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

KOOKJIN LEE* AND HOWARD C. ELMAN†

Abstract. We consider the numerical solution of the large systems of linear equations obtained from the stochastic Galerkin formulation of stochastic partial differential equations. We propose an iterative algorithm that exploits Kronecker product structure of the linear systems. Solutions in such settings often have a special low-rank structure, and Krylov subspace methods have been developed to take advantage of this property by truncating various terms constructed during the course of the iteration [3]. We show here that the costs of truncation can be reduced by identifying 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 multilevel method in that the important subspace can be identified in a coarse spatial grid setting. In addition, we construct preconditioners for the linear operators in tensor product format that accelerates the convergence of the low-rank iterative algorithms. The efficiency of the proposed method is illustrated by numerical experiments on benchmark problems.

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

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

where $\mathcal{L}$ is a linear elliptic operator and $a(\boldsymbol{x}, \xi)$ is a positive random field parameterized by a set of random variables $\xi = \{\xi_1, \ldots, \xi_M\}$. 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 where the coefficient is modeled as a random field [8].

As the solution method for (1.1), we consider the stochastic Galerkin method [1, 8], which after discretization, leads to a large coupled deterministic system

$$Au = f, \tag{1.2}$$

for which computations will be expensive for large-scale applications. When the coefficient $a(\boldsymbol{x}, \xi)$ 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 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 [2, 10, 11, 13]. 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]. If Krylov subspace methods are used to compute such a solution, however, it may happen that approximate solutions or other auxiliary terms obtained during the course of an iteration do not have low rank, and rank-reduction schemes are required to keep costs under control.

*Department of Computer Science, University of Maryland, College Park, MD 20742 (klee@cs.umd.edu).

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

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 solution. The strategy we propose is a multilevel 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. Moreover, 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.

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 multilevel rank-reduction scheme, which is the main contribution of this work. In section 5, we illustrate the effectiveness of the low-rank projection method combined with the proposed truncation scheme by numerical experiments on benchmark problems. Finally, in section 6, we draw some conclusions.

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

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

where the diffusion coefficient $a(\boldsymbol{x}, \omega)$ is a random field and ω is an elementary event in a probability space. The gradient operator ∇ only acts on the physical domain D. The weak form of (2.1) is to find u in $V = H_0^1(D) \otimes L^2(\Omega)$

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

If $a(\boldsymbol{x}, \omega)$ is bounded and strictly positive (i.e., $0 < a_{\min} \leq a(\boldsymbol{x}, \omega) \leq a_{\max} < +\infty$), then the Lax-Milgram lemma can be applied to establish existence and uniqueness of a solution $u(\boldsymbol{x}, \omega) \in V$ of the variational problem (2.2). For $a(\boldsymbol{x}, \xi)$ with mean a_0 and variance σ^2, we consider a truncated Karhunen-Loéve expansion [12],

$$a(\boldsymbol{x}, \omega) = a_0 + \sigma \sum_{i=1}^{M} \sqrt{\lambda_i} a_i(\boldsymbol{x}) \xi_i(\omega), \tag{2.3}$$

where (λ_i, a_i) is an eigenpair of the covariance kernel $C(\boldsymbol{x}, \boldsymbol{y})$ of the random field. In the following, we use the notation $a(\boldsymbol{x}, \xi)$ for the random field.

2.1. Stochastic Galerkin method. The stochastic Galerkin method [1, 8] seeks a finite-dimensional solution $u^{sg}(\boldsymbol{x}, \xi) \in W^h = X_h \otimes S_M$ such that

$$\left\langle \int_D a(\boldsymbol{x}, \xi)\nabla u^{sg}(\boldsymbol{x}, \xi) \cdot \nabla v(\boldsymbol{x}, \xi) d\boldsymbol{x} \right\rangle_\rho = \left\langle \int_D f v(\boldsymbol{x}, \xi) \right\rangle_\rho, \quad v(\boldsymbol{x}, \xi) \in W^h \tag{2.4}$$

where $X_h = \text{span}\{\phi_r(\boldsymbol{x})\}_{r=1}^{n_x}$ and $S_M = \text{span}\{\psi_s(\xi)\}_{s=1}^{n_\xi}$ are finite-dimensional subspaces of $H_0^1(D)$ and $L_\rho^2(\Gamma)$, and $u^{sg}(\boldsymbol{x}, \xi) = \sum_{s=1}^{n_\xi} \sum_{r=1}^{n_x} u_{r,s} \phi_r(\boldsymbol{x}) \psi_s(\xi)$. Here, $\{\phi_r\}$ are standard finite element basis functions and $\{\psi_s\}$ are basis functions for the generalized polynomial chaos expansion(PCE) [19]. The stochastic basis functions $\{\psi\}$ are

orthonormal with respect to the joint probability density function $\rho(\xi)$. We consider the total degree space where the maximal degree of $\{\psi_i\}_{i=1}^{n_\xi}$ is defined by p, and the number of PCE basis functions is $n_\xi = \frac{(M+p)!}{M!p!}$.

2.2. Stochastic Galerkin formulation in tensor notation. It follows from (2.4) that the linear system (1.2) can be represented in tensor product notation [14],

$$\left(G_0 \otimes K_0 + \sum_{l=1}^{M} G_l \otimes K_l\right) u = g_0 \otimes f_0 \tag{2.5}$$

where $\otimes$ is the Kronecker product, $\{K_i\}$ are weighted stiffness matrices defined via

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

where $\tilde{a}(\boldsymbol{x}) = \sigma\sqrt{\lambda_l}\, a_l(\boldsymbol{x})$, $\{G_i\}$ are "stochastic" matrices defined via

$$[G_0]_{ij} = \langle \psi_i(\xi)\psi_j(\xi)\rangle_\rho, \quad [G_l]_{ij} = \langle \xi_l\, \psi_i(\xi)\psi_j(\xi)\rangle_\rho, \quad l = 1, \ldots, M, \tag{2.6}$$

and the vectors f_0 and g_0 are defined via $[f_0]_i = \int_D f\phi_i(\boldsymbol{x})d\boldsymbol{x}$, $[g_0]_i = \langle \psi_i(\xi)\rangle_\rho$. Note that $\{G_l\}_{l=0}^{M}$ of (2.6) are highly sparse because of the orthogonality properties of the stochastic basis functions [7].

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 $\mathrm{vec}(\cdot)$ and $\mathrm{mat}(\cdot)$: $u = \mathrm{vec}(U)$, $U = \mathrm{mat}(u)$ where $u \in \mathbb{R}^{n_x n_\xi}, U \in \mathbb{R}^{n_x \times n_\xi}$. A solution u can be represented by a sum of vectors of tensor structure, or equivalently $U = \mathrm{mat}(u)$ can be represented by a sum of rank-one matrices,

$$u = \sum_{k=1}^{\kappa_u} z_k \otimes y_k \Leftrightarrow U = \sum_{k=1}^{\kappa_u} y_k z_k^T = Y_{\kappa_u} Z_{\kappa_u}^T \tag{2.7}$$

where $y_i \in \mathbb{R}^{n_x}$, $z_i \in \mathbb{R}^{n_\xi}$, and $Y_{\kappa_u} = [y_1, \ldots, y_{\kappa_u}] \in \mathbb{R}^{n_x \times \kappa_u}$ and $Z_{\kappa_u} = [z_1, \ldots, z_{\kappa_u}] \in \mathbb{R}^{n_\xi \times \kappa_u}$. If κ_u is the rank of U, then we use κ_u to refer to the rank of u given in tensor structure; thus, u is the sum of terms of rank-one tensor structure. With this notation, the stochastic Galerkin solution $u^{sg}(\boldsymbol{x}, \xi)$ can be represented as

$$u^{sg}(\boldsymbol{x}, \xi) = \Phi(\boldsymbol{x})^T Y_{\kappa_u} Z_{\kappa_u}^T \Psi(\xi) = \left(Y_{\kappa_u}^T \Phi(\boldsymbol{x})\right)^T \left(Z_{\kappa_u}^T \Psi(\xi)\right) \tag{2.8}$$

where $\Phi(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \ldots, \phi_{n_x}(\boldsymbol{x})]^T$ and $\Psi(\xi) = [\psi_1(\xi), \ldots, \psi_{n_\xi}(\xi)]^T$. Note that as shown in [18], (2.8) corresponds to a separated representation $u^{sg}(\boldsymbol{x}, \xi) = \sum_{i=1}^{\kappa_u} \hat{y}_i(\boldsymbol{x})\hat{z}_i(\xi)$ [4] where $\hat{y}_i(\boldsymbol{x}) = (\Phi(\boldsymbol{x}))^T y_i$ 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 (2.8), the rank of the solution is typically $\kappa_u = \min(n_x, n_\xi)$.

In [3, 9], it was shown that the solution to (2.5) can be approximated well by a quantity $\tilde{u}$ of rank $\kappa_{\tilde{u}} \ll \min(n_x, n_\xi)$. Thus, we seek a low-rank approximation to the solution to (2.5),

$$A\tilde{u} = \left(\sum_{l=0}^{M} G_l \otimes K_l\right)\left(\sum_{k=1}^{\kappa_{\tilde{u}}} z_i \otimes y_i\right) \approx g_0 \otimes f_0, . \tag{2.9}$$

2.3. Basic operations in tensor notation. We point out here 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 mth step of such methods results in the *Krylov subspace*, $\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, \ldots, A^{m-1}v_1\}$, which is generated using matrix-vector products and addition of vectors.

The matrix-vector product in (2.9) can be represented as a sum of rank-one tensors by exploiting the properties of the Kronecker product: $A\tilde{u} = \sum_{l=0}^{M}\sum_{k=1}^{\kappa_{\tilde{u}}} G_l z_k \otimes K_l y_k = \sum_{i=1}^{(M+1)\kappa_{\tilde{u}}} \tilde{z}_i \otimes \tilde{y}_i$. The later here 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 rank κ_u and κ_v in tensor notation gives $u + v = \sum_{i=1}^{\kappa_u} z_i \otimes y_i + \sum_{j=1}^{\kappa_v} \hat{z}_j \otimes \hat{y}_j = \sum_{i=1}^{\kappa_u+\kappa_v} z_i \otimes y_i$, where $y_{i+\kappa_u} = \hat{y}_i$ and $z_{i+\kappa_u} = \hat{z}_i$, $i = 1, \ldots, \kappa_v$, 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 (2.5) of low rank, two of the fundamental operations used in Krylov subspace methods tend to increase the rank of the quantities produced. Following [2], we will address this point in the next section.

3. A preconditioned projection method in tensor format. As is well known, the generalized minimum residual method (GMRES) [16] 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 $V_m = [v_1, \ldots, v_m]$, 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 $W_m = AV_m$. In this section, we discuss a variant of this method based on low-rank projection, where advantage is taken of the tensor format of the matrix A and low-rank structure of the solution u.

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* [2, 10, 11, 13], 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 1 summarizes the restarted low-rank projection method in tensor format [2]. A new vector w_j is constructed and then made orthogonal to an m-dimensional space exactly as for GMRES. 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 κ. Thus, all the basis vectors $\{v_i\}_{i=1}^m$ are of the same rank, κ, and $\mathcal{V}_m = \text{span}\{v_1, \ldots, v_m\}$ is not a Krylov subspace because of truncation . However, it is still possible to project the residual onto the subspace $\mathcal{W}_m = \text{span}\{w_1, \ldots, w_m\}$ to find out whether the residual can be decreased by forming a new iterate $\tilde{u}_k + V_m\beta$. Note that all the vectors used in the entire iteration process are stored as the product of two matrices in the form like that shown in the right side of (2.7). The ranks of these vectors will be discussed below.

3.2. Preconditioned low-rank projection method. As always for Krylov subspace methods, preconditioning can be used to speed convergence. For the stochastic diffusion problem, we choose $M = G_0 \otimes K_0$ as the precondtioner, which is known as the mean-based preconditioner [14]. For the practical application of the precon-

Algorithm 1 Restarted low-rank projection method in tensor format

set the initial solution $\tilde{u}_0$
for $k = 0, 1, \ldots$ **do**
  $r_k := f - A\tilde{u}_k$
  if $\|r_k\|/\|f\| < \epsilon$ **then**
    return $\tilde{u}_k$
  end if
  $\tilde{v}_1 := \mathcal{T}_\kappa(r_k)$
  $v_1 := \tilde{v}_1/\|\tilde{v}_1\|$
  for $j = 1, \ldots, m$ **do**
    $w_j := Av_j$
    solve $(V_j^T V_j)\alpha = V_j^T w_j$
    $\tilde{v}_{j+1} := \mathcal{T}_\kappa\left(w_j - \sum_{i=1}^{j} \alpha_i v_i\right)$
    $v_{j+1} := \tilde{v}_{j+1}/\|\tilde{v}_{j+1}\|$
  end for
  solve $(W_m^T A V_m)\beta = W_m^T r_k$
  $\tilde{u}_{k+1} := \mathcal{T}_\kappa(\tilde{u}_k + V_m\beta)$
end for

ditioner, we employ algebraic multigrid methods [15], 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. We use a right-oriented preconditioner, so that the preconditioned matrix-vector product is $AM^{-1}\hat{u} = \sum_{l=0}^{M}\sum_{k=1}^{\kappa_{\hat{u}}} G_l\hat{z}_k \otimes K_l\tilde{K}_0^{-1}\hat{y}_k$, and $\hat{u} = M\tilde{u} = \sum_{i=1}^{\kappa_{\hat{u}}} \hat{z}_i \otimes \hat{y}_i$. 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 low-rank projection method, truncation of tensors is essential for the efficient computation of approximate solutions. In this section, we discuss the conventional approach for truncation and we introduce a new multilevel 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 κ', a standard approach for truncation [2, 13] 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 $Z_{\kappa'}$: $Y_{\kappa'} = Q_Y R_Y \in \mathbb{R}^{n_x\times\kappa'}$, $Z_{\kappa'} = Q_Z R_Z \in \mathbb{R}^{n_\xi\times\kappa'}$. Then, one can compute the SVD $R_Y R_Z^T = \hat{U}_{\kappa'}\hat{\Sigma}_{\kappa'}\hat{V}_{\kappa'}^T = \sum_{k=1}^{\kappa'}\hat{\sigma}_k\hat{u}_k\hat{v}_k^T$ and truncate the sum with κ terms to produce $\tilde{Y}_\kappa = Q_Y\hat{U}_\kappa\hat{\Sigma}_\kappa \in \mathbb{R}^{n_x\times\kappa}$, $\tilde{Z}_\kappa = Q_Z\hat{V}_\kappa \in \mathbb{R}^{n_\xi\times\kappa}$. The truncated approximation of U is then $\tilde{U} = \tilde{Y}_\kappa\tilde{Z}_\kappa^T$. The computational complexity of the truncation is $O((n_x + n_\xi + \kappa)(\kappa')^2)$, which grows quadradically with respect to κ'. In the next section, we introduce a new truncation method that avoids this computation.

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

Let $u^c(\boldsymbol{x}, \xi)$ represent a solution obtained on a coarse-spatial grid (i.e., n_x is small). As in (2.8), $u^c(\boldsymbol{x}, \xi)$ can be represented as $u^c(\boldsymbol{x}, \xi) = (\Phi^c(\boldsymbol{x}))^T U^c\Psi(\xi) =$

$\left((Y^c)^T\Phi^c(\boldsymbol{x})\right)^T\left((Z^c)^T\Psi(\xi)\right)$. Here, we propose to use Z^c to define a truncation operator in the projection method when we compute a solution for the problem on a finer grid. That is, the truncation operator is defined such that, given a matricized vector $U = Y_{\kappa'}Z_{\kappa'}^T$ of rank κ', $\mathcal{T}_\kappa(U) \equiv \left(Y_{\kappa'}Z_{\kappa'}^TZ_\kappa^c\right)(Z_\kappa^c)^T = \tilde{U}$ where the resulting quantity $\tilde{U} = \tilde{Y}_\kappa\tilde{Z}_\kappa^T$ is of rank κ (i.e., $\tilde{Y}_\kappa = Y_{\kappa'}Z_{\kappa'}^TZ_\kappa^c \in \mathbb{R}^{n_x\times\kappa}$ and $\tilde{Z}_\kappa = Z_\kappa^c \in \mathbb{R}^{n_\xi\times\kappa}$). The desired rank κ is determined such that the relative residual $\|f^c - A^cu^{c,\kappa}\|_2/\|f^c\|_2$ is smaller than a certain tolerance where $u^{c,\kappa}$ is a κ-term approximation of u^c. This truncation operation requires two matrix-matrix products, and the computational complexity of truncating a vector from κ' to κ is $O(\kappa'\kappa(n_x+n_\xi))$.

For efficient coarse-grid computation, we use the Proper Generalized Decomposition (PGD) method developed in [18], which computes the separated representation of a coarse-grid solution: $u^{c,\kappa}(\boldsymbol{x},\,\xi) = \sum_{i=1}^{\kappa}\tilde{y}_i(\boldsymbol{x})\tilde{z}_i(\xi)$. With the stochastic Galerkin discretization, each function can be represented as $\tilde{y}_i(\boldsymbol{x}) = \sum_{k=1}^{n_x}\tilde{y}_k^{(i)}\phi_k^c(\boldsymbol{x})$ and $\tilde{z}_i(\xi) = \sum_{l=1}^{n_\xi}\tilde{z}_l^{(i)}\psi_l(\xi)$, and, as a result as in (2.8), $u^{c,\kappa}(\boldsymbol{x},\,\xi) = \left((\tilde{Y}_\kappa^c)^T\Phi^c(\boldsymbol{x})\right)^T\left((\tilde{Z}_\kappa^c)^T\Psi(\xi)\right)$ where $\tilde{Y}_\kappa^c = [\tilde{y}^{(1)},\,\cdots,\,\tilde{y}^{(\kappa)}] \in \mathbb{R}^{n_x\times\kappa}$ and $\tilde{Z}_\kappa^c = [\tilde{z}^{(1)},\,\cdots,\,\tilde{z}^{(\kappa)}] \in \mathbb{R}^{n_\xi\times\kappa}$ are coefficient matrices such that the ith elements of $\tilde{y}^{(j)}$ and $\tilde{z}^{(j)}$ are $\tilde{y}_i^{(j)}$ and $\tilde{z}_i^{(j)}$, respectively. Now, the solution U^c is approximated by $U^{c,\kappa} = \tilde{Y}_\kappa^c(\tilde{Z}_\kappa^c)^T$, and we can obtain Z_κ^c by computing the SVD of $U^{c,\kappa}$ (i.e., $U^{c,\kappa} = \hat{U}\hat{\Sigma}\hat{V}^T = Y_\kappa^c(Z_\kappa^c)^T$). We briefly explain how the PGD method computes a κ-term approximation in the next section.

4.3. Proper Generalized Decomposition method. The PGD method incrementally identifies the function pairs $(\tilde{y}_i(\boldsymbol{x}),\,\tilde{z}_i(\xi))$ 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 $\{\tilde{y}_{i+1}\zeta + \upsilon\tilde{z}_{i+1};\,\upsilon \in X_h,\,\zeta \in S_M\}$: find $\tilde{y}_{i+1}\tilde{z}_{i+1}$ such that $\forall(\upsilon,\,\zeta) \in X_h \times S_M$

$$\left\langle \int_D a(\boldsymbol{x},\,\xi)\nabla(u^{c,\,i} + \tilde{y}_{i+1}\tilde{z}_{i+1})\cdot\nabla(\tilde{y}_{i+1}\zeta + \upsilon\tilde{z}_{i+1})\right\rangle = \left\langle \int_D f(\tilde{y}_{i+1}\zeta + \upsilon\tilde{z}_{i+1})\right\rangle. \tag{4.1}$$

It follows from (4.1) 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 and a stochastic problem. The deterministic problem is as follows: given $\tilde{z}_{i+1}$, find $\tilde{y}_{i+1} \in X_h$ such that $\left\langle \int_D a(\boldsymbol{x},\,\xi)\nabla(u^{c,\,i} + \tilde{y}_{i+1}\tilde{z}_{i+1})\cdot\nabla(\phi_j^c\tilde{z}_{i+1})\right\rangle = \left\langle \int_D f\phi_j^c\tilde{z}_{i+1}\right\rangle$, $j = 1,\,\ldots,\,n_x^c$. 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 $\left\langle \int_D a(\boldsymbol{x},\,\xi)\nabla(u^{c,\,i} + \tilde{y}_{i+1}\tilde{z}_{i+1})\cdot\nabla(\tilde{y}_{i+1}\psi_j)\right\rangle = \left\langle \int_D f\tilde{y}_{i+1}\psi_j\right\rangle$, $j = 1,\,\ldots,\,n_\xi$. Since $\tilde{z}_{i+1}$ is approximated by the PCE, n_ξ unknowns have to be determined by solving a linear system of order n_ξ.

Solutions of these sets of κ systems of order n_x^c and κ systems of order n_ξ produce the κ-term approximation to the solution. The PGD method seeks solution pairs until the relative residual of the computed solution satisfies a given tolerance, $\|f^c - A^cu^{c,\kappa}\|_2/\|f^c\|_2 < \epsilon$. The accuracy of the κ-term approximation can also be improved by solving a set of κ coupled equations: given $\{\tilde{y}_i\}_{i=1}^{\kappa}$, find $\{\tilde{z}_i\}_{i=1}^{\kappa}$ such that $\left\langle \int_D a(\boldsymbol{x},\,\xi)\nabla(u^{(\kappa)})\cdot\nabla(\tilde{y}_i\psi_j)\right\rangle = \left\langle \int_D f\tilde{y}_i\psi_j\right\rangle$, $i = 1,\,\ldots,\,\kappa,\ j = 1,\,\ldots,\,n_\xi$. The update problem requires a linear system of order κn_ξ to be solved.

With the proposed truncation strategy, Algorithm 2 summarizes the entire procedure to compute a solution on a finer grid.

Algorithm 2 Preconditioned low-rank projection method with the multilevel rank-reduction

1: Compute $u^{c,\kappa}$ which satisfies $\frac{\|f^c - A^c u^{c,\kappa}\|_2}{\|f^c\|_2} < \epsilon$ using the PGD method
2: Compute Z^c_κ such that $U^{c,\kappa} = Y^c_\kappa (Z^c_\kappa)^T$ and define $\mathcal{T}_\kappa(U) \equiv (UZ^c_\kappa)(Z^c_\kappa)^T$
3: Run Algorithm 1 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 [17]. Example problems are posed on a square domain and ℓ is the spatial discretization parameter (i.e., $n_x = (2^\ell + 1)^2$).

For $a(\boldsymbol{x}, \xi)$ in (2.3), we consider independent random variables $\{\xi_i\}_{i=1}^M$ uniformly distributed over $[-\sqrt{3}, \sqrt{3}]$, $a_0 = 1$ and $\sigma = 0.05$. As the covariance kernel, we use $C(\boldsymbol{x}, \boldsymbol{y}) = \sigma^2 \exp\left(-\frac{|x_1 - y_1|}{c} - \frac{|x_2 - y_2|}{c}\right)$ where c is the correlation length. The number of terms M in (2.3) is decided 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 standard finite element basis and Legendre polynomials as the stochastic basis functions. 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 (2.4) with forcing term $f(\boldsymbol{x}) = 1$.

Coarse spatial grid computation. We compute κ-term approximations using the PGD method on a coarser spatial grid. We choose the refinement level ℓ^c for the coarse grid such that ℓ^c is the smallest integer for which $n_x^c = (2^{\ell^c} - 1)^2 \geq n_\xi$; here n_x^c is the number of degrees of freedom in the spatial domain excluding boundary nodes. Table 5.1 shows the rank κ of solutions that satisfy the tolerance ϵ for varying correlation lengths c and M and the corresponding computation time t_c.

TABLE 5.1
Rank (κ) of coarse-grid solutions satisfying a specified tolerance ϵ for the PGD computation, and CPU time (t_c) for coarse-grid computation using the PGD method, for varying c and M

	$\epsilon = 10^{-5}$				$\epsilon = 10^{-6}$			
c	4	3	2.5	2	4	3	2.5	2
M, n_ξ	5, 56	7, 120	10, 286	15, 816	5, 56	7, 120	10, 286	15, 816
n_x^c (ℓ^c)	225 (4)	225 (4)	961 (5)	961 (5)	225 (4)	225 (4)	961 (5)	961 (5)
Rank(κ)	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

Fine spatial grid computation. With the truncation operator Z^c_κ obtained from the coarse-grid solution, we solve the same stochastic diffusion problems on finer spatial grids $\ell = \{8, 9\}$. For the fine-grid low-rank solutions, we use the rank κ obtained from the coarse-grid solutions. For example, the third column of Table 5.2 shows the time required to find solutions of rank 25 when the number of terms in the KL-expansion is $M = 5$.

Comparison to a truncation operator based on singular values. We compare the performance of the proposed solver to the preconditioned low-rank projection method combined with the conventional truncation operator. Table 5.3 shows the computation time required to compute approximate solutions using the conventional

TABLE 5.2
CPU time to compute approximate solutions satisfying $\epsilon = 10^{-5}$, 10^{-6} using the preconditioned low-rank projection method with the multilevel rank-reduction. Here, t is the total time and t_f excludes the time to compute the coarse-grid solution, t_c

n_x		$\epsilon = 10^{-5}$				$\epsilon = 10^{-6}$			
(ℓ)	M	5	7	10	15	5	7	10	15
257^2	t_f	22.69	34.90	84.85	340.51	27.61	56.36	148.07	1014.97
(8)	t	25.17	38.37	93.20	385.59	30.55	61.41	162.90	1177.68
513^2	t_f	144.69	194.41	445.36	2809.54	163.31	310.14	1318.79	-
(9)	t	147.17	197.87	453.71	2854.62	166.24	315.18	1333.63	-

TABLE 5.3
CPU time to compute approximate solutions satisfying $\epsilon = 10^{-5}$ using the preconditioned low-rank projection (LRP) methods with the multilevel rank-reduction and the singular value based truncation on the level 8 spatial grid

Solver	M	5	7	10	15	20
LRP-SVD	t	55.04	108.11	284.27	1280.65	8270.58
LRP-Multilevel	t	25.17	38.37	93.20	385.59	1943.49

and new truncation strategies.

PGD as a solver on a finer spatial grid. Note that 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. Table 5.4 shows the rank and the computation time for computing approximate solutions that satisfy the tolerance 10^{-5} when the PGD method is used on a finer spatial grid. We compare the rank and the computation time for computing solutions using the PGD method and the proposed projection method. The proposed low-rank projection method runs faster and requires somewhat smaller ranks than the PGD method.

TABLE 5.4
Computation time to obtain approximate solutions satisfying $\epsilon = 10^{-5}$ using the PGD method and the preconditioned low-rank projection method on the level 8 spatial grid

Solver	M	5	7	10	15	20
PGD	κ	25	45	65	125	195
	t	43.78	109.72	228.73	940.69	3066.87
LRP-Multilevel	κ	25	40	65	115	180
	t	25.17	38.37	93.20	385.59	1943.49

5.2. Stochastic convection-diffusion problem. For the next benchmark problem, we consider the steady-state convection-diffusion equation with non-homogeneous Dirichlet boundary condition with the constant vertical wind $\vec{w} = (0, 1)$ and $f = 0$,

$$\nu\nabla \cdot (a(\boldsymbol{x}, \xi)\nabla u(\boldsymbol{x}, \xi)) + \vec{w} \cdot \nabla u(\boldsymbol{x}, \xi) = f(\boldsymbol{x}, \xi) \text{ in } D \times \Gamma, \tag{5.1}$$

and the boundary condition is determined by $u(x, -1) = x$, $u(x, 1) = 0$, $u(-1, y) \approx -1$, $u(1, y) \approx 1$ where the latter two approximations hold except near $y = 1$, and ν is the viscosity parameter. We consider the convection-dominated case (i.e., $\nu < 1$) and employ the streamline diffusion method for stabilization [5]. Here, we define the element *Peclet* number $\mathcal{P}_k = \frac{\|\vec{w}_k\|_2 h_k}{2\nu}$ where $\|\vec{w}_k\|_2$ is the ℓ_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 [6].

Given $a(\boldsymbol{x}, \xi)$ in (2.3), we again discretize (5.1) using the finite element method and the generalized PCE. After the discretization, we obtain a linear system in tensor product notation

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

where the convection term N and the streamline-diffusion term S can be defined via $[N]_{ij} = \int_D \vec{w} \cdot \nabla\phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})d\boldsymbol{x}$, and $[S]_{ij} = \sum_{l=1}^{n_e} \delta_l \int_D (\vec{w} \cdot \nabla\phi_i)(\vec{w} \cdot \nabla\phi_j)d\boldsymbol{x}$ where n_e is the number of element in the finite element discretization and $\frac{h_k}{2\|\vec{w}\|_2}\left(1 - \frac{1}{\mathcal{P}_k}\right)$ if $\mathcal{P}_k > 1$. As the preconditioner, we choose $M = G_0 \otimes (K_0 + N + S)$, and the action of $(K_0 + N + S)^{-1}$ is replaced by application of a single V-cycle of an AMG method.

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 low-rank projection method for varying viscosity parameter ν, and we set $m = 10$ for Algorithm 1. Table 5.5 shows κ 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 $\ell = \{8, 9\}$ satisfying 10^{-5} and 10^{-6}. Underlined numbers in the spatial grid level indicates cases where streamline diffusion is not needed.

TABLE 5.5

CPU time and the number of projection cycles (k in Algorithm 1) to compute approximate solutions with $\epsilon = 10^{-5}$ and 10^{-6} using the preconditioned low-rank projection methods with the multilevel rank-reduction method for varying ν

			$\epsilon = 10^{-5}$			$\epsilon = 10^{-6}$		
ν	ℓ	M	5	7	10	5	7	10
$\frac{1}{20}$	4	κ	25	35	55	35	50	75
		t_c	2.56	4.83	26.34	3.31	9.17	60.51
	<u>8</u>	t_f	51.50 (1)	67.24 (1)	128.45 (1)	83.43 (2)	136.69 (2)	341.32 (2)
	<u>9</u>	t_f	424.17 (1)	489.96 (1)	929.74 (1)	559.78 (2)	871.88 (2)	3031.37 (2)
$\frac{1}{200}$	5	κ	20	25	45	25	40	60
		t_c	2.91	4.79	16.54	3.46	8.57	38.35
	8	t_f	52.51 (1)	61.29 (1)	99.14 (1)	74.33 (2)	120.39 (2)	254.95 (2)
	9	t_f	468.97 (2)	543.20 (2)	949.72 (2)	572.07 (3)	890.72 (3)	2972.20 (3)
$\frac{1}{600}$	6	κ	20	20	35	30	35	45
		t_c	9.79	13.20	34.47	17.99	22.03	47.44
	8	t_f	65.48 (2)	73.28 (2)	142.46 (2)	81.93 (2)	107.84 (2)	186.56 (2)
	9	t_f	473.52 (2)	505.27 (2)	849.07 (2)	625.07 (3)	803.89 (3)	1527.38 (3)

When the viscosity parameter is small (i.e., $\nu = 1/600$), the coarse-grid computation requires the κ-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.

6. Conclusion. We have studied iterative solvers for low-rank solutions of stochastic Galerkin systems of stochastic partial differential equations. In particular, we have explored low-rank projection methods in tensor format for linear systems of Kronecker-product 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 multilevel rank-reduction strategy. We have examined the performance of this strategy with two benchmark problems: stochastic diffusion problems and stochastic convection-diffusion problems. Numerical experiments showed that the rank structure of the solution can be identified by an inexpensive coarse-grid computation. Moreover, numerical results showed that the low-rank projection method combined with the multilevel rank-reduction strategy outperformed methods for which the truncation operator is based on singular values.

REFERENCES

[1] I. Babuška, R. Tempone, and G. E. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 800–825.

[2] J. Ballani and L. Grasedyck, *A projection method to solve linear systems in tensor format*, Numerical Linear Algebra with Applications, 20 (2013), pp. 27–43.

[3] P. Benner, A. Onwunta, and M. Stoll, *Low-rank solution of unsteady diffusion equations with stochastic coefficients*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 622–649.

[4] G. Beylkin and M. J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM Journal on Scientific Computing, 26 (2005), pp. 2133–2159.

[5] 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*, Computer Methods in Applied Mechanics and Engineering, 32 (1982), pp. 199–259.

[6] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, 2014.

[7] O. G. Ernst and E. Ullmann, *Stochastic Galerkin matrices*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1848–1872.

[8] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: a Spectral Approach*, Dover Publications, 2003.

[9] L. Grasedyck, *Existence and computation of low kronecker-rank approximations for large linear systems of tensor product structure*, Computing, 72 (2004), pp. 247–265.

[10] D. Kressner and C. Tobler, *Krylov subspace methods for linear systems with tensor product structure*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 1688–1714.

[11] ———, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 1288–1316.

[12] M. Loeve, *Probability Theory, Vol. II*, Graduate texts in Mathematics, 46 (1978), pp. 0–387.

[13] H. G. Matthies and E. Zander, *Solving stochastic systems with low-rank tensor compression*, Linear Algebra and its Applications, 436 (2012), pp. 3819–3838.

[14] C. E. Powell and H. C. Elman, *Block-diagonal preconditioning for spectral stochastic finite-element systems*, IMA Journal of Numerical Analysis, 29 (2009), pp. 350–375.

[15] J. Ruge and K. Stüben, *Algebraic multigrid*, Multigrid Methods, 3 (1987), pp. 73–130.

[16] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[17] D. Silvester, H. Elman, and A. Ramage, *Incompressible Flow and Iterative Solver Software (IFISS) version 3.4*, August 2015. `http://www.manchester.ac.uk/ifiss/`.

[18] L. Tamellini, O. Le Maitre, and A. Nouy, *Model reduction based on proper generalized decomposition for the stochastic steady incompressible Navier-Stokes equations*, SIAM Journal on Scientific Computing, 36 (2014), pp. A1089–A1117.

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