TRUNCATED HIERARCHICAL PRECONDITIONING FOR THE STOCHASTIC GALERKIN FEM

***Bedřich Sousedík*[1,2,*] *&* *Roger G. Ghanem*[1]**

[1] *Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90089-2531, USA*

[2] *Institute of Thermomechanics, Academy of Sciences of the Czech Republic, Dolejškova 1402/5, 182 00 Prague 8, Czech Republic*

Original Manuscript Submitted: 03/05/2013; Final Draft Received: 09/03/2013

Stochastic Galerkin finite element discretizations of partial differential equations with coefficients characterized by arbitrary distributions lead, in general, to fully block dense linear systems. We propose two novel strategies for constructing preconditioners for these systems to be used with Krylov subspace iterative solvers. In particular, we present a variation of the hierarchical Schur complement preconditioner, developed recently by the authors, and an adaptation of the symmetric block Gauss-Seidel method. Both preconditioners take advantage of the hierarchical structure of global stochastic Galerkin matrices, and also, when applicable, of the decay of the norms of the stiffness matrices obtained from the polynomial chaos expansion of the coefficients. This decay allows to truncate the matrix-vector multiplications in the action of the preconditioners. Also, throughout the global matrix hierarchy, we approximate solves with certain submatrices by the associated diagonal block solves. The preconditioners thus require only a limited number of stiffness matrices obtained from the polynomial chaos expansion of the coefficients, and a preconditioner for the diagonal blocks of the global matrix. The performance is illustrated by numerical experiments.

KEY WORDS: *stochastic Galerkin finite element methods, iterative methods, Schur complement method, Gauss-Seidel method, hierarchical and multilevel preconditioning*

1. INTRODUCTION

Precise values of coefficients in the setup of physical models using partial differential equations (PDEs) are often not known. In such situations, the coefficients are typically treated as random variables or processes in attempt to quantify uncertainty in the underlying problem. Probably the most popular and widely used class of methods for a solution of these problems are the Monte Carlo techniques, because they require only solutions of the PDE for a given set of realizations of the input random coefficients. These methods are well-known for their robustness, versatility, and quite slow convergence. Therefore, in the last two decades, a significant effort has been devoted to the development of methods that leverage regularity of the solution, and outperform the Monte Carlo methods at least for problems with stochastic dimensions that are not too large. The most promising developments include the stochastic finite element methods. There are two main variants of the stochastic finite elements: collocation methods [1, 2], and stochastic Galerkin methods [3–5]. The first approach samples the stochastic PDE at a predetermined set of collocation points, which yields a set of uncoupled deterministic systems. The solution at these collocation points is then used to interpolate the solution in the entire random input domain. Because extending legacy software to support the collocation points is relatively simple, these methods are often regarded as non-intrusive. On the other hand, the second approach, using the stochastic Galerkin methods, translates the stochastic PDE into one large coupled deterministic system. Consequently, the Galerkin methods require a development of new solvers and therefore they are commonly regarded

[*] Correspond to Bedřich Sousedík, E-mail: sousedik@usc.edu

as intrusive. Since use of direct solvers for the stochastic Galerkin systems might be prohibitive due to their large size, iterative solvers are often preferred. We then seek preconditioners to speed up convergence.

There are two main types of expansion of the random field representing the uncertainty in the coefficients of the underlying model. The first one, known as Karhunen-Loève (KL) expansion [6], is suitable for a finite-dimensional representation of Gaussian random fields, and leads in conjunction with the stochastic Galerkin finite element discretizations to block sparse structure of global matrices. Random fields with general probability distributions are also sometimes represented using the KL expansion; however, it might be more suitable to use so-called generalized polynomial chaos (gPC) expansion [7] for their representation. This in general leads to block dense structure of the global matrices. In particular, in this paper we will focus on the random coefficients of the model elliptic PDE with lognormal distribution. Such discretization is done within the gPC framework, using Hermite polynomials [8]. It has been shown in [9, Theorem 18] that in order to guarantee a complete representation of the lognormal random field, one has to use twice the order of polynomial expansion of the coefficient than of the solution, which indeed leads to a fully block dense structure of the global stochastic Galerkin matrix. Clearly, based on the block sparsity pattern of the global stochastic Galerkin matrix, a certain class of preconditioners might be more suitable, compared to the others, for a particular problem to be solved.

Probably the most simple, yet quite powerful method is the mean-based preconditioner proposed by Pellissetti and Ghanem [10] and analyzed by Powell and Elman [11]. It is a block diagonal preconditioner which uses the information carried by the mean of the random coefficient and the corresponding stiffness matrix, i.e., the zeroth order term of the coefficient expansion. Subsequently, Ullmann has developed a Kronecker product preconditioner [12], cf. also [13], that makes use of information carried by higher order terms and improves the convergence quite significantly. The preconditioner has the Kronecker product structure of the global stochastic Galerkin matrix. Various iterative methods and preconditioners including multigrid methods, based on matrix splitting, were compared by Rosseel and Vandewalle in [14]. Most recently, the authors have developed in [15] a hierarchical Schur complement preconditioner that takes advantage of the recursive hierarchy in the structure of the global stochastic Galerkin matrix. We note that an interesting approach to solver parallelization has been proposed by Keese and Matthies [16], and a block-triangular preconditioner for the block sparse case has been proposed recently by Zheng et al. [17]. Finally, we refer to Ernst and Ullmann [18] for a more general study of the stochastic Galerkin matrices.

In this paper, we propose two novel strategies for constructing preconditioners for solution of the linear systems with block dense global matrices to be used with Krylov subspace iterative methods. In particular, we present a variation on the hierarchical Schur complement preconditioner developed recently by the authors [15], and an adaptation of the symmetric block Gauss-Seidel method. Both preconditioners are built in a way to account for the recursive hierarchy in the structure of the global stochastic Galerkin matrix. Because, unlike in the block sparse case, neither of the submatrices in this hierarchy is block diagonal, we approximate solves with submatrices by the associated diagonal block solves. This variant yields versions of the two preconditioners, which will be called as *approximate*. We note that the approximate versions thus use "locally" the block diagonal preconditioning, which allows for a certain level of decoupling. Our numerical experiments indicate that this approximation might not be necessarily traded off for slower convergence rates. In the second variation, we take take advantage of the decay of the norms of the stiffness matrices obtained from the polynomial chaos expansion of the coefficients. The decay of the matrices allows to truncate the matrix-vector multiplications in the action of the preconditioners, and therefore these versions of the preconditioners will be called as *truncated*. We also note that one can replace the direct solvers of the diagonal blocks by iterative ones, possibly with the same tolerance as for the outer iterations. Doing so, the preconditioners become variable and one has to make a careful choice of the Krylov iterative method used for the global (outer) iterations. In general, it is recommended to use flexible methods such as the flexible conjugate gradients [19], FGMRES [20], or GMRESR [21]. Thus neither the global matrix, nor the matrix of the preconditioner need to be formed explicitly, and we can use the so called MAT-VEC operations introduced in [10] for all matrix-vector multiplications. Provided that we have a preconditioner for the diagonal block solves available, the ingredients of our methods include only the number of stiffness matrices from the expansion of the random coefficient. Therefore, the proposed methods can be viewed as minimally intrusive because they can be built as wrappers around existing solvers for the corresponding deterministic problem. Finally, we note that choice of a preconditioner for the diagonal block solves would not change the convergence in terms of outer iterations, and we will address this topic elsewhere.

The paper is organized as follows. In Section 2 we introduce the model problem and its discretization, in Section 3 we discuss the structure of the stochastic Galerkin matrices, in Section 4 we formulate our algorithms, in Section 5 we present some numerical experiments, and finally in Section 6 we summarize and conclude our work.

2. MODEL PROBLEM

Let $(\Omega, \mathcal{F}, \mu)$ denote the probability space associated with a physical experiment. We are interested in the solution of the stochastic linear elliptic boundary value problem, with stochastic input and deterministic data, given in a bounded domain $D \subset \mathbb{R}^d$, $d = 2, 3$. The solution is a random function $u(x, \omega) : \overline{D} \times \Omega \to \mathbb{R}$ that almost surely (a.s.) satisfies the equation

$$-\nabla \cdot (k(x, \omega)\ \nabla u(x, \omega)) = f(x) \qquad \text{in } D \times \Omega, \tag{1}$$

$$u(x, \omega) = 0 \qquad \text{on } \partial D \times \Omega, \tag{2}$$

where $f \in L^2(D)$, the gradient symbol ∇ denoting the differentiation with respect to the spatial variables, and

$$0 < k_{\min} \leq k(x, \omega) \qquad \text{a.s. in } \Omega,\ \forall x \in \overline{D}.$$

The function $k(x, \omega) : \overline{D} \times \Omega \to \mathbb{R}$ is a random scalar field with a continuous and square-integrable covariance function. We will assume that the randomness of $k(x, \omega)$ is induced by a set of random variables $\xi(\omega) = \{\xi_i(\omega)\}_{i=1}^N$ that are assumed to be independent Gaussian with zero mean and unit variance. Thus the domain of $k(\cdot, \omega)$ is restricted to the subset $(\Omega, \mathcal{F}(\xi), \mu)$ of $(\Omega, \mathcal{F}, \mu)$ where $\mathcal{F}(\xi)$ is the $\sigma-$algebra generated by ξ. We will henceforth use $k(x, \xi)$ instead of $k(x, \omega)$, and the solution $u(x, \omega)$ can be also written as $u(x, \xi)$.

In the variational formulation of problem (1) and (2), we consider the solution of the equation

$$u \in U :\ a(u, v) = \langle f, v \rangle, \qquad \forall v \in U. \tag{3}$$

where U is a tensor product space defined as

$$U = H_0^1(D) \otimes L^2(\Omega), \qquad \|u\|_U = \sqrt{\mathbb{E}\left[\int_D |\nabla u|^2\, dx\right]},$$

where $\mathbb{E}[\cdot]$ denotes mathematical expectation, and the bilinear form $a(\cdot, \cdot)$ along with the right-hand side are defined as

$$a(u, v) = \mathbb{E}\left[\int_D k(x, \xi)\ \nabla u \cdot \nabla v\, dx\right], \qquad \langle f, v \rangle = \mathbb{E}\left[\int_D f\, v\, dx\right].$$

We assume that $k(x, \xi)$ can be represented, using a generalized polynomial chaos (gPC) expansion, as

$$k(x, \xi) = \sum_{i=0}^{M'} k_i(x)\, \psi_i(\xi), \quad \text{where } k_i(x) = \frac{\mathbb{E}[k(x)\, \psi_i(\xi)]}{\mathbb{E}[\psi_i^2]}, \tag{4}$$

and $\{\psi_i(\xi)\}_{i=0}^{M'}$ is a set of N-dimensional Hermite polynomials [5]. Similarly, we will look for an expansion of the solution as

$$u(x, \xi) = \sum_{i=0}^{M} u_i(x)\, \psi_i(\xi), \tag{5}$$

that converges in $L^2(\Omega \times D)$ as $M \to \infty$. We will, in particular, model $k(x, \xi)$ as a truncated lognormal process. To this end, let $g(x, \xi) = g_0(x) + \sum_{i=1}^N \xi_i g_i(x)$ be a truncated Karhunen-Loève expansion of a Gaussian process defined on D with known mean $g_0(x)$ and covariance function $C_g(x, y)$, i.e., $g_i(x)$ are weighted eigenfunctions of

$$\int_D C_g(x, y)\, \varphi_i(y)\ dy = \lambda_i \varphi_i(x), \quad \forall x \in D, \qquad \text{and} \quad g_i(x) = \sqrt{\lambda_i}\varphi_i(x). \tag{6}$$

Then $k(x,\xi)$ is defined as $k(x,\xi) = \exp[g(x,\xi)]$. The procedure for computing the coefficients $k_i(x)$ in (4) has been derived by Ghanem in [8]. Denoting $\eta_j = \xi_j - g_j$, the coefficients are precisely given, cf. [8, eq. (33)], as

$$k_i(x) = \frac{\mathbb{E}[\psi_i(\eta)]}{\mathbb{E}[\psi_i^2]} \exp\left[g_0(x) + \frac{1}{2}\sum_{j=1}^{N}(g_j(x))^2\right]. \tag{7}$$

According to [9], in order for (4) to guarantee a complete representation of the lognormal random field, the order of polynomial expansion of $k(x,\xi)$ in (4) should be twice the order of the expansion of the solution $u(x,\xi)$ in (5). Denoting the two orders of the polynomial expansions of $u(x,\xi)$ and $k(x,\xi)$ as P and P', resp., with $P' = 2P$, the total numbers of the gPC polynomials are

$$M+1 = \frac{(N+P)!}{N!P!}, \quad M'+1 = \frac{(N+P')!}{N!P'!} = \frac{(N+2P)!}{N!(2P)!}. \tag{8}$$

We will consider approximations to the variational problem (3) given by the finite element discretizations of $H_0^1(D)$. The solution $u(x,\xi)$ from (5) will be thus approximated by

$$u(x,\xi) = \sum_{i=1}^{N_{\text{dof}}}\sum_{j=0}^{M} u_{ij}\phi_i(x)\psi_j(\xi), \tag{9}$$

where $\{\phi_i(x)\}_{i=1}^{N_{\text{dof}}}$ is a finite element basis, and $\{\psi_j(\xi)\}_{j=0}^{M}$ is the basis of the Hermite polynomials described above. Substituting the expansions (4) and (9) into (3) yields a deterministic system of linear equations

$$\sum_{j=0}^{M}\sum_{i=0}^{M'} c_{ijk}K_i u_j = f_k, \qquad k = 0,\ldots,M, \tag{10}$$

where $(f_k)_l = \mathbb{E}\left[\int_D f(x)\phi_l(x)\psi_k\,dx\right]$, $(K_i)_{lm} = \int_D k_i(x)\phi_l(x), \phi_m(x)\,dx$, and $c_{ijk} = \mathbb{E}[\psi_i\psi_j\psi_k]$. Each one of the blocks K_i is thus a deterministic stiffness matrix given by $k_i(x)$ of size $N_{\text{dof}} \times N_{\text{dof}}$, where N_{dof} is the number of spatial degrees of freedom. The system (10) is given by a global stochastic Galerkin matrix of size $(M+1)N_{\text{dof}} \times (M+1)N_{\text{dof}}$, consisting of $N_{\text{dof}} \times N_{\text{dof}}$ blocks $K^{(j,k)}$, and it can be written as

$$\begin{bmatrix} K^{(0,0)} & K^{(0,1)} & \cdots & & K^{(0,M)} \\ & \ddots & & & \\ \vdots & & K^{(k,k)} & & \vdots \\ & & & \ddots & \\ K^{(M,0)} & K^{(M,1)} & \cdots & & K^{(M,M)} \end{bmatrix} \begin{bmatrix} u_{(0)} \\ \vdots \\ u_{(k)} \\ \vdots \\ u_{(M)} \end{bmatrix} = \begin{bmatrix} f_{(0)} \\ \vdots \\ f_{(k)} \\ \vdots \\ f_{(M)} \end{bmatrix}, \tag{11}$$

where each of the blocks $K^{(j,k)}$ is obtained as

$$K^{(j,k)} = \sum_{i=0}^{M'} c_{ijk}K_i. \tag{12}$$

We note that the first diagonal block is obtained by the 0th order polynomial chaos expansion and therefore corresponds to the deterministic problem obtained using the mean value of the coefficient k_0. In particular,

$$K^{(0,0)} = K_0.$$

In the next section we discuss the structure of the global stochastic Galerkin matrix from (11) in somewhat more detail.

3. HIERARCHICAL STRUCTURE OF THE MATRICES

The stochastic Galerkin matrices have recently received quite a lot of attention; cf., e.g., [9, 14, 18]. The key role in their block structure is played by the constants c_{ijk} and the upper bound of the summation in (12). In general, there are two types of block sparsity patterns. The first type, typically regarded as block sparse, is associated with use of only the linear terms $\xi_1, \ldots, \xi_N$ such as appearing in a Karhunen-Loève expansion of $k\left(x, \xi\right)$; cf. Fig. 1(b). The second type, block dense, is associated with a general (nonlinear in ξ's) form of the expansion as in (4). We note that due to our setup with $P' = 2P$, we obtain that $M' \gg M$ from (8), and the matrices are in general fully dense; cf. Fig. 1(f). Recently, we proposed a preconditioner suitable for iterative solution of block sparse stochastic Galerkin systems [15]. Here, we will focus on a design of preconditioners for more general, block dense, linear systems.

The structure of the global stochastic Galerkin matrix can be understood through knowledge of the coefficient matrix c_P with entries $c_P\left(j,k\right) = \sum_{i=0}^{M'} c_{ijk}$ where $j, k = 0, \ldots, M$, and the value of M follows from the fact that we use the Pth order polynomial expansion; cf. (8). In general, let us consider some ℓth order polynomial expansion, such that $1 \leq \ell \leq P$. It is easy to see that the corresponding coefficient matrix c_ℓ will have a hierarchical structure,

$$c_\ell = \begin{bmatrix} c_{\ell-1} & b_\ell^T \\ b_\ell & d_\ell \end{bmatrix}, \qquad \ell = 1, \ldots, P, \tag{13}$$

where $c_{\ell-1}$ are the first principal submatrices corresponding to the the $(\ell-1)$th order polynomials expansion. We note that even though the matrices c_ℓ are symmetric, the global stochastic Galerkin matrix in (11) will be symmetric only if each one of the stiffness matrices K_i is symmetric. Clearly, all matrices K_i will have the same sparsity pattern.

In either case, the block sparse or the block dense, the linear system (11) can be written as

$$A_P u_P = f_P, \tag{14}$$

where the global Galerkin matrix A_P has the hierarchical structure, cf. Fig. 3(a), given as

$$A_\ell = \begin{bmatrix} A_{\ell-1} & B_\ell \\ C_\ell & D_\ell \end{bmatrix}, \qquad \ell = P, \ldots, 1, \tag{15}$$

and $A_0 = K_0$ is the matrix of the mean. Although for the model problem (1) and (2) it holds that $C_\ell = B_\ell^T$, for $\ell = 1, \ldots P$, we will use the general notation of (15) for the sake of generality.

Alternatively, let us consider a hierarchical splitting of the matrix A_P, cf. Fig. 3(b), as

$$A_P = \begin{bmatrix} D_0 & & & F_0 & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & E_\ell & D_\ell & F_\ell & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & E_P & & & D_P \end{bmatrix}. \tag{16}$$

The blocks D_ℓ, $\ell = 1, \ldots, P$, are the same in (15) and (16), and we have also, for convenience, denoted $D_0 = A_0$.

The decomposition (15) has served as a starting point for the *hierarchical Schur complement preconditioner* in [15], and we will use the decomposition (16) to formulate the *hierarchical block symmetric Gauss-Seidel preconditioner*. Since we are interested here only in the preconditioners that are *block* and *symmetric* we will drop the two words for brevity as this shall not cause any confusion. We only note that in the numerical experiments section we compare a variant of the *hierarchical* Gauss-Seidel method with the block (non-hierarchical) Gauss-Seidel method, and the usual (row) Gauss-Seidel method is not considered here at all.

In this paper we make an assumption that it is possible to factorize, e.g., by the LU-decomposition, the diagonal blocks of the global stochastic Galerkin matrix or, at least, that we have a preconditioner readily available.

4. APPROXIMATE AND TRUNCATED HIERARCHICAL PRECONDITIONERS

In the block sparse case, the matrices D_ℓ are block diagonal, as in Fig. 1(b). However, considering the underlying coefficient k as a random field with lognormal distribution, all of the matrices D_ℓ are block dense, as in Fig. 1(f). The word *approximate* will refer to the fact that we will approximate the (inverses of the) matrices D_ℓ by (inverses of) their diagonal blocks in the action of our preconditioners. Next, a global matrix-vector multiplication in an iterative solver can be performed using formula (12); i.e., one needs to store only the constants c_{ijk} and the matrices K_i for the so-called called MAT-VEC operations [10]. Here, we are interested in preconditioners that would rely only on the MAT-VEC operations as well. The top panels in Fig. 2 show a decaying trend of the norms of matrices K_i with increasing index i. This indicates, with respect to the structure of the coefficients c_{ijk}, that the lower order terms in the general P'th order expansion might dominate, depending on the *coefficient of variation* CoV, over the higher order terms; cf. lower panels in Fig. 2. Now, the idea is to *truncate*, i.e., selectively drop some of the matrices K_i from the MAT-VEC operations in the action of the preconditioner. There are two possible strategies of this truncation.

First, the *standard truncation* is defined as follows. Let ℓ_t be maximal order of the polynomial expansion that we would like to include in the MAT-VEC operations. Then, we define the degree of truncation $M_t \leq M'$ as

$$M_t + 1 = \frac{(N + \ell_t)!}{N!\,\ell_t!}, \tag{17}$$

and the matrices K_i, for all $i = 0, \ldots, M_t$ are included in the MAT-VEC operations.

Alternatively, in the *adaptive truncation* we select matrices K_i that a priori satisfy certain criteria, e.g., such that their norm is greater than a given threshold. This is computationaly slightly more demanding because it requires finding and bookkeeping a subset of matrices K_i where $i \in \mathcal{M}_t \subseteq \{0, \ldots, M'\}$; however, for higher values of CoV, or in the case of a non-monotonous decay of the matrices K_i, this might be the preferred truncation strategy.

The algorithm of the truncated block matrix-vector multiplication (tMAT-VEC) is precisely defined as follows:

Algorithm 1. [tMAT-VEC]
The truncated MAT-VEC multiplication $w = \mathcal{P}_\ell v$ is performed as

$$w_{(j)} = \sum_{k=0}^{M} \sum_{i \in \mathcal{M}_t} c_{ijk} K_i v_{(k)}, \tag{18}$$

where j, k are suitable subsets of $\{0, \ldots, M\}$ selecting blocks of the global Galerkin matrix to be multiplied with, and the truncation is defined by a set $\mathcal{M}_t \subseteq \{0, \ldots, M'\}$ selecting the matrices K_i for the MAT-VEC operation.

The notation $\mathcal{P}_\ell$ used in Algorithm 1 will correspond in Algorithms 2 and 3 to either $\mathcal{B}_\ell$, $\mathcal{C}_\ell$, $\mathcal{E}_\ell$, or $\mathcal{F}_\ell$, and it will denote the corresponding truncated variants of the block matrix-vector multiplications by the blocks B_ℓ, C_ℓ, E_ℓ, or F_ℓ from (15) and (16). We note that the full MAT-VEC operation is obtained by setting $\ell_t = P'$ in (17) so that $M_t = M'$.

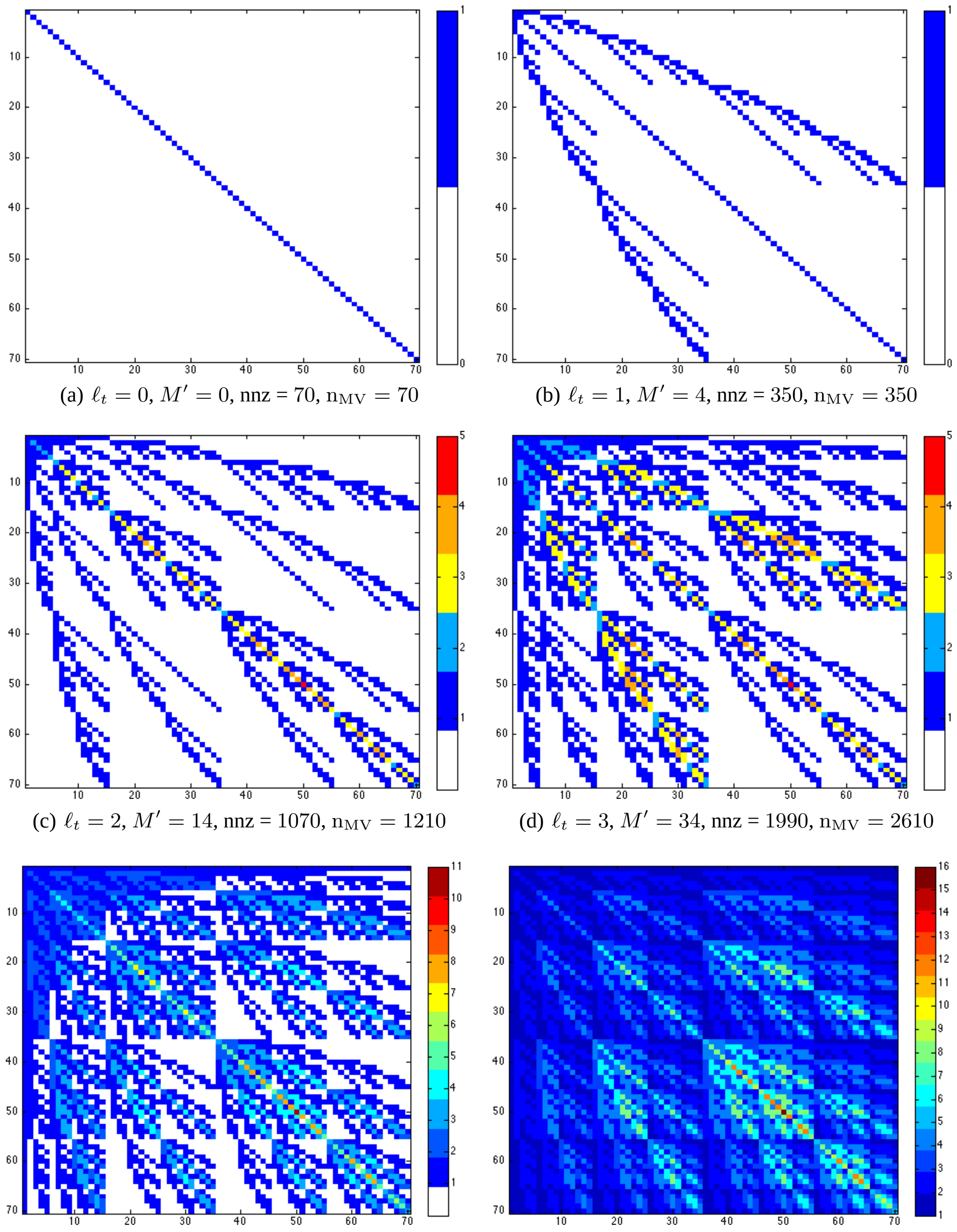
Next, let us introduce some additional notation. Considering (15), problem (14) can be rewritten as

$$\begin{bmatrix} A_{P-1} & B_P \\ C_P & D_P \end{bmatrix} \begin{bmatrix} u_P^{P-1} \\ u_P^P \end{bmatrix} = \begin{bmatrix} f_P^{P-1} \\ f_P^P \end{bmatrix}, \tag{19}$$

which motivates the following notation for a vector x_ℓ, where $\ell = 1, \ldots, P$, cf. Fig. 3, as

$$x_\ell = \begin{bmatrix} x_\ell^{\ell-1} \\ x_\ell^\ell \end{bmatrix}, \text{ or } x_\ell = \begin{bmatrix} x_{(0)} \\ \vdots \\ x_{(\ell)} \end{bmatrix}. \quad \text{So then } x_\ell^{\ell-1} = \begin{bmatrix} x_{(0)} \\ \vdots \\ x_{(\ell-1)} \end{bmatrix}, \text{ and } x_\ell^\ell = x_{(\ell)}. \tag{20}$$

Note that $x_{(0)} = x_1^0 = x_0$, and we will also for brevity denote $x_\ell = x_{(0:\ell)}$. We are ready to formulate now the two preconditioners. First, we recall the algorithm of the hierarchical Schur complement preconditioner (hS) from [15, Algorithm 5]. Then, we formulate the algorithm of the hierarchical Gauss-Seidel preconditioner (hGS).



(a) $\ell_t = 0$, $M' = 0$, nnz = 70, $\mathrm{n_{MV}} = 70$

(b) $\ell_t = 1$, $M' = 4$, nnz = 350, $\mathrm{n_{MV}} = 350$

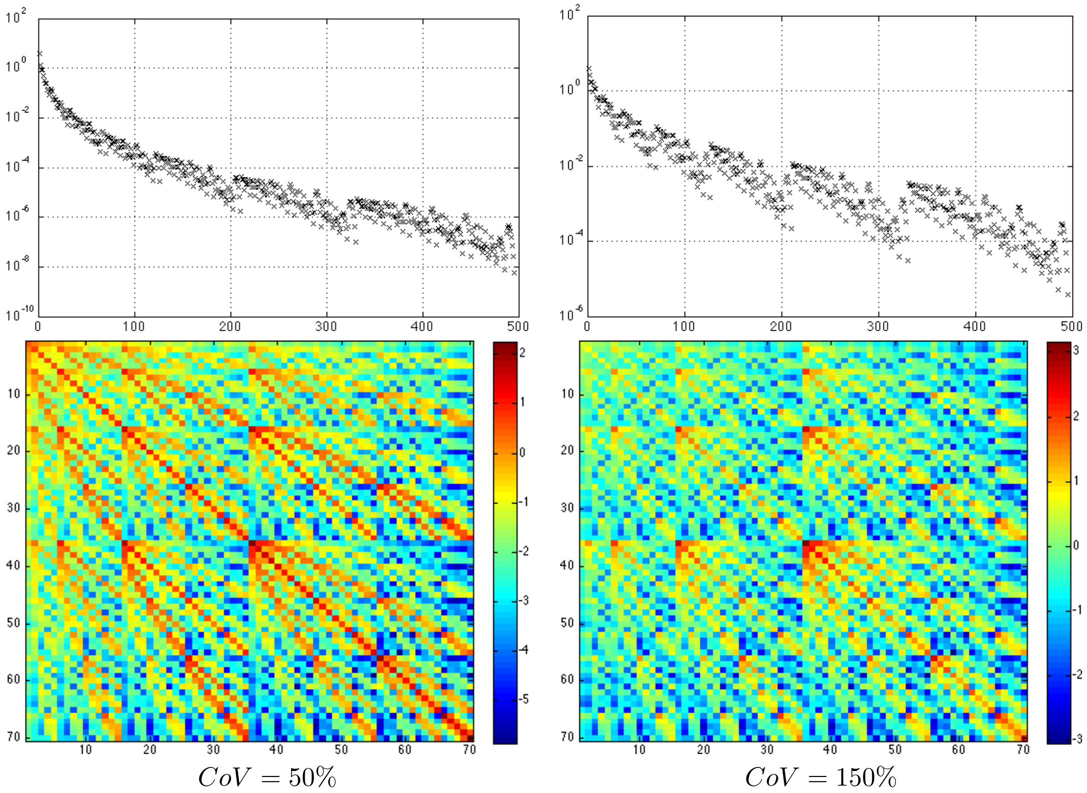
(c) $\ell_t = 2$, $M' = 14$, nnz = 1070, $\mathrm{n_{MV}} = 1210$

(d) $\ell_t = 3$, $M' = 34$, nnz = 1990, $\mathrm{n_{MV}} = 2610$

(e) $\ell_t = 4$, $M' = 69$, nnz = 3090, $\mathrm{n_{MV}} = 4980$

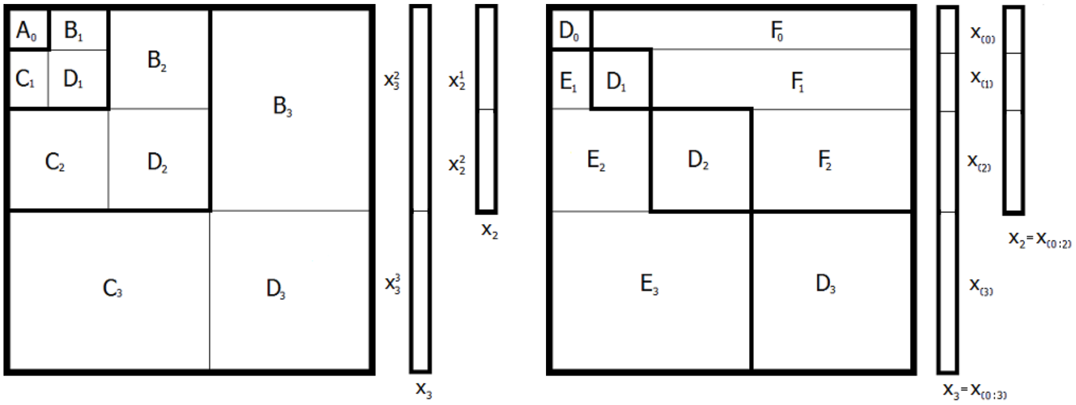
(f) $\ell_t = 8$, $M' = 494$, nnz = 4900, $\mathrm{n_{MV}} = 12{,}585$

FIG. 1: Block sparsity structures of the coefficient matrices c_P with entries $c^{(j,k)} = \sum_{i=0}^{M'} c_{ijk}$, where $j, k = 0, \ldots, M$ and $M + 1 = (N + P)!/N!P!$. Here $N = P = 4$, so the size of all the matrices c_P is $M + 1 = 70$, and $M' + 1 = (N + \ell_t)!/N!\ell_t!$, where ℓ_t is different in each panel. The colors indicate number of summations in a position (j, k), nnz is the number of nonzero coefficients in the matrix c_P, and $\mathrm{n_{MV}}$ is the total number of summations (=MAT-VEC operations).



$CoV = 50\%$ $CoV = 150\%$

FIG. 2: Norms of the $M'+1 = 495$ stiffness matrices (top panels) in the representation of the lognormal random field using $N = 4$ random variables and polynomial expansion of order $P' = 8$, and the decadic logarithm of the weighted coefficient matrix where in each position (j, k) the coefficient is obtained as weighted sums $\sum_{i=0}^{494} c_{ijk} \times \mathrm{norm}\,(K_i)$ for two coefficients of variation $CoV = 50\%$ (left panel) and $CoV = 150\%$ (right panel). The decay of norms of the matrices K_i illustrated by the top panels motivates the truncation of the MAT-VEC operations introduced in Algorithm 1, which is used in the construction of the truncated preconditioners defined in Algorithms 2 and 3. In particular, the construction is guided by the weighted sums illustrated by the lower panels.



(a) hierarchical Schur complement preconditioner (b) hierarchical Gauss-Seidel preconditioner

FIG. 3: Structure of the stochastic Galerkin matrix (15) and of the hierarchical Schur complement preconditioner from Algorithm 2 (left panel), and the structure of the hierarchical Gauss-Seidel preconditioner from Algorithm 3 (right panel). Both panels also illustrate the structure of vectors introduced in equation (20).

Algorithm 2. [Hierarchical Schur complement preconditioner (hS)]
The preconditioner $M_{hS} : r_P \longmapsto v_P$ for system (14) is defined as follows: **for** $\ell = P, \ldots 1$**,**

split the residual r_ℓ, based on the hierarchical structure of matrices, as

$$r_\ell = \begin{bmatrix} r_\ell^{\ell-1} \\ r_\ell^{\ell} \end{bmatrix}, \tag{21}$$

compute the pre-correction as

$$g_{\ell-1} = r_\ell^{\ell-1} - \mathcal{B}_\ell \mathcal{D}_\ell^{-1} r_\ell^\ell. \tag{22}$$

If $\ell > 1$, set

$$r_{\ell-1} = g_{\ell-1}. \tag{23}$$

Else (if $\ell = 1$), solve the system $A_0 v_0 = g_0$.

end for $\ell = 1, \ldots P$**,**

compute the post-correction, i.e., set $v_\ell^{\ell-1} = v_{\ell-1}$, solve

$$v_\ell^\ell = \mathcal{D}_\ell^{-1} \left(r_\ell^\ell - \mathcal{C}_\ell v_\ell^{\ell-1} \right), \tag{24}$$

and concatenate

$$v_\ell = \begin{bmatrix} v_\ell^{\ell-1} \\ v_\ell^{\ell} \end{bmatrix}. \tag{25}$$

If $\ell < P$, set $v_{\ell+1}^\ell = v_\ell$.

end

Algorithm 3. [Hierarchical Gauss-Seidel preconditioner (hGS)]
The preconditioner $M_{hGS} : r_P \longmapsto v_P$ for system (14) is defined as follows:
set the initial solution u_P and update it in the following steps,

$$v_{(0)} = \mathcal{D}_0^{-1} \left(r_{(0)} - \mathcal{F}_0 v_{(1:P)} \right), \tag{26}$$

for $\ell = 1, \ldots P - 1$**,**

$$v_{(\ell)} = \mathcal{D}_\ell^{-1} \left(r_{(\ell)} - \mathcal{E}_\ell v_{(0:\ell-1)} - \mathcal{F}_\ell v_{(\ell+1:P)} \right), \tag{27}$$

end

$$v_{(P)} = \mathcal{D}_P^{-1} \left(r_{(P)} - \mathcal{E}_P v_{(0:P-1)} \right), \tag{28}$$

for $\ell = P - 1, \ldots 1$**,**

$$v_{(\ell)} = \mathcal{D}_\ell^{-1} \left(r_{(\ell)} - \mathcal{E}_\ell v_{(0:\ell-1)} - \mathcal{F}_\ell v_{(\ell+1:P)} \right), \tag{29}$$

end

$$v_{(0)} = \mathcal{D}_0^{-1} \left(r_{(0)} - \mathcal{F}_0 v_{(1:P)} \right). \tag{30}$$

In our implementation, we initialize $u_P = 0$ in Algorithm 3. Then the multiplications of $\mathcal{F}_\ell$, $\ell = 0, \ldots, P - 1$, by this zero vector will vanish from (26)–(27). This also reduces the computational cost of the MAT-VEC operations.

We have already mentioned that the *approximate* variants of the two preconditioners are obtained by replacing the solves with the blocks $\mathcal{D}_\ell$, $\ell = 1, \ldots, P$, by the corresponding diagonal block solves; cf. the right panel of Fig. 4.

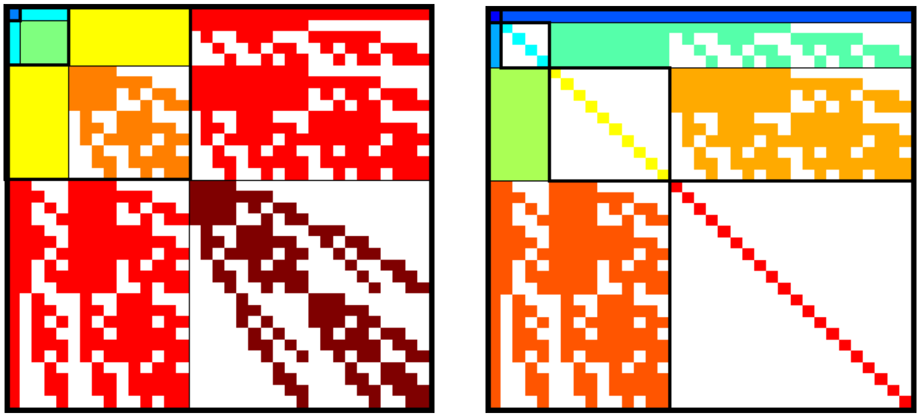


FIG. 4: Block sparsity structure of the hierarchical Schur complement preconditioner from Algorithm 2 (left panel), and the block sparsity structure of the approximate hierarchical Gauss-Seidel preconditioner derived from Algorithm 3 (right panel). Both preconditioners are also truncated, setting $\ell_t = P$; cf. Fig. 1(e).

4.1 Computational Cost Considerations

In order to compare the computational cost of the preconditioners, we first recall two common solution algorithms against which comparisons will be discussed. First, let us define matrices G_α, $\alpha = 0, \dots, M'$, using the coefficients c_{ijk} as

$$G_\alpha = c_{\alpha jk}, \qquad j = 0, \dots, M, \quad k = 0, \dots, M.$$

Let the symbol $\otimes$ denote the Kronecker product. The mean-based preconditioner [10, 11] is defined as

$$M_{mb} = \text{diag}\,(G_0) \otimes A_0. \tag{31}$$

Next, let us define a matrix

$$G = \sum_{\alpha=0}^{M'} \frac{\text{tr}\left(A_\alpha^T A_0\right)}{\text{tr}\left(A_0^T A_0\right)} G_\alpha.$$

The Kronecker product preconditioner [12] is defined as

$$M_K = G \otimes A_0 = (G \otimes I_{N_\text{dof}})\,(I_{M+1} \otimes A_0) = \left(I_{M+1} \otimes A_0^{-1}\right)\left(G^{-1} \otimes I_{N_\text{dof}}\right). \tag{32}$$

The multiplication of $\left(I_{M+1} \otimes A_0^{-1}\right)$ by a vector is essentially the same as an application of the mean-based preconditioner, and an efficient implementation of $\left(G^{-1} \otimes I_\text{ndof}\right)$ is described in [22]. We can immediately see from (31) that the mean based-preconditioner requires $M+1$ solves with the matrix A_0 of size $(N_\text{dof} \times N_\text{dof})$, and from (32) that the Kronecker product preconditioner requires in addition N_dof solves with the matrix G of size $((M+1) \times (M+1))$, which requires $\mathcal{O}\left(N_\text{dof} \times (M+1)^2\right)$ operations if a Cholesky decomposition of G is available. Both of our approximate hierarchical preconditioners require $2(M+1)$ solves with the diagonal blocks of the global Galerkin matrix, and a certain number of matrix-vector multiplications with stiffness matrices K_i. The number of these multiplications depends on the degree of truncation of the MAT-VEC operations, and it is denoted by $nz\,(c_{ijk})$ in Tables 5 and 6. Clearly, the Kronecker product preconditioner is very efficient if (i) it is possible to obtain a Cholesky decomposition of G, and (ii) the number of spatial degrees of freedom N_dof is not too large. Moreover, it was assumed in [11, 12] that a solve with A_0 can be performed in $\mathcal{O}\left(N_\text{dof}\right)$ operations using a multigrid solver. However, in situations when the multigrid solver is not suitable or the size of the discretized spatial problem is too large, the question of selecting block solvers becomes much more delicate. For example, we might even consider "inner" Krylov iterations for the solves with the diagonal blocks. The preconditioner would thus become variable and we would also need to make a careful choice of the Krylov iterative method used for the global ("outer") iterations. Our initial experiments with such choice were successful, and the present work is the first step in the development of such solvers.

5. NUMERICAL EXPERIMENTS

We have implemented the stochastic Galerkin finite element method for the model problem (1)–(2) on a two-dimensional domain with dimensions $[0,1] \times [0,1]$ uniformly discretized by 10×10 Lagrangian finite elements. The mean value of the lognormal field was set to $\mu_{\log} = 1$, the correlation length $L = 0.5$, and we have used the covariance kernel in (6) as

$$C_g(x,y) = \sigma^2 \exp\left(-\frac{\|x-y\|_1}{L}\right), \tag{33}$$

where σ denotes the standard deviation of the underlying Gaussian field. In the experiments reported in Tables 1, 2, and 4 we have set $\sigma_{\log} = 1$, and so the coefficient of variation $CoV = \sigma_{\log}/\mu_{\log} = 100\%$. We have compared convergence of the flexible conjugate gradients, the mean-based preconditioner (mb), the Kronecker product preconditioner (K), the hierarchical Schur complement preconditioner (hS), the approximate hierarchical Schur complement preconditioner (ahS), symmetric (block) Gauss-Seidel preconditioner (GS), and the approximate hierarchical Gauss-Seidel preconditioner (ahGS). We note that in all cases we have observed essentially the same convergence of the standard and flexible conjugate gradients [19]. We have also tested direct and iterative solvers, with the same tolerance as for the outer iterations, for the inner block solves. Our experiments indicate that the convergence in terms of outer (global) iteration counts is not sensitive to the particular choice of inner block solves. The results are summarized in Tables 1–6. First, we have compared the convergence using the preconditioners with no truncation, varying either of the stochastic dimension N, the order of polynomial expansion P, the coefficient of variation CoV or the mesh size h, and keeping other parameters fixed. These results are, respectively, reported in Tables 1–4. Looking at

TABLE 1: Convergence of flexible conjugate gradients for the global matrix A obtained by the gPC expansion of the lognormal field with $CoV = \sigma_{\log}/\mu_{\log} = 100\%$, and $\mu_{\log} = 1$, preconditioned by the mean-based preconditioner (mb), Kronecker product preconditioner (K), hierarchical Schur complement (hS), approximate hierarchical Schur complement (ahS), block Gauss-Seidel (GS). and approximate hierarchical block Gauss-Seidel (ahGS) preconditioners. Polynomial degree is fixed to $P = 4$, and the stochastic dimension N is variable. Here, $ndof$ is the dimension of A, it is the number of iterations with the relative residual tolerance 10^{-8}, and κ is the condition number estimate from the Lánczos sequence in conjugate gradients

Setup		**mb**		**K**		**hS**		**ahS**		**GS**		**ahGS**	
N	$ndof$	it	κ	it	κ	it	κ	it	κ	it	κ	it	κ
1	605	48	28.76	18	3.87	15	3.40	15	3.40	15	3.42	15	3.42
2	1815	61	37.16	32	10.10	16	3.62	27	8.06	17	3.75	16	3.45
3	4235	62	38.07	32	10.30	16	3.76	31	10.77	17	3.74	18	4.35
4	8470	66	43.65	37	13.60	16	4.17	38	15.28	19	4.29	19	4.74

TABLE 2: Convergence of flexible conjugate gradients for the global matrix A preconditioned by the mean-based preconditioner (mb), Kronecker product preconditioner (K), hierarchical Schur complement (hS), approximate hierarchical Schur complement (ahS), block Gauss-Seidel (GS), and approximate hierarchical block Gauss-Seidel (ahGS) preconditioners. Stochastic dimension is fixed to $N = 4$, and the polynomial degree P is variable. The other headings are the same as in Table 1

Setup		**mb**		**K**		**hS**		**ahS**		**GS**		**ahGS**	
P	$ndof$	it	κ	it	κ	it	κ	it	κ	it	κ	it	κ
1	605	15	3.50	14	2.44	7	1.39	11	1.76	8	1.39	8	1.31
2	18	28	8.95	21	4.74	10	1.93	16	3.04	12	1.97	11	1.76
3	4235	44	20.04	29	8.28	13	2.80	24	6.09	15	2.87	14	2.58
4	8470	66	43.65	37	13.60	16	4.17	38	15.28	19	4.29	19	4.74

TABLE 3: Convergence of flexible conjugate gradients for the global matrix A preconditioned by the mean-based preconditioner (mb), Kronecker product preconditioner (K), hierarchical Schur complement (hS), approximate hierarchical Schur complement (ahS), block Gauss-Seidel (GS), and approximate hierarchical block Gauss-Seidel (ahGS) preconditioners. Stochastic dimension and polynomial degree are fixed to $N = P = 4$, and the coefficient of variation CoV is variable. The other headings are the same as in Table 1

Setup	**mb**		**K**		**hS**		**ahS**		**GS**		**ahGS**	
CoV (%)	it	κ	it	κ	it	κ	it	κ	it	κ	it	κ
25	16	3.24	14	2.37	7	1.18	8	1.25	7	1.18	6	1.12
50	29	9.36	22	4.96	10	1.78	14	2.45	11	1.77	10	1.62
75	46	22.21	30	8.73	13	2.85	23	6.01	15	2.82	14	2.69
100	66	43.65	37	13.60	16	4.17	38	15.28	19	4.29	19	4.74
125	85	72.76	45	19.79	19	5.54	58	36.34	23	5.98	26	8.21
150	103	107.07	52	27.02	21	6.85	84	77.73	26	7.75	35	13.74

TABLE 4: Convergence of flexible conjugate gradients for the global matrix A preconditioned by the mean-based preconditioner (mb), Kronecker product preconditioner (K), hierarchical Schur complement (hS), approximate hierarchical Schur complement (ahS), block Gauss-Seidel (GS), and approximate hierarchical block Gauss-Seidel (ahGS) preconditioners. Stochastic dimension and polynomial degree are fixed to $N = P = 4$, the coefficient of variation is $CoV = 100\%$, and the mesh size h is variable. The other headings are the same as in Table 1

Setup		**mb**		**K**		**hS**		**ahS**		**GS**		**ahGS**	
h	$ndof$	it	κ	it	κ	it	κ	it	κ	it	κ	it	κ
$1/5$	2520	59	40.62	35	14.06	15	3.84	35	15.43	18	3.99	19	4.91
$1/10$	8470	66	43.65	37	13.60	16	4.17	38	15.28	19	4.29	19	4.74
$1/15$	17920	68	44.42	39	13.87	16	4.24	39	15.81	19	4.38	20	4.72
$1/20$	30870	69	44.89	39	14.07	17	4.25	40	16.23	19	4.37	20	4.78
$1/25$	47320	69	44.94	40	14.13	17	4.26	41	16.38	20	4.40	20	4.81
$1/30$	67270	71	45.11	40	14.04	17	4.26	41	16.38	19	4.37	20	4.75

the tables, we see that the hS preconditioner performs best, followed by the GS preconditioner. On the other hand, the convergence of the ahS preconditioner quickly deteriorates, whereas the convergence of the aGS is quite similar to the GS. In fact, it is quite interesting to note from Table 3 that the convergence of the ahS begins to deteriorate for values of $CoV > 25\%$; however, the aGS remains comparable to the GS preconditioner for values of CoV at around 100%. From Table 4 we see that the dependence on the mesh size h is not very significant. Tables 5 and 6 contain results obtained for variable coefficients of variation CoV and different truncation strategy of the MAT-VEC operations in the action of the preconditioners, which is guided by the decay of the norms of the stiffness matrices K_i obtained from the finite element discretization of the generalized polynomial chaos expansion of the random coefficient $k\left(x,\omega\right)$; cf. Fig. 2. Table 5 contains results obtained using the *standard truncation*. First, we note that setting $\ell_t = 0$ yields to use of only the matrix K_0 in the action of the ahS, GS, and ahGS preconditioners. So the resulting preconditioners are in this case block diagonal and because they are symmetric, their application is the same as using twice the mean-based preconditioner. On the contrary, the performance of the hS preconditioner might be slightly better even with $\ell_t = 0$, in particular for higher values of CoV, but this is because the hS preconditioner performs solves with the full submatrices D_ℓ. However, we can see that this does not correspondingly improve the convergence, and use of the off-diagonal blocks seems to be warranted. Indeed, the convergence improves with more included into the MAT-VEC operations in the action of all of the preconditioners. In particular, we see that for values of CoV at around 25% it is sufficient to include only the five matrices corresponding to the linear terms of the coefficient expansion. On the other hand, even for higher values of CoV there seems to be no reason for the GS and aGS preconditioners to use the

TABLE 5: Convergence of the flexible conjugate gradients for the global stochastic Galerkin matrix preconditioned by the mean-based preconditioner (mb), Kronecker product preconditioner (K), hierarchical Schur complement (hS), approximate hierarchical Schur complement (ahS), block Gauss-Seidel (GS), and approximate hierarchical block Gauss-Seidel (ahGS) preconditioners with a variable degree of truncation of the MAT-VEC operations in the action of the preconditioner. Here ℓ_t is the maximum polynomial order of the coefficient expansion used in the construction of the preconditioner, so that $M_t + 1$ is the degree of truncation of the MAT-VEC operations, i.e., the number of retained matrices, $nz(c_{ijk})$ is the number of nonzeros in the truncated tensor c_{ijk}, it is the number of iterations with the relative residual tolerance 10^{-8}, and κ is the condition number estimate from the Lánczos sequence in flexible conjugate gradients

Setup			**hS**		**ahS**		**GS**		**ahGS**	
ℓ_t	$M_t + 1$	$nz(c_{ijk})$	it	κ	it	κ	it	κ	it	κ
$CoV = 25\%$ (mb: $it = 16$ $\kappa = 3.24$, K: $it = 14$ $\kappa = 2.37$)										
0	1	70	16	3.20	16	3.19	16	3.19	16	3.19
1	5	350	8	1.27	8	1.33	7	1.23	7	1.23
2	15	1210	7	1.21	8	1.25	7	1.20	6	1.17
3	35	2610	7	1.18	8	1.25	7	1.17	6	1.11
4	70	4980	7	1.18	8	1.25	7	1.18	6	1.12
8	495	12585	7	1.18	8	1.25	7	1.18	6	1.12
$CoV = 50\%$ (mb: $it = 29$ $\kappa = 9.36$, K: $it = 22$ $\kappa = 4.96$)										
0	1	70	28	8.90	28	8.99	28	8.99	28	8.99
1	5	350	14	2.54	15	2.83	14	2.54	14	2.54
2	15	1210	13	2.47	14	2.74	12	2.15	11	2.12
3	35	2610	10	1.79	14	2.44	10	1.68	10	1.69
4	70	4980	10	1.79	13	2.39	11	1.78	10	1.68
8	495	12585	10	1.78	14	2.45	11	1.77	10	1.62
$CoV = 100\%$ (mb: $it = 66$ $\kappa = 43.65$, K: $it = 37$ $\kappa = 13.60$)										
0	1	70	53	32.88	58	39.94	58	39.94	58	39.94
1	5	350	32	11.83	35	14.29	33	12.88	33	12.88
2	15	1210	32	12.31	35	15.13	28	9.35	29	10.05
3	35	2610	20	5.03	36	14.32	19	4.64	22	6.23
4	70	4980	20	5.24	32	11.92	19	4.55	20	5.33
8	495	12585	16	4.17	38	15.28	19	4.29	19	4.74
$CoV = 150\%$ (mb: $it = 103$ $\kappa = 107.067$, K: $it = 52$ $\kappa = 27.0203$)										
0	1	70	71	61.44	89	90.18	89	90.18	89	90.18
1	5	350	51	29.92	59	39.66	57	36.85	57	36.85
2	15	1210	51	30.18	60	42.06	46	24.26	50	27.53
3	35	2610	31	11.06	71	55.98	30	10.17	40	19.53
4	70	4980	32	12.05	58	38.08	28	9.42	34	13.86
8	495	12585	21	6.85	84	77.73	26	7.75	35	13.74

matrices obtained from the polynomials of higher order than the expansion of the solution; i.e., the computational cost of these preconditioners can be reduced more than twice. Finally, Table 6 contains results obtained with the *adaptive truncation*, where the tolerance τ has been set such that the matrices K_i for which $\max_{jk}(c_{ijk}) \times \mathrm{norm}(K_i) < \tau$ are dropped from the MAT-VEC operations in the action of the preconditioners. As before, only a few matrices need to be used in order to significantly improve the convergence. Moreover, it appears that setting the threshold τ too low might have a negative impact on the convergence of the ahS and ahGS preconditioners. In particular, rather surprisingly,

TABLE 6: Convergence of the flexible conjugate gradients for the global stochastic Galerkin matrix preconditioned by the mean-based preconditioner (mb), Kronecker product preconditioner (K), hierarchical Schur complement (hS), approximate hierarchical Schur complement (ahS), block Gauss-Seidel (GS), and approximate hierarchical block Gauss-Seidel (ahGS) preconditioners with a variable degree of truncation of the MAT-VEC operations in the action of the preconditioner. Here τ is the tolerance such that the matrices K_i for which $\max_{jk}(c_{ijk}) \times \mathrm{norm}(K_i) < \tau$ are dropped from the MAT-VEC operations in the action of the preconditioner and N_{adapt} is the number of retained matrices, $nz(c_{ijk})$ is the number of nonzeros in the truncated tensor c_{ijk}, it is the number of iterations with the relative residual tolerance 10^{-8}, and κ is the condition number estimate from the Lánczos sequence in flexible conjugate gradients

Setup			**hS**		**ahS**		**GS**		**ahGS**	
τ	N_{adapt}	$nz(c_{ijk})$	it	κ	it	κ	it	κ	it	κ
$CoV = 25\%$ mb: $it = 16$ $\kappa = 3.24$ K: $it = 14$ $\kappa = 2.37$										
10	5	345	10	1.63	10	1.58	10	1.57	10	1.57
1	13	877	7	1.20	8	1.24	7	1.18	6	1.12
0.1	32	2057	7	1.18	8	1.24	7	1.17	6	1.12
0	495	12585	7	1.18	8	1.25	7	1.18	6	1.12
$CoV = 50\%$ mb: $it = 29$ $\kappa = 9.36$ K: $it = 22$ $\kappa = 4.96$										
10	11	677	12	2.18	13	2.24	11	1.83	11	1.83
1	32	1958	11	1.91	13	2.25	11	1.77	10	1.62
0.1	86	4765	10	1.78	14	2.41	11	1.77	10	1.62
0	495	12585	10	1.78	14	2.45	11	1.77	10	1.62
$CoV = 100\%$ mb: $it = 66$ $\kappa = 43.65$ K: $it = 37$ $\kappa = 13.60$										
100	2	92	54	34.95	63	47.94	65	51.28	65	51.28
10	25	1241	27	9.12	25	7.14	23	6.98	19	4.56
1	97	4731	18	4.48	33	12.29	19	4.55	18	4.10
0.1	219	8202	17	4.18	37	14.90	19	4.31	19	4.66
0	495	12585	16	4.17	38	15.28	19	4.29	19	4.74
$CoV = 150\%$ mb: $it = 103$ $\kappa = 107.07$ K: $it = 52$ $\kappa = 27.02$										
100	10	336	66	50.22	68	48.84	70	50.68	70	50.68
10	55	2450	30	10.89	44	20.38	29	9.49	25	6.95
1	171	6338	23	7.00	73	59.57	27	7.98	32	11.48
0.1	313	9714	22	6.86	83	76.38	26	7.74	35	13.54
0.01	436	11741	21	6.84	84	77.69	26	7.75	35	13.72
0	495	12585	21	6.85	84	77.73	26	7.75	35	13.74

even with higher values of τ the convergence of the truncated ahGS is comparable to the convergence of GS with no truncation (setting $\tau = 0$). We can in fact see that, e.g., for the values of CoV equal to 100% and 150%, with value $\tau = 10$, the convergence of the ahGS is essentially the same as the convergence of the GS preconditioner with no truncation.

6. CONCLUSION

We have proposed two novel strategies for constructing preconditioners for the iterative solution of the systems of linear algebraic obtained from the stochastic Galerkin finite element discretizations. Our main focus was on a class of problems with coefficients characterized by arbitrary distributions that generally yield fully block dense structure of global stochastic Galerkin matrices. The preconditioners take an advantage of the hierarchical structure of these matrices. We have, in particular, examined variants of the hierarchical Schur complement preconditioner proposed recently by the authors [15], and variants of the hierarchical block symmetric Gauss-Seidel preconditioner.

The first variant, called an *approximate*, replaces solves with submatrices by the associated diagonal block solves. This variant thus combines the global Gauss-Seidel method with "local" block-diagonal preconditioner, which allows decoupling of blocks, and introduces a possibility for a parallelization in an implementation. Numerical experiments with our model problem indicate that whereas the performance of the approximate version of the hierarchical Schur complement preconditioner deteriorates with the increasing values of the coefficient of variation, the convergence of the approximate hierarchical Gauss-Seidel preconditioner is quite comparable to the (non-hierarchical) block Gauss-Seidel preconditioner for the values of the coefficient of variation equal up to 100%.

The second variant, called *truncated*, is based on a truncation of the sequence of block matrix-vector multiplications, called MAT-VEC operations, used in the action of the preconditioners. The truncation can be performed using either a standard or adaptive strategy, based on the monotonicity in the decay of the stiffness matrices, and further alleviates the computational cost of the preconditioners. Our numerical experiments indicate that truncation, in particular with adaptive strategy, might not be necessarily traded off for slower convergence rates.

We have therefore proposed two strategies that are combined for optimal performance. One strategy introduces decoupled diagonal block solves, and the other reduces the overall computational cost associated with the action of a preconditioner. Considering that a multiplication of a vector by the global stochastic Galerkin matrix in each iteration of a Krylov subspace method is, of course, performed by the full MAT-VEC multiplication with no truncation, in particular the truncated version of the approximate hierarchical Gauss-Seidel preconditioner thus offers an appealing combination of good convergence rates and a reasonable computational cost.

ACKNOWLEDGMENT

Support from DOE/ASCR is gratefully acknowledged. B. Sousedík has been also supported in part by the Grant Agency of the Czech Republic GA ČR 106/08/0403.

REFERENCES

1. Babuška, I., Nobile, F., and Tempone, R., A stochastic collocation method for elliptic partial differential equations with random input data, *SIAM Rev.*, 52(2):317–355, 2010.
2. Xiu, D. and Hesthaven, J. S., High-order collocation methods for differential equations with random inputs, *SIAM J. Sci. Comput.*, 27(3):1118–1139, 2005.
3. Babuška, I., Tempone, R., and Zouraris, G. E., Galerkin finite element approximations of stochastic elliptic partial differential equations, *SIAM J. Num. Anal.*, 42(2):800–825, 2005.
4. Babuška, I., Tempone, R., and Zouraris, G., Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation, *Comput. Methods Appl. Mech. Eng.*, 194(12-16):1251–1294, 2005.
5. Ghanem, R. G. and Spanos, P. D., *Stochastic Finite Elements: A Spectral Approach*, Springer, New York, 1991.
6. Loève, M., *Probability theory. II*, Vol. 46, 4th ed. Graduate Texts in Mathematics, Springer, New York, 1978.
7. Xiu, D. and Karniadakis, G. E., The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, 24(2):619–644, 2002.
8. Ghanem, R., The nonlinear Gaussian spectrum of log-normal stochastic processes and variables, *J. Appl. Mech.*, 66(4):964–973, 1999.
9. Matthies, H. G. and Keese, A., Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations, *Comput. Methods Appl. Mech. Eng.*, 194(12-16):1295–1331, 2005.
10. Pellissetti, M. F. and Ghanem, R. G., Iterative solution of systems of linear equations arising in the context of stochastic finite elements, *Adv. Eng. Software*, 31(8-9):607–616, 2000.
11. Powell, C. E. and Elman, H. C., Block-diagonal preconditioning for spectral stochastic finite-element systems, *IMA J. Numer. Anal.*, 29(2):350–375, 2009.

12. Ullmann, E., A Kronecker product preconditioner for stochastic Galerkin finite element discretizations, *SIAM J. Sci. Comput.*, 32(2):923–946, 2010.
13. Ullmann, E., Elman, H., and Ernst, O., Efficient iterative solvers for stochastic Galerkin discretizations of log-transformed random diffusion problems, *SIAM J. Sci. Comput.*, 34(2):A659–A682, 2012.
14. Rosseel, E. and Vandewalle, S., Iterative solvers for the stochastic finite element method, *SIAM J. Sci. Comput.*, 32(1):372–397, 2010.
15. Sousedík, B., Ghanem, R. G., and Phipps, E. T., Hierarchical Schur complement preconditioner for the stochastic Galerkin finite element methods, *Numer. Linear Algebra Appl.*, 21(1):136–151, 2013.
16. Keese, A. and Matthies, H. G., Hierarchical parallelisation for the solution of stochastic finite element equations, *Comput. Struct.*, 83(14):1033–1047, 2005.
17. Zheng, B., Lin, G., and Xu, J., Block triangular preconditioning for stochastic Galerkin method, *SIAM J. Sci. Comput.*, in Review, Arxiv:1304.1755, 2013.
18. Ernst, O. G. and Ullmann, E., Stochastic Galerkin matrices, *SIAM J. Matrix Anal. Appl.*, 31(4):1848–1872, 2010.
19. Notay, Y., Flexible conjugate gradients, *SIAM J. Sci. Comput.*, 22(4):1444–1460, 2000.
20. Saad, Y., A flexible inner-outer preconditioned GMRES algorithm, *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.
21. Van der Vorst, H. A. and Vuik, C., GMRESR: A family of nested GMRES methods, *Numer. Linear Algebra Appl.*, 1(4):369–386, 1994.
22. Moravitz Martin, C. D. and Van Loan, C. F., Shifted Kronecker product systems, *SIAM J. Matrix Anal. Appl.*, 29(1):184–198, 2006.