean of the reference solution and the solution generated by (39-40) in three subdomains with the number of the separated terms $M = 3$ for the stochastic interface unknowns, the first row is the reference solution, and the second row is the solution generated by our proposed SDD-VS method. We find that our method works well for this problem.

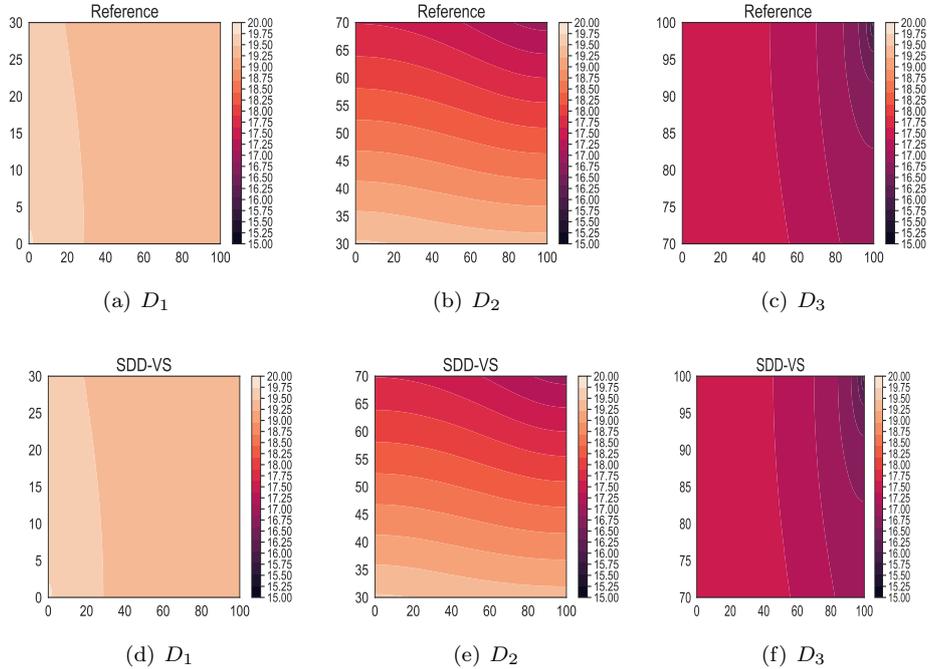


Figure 10: Comparison of the mean solution on three subdomains for reference and the SDD-VS method with the number of the separated terms $M = 3$.

Finally, since the VS method proposed in [34] is applicable to solve SPDEs directly, we compare the accuracy and the computational cost of the SDD-VS method against the VS method. To this end, $|\Xi| = 20$ samples are selected to construct the surrogate model for the VS method. In Table 1, we list the relative mean error, CPU time (offline CPU time \mathcal{T}_{off} , online CPU time \mathcal{T}_{on} , total CPU time \mathcal{T}_{tot}

and average online CPU time T_{on}) for SDD-VS method with the number of the separated terms $M = 10$ for the interface unknowns, VS method with the number of the separated terms $M_{VS} = 3, 6, 12$ and the reference method. Note that the online CPU time for the SDD-VS method can become smaller if we use the parallel strategy. From the table, we conclude that (1) as the number of separated terms M_{VS} increases, the CPU times needed for the VS method increase steadily; (2) the magnitude of average online CPU time by the SDD-VS method and VS method are much smaller than that of the reference method; (3) the SDD-VS method achieves much better approximation than the VS method with similar online computational cost, besides, the SDD-VS method uses much less computational cost than the VS method with similar approximation accuracy. In summary, the proposed SDD-VS method for this problem renders a more robust and accurate approximation than the VS method applied here directly.

Table 1: Comparison of average relative errors and the CPU times for SDD-VS, VS and the reference based on 10^4 parameter samples.

	SDD-VS	VS			Reference
		$M_{VS} = 3$	$M_{VS} = 6$	$M_{VS} = 12$	
ε	2.40×10^{-13}	1.37×10^{-5}	8.14×10^{-9}	1.37×10^{-13}	\
\mathcal{T}_{off}	10.39s	222.14s	223.23s	225.95s	\
\mathcal{T}_{on}	3.98s	9.53s	22.65s	59.67s	\
\mathcal{T}_{tot}	14.37s	231.67s	245.88s	285.62s	94507.32s
T_{on}	$3.98 \times 10^{-4}s$	$9.53 \times 10^{-4}s$	$2.27 \times 10^{-3}s$	$5.97 \times 10^{-3}s$	9.45s

4.3 2D stochastic convection-diffusion equation with high-dimensional random inputs

In the final example, we consider the stochastic convection-diffusion equation defined on domain $D = [0, 1] \times [0, 1]$ with high-dimensional random variables, which is defined by

$$\begin{cases} -\text{div}(c(x, y; \boldsymbol{\xi}) \nabla u(x, y; \boldsymbol{\xi})) + d(x, y; \boldsymbol{\xi}) \cdot \nabla u(x, y; \boldsymbol{\xi}) = f(x, y; \boldsymbol{\xi}), \quad \forall (x, y) \in D, \\ u(x, y; \boldsymbol{\xi}) = g(x, y; \boldsymbol{\xi}), \quad \forall (x, y) \in \partial D, \end{cases}$$

where the boundary condition function $g(x, y; \boldsymbol{\xi}) = 0$, the random diffusivity $c(x, y; \boldsymbol{\xi})$ and velocity $d(x, y; \boldsymbol{\xi})$ are defined as

$$c(x, y; \boldsymbol{\xi}) = \boldsymbol{\xi}_{33} + y\boldsymbol{\xi}_{34} + 3, \quad d(x, y; \boldsymbol{\xi}) = 1.$$

For this problem, we divide D into two subdomains $D_1 = [0, 0.5] \times [0, 1]$ and $D_2 = (0.5, 1] \times [0, 1]$. In each subdomain D_i , the source function $f(x, y; \boldsymbol{\xi})$ is taken as a random field, which is characterized by a two-point exponential covariance function $\text{cov}[f]$, i.e.,

$$\text{cov}[f](x_1, y_1; x_2, y_2) = \sigma^2 \exp\left(-\frac{|x_1 - x_2|^2}{2l_x^2} - \frac{|y_1 - y_2|^2}{2l_y^2}\right), \quad (41)$$

where $(x_i, y_i), i = 1, 2$ is the spatial coordinate, the variance $\sigma = 0.1$, correlation length $l_y = 0.5$ and we take different correlation lengths in the x direction for different subdomains, specifically, $l_x = 0.5$ in subdomain D_1 while $l_x = 0.05$ in subdomain D_2 . Then the random source term $f(x, y; \boldsymbol{\xi})$ is obtained by truncating a Karhunen-Loève expansion, i.e.,

$$f(x, y; \boldsymbol{\xi}) := E[f] + \sum_{i=1}^{16} \sqrt{\gamma_i} b_i(x, y) \boldsymbol{\xi}_i,$$

where $E[f] = 1$, the random vector $\boldsymbol{\xi} := (\xi_1, \xi_2, \dots, \xi_{34}) \in \mathbb{R}^{34}$, we assumed that $\xi_i, i = 1, 2, \dots, 34$ are i.i.d uniform random variables range $[-1, 1]$. The reference solution is obtained by the finite element method with mesh size $h_x = h_y = 1/60$.

We focus on the numerical result of the stochastic interface problem solved by the SDD-VS method with $N_{S_1} = N_{S_2} = 20$, $N_{F_1} = N_{F_2} = 80$. And we choose $|\Xi| = 120$ samples for training in Algorithm 1.

First, we randomly choose 10^3 samples and plot the relative mean error calculated by equation (36) versus the number of the separated terms M for the interface numerical solution in Figure 11, which shows that the SDD-VS method is suited for the high-dimensional stochastic problem, and the relative mean error becomes smaller when the number of the separated terms M increases.

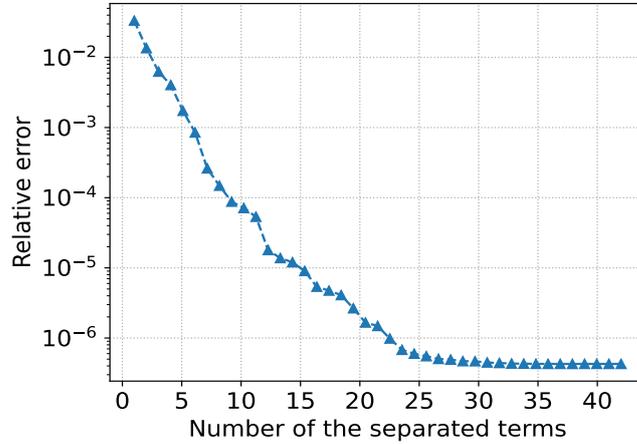


Figure 11: Comparison of the relative mean error corresponding to the different numbers of the separated terms M .

We plot the relative error for the number of the separated terms $M = 20$ in Figure 12 to visualize the individual relative error by choosing the first 100 samples out of 10^3 random samples. From the figure, we can see that the SDD-VS method can give a good approximation for each sample.

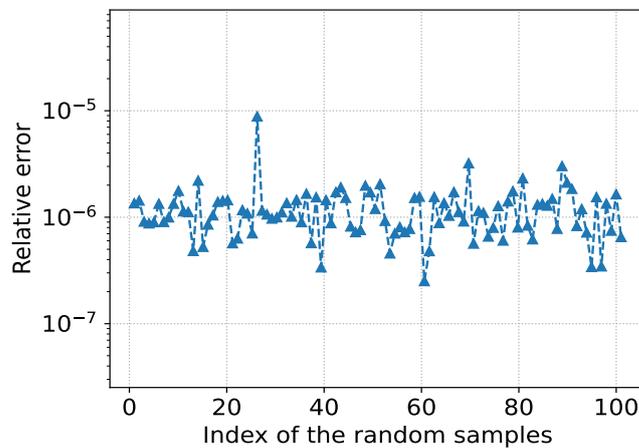


Figure 12: The relative mean error for 100 samples with the number of the separated terms $M = 20$.

In Figure 13, we plot the mean of the solution generated by our proposed SDD-VS method with the number of the separated terms $M = 4, 8, 12$ and the reference in the interface based on 10^3 random samples. As we can see, the approximation becomes better when the number of the separated terms M increases.

5 Conclusions

This paper presented the SDD-VS method for solving linear steady-state convection-diffusion equations with random coefficients. The proposed method is devoted to building a relationship between the random

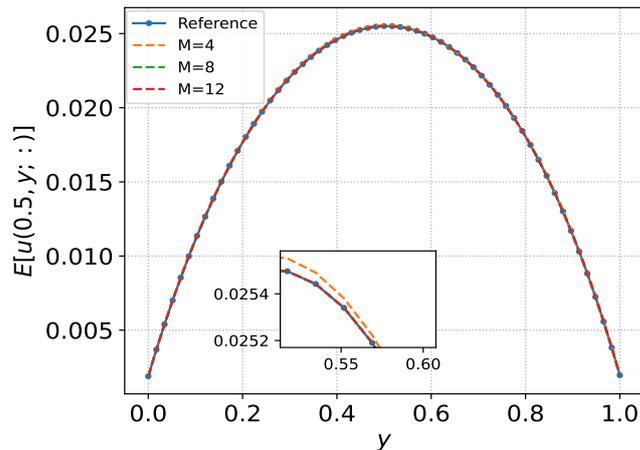


Figure 13: Comparison of the mean solution in the interface for reference and the SDD-VS method with the number of the separated terms $M = 4, 8, 12$.

inputs and the interface problem of SPDEs, in which the VS method plays a key role. The whole computation of the SDD-VS method admits an offlineonline decomposition. In the offline phase, we applied the direct Schur complement method to construct equation (15), which is dense and depends on the inversion of $A_{II}^i(\boldsymbol{\xi})$, for $1 \leq i \leq N_s$. It brings great challenges to solving the stochastic interface system (15) repeatedly, especially for many samples. To improve computational efficiency, we adopted the extended VS method to reconstruct $S(\boldsymbol{\xi})$ and $F(\boldsymbol{\xi})$ as in equations (19) and (20), i.e., having affine decomposition. This renders the reduced model (32), which needs much less computation effort than the original stochastic Schur complement system (15) with the aid of affine decomposition. However, it may be not a very small-scale problem especially when the number of subdomains N_s is large. To further improve efficiency, the VS method was used again to get the surrogate model for equation (32).

In the online phase, we used the functional decomposition expression (34) to recover the solution to the stochastic interface problem for a large number of new samples. Moreover, the efficient surrogate model of the stochastic subproblem (35) can be obtained by the VS method in [34]. The online phase is efficient thanks to the reduced model representation for the stochastic interface solution. We applied the proposed method to a few numerical models with random inputs. Careful numerical analysis was carried out for these numerical examples. We found that the SDD-VS method renders an efficient and robust reduced model.

In the future, we plan to extend the proposed method to tackle nonlinear unsteady problems. This will involve adapting the SDD-VS method to handle nonlinearity and time-dependent behavior. We also aim to conduct a rigorous convergence analysis for both the VS method and the SDD-VS method. This analysis will provide insights into the convergence properties and accuracy of the methods, enabling us to establish theoretical guarantees. Additionally, we will focus on applying the proposed method to models in dynamical systems, such as biological systems and petroleum engineering problems. By tailoring the methodology to these specific domains, we can address important challenges and develop efficient computational tools. In summary, our future research will involve extending the method to nonlinear unsteady problems, conducting convergence analysis, and exploring applications in dynamical systems. These efforts will contribute to advancing reduced-order modeling for stochastic partial differential equations and its practical use in various fields.

References

- [1] I. BABUŠKA, R. TEMPONE, AND G. ZOURARIS, *Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation*, *Comput. Methods Appl. Mech. Engrg.*, 194 (2005), pp. 1251-1294.
- [2] P. FRAUENFELDER, C. SCHWAB, AND R. TODOR, *Finite elements for elliptic problems with stochastic coefficients*, *Comput. Methods Appl. Mech. Engrg.*, 194 (2005), pp. 205-228.

- [3] H. G. MATTHIES AND A. KEESE, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, *Comput. Methods Appl. Mech. Engrg.*, 194 (2005), pp. 1295-1331.
- [4] I. BABUŠKA, F. NOBILE, AND R. TEMPONE, *A stochastic collocation method for elliptic partial differential equations with random input data*, *SIAM J. Numer. Anal.*, 45 (2007), pp. 1005-1034.
- [5] F. NOBILE, R. TEMPONE, AND C. G. WEBSTER, *A sparse grid stochastic collocation method for partial differential equations with random input data*, *SIAM J. Numer. Anal.*, 46 (2008), pp. 2309-2345.
- [6] D. XIU, *Efficient collocational approach for parametric uncertainty analysis*, *Commun. Comput. Phys.*, 2 (2007), pp. 293-309.
- [7] 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.
- [8] Y. ZHU AND N. ZABARAS, *Bayesian deep convolutional encoderdecoder networks for surrogate modeling and uncertainty quantification*, *J. Comput. Phys.*, 366 (2018), pp. 415-447.
- [9] Z. WANG AND Z. ZHANG, *A mesh-free method for interface problems using the deep learning approach*, *J. Comput. Phys.*, 400 (2020), pp. 108963.
- [10] M. PAPADRAKAKIS, G. STAVROULAKIS AND A. KARATARAKIS, *A new era in scientific computing: domain decomposition methods in hybrid CPU-GPU architectures*, *Comput. Methods Appl. Mech. Engrg.*, 200 (2011), pp. 14901508.
- [11] T. F. CHAN AND T. P. MATHEW, *Domain decomposition algorithms*, *Acta Numer.*, 3 (1994), pp. 61-143.
- [12] I. M. NAVON AND Y. CAI, *Domain decomposition and parallel processing of a finite element model of the shallow water equations*, *Comput. Methods Appl. Mech. Engrg.*, 106 (1993), pp. 179-212.
- [13] B. F. SMITH, P. BJORSTAD, AND W. GROPP, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, New York, 1996.
- [14] H. VEREECKEN, O. NEUENDORF, G. LINDENMAYR, AND A. BASERMANN, *A Schwarz domain decomposition method for solution of transient unsaturated water flow on parallel computers*, *Ecol. Model.*, 93 (1996), pp. 275-289.
- [15] A. QUARTERONI AND A. VALLI, *Domain decomposition methods for partial differential equations*, The Clarendon Press, Oxford University Press, New York, 1999.
- [16] A. TOSELLI AND O. B. WIDLUND, *Domain decomposition methods - algorithms and theory*, Springer Ser. Comput. Math., 34, Springer, Berlin Heidelberg, 2005.
- [17] X. CAI, *An optimal two-level overlapping domain decomposition method for elliptic problems in two and three dimensions*, *SIAM J. Sci. Comput.*, 14 (1993), pp. 239247.
- [18] M. JUNG, *On the parallelization of multi-grid methods using a non-overlapping domain decomposition data structure*, *Appl. Numer. Math.*, 23 (1997), pp. 119-137.
- [19] T. SUN AND K. MA, *Parallel Galerkin domain decomposition procedures for wave equation*, *J. Comput. Appl. Math.*, 233 (2010), pp. 1850-1865.
- [20] T. F. CHAN AND D. GOOVAERTS, *On the relationship between overlapping and nonoverlapping domain decomposition methods*, *SIAM J. Matrix Anal. Appl.*, 13 (1992), pp. 663-670.
- [21] T. F. CHAN, *Analysis of preconditioners for domain decomposition*, *SIAM J. Numer. Anal.*, 24 (1987), pp. 382-390.
- [22] J. MANDEL, *Balancing domain decomposition*, *Comm. Numer. Methods Engrg.*, 9 (1993), pp. 233-241.
- [23] M. HEINKENSCHLOSS, H. NGUYEN, *Neumann–Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems*, *SIAM J. Sci. Comput.*, 28 (2006), pp. 1001-1028.
- [24] J. SCHBERL, *Efficient contact solvers based on domain decomposition techniques*, *Comput. Math. Appl.*, 42 (2001), pp. 1217-1228.
- [25] A. SARKAR, N. BENABBOU, AND R. GHANEM, *Domain decomposition of stochastic PDEs: theoretical formulations*, *Internat. J. Numer. Methods Engrg.*, 77 (2009), pp. 689-701.
- [26] W. SUBBER AND S. LOISEL, *Schwarz preconditioners for stochastic elliptic PDEs*, *Comput. Methods Appl. Mech. Engrg.*, 272 (2014), pp. 34-57.
- [27] W. SUBBER AND A. SARKAR, *A domain decomposition method of stochastic PDEs: An iterative solution techniques using a two-level scalable preconditioner*, *J. Comput. Phys.*, 257(2014), pp. 298-317.
- [28] 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.
- [29] M. HADIGOL, A. DOOSTAN, H. G. MATTHIES, AND R. NIEKAMP, *Partitioned treatment of uncertainty in coupled domain problems: A separated representation approach*, *Comput. Methods Appl. Mech. Engrg.*, 274 (2014), pp. 103-124.

- [30] Q. LIAO AND K. WILLCOX, *A domain decomposition approach for uncertainty analysis*, SIAM J. Sci. Comput., 37 (2015), pp. A103-A133.
- [31] 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.
- [32] D. ZHANG, H. BABAEI, AND G. E. KARNIADAKIS, *Stochastic domain decomposition via moment minimization*, SIAM J. Sci. Comput., 40 (2018), pp. A2152-A2173.
- [33] L. MU AND G. ZHANG, *A domain decomposition model reduction method for linear convection-diffusion equations with random coefficients*, SIAM J. Sci. Comput., 41 (2019), pp. A1984-A2011.
- [34] Q. LI AND L. JIANG, *A novel variable-separation method based on sparse and low rank representation for stochastic partial differential equations*, SIAM J. Sci. Comput., 39 (2017), pp. A2879-A2910.
- [35] L. JIANG AND Q. LI, *Model reduction method using variable-separation for stochastic saddle point problems*, J. Comput. Phys., 354 (2018), pp. 43-66.
- [36] Q. LI AND P. ZHANG, *A variable-separation method for nonlinear partial differential equations with random inputs*, SIAM J. Sci. Comput., 42 (2020), pp. A723-A750.
- [37] L. TAMELLINI, O. LE MAÎTRE, AND A. NOUY, *Model reduction based on proper generalized decomposition for the stochastic steady incompressible Navier–Stokes equations*, SIAM J. Sci. Comput., 36 (2014), pp. A1089-A1117.
- [38] Z. ZHANG, M. CI, AND T. Y. HOU, *A multiscale data-driven stochastic method for elliptic PDEs with random coefficients*, Multiscale Model. Simul., 13 (2015), pp. 173204.
- [39] S. LI, Z. ZHANG, AND H. ZHAO, *A data-driven approach for multiscale elliptic PDEs with random coefficients based on intrinsic dimension reduction*, Multiscale Model. Simul., 18 (2020), pp. 12421271.
- [40] C. CANUTO, T. TONN, AND K. URBAN, *A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear pdes*, SIAM J. Numer. Anal., 47 (2009), pp. 2001-2022.
- [41] H. C. ELMAN AND Q. LIAO, *Reduced basis collocation methods for partial differential equations with random coefficients*, SIAM/ASA J. Uncertain. Quantificat., 1 (2013), pp. 192217.
- [42] P. CHEN, A. QUARTERONI, AND G. ROZZA, *A weighted reduced basis method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 51 (2013), pp. 3163-3185.
- [43] M. DROHMANN, B. HAASDONK, AND M. OHLBERGER, *Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. A937-A969.
- [44] L. JIANG AND Q. LI, *Model’s sparse representation based on reduced mixed GMsFE basis methods*, J. Comput. Phys., 338 (2017), pp. 285-312.
- [45] A. QUARTERONI, A. MANZONI, AND F. NEGRI, *Reduced basis methods for partial differential equations*, Springer, New York, 2015.
- [46] C. HOANG, Y. CHOI, AND K. CARLBERG, *Domain-decomposition least-squares PetrovGalerkin (DD-LSPG) nonlinear model reduction*, Comput. Methods Appl. Mech. Engrg., 384 (2021), 113997.
- [47] H. CHO, D. VENTURI, AND G. E. KARNIADAKIS, *Numerical methods for high-dimensional probability density function equations*, J. Comput. Phys., 305 (2016), pp. 817-837.
- [48] A. NOUY AND O. P. LE MAÎTRE, *Generalized spectral decomposition for stochastic nonlinear problems*, J. Comput. Phys., 228 (2009), pp. 202-235.
- [49] A. NOUY, *Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems*, Arch. Comput. Methods Eng., 17 (2010), pp. 403-434.