

# A PRECONDITIONED LOW-RANK PROJECTION METHOD WITH A RANK-REDUCTION SCHEME FOR STOCHASTIC PARTIAL DIFFERENTIAL EQUATIONS\*

KOOKJIN LEE<sup>†</sup> AND HOWARD C. ELMAN<sup>‡</sup>

**Abstract.** In this study, we consider the numerical solution of large systems of linear equations obtained from the stochastic Galerkin formulation of stochastic partial differential equations (PDEs). We propose an iterative algorithm that exploits the Kronecker product structure of the linear systems. The proposed algorithm efficiently approximates the solutions in low-rank tensor format. Using standard Krylov subspace methods for the data in tensor format is computationally prohibitive due to the rapid growth of tensor ranks during the iterations. To keep tensor ranks low over the entire iteration process, we devise a rank-reduction scheme that can be combined with the iterative algorithm. The proposed rank-reduction scheme identifies an important subspace in the stochastic domain and compresses tensors of high rank on-the-fly during the iterations. The proposed reduction scheme is a coarse-grid method in that the important subspace can be identified inexpensively in a coarse spatial grid setting. The efficiency of the proposed method is illustrated by numerical experiments on benchmark problems.

**Key words.** low-rank approximation, tensor format, stochastic Galerkin method, finite elements, GMRES, preconditioning, algebraic multigrid

**AMS subject classifications.** 35R60, 60H15, 60H35, 65F08, 65F10, 65N30

**DOI.** 10.1137/16M1075582

**1. Introduction.** Consider the stochastic elliptic boundary value problem, to find a random function  $u(\mathbf{x}, \xi) : D \times \Gamma \rightarrow \mathbb{R}$  that satisfies

$$(1.1) \quad \mathcal{L}(a(\mathbf{x}, \xi))(u(\mathbf{x}, \xi)) = f(\mathbf{x}) \quad \text{in } D \times \Gamma,$$

where  $\mathcal{L}$  is a linear elliptic operator and  $a(\mathbf{x}, \xi)$  is a positive random field parameterized by a set of random variables  $\xi = \{\xi_1, \dots, \xi_M\}$ . The problem is posed on a bounded domain  $D \subset \mathbb{R}^2$  with appropriate boundary conditions. Such problems arise, for example, from fluid flow and the transport of chemicals in flows in heterogeneous porous media, where the permeability coefficient is modeled as a random field [14, 33].

As the solution method for (1.1), we consider the stochastic Galerkin method [1, 2, 14], which, after discretization, leads to a large coupled deterministic system

$$(1.2) \quad Au = f,$$

for which computations will be expensive for large-scale applications. When the coefficient  $a(\mathbf{x}, \xi)$  has an affine structure depending on a finite number of random variables, the system matrix  $A$  can be represented by a sum of Kronecker products of smaller matrices. Matrix operations such as matrix-vector products that take advantage of

---

\*Received by the editors May 18, 2016; accepted for publication (in revised form) June 7, 2017; published electronically October 26, 2017.

<http://www.siam.org/journals/sisc/39-5/M107558.html>

**Funding:** This work was supported by the U.S. Department of Energy Office of Advanced Scientific Computing Research, Applied Mathematics program under award DEC-SC0009301, and by the U.S. National Science Foundation under grant DMS-1418754.

<sup>†</sup>Department of Computer Science, University of Maryland, College Park, MD 20742 (klee@cs.umd.edu).

<sup>‡</sup>Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (elman@cs.umd.edu).

the tensor format can be performed efficiently, which makes the use of iterative solvers attractive. In this study, we develop a new efficient iterative solver for systems represented in the Kronecker-product structure.

In recent years, many authors started to explore the Kronecker-product structure of such problems and developed iterative algorithms that exploit the structure to reduce computational efforts [5, 18, 19, 21, 24, 29]. In addition, it has been shown that the solution of (1.1) in the stochastic Galerkin setting can be approximated by a tensor of low rank, which further reduces computational effort [3, 4]. If Krylov subspace methods are used to compute such a solution, however, approximate solutions or other auxiliary terms obtained during the course of an iteration may not have low rank, and thus rank-reduction schemes are required to keep costs under control.

In this study, we will explore a variant of the generalized minimum residual (GMRES) method combined with a rank-reduction strategy that exploits specific features of the stochastic Galerkin formulation. The strategy we propose is a two-level scheme that first identifies a low-dimensional subspace, obtained from a coarse-grid spatial discretization, on which a low-rank coarse-grid tensor solution is computed. This solution can be used to estimate the rank of the tensor solution for the desired fine-grid solution. This information is used to define a strategy for rank reduction to be used with iteration on the fine-grid space. We show that this strategy enhances the efficiency of preconditioned GMRES for computing the solution.

The proposed method can be viewed as a dimension-reduction method because it identifies a dominant subspace and computes an approximate solution in that subspace. Other approaches developed for dimension reduction for the solutions of stochastic partial differential equations (PDEs) include reduced basis methods [17, 25], which construct a dominant subspace associated with parameterized models using greedy search methods, and active subspace methods [10], which detect a subspace of strong variability for a scalar-valued multivariate function using gradient computations. Another model reduction approach, developed in [11], identifies a dominant subspace based on the covariance structure of the solution on the coarse grid and uses the subspace for the fine-grid computation. The approach developed here uses an inexpensive low-rank approximation technique to construct the desired subspace on coarse-grid computations. Then the identified subspace is used to truncate tensors of high ranks in the iteration process to construct a solution on a finer spatial discretization.

An outline of the paper is as follows. In section 2, we review the stochastic Galerkin method and present the Kronecker-product structure of Galerkin systems. In section 3, we present a preconditioned projection method for computing approximate solutions in low-rank tensor format. In section 4, we review the conventional approaches and propose a coarse-grid rank-reduction scheme, which is the main contribution of this work. In section 5, we illustrate the effectiveness of the low-rank projection (LRP) method combined with the proposed truncation scheme by numerical experiments on benchmark problems. In section 6, we discuss the impact of truncation on solution statistics. Finally, in section 7 we draw some conclusions.

**2. Model problems with random inputs.** Consider the steady-state stochastic diffusion equation with homogeneous Dirichlet boundary conditions,

$$(2.1) \quad \begin{cases} -\nabla \cdot (a(\mathbf{x}, \omega) \nabla u(\mathbf{x}, \omega)) & = f(\mathbf{x}) & \text{in } D \times \Omega, \\ u(\mathbf{x}, \omega) & = 0 & \text{on } \partial D \times \Omega, \end{cases}$$

where the diffusion coefficient  $a(\mathbf{x}, \omega)$  is a random field and  $\omega$  is an elementary event in a probability space  $(\Omega, \mathcal{F}, P)$ . Here,  $\mathcal{F}$  and  $P$  are a  $\sigma$ -algebra on  $\Omega$  and a probability measure on  $\Omega$ , respectively. The gradient operator  $\nabla$  acts only on the physical domain  $D$ . The weak form of (2.1) is to find  $u$  in  $V = H_0^1(D) \otimes L^2(\Omega)$  such that

$$(2.2) \quad \left\langle \int_D a(\mathbf{x}, \omega) \nabla u(\mathbf{x}, \omega) \cdot \nabla v(\mathbf{x}, \omega) d\mathbf{x} \right\rangle = \left\langle \int_D f v(\mathbf{x}, \omega) \right\rangle \quad \forall v(\mathbf{x}, \omega) \in V,$$

where  $\langle \cdot \rangle$  refers to expected value with respect to the probability measure on  $L_2(\Omega)$  and  $V$  is equipped with the gradient norm

$$(2.3) \quad \|v\|_V^2 = \int_{\Omega} \int_D a(\mathbf{x}, \omega) |\nabla v(\mathbf{x}, \omega)|^2 d\mathbf{x} dP(\omega).$$

If  $a(\mathbf{x}, \omega)$  is bounded and strictly positive,

$$(2.4) \quad 0 < a_{\min} \leq a(\mathbf{x}, \omega) \leq a_{\max} < +\infty \quad \text{a.e. in } D \times \Omega,$$

then the Lax–Milgram lemma can be applied to establish existence and uniqueness of a solution  $u(\mathbf{x}, \omega) \in V$  of the variational problem (2.2). For the random field  $a(\mathbf{x}, \omega)$  with mean  $a_0$  and variance  $\sigma^2$ , we consider a truncated Karhunen–Loève (KL) expansion [20],

$$(2.5) \quad a(\mathbf{x}, \omega) \approx a_0 + \sigma \sum_{i=1}^M \sqrt{\lambda_i} a_i(\mathbf{x}) \xi_i(\omega),$$

where  $(\lambda_i, a_i)$  is an eigenpair of the covariance kernel  $C(\mathbf{x}, \mathbf{y})$  of the random field. In what follows, we change notation slightly and use  $a(\mathbf{x}, \xi)$ , as specified on the right-hand side of (2.5), to refer to the random diffusion coefficient, where  $\xi$  is an  $M$ -dimensional random variable with joint probability density function  $\rho(\xi)$ . We let  $\Gamma = \prod_{i=1}^M \Gamma_i$  denote the joint image of  $\xi$ , which we refer to as the *stochastic domain*. The expected value of a random variable  $v(\xi)$  on  $\Gamma$  is then  $\langle v(\xi) \rangle_{\rho} = \int_{\Gamma} v(\xi) \rho(\xi) d\xi$ .

**2.1. Stochastic Galerkin method.** The stochastic Galerkin method [1, 2, 14] seeks a finite-dimensional solution  $u_{hp}(\mathbf{x}, \xi) \in W^h = X_h \otimes S_M$  such that

$$(2.6) \quad \left\langle \int_D a(\mathbf{x}, \xi) \nabla u_{hp}(\mathbf{x}, \xi) \cdot \nabla v(\mathbf{x}, \xi) d\mathbf{x} \right\rangle_{\rho} = \left\langle \int_D f v(\mathbf{x}, \xi) \right\rangle_{\rho}, \quad v(\mathbf{x}, \xi) \in W^h,$$

where  $X_h$  and  $S_M$  are finite-dimensional subspaces of  $H_0^1(D)$  and  $L_{\rho}^2(\Gamma)$ ,

$$X_h = \text{span}\{\phi_r(\mathbf{x})\}_{r=1}^{n_x} \subset H_0^1(D), \quad S_M = \text{span}\{\psi_s(\xi)\}_{s=1}^{n_{\xi}} \subset L_{\rho}^2(\Gamma),$$

and

$$(2.7) \quad u_{hp}(\mathbf{x}, \xi) = \sum_{s=1}^{n_{\xi}} \sum_{r=1}^{n_x} u_{rs} \phi_r(\mathbf{x}) \psi_s(\xi).$$

Here,  $\{\phi_r\}$  is a set of standard finite element basis functions, and  $\{\psi_s\}$  is a set of basis functions for the generalized polynomial chaos expansion (PCE) [32] consisting of products of orthonormal univariate polynomials  $\psi_s(\xi) = \psi_{\alpha(s)}(\xi) = \prod_{i=1}^M \pi_{\alpha_i(s)}(\xi_i)$ , where  $\{\pi_{\alpha_i(s)}(\xi_i)\}_{i=1}^M$  is a set of univariate polynomials, and  $\alpha(s) = (\alpha_1(s), \dots, \alpha_M(s))$

is a multi-index, where  $\alpha_i$  represents the degree of a polynomial in  $\xi_i$ . In this study, we set the total degree (TD) space  $\Lambda_{M,p}$ ,

$$(2.8) \quad \Lambda_{M,p} = \{\alpha(s) \in \mathbb{N}_0^M : \|\alpha(s)\|_1 \leq p\},$$

where  $\mathbb{N}_0$  is the set of nonnegative integers,  $\|\alpha(s)\|_1 = \sum_{k=1}^M \alpha_k(s)$ , and  $p$  defines the maximal degree of  $\{\psi_i\}_{i=1}^{n_\xi}$ . Then the number of PCE basis functions is  $n_\xi = \dim(\Lambda_{M,p}) = \frac{(M+p)!}{M!p!}$ .

**2.2. Stochastic Galerkin formulation in tensor notation.** If the coefficients of (2.7) are ordered by grouping spatial indices together as  $u_{11}, u_{21}, \dots, u_{n_x 1}, u_{12}, \dots, u_{n_x n_\xi}$ , then the linear system  $Au = f$  of (1.2) can be represented in tensor product notation [23],

$$(2.9) \quad \left( G_0 \otimes K_0 + \sum_{l=1}^M G_l \otimes K_l \right) u = g_0 \otimes f_0,$$

where  $\otimes$  is the Kronecker product,  $K_i$  refers to the  $i$ th weighted stiffness matrix defined via

$$[K_0]_{ij} = \int_D a_0 \nabla \phi_i(\mathbf{x}) \nabla \phi_j(\mathbf{x}) d\mathbf{x}, \quad [K_l]_{ij} = \int_D \tilde{a}_l(\mathbf{x}) \nabla \phi_i(\mathbf{x}) \nabla \phi_j(\mathbf{x}) d\mathbf{x}, \quad l = 1, \dots, M,$$

$\tilde{a}_l(\mathbf{x}) = \sigma \sqrt{\lambda_l} a_l(\mathbf{x})$ ,  $G_i$  refers to the  $i$ th ‘‘stochastic’’ matrix defined via

$$(2.10) \quad [G_0]_{ij} = \langle \psi_i(\xi) \psi_j(\xi) \rangle_\rho, \quad [G_l]_{ij} = \langle \xi_l \psi_i(\xi) \psi_j(\xi) \rangle_\rho, \quad l = 1, \dots, M,$$

and the vectors  $f_0$  and  $g_0$  are defined via

$$[f_0]_i = \int_D f \phi_i(\mathbf{x}) d\mathbf{x}, \quad [g_0]_i = \langle \psi_i(\xi) \rangle_\rho.$$

Here the Kronecker product of two matrices  $G \in \mathbb{R}^{n_\xi \times n_\xi}$  and  $K \in \mathbb{R}^{n_x \times n_x}$  is

$$G \otimes K = \begin{bmatrix} [G]_{11} K & \dots & [G]_{1n_\xi} K \\ \vdots & & \vdots \\ [G]_{n_\xi 1} K & \dots & [G]_{n_\xi n_\xi} K \end{bmatrix}.$$

Note that  $\{G_l\}_{l=0}^M$  of (2.10) are highly sparse because of the orthogonality properties of the stochastic basis functions [13].

We will make use of an isomorphism between  $\mathbb{R}^{n_x n_\xi}$  and  $\mathbb{R}^{n_x \times n_\xi}$  determined by the operators  $\text{vec}(\cdot)$  and  $\text{mat}(\cdot)$ :  $u = \text{vec}(U)$ ,  $U = \text{mat}(u)$ , where  $u = [u_1^T, \dots, u_{n_\xi}^T]^T \in \mathbb{R}^{n_x n_\xi}$  with each  $u_i$  of length  $n_x$ , and  $U = [u_1, \dots, u_{n_\xi}] \in \mathbb{R}^{n_x \times n_\xi}$ . In particular, (2.9) is equivalent to its ‘‘matricized’’ form  $\sum_{l=0}^M K_l U G_l^T = f_0 g_0^T$ , and  $u$  and  $U$  can be used interchangeably to represent a solution of the Galerkin system. A solution  $u$  can be represented by a sum of vectors of Kronecker structure or, equivalently,  $U = \text{mat}(u)$  can be represented by a sum of rank-one matrices,

$$(2.11) \quad u = \sum_{k=1}^{\kappa_u} z_k \otimes y_k$$

$$(2.12) \quad \Leftrightarrow U = \sum_{k=1}^{\kappa_u} y_k z_k^T = Y_{\kappa_u} Z_{\kappa_u}^T,$$

where  $y_i \in \mathbb{R}^{n_x}$ ,  $z_i \in \mathbb{R}^{n_\xi}$ ,  $Y_{\kappa_u} = [y_1, \dots, y_{\kappa_u}] \in \mathbb{R}^{n_x \times \kappa_u}$ , and  $Z_{\kappa_u} = [z_1, \dots, z_{\kappa_u}] \in \mathbb{R}^{n_\xi \times \kappa_u}$ . A tensor of the form (2.11) is often referred to as having a *canonical decomposition* [9] (e.g.,  $x = \sum_{i=1}^{\kappa_x} \otimes_{j=1}^d x_i^j$ , where  $x \in \mathbb{R}^{n_1 \cdots n_d}$ ,  $x_i^j \in \mathbb{R}^{n_j}$  for  $i = 1, \dots, \kappa_x$ ,  $j = 1, \dots, d$ , and  $d$  refers to the dimension of the tensor). The tensor rank  $\kappa_u$  is defined as the smallest number of terms needed to represent  $u$ . In this study, the dimension of the tensor  $u$  is two, and the tensor rank  $\kappa_u$  of the tensor  $u$  coincides with the rank of the matrix  $U$ . Thus, in what follows we also use  $\kappa_u$  to refer to the rank of  $u$ . With this notation, the stochastic Galerkin solution  $u_{hp}(\mathbf{x}, \xi)$  can be represented as

$$(2.13) \quad u_{hp}(\mathbf{x}, \xi) = \Phi(\mathbf{x})^T Y_{\kappa_u} Z_{\kappa_u}^T \Psi(\xi) = (Y_{\kappa_u}^T \Phi(\mathbf{x}))^T (Z_{\kappa_u}^T \Psi(\xi)),$$

where  $\Phi : D \rightarrow \mathbb{R}^{n_x}$  is given by  $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_{n_x}(\mathbf{x})]^T$  and  $\Psi : \Gamma \rightarrow \mathbb{R}^{n_\xi}$  is given by  $\Psi(\xi) = [\psi_1(\xi), \dots, \psi_{n_\xi}(\xi)]^T$ . As shown in [31], (2.13) corresponds to a separated representation [7],

$$(2.14) \quad u_{hp}(\mathbf{x}, \xi) = \sum_{i=1}^{\kappa_u} \hat{y}_i(\mathbf{x}) \hat{z}_i(\xi),$$

where  $\hat{y}_i(\mathbf{x}) = (\Phi(\mathbf{x}))^T y_i$  and  $\hat{z}_i(\xi) = (\Psi(\xi))^T z_i$ . We will use this representation to construct a new rank-reduction operator. In the discrete model (2.13), the rank of the solution is typically  $\kappa_u = \min(n_x, n_\xi)$ .

In [6, 16], it was shown that the solution to (2.9) can be approximated well by a quantity  $\tilde{u}$  of rank  $\kappa_{\tilde{u}} \ll \min(n_x, n_\xi)$  if the system matrix and the right-hand side have Kronecker-product structure. Thus, we seek a low-rank approximation to the solution  $\tilde{u}$  to (2.9), for which

$$(2.15) \quad A\tilde{u} = \left( \sum_{l=0}^M G_l \otimes K_l \right) \left( \sum_{k=1}^{\kappa_{\tilde{u}}} \tilde{z}_k \otimes \tilde{y}_k \right) \approx g_0 \otimes f_0.$$

**2.3. Basic operations in tensor notation.** Here we point out a feature of the basic operations required by Krylov subspace methods in the setting we are considering, where the operators and data of interest have tensor format. The  $m$ th step of such methods results in the *Krylov subspace*,  $\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, \dots, A^{m-1}v_1\}$ , which is generated using matrix-vector products and addition/subtraction of vectors.

The matrix-vector product in (2.15) can be represented as a sum of rank-one tensors by exploiting the properties of the Kronecker product,

$$(2.16) \quad Au = \sum_{l=0}^M \sum_{k=1}^{\kappa_u} G_l z_k \otimes K_l y_k = \sum_{i=1}^{(M+1)\kappa_u} \hat{z}_i \otimes \hat{y}_i.$$

The latter expression in (2.16) suggests that in tensor notation, the matrix-vector product typically results in a vector with a higher rank. Similarly, the addition of two vectors  $u$  and  $v$  of ranks  $\kappa_u$  and  $\kappa_v$  in tensor notation gives

$$(2.17) \quad u + v = \sum_{i=1}^{\kappa_u} z_i \otimes y_i + \sum_{j=1}^{\kappa_v} \hat{z}_j \otimes \hat{y}_j = \sum_{i=1}^{\kappa_u + \kappa_v} z_i \otimes y_i,$$

where  $y_{i+\kappa_u} = \hat{y}_i$  and  $z_{i+\kappa_u} = \hat{z}_i$ ,  $i = 1, \dots, \kappa_v$ , so that the resulting sum may have rank as large as  $\kappa_u + \kappa_v$ . Thus, although the goal is to find an approximate solution to (2.9) of low rank, two of the fundamental operations used in Krylov subspace methods tend to increase the rank of the quantities produced. Following [5], we will address this point in the next section.

**3. A preconditioned projection method in tensor format.** As is well known, the GMRES method [28] constructs an approximate solution  $u_m \in u_0 + \mathcal{K}_m(A, v_1)$ , where  $u_0$  is an initial vector with residual  $r_0 = f - Au_0$ ,  $v_1 = r_0/\|r_0\|_2$ , and  $\mathcal{K}_m$  is the Krylov space. This is done by generating  $V_m = [v_1, \dots, v_m]$ , where  $\{v_j\}_{j=1}^m$  is an orthogonal basis for  $\mathcal{K}_m$ , and then computing  $u_m$ , whose residual  $r_m$  is orthogonal to  $W_m = AV_m$ . The method is shown in Algorithm 1. In this section, we discuss a variant of this method based on LRP, where advantage is taken of the Kronecker format of the matrix  $A$  and the fact that we seek an approximation of  $u$  with low-rank structure.

---

**Algorithm 1** GMRES method without restarting [27].

---

- 1: set the initial solution  $u_0$
  - 2:  $r_0 := f - Au_0$
  - 3:  $\tilde{v}_1 := r_0$
  - 4:  $v_1 := \tilde{v}_1/\|\tilde{v}_1\|$
  - 5: **for**  $j = 1, \dots, m$  **do**
  - 6:  $w_j := Av_j$
  - 7: solve  $(V_j^T V_j)\alpha = V_j^T w_j$
  - 8:  $\tilde{v}_{j+1} := w_j - \sum_{i=1}^j \alpha_i v_i$
  - 9:  $v_{j+1} := \tilde{v}_{j+1}/\|\tilde{v}_{j+1}\|$
  - 10: **end for**
  - 11: solve  $(W_m^T AV_m)y = W_m^T r_0$
  - 12:  $u_m := u_0 + V_m y$
- 

**3.1. Low-rank projection method with restarting.** As we observed in section 2, matrix-vector products and vector sums in tensor structure tend to increase the rank of the resulting objects. Thus, although we seek a solution of low rank, straightforward use of the GMRES method may lead to approximate solutions of higher rank than the desired solutions. This complication can be addressed using *truncation operators* [5, 18, 19, 21, 29], whereby vectors of high rank are replaced by ones of low rank. The truncation is inserted into the GMRES algorithm and is interleaved with the basic operations such as matrix-vector product and addition so that the ranks of the vectors used in the algorithm are kept low.

Algorithm 2 summarizes the restarted LRP method in tensor format [5]. As in the standard Arnoldi iteration used by GMRES, a new vector is constructed by applying the linear operator  $A$  to the previous basis vector  $v_j$  and orthogonalizing the new basis vector  $w_j$  with respect to the previous basis vectors  $\{v_i\}_{i=1}^j$ . The resulting vector is truncated to a vector  $\tilde{v}_{j+1}$  of low rank and normalized to  $v_{j+1}$ , which is then added to the set of basis vectors. The truncation operator  $\mathcal{T}_\kappa$  truncates a tensor of higher rank to one of rank  $\kappa$ . Thus, all the basis vectors  $\{v_i\}_{i=1}^m$  are of the same rank,  $\kappa$ . The basis vectors determine the subspace  $\mathcal{K}_m = \text{span}\{v_1, \dots, v_m\}$ , but because of truncation the basis vectors are not orthogonal, and  $\mathcal{K}_m$  is not a Krylov subspace. However, it is still possible to project the residual onto the subspace  $\mathcal{W}_m = \text{span}\{w_1, \dots, w_m\}$  to find out whether the residual can be decreased by forming a new iterate  $\tilde{u}_k + V_m \beta$ . Note that all of the vectors used in the entire iteration process are stored as the product of two matrices in the form shown in (2.12). The ranks of these vectors will be discussed below.

---

**Algorithm 2** Restarted LRP method in tensor format.

---

```

1: set the initial solution  $\tilde{u}_0$ 
2: for  $k = 0, 1, \dots$  do
3:    $r_k := f - A\tilde{u}_k$ 
4:   if  $\|r_k\|/\|f\| < \epsilon$  then
5:     return  $\tilde{u}_k$ 
6:   end if
7:    $\tilde{v}_1 := \mathcal{T}_\kappa(r_k)$ 
8:    $v_1 := \tilde{v}_1/\|\tilde{v}_1\|$ 
9:   for  $j = 1, \dots, m$  do
10:     $w_j := Av_j$ 
11:    solve  $(V_j^T V_j)\alpha = V_j^T w_j$ 
12:     $\tilde{v}_{j+1} := \mathcal{T}_\kappa\left(w_j - \sum_{i=1}^j \alpha_i v_i\right)$ 
13:     $v_{j+1} := \tilde{v}_{j+1}/\|\tilde{v}_{j+1}\|$ 
14:   end for
15:   solve  $(W_m^T AV_m)\beta = W_m^T r_k$ 
16:    $\tilde{u}_{k+1} := \mathcal{T}_\kappa(\tilde{u}_k + V_m\beta)$ 
17: end for

```

---

**3.2. Preconditioned low-rank projection method.** To speed the convergence of the projection method, we consider a right-preconditioned system,

$$(3.1) \quad AM^{-1}\hat{u} = f, \quad \hat{u} = Mu.$$

For the stochastic diffusion problem, we consider  $M = G_0 \otimes \tilde{K}_0 \approx G_0 \otimes K_0$  as the preconditioner, a mean-based preconditioner [23]. For the practical application of the preconditioner, we employ algebraic multigrid methods [26], where the action of  $K_0^{-1}$  is replaced by  $\tilde{K}_0^{-1}$ , an application of a single V-cycle of an algebraic multigrid method. The multigrid algorithm used point damped Jacobi smoothing with damping parameter 0.5 and two presmoothing and two postsmoothing steps, together with bilinear interpolation for grid transfer (as implemented in [30]). The preconditioned matrix-vector product is then

$$AM^{-1}\hat{u} = \sum_{l=0}^M \sum_{k=1}^{\kappa_{\hat{u}}} G_l \hat{z}_k \otimes K_l \tilde{K}_0^{-1} \hat{y}_k, \quad \hat{u} = Mu = \sum_{i=1}^{\kappa_{\hat{u}}} \hat{z}_i \otimes \hat{y}_i.$$

Note that  $G_0^{-1}$  is the identity matrix because of the orthonormality of the stochastic basis functions. With right preconditioning and this preconditioner, the strategy for handling tensor rank is largely unaffected by preconditioning.

**4. Truncation methods.** As discussed in section 3.1, in the LRP method, truncation of tensors is essential for the efficient computation of approximate solutions. In this section, we discuss the conventional approach for truncation and introduce a new coarse-grid truncation method based on a coarse-grid solution.

**4.1. Truncation based on singular values.** Given a matricized vector  $U = Y_{\kappa'} Z_{\kappa'}^T$  of rank  $\kappa'$ , a standard approach for truncation [5, 21] is to compute the singular value decomposition (SVD) of  $U$  and compress  $U$  into an approximation of desired

rank  $\kappa \ll \kappa'$ . This can be done efficiently by computing QR factorizations of  $Y_{\kappa'}$  and  $Z_{\kappa'}$ ,

$$Y_{\kappa'} = Q_Y R_Y \in \mathbb{R}^{n_x \times \kappa'}, \quad Z_{\kappa'} = Q_Z R_Z \in \mathbb{R}^{n_\xi \times \kappa'}.$$

Then, one can compute the SVD of  $R_Y R_Z^T$ ,

$$R_Y R_Z^T = \hat{U}_{\kappa'} \hat{\Sigma}_{\kappa'} \hat{V}_{\kappa'}^T = \sum_{k=1}^{\kappa'} \hat{\sigma}_k \hat{u}_k \hat{v}_k^T,$$

and truncate the sum with  $\kappa$  terms to produce

$$\tilde{Y}_\kappa = Q_Y \hat{U}_\kappa \hat{\Sigma}_\kappa \in \mathbb{R}^{n_x \times \kappa}, \quad \tilde{Z}_\kappa = Q_Z \hat{V}_\kappa \in \mathbb{R}^{n_\xi \times \kappa}.$$

The truncated approximation of  $U$  is then  $\tilde{U} = \tilde{Y}_\kappa \tilde{Z}_\kappa^T$ . The computational complexity of the truncation is  $O((n_x + n_\xi + \kappa)(\kappa')^2)$  [15], which grows quadratically with respect to  $\kappa'$ . In the next section, we introduce a new truncation method that avoids this computation.

**4.2. Truncation based on coarse-grid rank reduction.** We now propose a coarse-grid rank-reduction strategy. We obtain insight into the rank structure of the solution by using a coarse spatial grid computation. Then, we define a truncation operator based on the information obtained from this coarse-grid computation.

Let  $u^c(\mathbf{x}, \xi)$  represent a solution obtained on a coarse spatial grid (i.e.,  $n_x$  is small). As in (2.13),  $u^c(\mathbf{x}, \xi)$  can be represented as

$$u^c(\mathbf{x}, \xi) = (\Phi^c(\mathbf{x}))^T U^c \Psi(\xi) = ((Y^c)^T \Phi^c(\mathbf{x}))^T ((Z^c)^T \Psi(\xi)).$$

Here, we propose using  $Z^c$  to define a truncation operator for use in the projection method to compute a solution for the problem on a finer grid. That is, the truncation operator  $\mathcal{T}_\kappa$  is defined such that, given a matricized vector  $U = Y_{\kappa'} Z_{\kappa'}^T$  of rank  $\kappa'$ ,

$$(4.1) \quad \mathcal{T}_\kappa(U) \equiv (Y_{\kappa'} Z_{\kappa'}^T Z_\kappa^c) (Z_\kappa^c)^T = \tilde{U},$$

where the resulting quantity  $\tilde{U} = \tilde{Y}_\kappa \tilde{Z}_\kappa^T$  is of rank  $\kappa$ ,

$$\tilde{Y}_\kappa = Y_{\kappa'} Z_{\kappa'}^T Z_\kappa^c \in \mathbb{R}^{n_x \times \kappa}, \quad \tilde{Z}_\kappa = Z_\kappa^c \in \mathbb{R}^{n_\xi \times \kappa}.$$

The desired rank  $\kappa$  is determined such that the relative residual  $\|f^c - A^c u^{c, \kappa}\|_2 / \|f^c\|_2$  is smaller than a certain tolerance  $\epsilon^c$ , where  $u^{c, \kappa}$  is a  $\kappa$ -term approximation of  $u^c$ . This truncation operation requires two matrix-matrix products, and the computational complexity of truncating a vector from  $\kappa'$  to  $\kappa$  is  $O(\kappa' \kappa (n_x + n_\xi))$ . Note that with the proposed truncation strategy, the fine-grid computation is equivalent to applying GMRES to  $\sum_{i=0}^M K_i U G_i Z_\kappa^c (Z_\kappa^c)^T = f_0 g_0^T$ .

For efficient coarse-grid computation, we use the proper generalized decomposition (PGD) method developed in [22, 31], which computes a separated representation of a coarse-grid solution,

$$(4.2) \quad u^{c, \kappa}(\mathbf{x}, \xi) = \sum_{i=1}^{\kappa} \tilde{y}_i(\mathbf{x}) \tilde{z}_i(\xi).$$



With the stochastic Galerkin discretization, each function can be represented as

$$\tilde{y}_i(\mathbf{x}) = \sum_{k=1}^{n_x} \tilde{y}_k^{(i)} \phi_k^c(\mathbf{x}), \quad \tilde{z}_i(\xi) = \sum_{l=1}^{n_\xi} \tilde{z}_l^{(i)} \psi_l(\xi).$$

As a result, as in (2.13),

$$u^{c, \kappa}(\mathbf{x}, \xi) = \left( (\tilde{Y}_\kappa^c)^T \Phi^c(\mathbf{x}) \right)^T \left( (\tilde{Z}_\kappa^c)^T \Psi(\xi) \right),$$

where  $\tilde{Y}_\kappa^c = [\tilde{y}^{(1)}, \dots, \tilde{y}^{(\kappa)}] \in \mathbb{R}^{n_x \times \kappa}$  and  $\tilde{Z}_\kappa^c = [\tilde{z}^{(1)}, \dots, \tilde{z}^{(\kappa)}] \in \mathbb{R}^{n_\xi \times \kappa}$  are coefficient matrices such that the  $i$ th elements of  $\tilde{y}^{(j)}$  and  $\tilde{z}^{(j)}$  are  $\tilde{y}_i^{(j)}$  and  $\tilde{z}_i^{(j)}$ , respectively. Now, the discrete solution  $U^c$  in (4.2) is approximated by  $U^{c, \kappa} = \tilde{Y}_\kappa^c (\tilde{Z}_\kappa^c)^T$ , and we can obtain  $Z_\kappa^c$  by computing the SVD of  $U^{c, \kappa} = \hat{U} \hat{\Sigma} \hat{V}^T$ , and, as a result,  $Z_\kappa^c = \hat{V}$ . We briefly explain how the PGD method computes a  $\kappa$ -term approximation in the next section.

**4.3. Proper generalized decomposition method.** The PGD method is a successive rank-one approximation method. That is, the method incrementally identifies the function pairs  $(\tilde{y}_i(\mathbf{x}), \tilde{z}_i(\xi))$  of (4.2) one at a time. Once  $i$  such pairs have been computed, the next pair  $(\tilde{y}_{i+1}, \tilde{z}_{i+1})$  is sought in  $X_h \times S_M$  by imposing Galerkin orthogonality with respect to the tangent manifold of the set of rank-one elements at  $\tilde{y}_{i+1} \tilde{z}_{i+1}$ , which is  $\{\tilde{y}_{i+1} \zeta + v \tilde{z}_{i+1}; v \in X_h, \zeta \in S_M\}$ : find  $\tilde{y}_{i+1} \tilde{z}_{i+1}$  such that for all  $(v, \zeta) \in X_h \times S_M$ ,

$$(4.3) \quad \left\langle \int_D a(\mathbf{x}, \xi) \nabla(u^{c, i} + \tilde{y}_{i+1} \tilde{z}_{i+1}) \cdot \nabla(\tilde{y}_{i+1} \zeta + v \tilde{z}_{i+1}) \right\rangle = \left\langle \int_D f(\tilde{y}_{i+1} \zeta + v \tilde{z}_{i+1}) \right\rangle.$$

It follows from (4.3) that each component of a pair  $(\tilde{y}_{i+1}, \tilde{z}_{i+1})$  can be computed by solving two coupled problems: a deterministic problem (4.4) and a stochastic problem (4.5). The deterministic problem is as follows: given  $\tilde{z}_{i+1}$ , find  $\tilde{y}_{i+1} \in X_h$  such that

$$(4.4) \quad \left\langle \int_D a(\mathbf{x}, \xi) \nabla(u^{c, i} + \tilde{y}_{i+1} \tilde{z}_{i+1}) \cdot \nabla(\phi_j^c \tilde{z}_{i+1}) \right\rangle = \left\langle \int_D f \phi_j^c \tilde{z}_{i+1} \right\rangle, \quad j = 1, \dots, n_x^c.$$

The first basis function  $\tilde{z}_1$  can be chosen arbitrarily at the beginning of the PGD method. The finite element discretization of  $u_{i+1}$  yields a linear system of order  $n_x^c$ . Analogously, the stochastic problem starts with  $\tilde{y}_{i+1}$  and finds  $\tilde{z}_{i+1} \in S_M$  such that

$$(4.5) \quad \left\langle \int_D a(\mathbf{x}, \xi) \nabla(u^{c, i} + \tilde{y}_{i+1} \tilde{z}_{i+1}) \cdot \nabla(\tilde{y}_{i+1} \psi_j) \right\rangle = \left\langle \int_D f \tilde{y}_{i+1} \psi_j \right\rangle, \quad j = 1, \dots, n_\xi.$$

Since  $\tilde{z}_{i+1}$  is approximated by the PCE,  $n_\xi$  unknowns have to be determined by solving a linear system of order  $n_\xi$ .

Solutions of these sets of  $\kappa$  systems of order  $n_x^c$  and  $\kappa$  systems of order  $n_\xi$  produce the  $\kappa$ -term approximation to the solution. The PGD method seeks solution pairs until the relative residual of the computed solution satisfies a given tolerance,

$$(4.6) \quad \|f^c - A^c u^{c, \kappa}\|_2 / \|f^c\|_2 < \epsilon^c.$$

The accuracy of the  $\kappa$ -term approximation can also be improved by solving a set of  $\kappa$  coupled equations: given  $\{\tilde{y}_i\}_{i=1}^\kappa$ , find  $\{\tilde{z}_i\}_{i=1}^\kappa$  such that

$$(4.7) \quad \left\langle \int_D a(\mathbf{x}, \xi) \nabla(u^{(\kappa)}) \cdot \nabla(\tilde{y}_i \psi_j) \right\rangle = \left\langle \int_D f \tilde{y}_i \psi_j \right\rangle, \quad i = 1, \dots, \kappa, j = 1, \dots, n_\xi.$$

This update requires the solution of a linear system of order  $\kappa n_\xi$ . For the stochastic diffusion problems, the update problem is solved once at the end of the PGD method. Note that the update problem could also be formulated for finding the deterministic parts  $\{u_i\}_{i=1}^\kappa$  if  $n_x \ll n_\xi$ , which requires a solution of a linear system of order  $\kappa n_x$ .

With the proposed truncation strategy, Algorithm 3 summarizes the entire procedure to compute a solution on a finer grid.

---

**Algorithm 3** Preconditioned LRP method with the coarse-grid rank reduction.

---

- 1: Compute  $u^{c, \kappa}$  that satisfies  $\frac{\|f^c - A^c u^{c, \kappa}\|_2}{\|f^c\|_2} < \epsilon^c$  using the PGD method
  - 2: Compute  $Z_\kappa^c$  such that  $U^{c, \kappa} = Y_\kappa^c (Z_\kappa^c)^T$  and define  $\mathcal{T}_\kappa(U) \equiv (U Z_\kappa^c) (Z_\kappa^c)^T$
  - 3: Run Algorithm 2 with  $\mathcal{L} = AM^{-1}$ ,  $f$ , and  $\mathcal{T}_\kappa$
- 

**5. Numerical experiments.** In this section, we present the results of numerical experiments in which the proposed iterative solver is applied to some benchmark problems. The implementation of the spatial discretization is based on the incompressible flow and iterative solver software (IFISS) package [30]. Example problems are posed on a square domain, and  $\ell$  is the spatial discretization parameter (i.e.,  $n_x = (2^\ell + 1)^2$ ).

For  $a(\mathbf{x}, \xi)$  in (2.5), we consider independent random variables  $\{\xi_i\}_{i=1}^M$  that are uniformly distributed over  $[-\sqrt{3}, \sqrt{3}]$ ,  $a_0 = 1$ , and, unless otherwise specified,  $\sigma = 0.05$ . As the covariance kernel, we use

$$(5.1) \quad C(\mathbf{x}, \mathbf{y}) = \sigma^2 \exp\left(-\frac{|x_1 - y_1|}{\gamma} - \frac{|x_2 - y_2|}{\gamma}\right),$$

where  $\gamma$  is the correlation length. The number of terms  $M$  in the truncated expansion (2.5) is determined such that 95% of the total variance is captured by  $M$  terms (i.e.,  $(\sum_{i=1}^M \lambda_i) / (\sum_{i=1}^{n_x} \lambda_i) > 0.95$ ). We use bilinear  $Q_1$  elements to generate the finite element basis and Legendre polynomials as the stochastic basis functions because the underlying random variables have a uniform distribution. The default setting of the maximal polynomial degree  $p$  is 3.

**5.1. Stochastic diffusion problem.** We consider the steady-state stochastic diffusion equation in (2.6) on a domain  $D = [0, 1] \times [0, 1]$  with forcing term  $f(\mathbf{x}) = 1$  and homogeneous Dirichlet boundary conditions  $u(\mathbf{x}, \omega) = 0$  on  $\partial D \times \Gamma$ .

**Coarse spatial grid computation.** We compute  $\kappa$ -term approximations using the PGD method on a coarser spatial grid. Here  $\ell^c$  is the refinement level for the coarse grid, and  $n_x^c$  is the number of degrees of freedom in the corresponding spatial domain excluding boundary nodes. We discuss choices of coarse spatial grid in section 5.3. Table 1 shows the rank  $\kappa$  of solutions that satisfy the tolerance  $\epsilon^c$  for varying correlation lengths  $\gamma$  and  $M$  and the computation time  $t_c$ . In PGD, the linear systems arising from (4.4), (4.5), and (4.7) are solved using the MATLAB backslash operator.

**Fine spatial grid computation.** With the truncation operator  $\mathcal{T}_\kappa$  from (4.1) obtained from the coarse-grid solution (i.e.,  $Z_\kappa^c$ ), we solve the same stochastic diffusion problems on finer spatial grids  $\ell = \{7, 8, 9\}$ . For the fine-grid low-rank solutions,

TABLE 1

Rank ( $\kappa$ ) of coarse-grid solutions satisfying  $\epsilon^c$  of (4.6), and CPU time ( $t_c$ ) for coarse-grid computation using the PGD method, for varying  $\gamma$  and  $M$ .

$\gamma$	$\epsilon^c = 10^{-5}$				$\epsilon^c = 10^{-6}$			
	4	3	2.5	2	4	3	2.5	2
$M, n_\xi$	5, 56	7, 120	10, 286	15, 816	5, 56	7, 120	10, 286	15, 816
$n_x^c(\ell^c)$	225(4)	225(4)	961(5)	961(5)	225(4)	225(4)	961(5)	961(5)
Rank( $\kappa$ )	25	40	65	115	35	65	100	210
CPU time( $t_c$ )	2.49	3.47	8.35	45.08	2.93	5.04	14.83	162.71

TABLE 2

CPU times to compute low-rank solutions of the diffusion equation for  $\epsilon^c = \epsilon = 10^{-5}$  using the preconditioned LRP method. Numbers of GMRES cycles are shown in parentheses.

$n_x(\ell)$		$M = 5$	$M = 7$	$M = 10$	$M = 15$	$t_{setup}$
$129^2$	$t_f$	4.12 (1)	7.22 (1)	18.79 (1)	86.29 (1)	1.76
(7)	$t$	8.35	12.43	28.88	132.15	
$257^2$	$t_f$	12.55 (1)	24.70 (1)	74.71 (1)	330.45 (1)	10.16
(8)	$t$	25.17	38.37	93.20	385.59	
$513^2$	$t_f$	92.83 (1)	102.42 (1)	353.07 (1)	2717.03 (1)	92.41
(9)	$t$	147.17	197.87	453.71	2854.62	

we use the rank  $\kappa$  obtained from the coarse-grid solutions. For example, the third column of Table 2 shows the time required to find solutions of rank 25 satisfying the relative residual tolerance  $10^{-5}$  when the number of terms in (2.5) is  $M = 5$ . In Algorithm 2, we set  $m = 8$  (like restarted GMRES(8)). In examining performance, we identify the number of cycles,  $k$ , performed for the outer for-loop in Algorithm 2; this means that the number of matrix-vector products (i.e., the number of times line 10 is executed) is  $mk$ . Tables 2 and 3 show the number of cycles,  $k$ , and the computation time in seconds needed to compute approximate solutions with  $\epsilon = 10^{-5}$  and  $10^{-6}$ , respectively (see line 4 of Algorithm 2). Here,  $t$  is the total time and  $t_f$  excludes the time to compute the coarse-grid solution,  $t_c$ . The fine-grid computation time,  $t_f$ , consists of algorithm execution time and preconditioner setup time,  $t_{setup}$ . The execution times show “textbook” behavior; i.e., they grow linearly with the size of the spatial grid.<sup>1</sup> Note that the computational cost for the coarse-grid computation becomes negligible as the size of the problem becomes larger. If the required memory for running Algorithm 2 exceeds the resources of our computing environment, solutions cannot be computed, and we denote these cases by OoM for “Out-of-Memory.” Table 4 shows the number of degrees of freedom of the fine spatial grid problems for varying stochastic dimensions,  $M$ .

In these experiments and in all those described below, we used  $\epsilon^c = \epsilon$  (the stopping tolerance specified in line 4 of Algorithm 2), and for this choice, the solver always satisfied the stopping criterion. We also tested both larger  $\epsilon^c$  and smaller  $\epsilon^c$ . For  $\epsilon^c > \epsilon$ , the solver sometimes failed to satisfy the stopping criterion. For  $\epsilon^c < \epsilon$ , the solver was robust but consistently more expensive.

**Example problems with varying  $\sigma$  and  $p$ .** We examine the rank structure of

<sup>1</sup>An exception to this statement is when both  $M$  and  $n_x$  are large. For these cases, the problem does not fit into physical memory, and memory swap-in/out time dominates the execution time.

TABLE 3

CPU times to compute low-rank solutions of the diffusion equation for  $\epsilon^c = \epsilon = 10^{-6}$  using the preconditioned LRP method. Numbers of GMRES cycles are shown in parentheses.

$n_x(\ell)$		$M = 5$	$M = 7$	$M = 10$	$M = 15$	$t_{setup}$
$129^2$ (7)	$t_f$	5.40 (1)	12.50 (1)	35.09 (1)	233.54 (1)	1.79
	$t$	10.14	19.32	51.69	398.06	
$257^2$ (8)	$t_f$	17.23 (1)	46.07 (1)	137.19 (1)	1004.40 (1)	10.53
	$t$	30.55	61.41	162.90	1177.68	
$513^2$ (9)	$t_f$	70.37 (1)	217.12 (1)	1225.77 (1)	OoM	92.81
	$t$	166.24	315.18	1333.63	OoM	

TABLE 4

Number of degrees of freedom of the fine-grid discretizations with  $p = 3$  for varying spatial grid refinement level,  $\ell$ , and number of random variables,  $M$ .

$\ell$	$M = 5$	$M = 7$	$M = 10$	$M = 15$
7	931,896	1,996,920	4,759,326	13,579,056
8	3,698,744	7,925,880	18,890,014	53,895,984
9	14,737,464	31,580,280	75,266,334	214,745,904

the numerical solutions of the stochastic diffusion problems and assess the performance of the proposed solution algorithm for different values of maximal degree of stochastic polynomial  $p$  in (2.8) and variance  $\sigma^2$  of the random field  $a(\mathbf{x}, \xi)$ . As in the previous numerical experiments, we first identify the rank structure and define the truncation operator from coarse-grid computation. Then we solve the same problems on a finer grid by using the proposed LRP method with the coarse-grid rank-reduction scheme.

Table 5 shows the computation time needed to compute approximate solutions of the stochastic diffusion problems with  $M = 7$  for varying maximal polynomial degree  $p$ . The required ranks of the approximate solutions are not affected by the number of terms in the polynomial expansion. However, the computation time is increased for the polynomial expansion with higher maximal polynomial degree because the size of  $\{G_i\}_{i=0}^M$  and the size of the stochastic part of the solution becomes larger as the number of terms in the PCE is increased.

Table 6 shows the computation time  $t$  needed to compute approximate solutions of the stochastic diffusion problems that satisfy tolerances  $10^{-5}$  and  $10^{-6}$  for varying variance,  $\sigma^2$ . In general, the example problem with a larger variance requires a higher rank to satisfy the stopping tolerance, which, therefore, requires more computational effort.

**Comparison to a truncation operator based on singular values.** We compare the performance of the proposed solver to the preconditioned LRP method combined with the conventional truncation operator from [19]. Table 7 shows the computation time required to compute approximate solutions using the conventional and new truncation strategies. The total computation time,  $t$ , of the LRP method with the coarse-grid rank reduction includes both coarse-grid,  $t_c$ , and fine-grid,  $t_f$ , computations. The LRP method with the SVD-based truncation operator, which is implemented based on [5], does not require a coarse-grid computation and can start with any arbitrary initial guess for rank  $\kappa$ . For these computations, we used the values of rank identified in the coarse-grid computations, which are illustrated in Table 1, for the initial rank.

TABLE 5

CPU time  $t$  to compute low-rank solutions of the diffusion equation for  $\epsilon^c = \epsilon = 10^{-5}$  and  $10^{-6}$  using the preconditioned LRP method for varying maximal polynomial degree  $p$  (stochastic degrees of freedom,  $n_\xi$ , are shown in parentheses).

$n_x(\ell)$	$\epsilon^c = \epsilon = 10^{-5} (\kappa = 40)$			$\epsilon^c = \epsilon = 10^{-6} (\kappa = 65)$		
	$p = 3$ (120)	$p = 4$ (330)	$p = 5$ (792)	$p = 3$ (120)	$p = 4$ (330)	$p = 5$ (792)
$129^2(7)$	12.43	15.55	21.56	19.32	23.42	38.49
$257^2(8)$	38.37	44.27	56.79	61.41	69.17	91.10
$513^2(9)$	197.87	217.38	252.39	315.18	322.86	383.89

TABLE 6

CPU time  $t$  and rank  $\kappa$  to compute low-rank solutions of the diffusion equation for  $\epsilon^c = \epsilon = 10^{-5}$  and  $10^{-6}$  using the preconditioned LRP method for varying  $\sigma$ .

$\sigma$	$n_x$	$\epsilon = 10^{-5}$				$\epsilon = 10^{-6}$			
		$M=5$	$M=7$	$M=10$	$M=15$	$M=5$	$M=7$	$M=10$	$M=15$
0.01		$\kappa = 15$	$\kappa = 20$	$\kappa = 35$	$\kappa = 55$	$\kappa = 20$	$\kappa = 30$	$\kappa = 50$	$\kappa = 85$
	$129^2$	7.28	8.65	15.01	45.69	7.87	10.81	20.76	83.07
	$257^2$	21.47	26.08	47.21	135.75	23.30	31.94	66.92	240.98
	$513^2$	130.93	150.85	236.34	922.87	137.98	173.03	333.70	1893.89
0.05		$\kappa = 25$	$\kappa = 40$	$\kappa = 65$	$\kappa = 115$	$\kappa = 35$	$\kappa = 65$	$\kappa = 100$	$\kappa = 210$
	$129^2$	8.35	12.43	28.88	132.15	10.14	19.32	51.69	398.06
	$257^2$	25.17	38.37	93.20	385.59	30.55	61.41	162.90	1177.68
	$513^2$	147.17	197.87	453.71	2854.62	166.24	315.18	1333.63	OoM
0.1		$\kappa = 35$	$\kappa = 60$	$\kappa = 100$	$\kappa = 180$	$\kappa = 50$	$\kappa = 85$	$\kappa = 145$	-
	$129^2$	9.78	17.24	50.70	297.35	8.79	28.37	113.53	OoM
	$257^2$	29.98	54.94	157.76	866.41	41.69	94.48	356.50	OoM
	$513^2$	164.48	273.33	1324.47	OoM	208.15	515.29	2902.95	OoM

**PGD as a solver on a finer spatial grid.** The PGD method could be applied directly to the fine-grid problems. We assess the performance of the PGD method for computing fine-grid solutions in Table 8, which shows the rank and computation time for computing approximate solutions that satisfy the tolerances  $10^{-5}$  and  $10^{-6}$  using PGD on a finer spatial grid. For the LRP method, we record total computation time,  $t$ , which includes coarse-grid computation,  $t_c$ ; AMG preconditioner setup,  $t_{setup}$ ; and fine-grid computation time,  $t_f$ . We compare the rank and the computation time for computing solutions using the PGD method and the proposed projection method. The proposed LRP method runs faster and requires somewhat smaller ranks than the PGD method.

*Remark.* We also tested the techniques compared in Tables 7 and 8 for different values of  $\sigma$ ,  $\sigma = 0.01$  and  $0.1$ , with similar results. Indeed, the performance of LRP-coarse is more favorable for the larger value  $\sigma = 0.1$ .

**5.2. Stochastic convection-diffusion problem.** For a second benchmark problem, we consider the steady-state convection-diffusion equation defined on  $D = [-1, 1] \times [-1, 1]$  with nonhomogeneous Dirichlet boundary conditions, constant vertical wind

TABLE 7

CPU times to compute low-rank solutions of the diffusion equation for  $\epsilon^c = \epsilon = 10^{-5}$  and  $10^{-6}$  using the preconditioned LRP methods with the coarse-grid rank reduction and the singular value-based truncation on the level 8 spatial grid (i.e.,  $n_x = 257^2$ ).

	Solver		M=5	M=7	M=10	M=15	M = 20
$\epsilon = 10^{-5}$	LRP-SVD	$t_{\text{SVD}}$	55.04	108.11	284.27	1280.65	5691.19
	LRP-coarse	$t$	25.17	38.37	93.20	385.59	1943.49
$\epsilon = 10^{-6}$	LRP-SVD	$t_{\text{SVD}}$	76.03	198.20	564.12	5131.32	OoM
	LRP-coarse	$t$	30.55	61.41	162.90	1177.68	OoM

TABLE 8

CPU times to compute low-rank solutions of the diffusion equation for  $\epsilon^c = \epsilon = 10^{-5}$  and  $10^{-6}$  using the PGD method and the preconditioned LRP methods on the level 8 spatial grid (i.e.,  $n_x = 257^2$ ).

	Solver		M = 5	M = 7	M = 10	M = 15	M = 20
$\epsilon = 10^{-5}$	PGD	$\kappa$	25	45	65	125	195
		$t$	43.78	109.72	228.73	940.69	3066.87
	LRP-coarse	$\kappa$	25	40	65	115	180
		$t$	25.17	38.37	93.20	385.59	1943.49
$\epsilon = 10^{-6}$	PGD	$\kappa$	40	70	110	225	OoM
		$t$	74.43	214.82	533.10	2713.70	OoM
	LRP-coarse	$\kappa$	35	65	100	210	OoM
		$t$	30.55	61.41	162.90	1177.68	OoM

$\vec{w} = (0, 1)$ , and  $f = 0$ ,

$$(5.2) \quad \begin{cases} \nu \nabla \cdot (a(\mathbf{x}, \xi) \nabla u(\mathbf{x}, \xi)) + \vec{w} \cdot \nabla u(\mathbf{x}, \xi) = f(\mathbf{x}, \xi) & \text{in } D \times \Gamma, \\ u(\mathbf{x}, \xi) = g_D(\mathbf{x}) & \text{on } \partial D \times \Gamma, \end{cases}$$

where  $g_D(\mathbf{x})$  is determined by

$$(5.3) \quad g_D(\mathbf{x}) = \begin{cases} g_D(x, -1) = x, & g_D(x, 1) = 0, \\ g_D(-1, y) = -1, & g_D(1, y) = 1, \end{cases}$$

where the latter two approximations hold except near  $y = 1$ , and  $\nu$  is the viscosity parameter. We consider the convection-dominated case (i.e.,  $\nu < 1$ ) and employ the streamline-diffusion method for stabilization [8]. Here, we define the element Péclet number

$$(5.4) \quad \mathcal{P}_k = \frac{\|\vec{w}_k\|_2 h_k}{2\nu},$$

where  $\|\vec{w}_k\|_2$  is the  $\ell_2$  norm of the wind at the element centroid and  $h_k$  is a measure of the element length in the direction of the wind. Note that the solution has an exponential boundary layer near  $y = 1$ , where the value of the solution dramatically changes essentially from  $-1$  to  $0$  on the left and from  $+1$  to  $0$  on the right [12]. Figure 1 illustrates the mean of solutions  $\langle u(\mathbf{x}, \xi) \rangle_\rho$  computed on the level 6 spatial grid and corresponding contour plots for varying viscosity parameter,  $\nu$ .

Given  $a(\mathbf{x}, \xi)$  in (2.5), we again discretize (5.2) using the finite element method

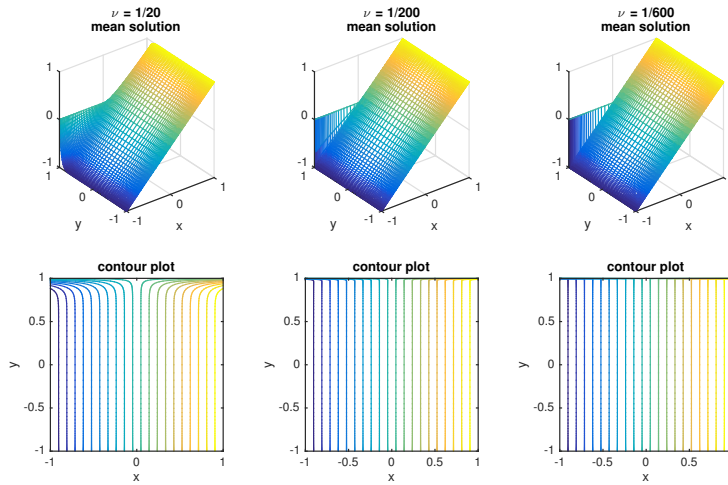


FIG. 1. Mean solutions and contour plots on the level 6 spatial grid for varying  $\nu$ .

and the generalized PCE. The result is a linear system in tensor product notation

$$(5.5) \quad \left( G_0 \otimes \nu K_0 + \sum_{l=1}^M G_l \otimes \nu K_l + G_0 \otimes N + G_0 \otimes S \right) u = g_0 \otimes f_0,$$

where the convection term  $N$  and the streamline-diffusion term  $S$  are given by

$$[N]_{ij} = \int_D \vec{w} \cdot \nabla \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) d\mathbf{x},$$

$$[S]_{ij} = \sum_{k=1}^{n_e} \delta_k \int_D (\vec{w} \cdot \nabla \phi_i)(\vec{w} \cdot \nabla \phi_j) d\mathbf{x},$$

$n_e$  is the number of elements in the finite element discretization, and

$$(5.6) \quad \delta_k = \begin{cases} \frac{h_k}{2\|\vec{w}\|_2} \left( 1 - \frac{1}{\mathcal{P}_k} \right) & \text{if } \mathcal{P}_k > 1, \\ 0 & \text{if } \mathcal{P}_k \leq 1. \end{cases}$$

As the preconditioner, we choose  $M \approx G_0 \otimes (K_0 + N + S)$ , where the action of  $(K_0 + N + S)^{-1}$  is replaced by application of a single V-cycle of an AMG method. In the PGD method, the nonhomogeneous Dirichlet boundary condition is handled by introducing an extended affine space [22]  $u^c \approx u_{bc} + u^{c,\kappa}$ , where  $u_{bc}$  is the boundary nodal function such as  $u_{bc} = \sum_{k \in \partial D} u_k^{(bc)} \phi_k(\mathbf{x})$ . For the stochastic convection-diffusion problems, the update problems (4.7) need to be solved more often to compute an approximate solution of a desired accuracy with fewer terms.

**Numerical results.** To cope with the existence of the exponential boundary layer in the solution, we use vertically stretched spatial grids. We examine the performance of the LRP method for varying viscosity parameter  $\nu$ , and we set  $m = 10$  for Algorithm 2. Tables 9 and 10 show  $\kappa$  computed by the PGD method, coarse-grid computation time  $t_c$ , and fine-grid computation time  $t_f$  to compute approximate solutions on fine spatial grids  $\ell = \{7, 8, 9\}$  satisfying  $10^{-5}$  and  $10^{-6}$ , respectively.

TABLE 9

CPU times to compute low-rank solutions of the convection-diffusion equation for  $\epsilon^c = \epsilon = 10^{-5}$  using the preconditioned LRP methods for varying  $\nu$ . Numbers of GMRES cycles are shown in parentheses.

$\nu$	$\ell$		$M = 5$	$M = 7$	$M = 10$	$M = 15$	$t_{setup}$
$\frac{1}{20}$	4	$\kappa$	25	35	55	65*	
		$t_c$	2.56	4.83	26.34	58.92*	
	<u>7</u>	$t_f$	5.73 (1)	9.47 (1)	24.86 (1)	72.29 (1)	6.14
	<u>8</u>	$t_f$	20.52 (1)	36.66 (1)	98.72 (1)	248.31 (1)	30.57
	<u>9</u>	$t_f$	84.55 (1)	152.69 (1)	592.63 (1)	1953.52 (2)	338.28
$\frac{1}{100}$	4	$\kappa$	20	25	45	55*	
		$t_c$	2.94	3.12	16.28	47.24*	
	7	$t_f$	5.06 (1)	7.28 (1)	18.90 (1)	60.66 (1)	6.34
	8	$t_f$	16.87 (1)	26.36 (1)	74.26 (1)	202.29 (1)	35.52
	<u>9</u>	$t_f$	121.98 (2)	201.62 (2)	745.92 (2)	3079.24 (2)	341.41
$\frac{1}{200}$	5	$\kappa$	20	25	45	50	
		$t_c$	2.91	4.79	16.54	46.85	
	7	$t_f$	5.16 (1)	7.21 (1)	16.57 (1)	53.97 (1)	6.35
	8	$t_f$	17.57 (1)	25.05 (1)	63.56 (1)	175.30 (1)	35.89
	9	$t_f$	123.73 (2)	200.10 (2)	605.50 (2)	2568.41 (2)	344.87
$\frac{1}{400}$	5	$\kappa$	20	20	35	45 <sup>†</sup>	
		$t_c$	2.94	3.79	12.49	82.06 <sup>†</sup>	
	7	$t_f$	8.61 (2)	9.84 (2)	26.97 (2)	85.01 (2)	6.09
	8	$t_f$	31.55 (2)	37.74 (2)	111.31 (2)	298.49 (2)	34.93
	9	$t_f$	133.45 (2)	158.01 (2)	512.88 (2)	2080.60 (2)	342.12
$\frac{1}{600}$	6	$\kappa$	20	20	35	45	
		$t_c$	9.79	13.20	34.47	94.79	
	7	$t_f$	8.27 (2)	10.07 (2)	26.91 (2)	82.30 (2)	6.14
	8	$t_f$	31.94 (2)	39.84 (2)	109.25 (2)	295.25 (2)	33.25
	9	$t_f$	343.80 (2)	163.90 (2)	506.42 (2)	1977.83 (2)	342.98

Underlined numbers in the spatial grid level indicate cases where streamline diffusion is not needed.

When the viscosity parameter is small (i.e.,  $\nu = 1/600$ ), the coarse-grid computation requires the  $\kappa$ -term approximation on a relatively fine spatial grid (i.e.,  $\ell = 6$ ). The exponential boundary layer gets narrower as the viscosity parameter gets smaller, which requires the use of a finer spatial grid for the coarse-grid computation. If the coarse-grid computation is performed on coarser spatial grids, it fails to identify the rank structure of solutions and to yield a proper truncation operator. Analogously, when the number of terms,  $M$ , in the KL expansion (2.5) is large, the coarse-grid computation has to be done on a relatively fine spatial grid because the KL expansion contains more spatially oscillatory terms. In the last columns of Table 9 and 10, \* and † indicate that the coarse-grid solutions are computed on the level 5 and the level 6 spatial grid, respectively.

**Comparison to a truncation operator based on singular values.** We again compare the performance of the proposed solver to the preconditioned LRP method combined with the conventional truncation operator, i.e., the SVD-based truncation operator. Table 11 shows the computation time required to compute approximate



TABLE 10

CPU times to compute approximate solutions of the convection-diffusion equation for  $\epsilon^c = \epsilon = 10^{-6}$  using the preconditioned LRP methods for varying  $\nu$ . Numbers of GMRES cycles are shown in parentheses.

$\nu$	$\ell$		$M = 5$	$M = 7$	$M = 10$	$M = 15$	$t_{setup}$
$\frac{1}{20}$	4	$\kappa$	35	50	75	105*	
		$t_c$	3.31	9.17	60.51	194.33*	
	<u>7</u>	$t_f$	13.92 (2)	27.47 (2)	80.78 (2)	275.96 (2)	6.14
	<u>8</u>	$t_f$	52.45 (2)	106.11 (2)	311.59 (2)	1042.40 (2)	30.57
	<u>9</u>	$t_f$	220.67 (2)	534.61 (2)	2694.26 (2)	8101.20 (2)	338.28
$\frac{1}{100}$	4	$\kappa$	30	40	65	95*	
		$t_c$	2.83	6.25	38.39	155.83*	
	7	$t_f$	12.34 (2)	21.28 (2)	65.02 (2)	239.91 (2)	6.34
	8	$t_f$	46.67 (2)	85.66 (2)	255.79 (2)	895.81 (2)	35.52
	<u>9</u>	$t_f$	273.45 (3)	549.82 (3)	3069.96 (3)	10963.03 (3)	341.41
$\frac{1}{200}$	5	$\kappa$	25	40	60	85	
		$t_c$	3.46	8.57	38.35	122.49	
	7	$t_f$	10.52 (2)	21.43 (2)	56.36 (2)	204.09 (2)	6.35
	8	$t_f$	39.39 (2)	84.14 (2)	219.36 (2)	732.88 (2)	35.89
	<u>9</u>	$t_f$	226.83 (3)	547.62 (3)	2627.98 (3)	9284.60 (3)	344.87
$\frac{1}{400}$	5	$\kappa$	25	35	55	75 <sup>†</sup>	
		$t_c$	3.49	6.63	30.50	151.46 <sup>†</sup>	
	7	$t_f$	10.44 (2)	17.96 (2)	50.96 (2)	161.58 (2)	6.09
	8	$t_f$	40.02 (2)	70.82 (2)	204.71 (2)	610.23 (2)	34.93
	<u>9</u>	$t_f$	239.04 (3)	441.73 (3)	2106.30 (3)	7817.82 (3)	342.12
$\frac{1}{600}$	6	$\kappa$	30	35	45	65	
		$t_c$	17.99	22.03	47.44	140.01	
	7	$t_f$	17.74 (3)	26.56 (3)	56.25 (3)	281.27 (3)	6.14
	8	$t_f$	48.39 (2)	74.40 (2)	153.35 (2)	506.84 (2)	33.25
	<u>9</u>	$t_f$	281.27 (3)	462.52 (3)	1184.74 (3)	6261.34 (3)	342.98

solutions using the conventional and the new truncation strategy. When the LRP method with SVD-based truncation operator is used, initial values for rank  $\kappa$  in Algorithm 2 are obtained from coarse-grid computations of the proposed rank-reduction strategy.

**5.3. Choices of coarse spatial grid.** Finally, we discuss criteria for choosing the coarse grid used to generate truncation operators. The basic idea is that the coarse grid needs to be fine enough so that important features of the problem are represented. This quality is problem dependent, and we outline what is needed for the two types of problems we examined.

First, consider the diffusion equation of section 5.1. The issue is the oscillatory nature of components of the random field  $a(\mathbf{x}, \xi)$ . In the KL expansion (2.5), the eigenpairs,  $\{(\lambda_i, a_i(\mathbf{x}))\}_{i=1}^M$ , can be obtained by solving the integral equation

$$(5.7) \quad \int_D C(\mathbf{x}, \mathbf{y}) a_i(\mathbf{y}) d\mathbf{y} = \lambda_i a_i(\mathbf{x}), \quad i = 1, \dots, M,$$

where  $C(\mathbf{x}, \mathbf{y})$  is the covariance kernel (5.1). Since the kernel is separable, the eigenfunctions of the integral problem (5.7) can be represented as  $a_i(\mathbf{x}) = a_k^1(x_1) a_j^2(x_2)$ ,

TABLE 11

CPU times to compute low-rank solutions of the convection-diffusion equation for  $\epsilon^c = \epsilon = 10^{-5}$  and  $10^{-6}$  using the preconditioned LRP methods with coarse-grid rank reduction and the singular value-based truncation on the level 8 spatial grid (i.e.,  $n_x = 257^2$ ).

	Viscosity ( $\nu$ )	Solver		$M = 5$	$M = 7$	$M = 10$	$M = 15$	
$\epsilon = 10^{-5}$	1/20	LRP-SVD	$t_{\text{SVD}}$	68.45	100.83	201.34	438.25	
		LRP-coarse	$t$	54.06	72.08	154.79	338.21	
	1/100	LRP-SVD	$t_{\text{SVD}}$	93.91	121.89	295.27	655.71	
		LRP-coarse	$t$	55.28	64.36	125.88	285.94	
	1/200	LRP-SVD	$t_{\text{SVD}}$	90.70	122.56	251.60	574.68	
		LRP-coarse	$t$	55.42	66.08	115.68	258.97	
	1/400	LRP-SVD	$t_{\text{SVD}}$	91.11	107.47	221.32	475.60	
		LRP-coarse	$t$	69.01	76.63	158.07	416.36	
	1/600	LRP-SVD	$t_{\text{SVD}}$	90.33	103.44	218.35	484.08	
		LRP-coarse	$t$	75.26	86.48	176.93	422.85	
	$\epsilon = 10^{-6}$	1/20	LRP-SVD	$t_{\text{SVD}}$	132.08	234.15	570.56	1748.43
			LRP-coarse	$t$	86.74	145.86	401.83	1267.71
1/100		LRP-SVD	$t_{\text{SVD}}$	121.88	196.66	471.11	1479.80	
		LRP-coarse	$t$	84.97	126.77	329.52	1088.05	
1/200		LRP-SVD	$t_{\text{SVD}}$	106.79	188.76	416.52	1203.78	
		LRP-coarse	$t$	77.79	128.96	293.30	892.18	
1/400		LRP-SVD	$t_{\text{SVD}}$	107.12	168.01	380.01	1015.88	
		LRP-coarse	$t$	78.04	112.55	269.48	797.50	
1/600		LRP-SVD	$t_{\text{SVD}}$	122.44	231.07	421.76	1208.88	
		LRP-coarse	$t$	97.00	129.87	234.00	670.90	

where  $\{a_k^1\}_{k=1}^\infty$  and  $\{a_j^2\}_{j=1}^\infty$  are the eigenfunctions of the one-dimensional integral problem (i.e.,  $\int_D \exp(-|x_l - y_l|/\gamma) a_k^l(y_l) dy_l = \lambda_k^l a_k^l(x_l)$ ,  $l = 1, 2$ ). The eigenvalues,  $\{\lambda_i\}_{i=1}^M$ , are in decreasing order, and  $\lambda_i$  is the  $i$ th largest value of products  $\lambda_k^1 \lambda_j^2$  for  $k, j = 1, 2, \dots$ . Analytic expressions for the one-dimensional eigenfunctions are given in [14] as, for  $l = 1, 2$ ,

$$\begin{aligned}
 (5.8) \quad a_k^l(x) &= \cos(\theta_k x) / \sqrt{\frac{1}{2} + \frac{\sin \theta_k}{2\theta_k}} && \text{for even } k, \\
 a_k^{l*}(x) &= \sin(\theta_k^* x) / \sqrt{\frac{1}{2} - \frac{\sin \theta_k}{2\theta_k}} && \text{for odd } k,
 \end{aligned}$$

where  $\theta_k$  and  $\theta_k^*$  are the solutions of

$$\frac{1}{c} - \theta \tan\left(\frac{\theta}{2}\right) = 0 \quad \text{and} \quad \theta^* + \frac{1}{c} \tan\left(\frac{\theta^*}{2}\right) = 0,$$

respectively, when the one-dimensional integral problem is posed on  $[-\frac{1}{2}, \frac{1}{2}]$ . As  $i$  in the KL expansion (2.5) increases, the eigenfunctions  $a_i(\mathbf{x})$  become more oscillatory over the spatial domain (i.e.,  $\theta_k$  or  $\theta_k^*$  becomes larger), so that finer coarse spatial grids are required to capture the oscillatory features of the KL expansion. Table 12 shows the largest value of  $\{\theta_k, \theta_k^*\}$  of the eigenfunctions in the KL expansion, the half-wavelength of the functions from (5.8), and our choice of coarse spatial grid refinement

TABLE 12

Largest values of  $\theta_k$  or  $\theta_k^*$  of eigenfunctions (5.8) in the KL expansion, required grid refinement level  $\ell^c$ , half-wavelength  $\pi/\theta$ , and element size  $h^c = 2^{-\ell^c}$  for different values of  $M$ .

$M$	3	5	7	10	15	20
$\max(\theta_k, \theta_k^*)$	3.25	6.36	9.49	12.63	18.90	25.19
wavelength/2	.97	.49	.33	.25	.17	.12
$\ell^c (h^c)$	3 ( $\frac{1}{8}$ )	4 ( $\frac{1}{16}$ )	4 ( $\frac{1}{16}$ )	5 ( $\frac{1}{32}$ )	5 ( $\frac{1}{32}$ )	6 ( $\frac{1}{64}$ )

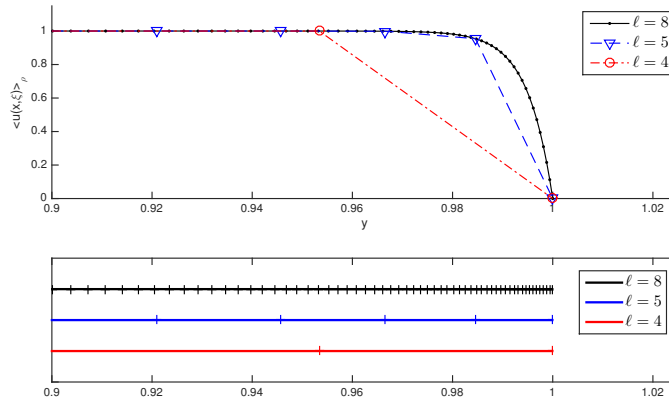


FIG. 2. Mean solutions  $\langle u(\mathbf{x}, \xi) \rangle_\rho$  at  $x = 1$  and  $y = [0.9, 1]$  illustrating the exponential boundary layer for varying spatial grid refinement level  $\ell = \{4, 5, 8\}$  (top), and lengths in the  $y$ -direction of the first few elements from  $y = 1$  (bottom).

levels,  $\ell^c$ , for different values of  $M$ . With these coarse grids, there are approximately eight grid points per half-wave, enough to capture the qualitative character of the wave.

We turn now to the convection-diffusion equation of section 5.2. This problem has the same diffusion coefficient (2.5) as the diffusion problem, but in addition its solution has an exponential boundary layer. In particular, for small  $\nu$ , the width of the layer is smaller than the finest interval needed to represent the eigenfunctions in (2.5), and in this case the coarse grid must be finer than that needed for the diffusion problem (whose solution is smooth). In Figure 2, the top plot illustrates the mean solutions  $\langle u(\mathbf{x}, \xi) \rangle_\rho$  of the weak formulation of (5.2) at  $x = 1$ , which are computed on two coarse spatial grids  $\ell = \{4, 5\}$  using PGD and a fine spatial grid  $\ell = 8$  using the proposed method, with viscosity parameter  $\nu = \frac{1}{200}$  and  $M = 10$  random variables. The bottom plot shows the lengths of the first few elements in the  $y$ -direction near  $y = 1$  for these refinement levels. If the level-4 spatial grid is used for the coarse-grid computation (i.e.,  $\ell = 4$ , bottom line in Figure 2), the width of the exponential boundary layer is much narrower than the length of the smallest element, and the coarse-grid solution gives a poor representation of the boundary layer. When this coarse grid is used to construct the truncation operator, the proposed scheme fails to compute an accurate approximate solution on a fine spatial grid (i.e.,  $\ell = 8$ , top line in Figure 2). On the other hand, the level-5 spatial grid (i.e.,  $\ell = 5$ , middle line in Figure 2) is fine enough for the coarse-grid solution to represent the character of the

exponential layer, and with this coarse grid, the resulting proposed scheme efficiently computes an accurate fine-grid solution.

Although this discussion shows that some a priori knowledge of the problem is needed to identify the coarse-grid operator, in general this information is not difficult to come by. In particular, we are assuming that the expansion (2.5) is known, and it is straightforward to identify the resolution needed to represent its components, for example, by examining one-dimensional cross-sections of them. If, as for the convection-diffusion problem, some knowledge of the solution is needed, this can be obtained cheaply from the solution of a deterministic problem derived from the mean of the diffusion coefficient; indeed, for the convection-diffusion problem, the boundary layer for the deterministic solution has essentially the same character as that of the stochastic solution, whose mean is shown in Figure 2.

**6. Statistical computation.** In this section, we explore the impact of truncation on statistical quantities associated with the solutions. In particular, we examine the mean and the variance of the solution  $u_{hp}(x, \xi)$ , which are defined as

$$(6.1) \quad \mu = E[u_{hp}], \quad \sigma_u^2 = E[(u_{hp} - \mu)^2],$$

where  $E[\cdot] = \int_{\Gamma} \cdot \rho(\xi) d\xi$  refers to the expectation. Let  $u_{hp}^{(\text{full})}$  refer to the discrete solution (of form (2.7)) obtained from a full-rank solution of (2.9) (i.e., with no truncation), and let  $u_{hp}^{(\text{low})}$  refer to that obtained using Algorithm 3. We will examine the accuracy of  $u_{hp}^{(\text{low})}$  by comparing its mean and variance to those of a reference solution  $u_{hp}^{(\text{ref})}$  as follows:

$$(6.2) \quad \eta_{\mu} \equiv \|\mu_{\text{ref}} - \mu_{\text{low}}\|_2 \leq \|\mu_{\text{ref}} - \mu_{\text{full}}\|_2 + \|\mu_{\text{full}} - \mu_{\text{low}}\|_2,$$

$$(6.3) \quad \eta_{\sigma} \equiv \|\sigma_{u,\text{ref}}^2 - \sigma_{u,\text{low}}^2\|_2 \leq \|\sigma_{u,\text{ref}}^2 - \sigma_{u,\text{full}}^2\|_2 + \|\sigma_{u,\text{full}}^2 - \sigma_{u,\text{low}}^2\|_2,$$

where the norm in (6.2)–(6.3) is the  $\ell_2$  norm (e.g.,  $\|\mu\|_2 = (\int_D \mu(x)^2 dx)^{\frac{1}{2}}$ ). For these tests,  $u_{hp}^{(\text{full})}$  and  $u_{hp}^{(\text{low})}$  were computed using a fixed discretization on a spatial grid ( $\ell = 7$ ) and polynomial degree  $p = 3$  for the stochastic discretization, and  $u_{hp}^{(\text{ref})}$  was computed using the larger polynomial degree  $p = 5$ .<sup>2</sup> Thus, for the means in (6.2),  $\mu_{\text{ref}} - \mu_{\text{full}}$  represents an approximate to the discretization error, and  $\mu_{\text{full}} - \mu_{\text{low}}$  is the error caused by the low-rank approximation, which we refer to as the bias. Note that the mean and the variance of the stochastic Galerkin solution (2.7) can be computed easily by exploiting the orthonormality of the basis functions (i.e., for  $u(\xi) = \sum_{i=1}^n u_i \psi_i(\xi)$ ,  $\mu = u_1 E[\psi_1] = u_1$ , and  $\sigma_u^2 = \sum_{i=2}^n u_i^2 E[\psi_i^2] = \sum_{i=2}^n u_i^2$ ).

Figure 3 shows the results for various tolerances  $\epsilon^c$  and two examples of the diffusion problem (2.1) (with  $M = 5$  and  $M = 7$  in (2.5)) and one example of the convection-diffusion problem (5.2) with  $M = 5$ . In all cases, it can be seen that the error for the low-rank solution is somewhat larger than the discretization error for large  $\epsilon^c$  (and this is caused by the bias), but the bias is significantly smaller than the tolerance  $\epsilon^c$ . The bias is negligible for  $\epsilon^c = 10^{-7}$ .

**7. Conclusion.** We have studied iterative solvers for low-rank solutions of stochastic Galerkin systems of stochastic partial differential equations. In particular, we have explored LRP methods in tensor format for linear systems of Kronecker-product

<sup>2</sup>We also computed a more accurate reference solution with  $p = 7$  for the moderate-dimensional problem (i.e., the diffusion problem (2.1) with  $M = 5$ ) and found the results to be virtually identical.

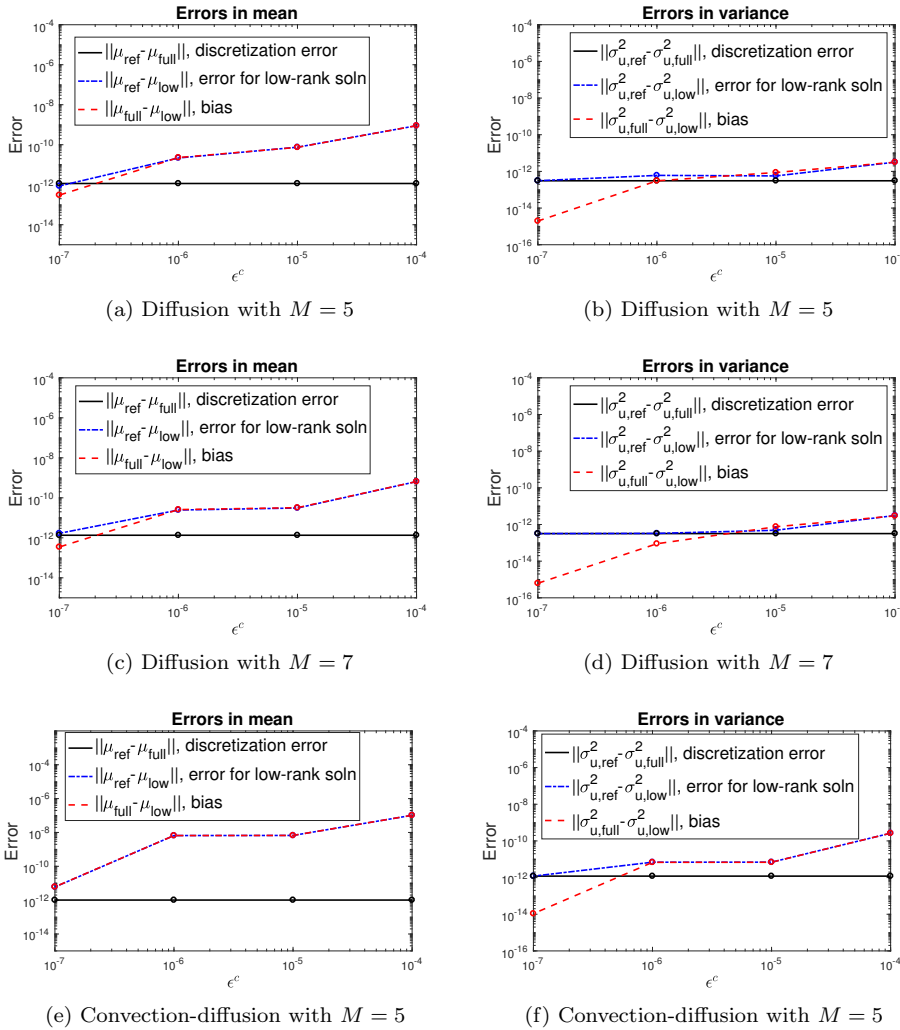


FIG. 3. Errors in the mean and the variance of the low-rank approximate solutions shown in (6.2) and (6.3) for the stochastic diffusion problem (a)–(d) and the stochastic convection-diffusion problem (e)–(f).

structure. For the computational efficiency of the projection methods, basis vectors and iterates in the projection methods are forced to have low rank, which is achieved by a coarse-grid rank-reduction strategy. We have examined the performance of this strategy with two classes of benchmark problems: stochastic diffusion problems and stochastic convection-diffusion problems. For both problem classes, the rank structure of the solution can be identified by an inexpensive coarse-grid computation, and with the resulting coarse-grid rank-reduction strategy, the LRP method is more efficient than methods for which the truncation operator is based on singular values.

## REFERENCES

- [1] I. BABUŠKA AND P. CHATZIPANTELIDIS, *On solving elliptic stochastic partial differential equations*, *Comput. Methods Appl. Mech. Engrg.*, 191 (2002), pp. 4093–4122.
- [2] I. BABUŠKA, R. TEMPONE, AND G. E. ZOURARIS, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, *SIAM J. Numer. Anal.*, 42 (2004), pp. 800–825, <https://doi.org/10.1137/S0036142902418680>.
- [3] M. BACHMAYR AND A. COHEN, *Kolmogorov widths and low-rank approximations of parametric elliptic PDEs*, *Math. Comp.*, 86 (2017), pp. 701–724.
- [4] M. BACHMAYR, A. COHEN, AND W. DAHMEN, *Parametric PDEs: Sparse or Low-Rank Approximations?*, preprint, <https://arxiv.org/abs/1607.04444>, 2016.
- [5] J. BALLANI AND L. GRASEDYCK, *A projection method to solve linear systems in tensor format*, *Numer. Linear Algebra Appl.*, 20 (2013), pp. 27–43.
- [6] P. BENNER, A. ONWUNTA, AND M. STOLL, *Low-rank solution of unsteady diffusion equations with stochastic coefficients*, *SIAM/ASA J. Uncertain. Quant.*, 3 (2015), pp. 622–649, <https://doi.org/10.1137/130937251>.
- [7] G. BEYLKIN AND M. J. MOHLENKAMP, *Algorithms for numerical analysis in high dimensions*, *SIAM J. Sci. Comput.*, 26 (2005), pp. 2133–2159, <https://doi.org/10.1137/040604959>.
- [8] A. N. BROOKS AND T. J. HUGHES, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, *Comput. Methods Appl. Mech. Engrg.*, 32 (1982), pp. 199–259.
- [9] J. D. CARROLL AND J.-J. CHANG, *Analysis of individual differences in multidimensional scaling via an  $N$ -way generalization of Eckart-Young decomposition*, *Psychometrika*, 35 (1970), pp. 283–319.
- [10] P. G. CONSTANTINE, E. DOW, AND Q. WANG, *Active subspace methods in theory and practice: Applications to kriging surfaces*, *SIAM J. Sci. Comput.*, 36 (2014), pp. A1500–A1524, <https://doi.org/10.1137/130916138>.
- [11] A. DOOSTAN, R. G. GHANEM, AND J. RED-HORSE, *Stochastic model reduction for chaos representations*, *Comput. Methods Appl. Mech. Engrg.*, 196 (2007), pp. 3951–3966.
- [12] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, 2nd ed., Oxford University Press, Oxford, UK, 2014.
- [13] O. G. ERNST AND E. ULLMANN, *Stochastic Galerkin matrices*, *SIAM J. Matrix Anal. Appl.*, 31 (2010), pp. 1848–1872, <https://doi.org/10.1137/080742282>.
- [14] R. G. GHANEM AND P. D. SPANOS, *Stochastic Finite Elements: A Spectral Approach*, Dover Publications, New York, 2003.
- [15] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 4th ed., Johns Hopkins Stud. Math. Sci., Johns Hopkins University Press, Baltimore, MD, 2013.
- [16] L. GRASEDYCK, *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, *Computing*, 72 (2004), pp. 247–265.
- [17] J. S. HESTHAVEN, G. ROZZA, AND B. STAMM, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer Briefs in Math., Springer, New York, 2016.
- [18] D. KRESSNER AND C. TOBLER, *Krylov subspace methods for linear systems with tensor product structure*, *SIAM J. Matrix Anal. Appl.*, 31 (2010), pp. 1688–1714, <https://doi.org/10.1137/090756843>.
- [19] D. KRESSNER AND C. TOBLER, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, *SIAM J. Matrix Anal. Appl.*, 32 (2011), pp. 1288–1316, <https://doi.org/10.1137/100799010>.
- [20] M. LOÈVE, *Probability Theory, Vol. II*, 4th ed., Grad. Texts in Math. 46, Springer-Verlag, New York, 1978.
- [21] H. G. MATTHIES AND E. ZANDER, *Solving stochastic systems with low-rank tensor compression*, *Linear Algebra Appl.*, 436 (2012), pp. 3819–3838.
- [22] A. NOUY AND O. P. LE MAITRE, *Generalized spectral decomposition for stochastic nonlinear problems*, *J. Comput. Phys.*, 228 (2009), pp. 202–235.
- [23] C. E. POWELL AND H. C. ELMAN, *Block-diagonal preconditioning for spectral stochastic finite-element systems*, *IMA J. Numer. Anal.*, 29 (2009), pp. 350–375.
- [24] C. E. POWELL, D. SILVESTER, AND V. SIMONCINI, *An efficient reduced basis solver for stochastic Galerkin matrix equations*, *SIAM J. Sci. Comput.*, 39 (2017), pp. A141–A163, <https://doi.org/10.1137/15M1032399>.
- [25] A. QUATERONI, A. MANZONI, AND F. NEGRI, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, Unitext 92, La Matematica per il 3+2, Springer, Cham, 2016.
- [26] J. RUGE AND K. STÜBEN, *Algebraic multigrid*, *Multigrid Methods*, 3 (1987), pp. 73–130.
- [27] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003, <https://doi.org/10.1137/1.9780898718003>.

- [28] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869, <https://doi.org/10.1137/0907058>.
- [29] C. SCHWAB AND C. J. GITTTELSON, *Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs*, Acta Numer., 20 (2011), pp. 291–467.
- [30] D. SILVESTER, H. ELMAN, AND A. RAMAGE, *Incompressible Flow and Iterative Solver Software (IFISS), Version 3.4*, <http://www.manchester.ac.uk/ifiss/>, 2015.
- [31] 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, <https://doi.org/10.1137/120878999>.
- [32] D. XIU AND G. E. KARNIADAKIS, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619–644, <https://doi.org/10.1137/S1064827501387826>.
- [33] D. ZHANG, *Stochastic Methods for Flow in Porous Media. Coping with Uncertainties*, Academic Press, San Diego, CA, 2002.