PARALLEL DOMAIN DECOMPOSITION METHODS FOR SOME STOCHASTIC PARTIAL DIFFERENTIAL EQUATIONS*

CHAO JIN†, XIAO-CHUAN CAI‡, AND CONGMING LI§

Abstract. We present a parallel multilevel domain decomposition preconditioned recycling Krylov subspace method for the numerical solution of stochastic elliptic problems, whose coefficients are assumed to be random fields with finite variance. The Karhunen-Loeve(KL) expansion and Galerkin method with double orthogonal polynomials are used to reformulate the stochastic elliptic problem into a sequence of uncoupled deterministic differential equations. We demonstrate that the parallel preconditioned recycling GMRES method for the sequence of algebraic systems can be used to compute the solutions efficiently. The efficiency and parallel scalability are discussed in paper.

Keywords Stochastic Partial Differential Equations, Domain Decomposition, Recycling Krylov Subspace Method.

1. Introduction. Many physical phenomena are modeled by partial differential equations whose coefficients are measured through experiments which often contain certain levels of randomness. To accurately model the physical phenomena, the randomness must be considered in the equations. In the last decade, many researchers have studied these so-called stochastic partial differential equations (SPDEs). One of the approaches for numerically solving SPDEs is known as stochastic Galerkin methods, see [1, 2, 8, 11, 15, 18] and references there. The method expands the random fields in the equation first. For example, [1, 2] use the Karhunen-Loeve expansion [14] while [11, 18] advocate the Wiener Chaos expansion [20] or generalized Chaos expansion of random fields. Then the SPDEs are reduced to a high dimensional deterministic equation. Once we obtain the deterministic equation, different numerical methods can be used to find the solution. Based on the solution, we can derive the statistics of the physical solution of the original stochastic partial differential equation, such as the mean, the variance, etc.

In this paper, we focus on elliptic problems with stochastic diffusion coefficients. For this type of SPDEs, we can use the double orthogonal basis [10, 3] to decouple the high dimensional deterministic equation and produce a sequence of independent equations. After the discretization, it becomes a sequence of linear systems:

$$A^{(i)}x^{(i)} = b^{(i)}, \quad i = 1, 2, \ldots \tag{1.1}$$

where the matrices $A^{(i)}$ and right-hand sides $b^{(i)}$ are similar but generally different from each other. The independency of these systems enables us to solve them in parallel. However, since these matrices are generally similar, we can do better in parallel processing by using a recycling Krylov subspace method [16]. This method retains a Krylov subspace while solving the previous system with GMRES and uses it to reduce the cost of solving the next system. In this paper, we extend the original algorithm in [16] to recycling Krylov subspace with Flexible GMRES, which allows the change of preconditioner during the iteration. We also modify the minimization of the least square error part of the original algorithm and present an efficient method to compute the minimizer y. For parallel processing, we use a recycling overlapping additive Schwarz domain decomposition method [17] to partition the computational domain, to distribute the work load and to precondition the linear systems. All the systems of (1.1) are solved by the additive Schwarz preconditioned restarted GMRES. We use the Portable Extensible Toolkit for Scientific computation (PETSc) package from Argonne National Laboratory [4] in our implementation. The scalability and parallel performance of our implementation are studied through numerical examples.

2. Stochastic Galerkin Method. In this section, we briefly describe the methodology to transform a stochastic elliptic partial differential equation to a sequence of independent systems. For our description, we use the model problem, an elliptic equation with a stochastic diffusion coefficient. See [10] for a complete description of the methodology.

2.1. The Problem and Stochastic Weak Formulation. First let us have a brief review of some notations.

*The research is supported in part by the National Science Foundation, CCR-0219190 and ACI-0305666, and in part by the Department of Energy, DE-FC02-01ER25479.

†Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309 (chao.jin@colorado.edu)

‡Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309 (cai@cs.colorado.edu)

§Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309 (cli@colorado.edu)

Given a probability space $(\Omega, \mathcal{A}, \mathcal{P})$ with sample space Ω, σ-algebra $\mathcal{A}$ and probability measure P, a complex-valued random variable is a function $\xi(\omega) : \Omega \to \mathbf{C}$. The probability distribution measure of ξ is defined on the Boreal sets B as $\mu(B) = P(\xi^{-1}(B))$. The mean or expected value of $\xi(\omega)$ is

$$\langle \xi \rangle = \int_\Omega \xi(\omega) dP(\omega) = \int_{\mathbf{C}} z d\mu(z) = \int_{\mathbf{C}} z\rho(z) dz, \tag{2.1}$$

where ρ is the probability density function of ξ. We also define the space

$$L^p(\Omega) = \left\{ \xi(\omega) | \int_\Omega |\xi|^p dP(\omega) < \infty \right\}.$$

A random filed $a(x, \omega) : D \times \Omega \to \mathbf{C}$, is a complex-valued function jointly measurable with respect to the Lebesgue measure on D and the probability measure P on Ω. Define the space

$$L^p(D \times \Omega) = \{ u(x, \omega) | \ \langle \| u(x, \omega) \|_{L^p(D)} \rangle < \infty \}.$$

The stochastic Sobolev spaces are defined analogously.

Now recall that for the classical deterministic 2-D elliptic equation:

$$\begin{cases} -\nabla \cdot (a(x) \nabla u(x)) &= f(x) \quad \text{on } D \in \mathbf{R}^2 \\ u(x) &= 0 \quad \text{on } \partial D, \end{cases} \tag{2.2}$$

the weak form of the problem is to find $u(x) \in H_0^1(D)$ such that

$$B[u, v] = (f, v) \quad \forall v \in H_0^1(D), \tag{2.3}$$

where

$$B[u, v] = \int_D a(x) \nabla u(x) \cdot \nabla v(x) dx, \quad (f, v) = \int_D f(x) v(x) dx.$$

If the diffusion coefficient is a random filed $a(x, \omega) \in L^2(D \times \Omega)$, the solution u will be involved with randomness too. Consequently, we have the stochastic elliptic equation

$$\begin{cases} -\nabla \cdot (a(x, \omega) \nabla u(x, \omega)) &= f(x) \quad \text{on } D \in \mathbf{R}^2, \ \omega \in \Omega \\ u(x, \omega) &= 0 \quad \text{on } \partial D, \ \omega \in \Omega \end{cases} \tag{2.4}$$

The weak form of (2.4) is to find $u(x, \omega) \in H_0^1(D \times \Omega)$ such that

$$\langle B[u, v] \rangle = \langle (f, v) \rangle \quad \forall v \in H_0^1(D \times \Omega). \tag{2.5}$$

We assume that $a(x, \omega) \in L^\infty(D \times \Omega)$ is strictly positive, with lower and upper bounds α and β respectively,

$$0 < \alpha \leq a(x, \omega) \leq \beta. \tag{2.6}$$

Under this assumption, the existence and uniqueness of a solution u to (2.4) follow from the Lax-Milgram Lemma. Note that we have assumed that the source term f is deterministic. This condition can be relaxed to include randomness.

2.2. The Karhunen-Loeve Expansion. Here, we use the Karhunen-Loeve expansion [14] of the random field $a(x, \omega)$ to separate the deterministic and stochastic components. We assume the mean and two-point correlation of $a(x, \omega)$ are known respectively as

$$a_0(x) = \int_\Omega a(x, \omega) dP(\omega) \ \text{ and } \ C_a(x, x') = \int_\Omega a(x, \omega) a(x', \omega) dP(\omega). \tag{2.7}$$

By Karhunen-Loeve expansion, a can be represented in the series form as

$$a(x, \omega) = k_0(x) + \sum_{m=1}^{\infty} \sqrt{\lambda_m} \, k_m(x) \, y_m(\omega), \tag{2.8}$$

where λ_m and $k_m(x)$ are the eigenvalues and orthogonal eigenfunctions of $C_a(x,x')$; i.e.,

$$\int_D C_a(x,x')k_m(x')d(x') = \lambda_m k_m(x). \tag{2.9}$$

This series converges in the mean-square sense.

By definition, $C_a(x,x')$ is symmetric and positive semidefinite. This implies that there exists a countable sequence of eigenpairs $\{(\lambda_m, k_m)\}$ where the eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots$ are nonnegative and the eigenfunctions $\{k_m(x)\}$ are orthogonal in $L^2(D)$. Moreover, $\{y_m\ \}$ is a set of uncorrelated random variables with mean value zero. If the eigenfunctions are normalized, ξ_m all have unit variance; i.e., $\langle y_m(\omega)\rangle = 0,\ \ \langle y_n(\omega)y_m(\omega)\rangle = \delta_{mn}$. For the computation, we approximate $a(x,\omega)$ by a truncation of (2.8),

$$a_M(x,\omega) = a_0(x) + \sum_{m=1}^{M} \sqrt{\lambda_m}\ k_m(x)\ y_m(\omega),$$

where M denotes the number of terms in the truncation.

In this paper, we also assume the $\{y_m\}$ are independent. Denote the probability density function of y_m as ρ_m, then the joint probability density function of $y = (y_1, \ldots, y_m)$ is $\rho = \rho_1 \times \ldots \times \rho_m$. Let Γ_i denote the image of y_i and $\Gamma = \Gamma_1 \times \Gamma_2 \ldots \Gamma_m$. Now we treat $a_M : D \times \Gamma \to R$ as

$$a_M(x,y) = a_0(x) + \sum_{m=1}^{M} \sqrt{\lambda_m}\ k_m(x)\ y_m. \tag{2.10}$$

Now, we consider the following deterministic elliptic problem, in the weak form: Find $u_M \in H_0^1(D) \otimes L^2(\Gamma,\rho)$ such that

$$-\nabla(a_M(x,y)\nabla u_M(x,y)) = f(x) \qquad \text{in } H_0^1(D) \otimes L^2(\Gamma,\rho). \tag{2.11}$$

2.3. Double Orthogonal Basis. For $y \in \Gamma$, we use the double orthogonal polynomial function space [3, 10] to approximate $L^2(\Gamma,\rho)$, which can decouple the system in the y-space, yielding a sequence of uncoupled systems. Each system is like one Monte Carlo realization of (2.4). The double orthogonal basis is constructed as following. For any $r \in \mathbf{N}$, the space of polynomials of degree at most r,

$$P_r := span\{1, t, t^2, ..., t^r\} \subset L^2(\Gamma,\rho), \tag{2.12}$$

and for $\mathbf{r} = (r_1, r_2, \ldots, r_M) \in \mathbf{N}^M$, we construct the polynomial space,

$$P_{\mathbf{r}} := P_{r_1} \otimes P_{r_2} \otimes \ldots \otimes P_{r_M} \in L^2(\Gamma,\rho). \tag{2.13}$$

For any M dimension of multiindex $\mathbf{i} = \{i_1, i_2, \ldots, i_M\}$, if $0 \leq i_j \leq r_j, \forall 1 \leq j \leq M$, we say that $\mathbf{i} \leq \mathbf{r}$. For any i_j, let $\{\psi^j_{k,i_j}(y_j)\}_{k=0}^{i_j}$ be a basis of the polynomial space P_{i_j} of order i_j.

We require that $\psi^j_{k_j,i_j}(y_j), k = 0, \ldots, i_j$ satisfy two orthogonal conditions:

$$\begin{cases} \int_{\Gamma_j} \psi^j_{p,\,i_j}(y_j)\psi^j_{q,\,i_j}(y_j)\rho_j(y_j)dy_j = \delta_{p,q}, & p,q = 0,\ldots,i_j, \\ \int_{\Gamma_j} y_j\psi^j_{p,\,i_j}(y_j)\psi^j_{q,\,i_j}(y_j)\rho_j(y_j)dy_j = C^j_{p,\,i_j}\delta_{p,q}, & p,q = 0,\ldots,i_j, \end{cases} \tag{2.14}$$

where $\{C^j_{p,\,i_j}\}_{p=0}^{i_j}$ are nonzero constants. Then we construct a basis function of $P_{\mathbf{r}}$ by selecting one polynomial basis function from each $P_{i_j}, j = 1, \ldots, M$, and then multiply these selected M basis functions together. So given a $\mathbf{r} = (r_1, r_2, \ldots, r_m) \in \mathbf{N}^M$, i.e., given the highest order for $\{y_1, y_2, \ldots, y_M\}$ respectively, there are total $N_y = \prod_{j=1}^{M}(r_j + 1)$ basis functions for $P_{\mathbf{r}}(y_1, y_2, \ldots, y_M)$. We denote these multi-variables polynomial basis functions as

$$\left\{ \psi_{\mathbf{i},\,\mathbf{r}}(y) : \psi_{\mathbf{i},\,\mathbf{r}}(y) = \prod_{j=1}^{m} \psi^j_{k,\,i_j}(y_j),\ \ k \in \{0,1,\ldots,i_j\} \right\}_{\mathbf{i}=1}^{N_y}. \tag{2.15}$$

Finding $\{\psi^j_{k,\,i_j}(y_j)\}_{k=0}^{i_j}$ for space P_{i_j}, $j = 1, \ldots, M$, results in an eigenproblem (c. f. section 8.7.2 in [12]). For the probability space of y, generally we do not need high order polynomials. So the computational work for these eigenprolems is negligible comparing the one required to solve for coupled systems.

2.4. Discretization. Now, we discuss a Galerkin method and double orthogonal basis to discretize (2.11) in the y-space. Given the polynomial order index $\mathbf{r} = (r_1, r_2, \ldots, r_M)$, with the constructed N_y double polynomial basis functions of (2.15), we write

$$u(x,y) = \sum_{1}^{N_y} u_{\mathbf{i}}(x)\psi_{\mathbf{i},\mathbf{r}}(y),$$

where $\mathbf{i} = (i_1, i_2, \cdots, i_M)$, $\mathbf{i} \le \mathbf{r}$. We define a finite element function $u(x,y) \in H_0^1(D) \times P_{\mathbf{r}}(D1)$, such that for all $v(x,y) = h(x)\psi_{\mathbf{j},\mathbf{r}}(y) \in H_0^1(D) \times P_{\mathbf{r}}(D1)$ where $h(x)$ is any function in $H_0^1(D)$, we have

$$\left\langle \int_D a_M(x,y)\nabla u(x,y)\cdot\nabla v(x,y)dx \right\rangle = \left\langle \int_D f(x)v(x,y)dx \right\rangle. \tag{2.16}$$

Writing in an explicit form,

$$\left\langle \int_D a_M(x,y)\nabla u(x,y)\cdot\nabla v(x,y)dx \right\rangle = \sum_{m=1}^{M}\sum_{\mathbf{i}=1}^{N_y} \sqrt{\lambda_m}\,(k_m(x)\nabla u_{\mathbf{i}}(x),\ \ \nabla h(x)) \int_\Gamma y_m \psi_{\mathbf{i},\mathbf{r}}(y)\psi_{\mathbf{j},\mathbf{r}}(y)\rho(y)dy.$$

The two orthogonalization conditions imply that

$$\int_\Gamma y_m \psi_{\mathbf{i},\mathbf{r}}(y)\psi_{\mathbf{j},\mathbf{r}}(y)\rho(y)dy = C^m_{k,j_m}\delta_{\mathbf{i},\mathbf{j}}.$$

Consequently,

$$\left\langle \int_D a_M(x,y)\nabla u(x,y)\cdot\nabla v(x,y)dx \right\rangle = \sum_{m=1}^{M}\sum_{\mathbf{i}=1}^{N_y} \sqrt{\lambda_m}\,(k_m(x)\nabla u_{\mathbf{i}}(x),\ \ \nabla h(x))\, C^m_{k,j_m}\delta_{\mathbf{i},\mathbf{j}}.$$

On the other hand,

$$\left\langle \int_D f(x)v(x,y)dx \right\rangle = \int_D f(x)h(x)dx\langle\psi_{\mathbf{j},\mathbf{r}}(y)\rangle.$$

Now, it is easy to see that

$$\langle B[u,v]\rangle = \langle (f,v)\rangle,$$

which implies that

$$-\nabla(a_{M,\mathbf{i}}\nabla u_{M,\mathbf{i}}) = f_{\mathbf{i}} \ \ in\ H^{-1}(D),\ \text{with} \tag{2.17}$$

$$\begin{cases} a_{M,\mathbf{i}} := a_0(x) + \sum\limits_{j=1}^{M} \sqrt{\lambda_j}\, k_j(x) C^j_{k,r_j}, \\ f_{\mathbf{i}}(x) := f(x)\cdot \prod\limits_{j=1}^{M} \int_{\Gamma_j} \psi^j_{k,r_j}(y_j)\rho_j(y_j)dy_j. \end{cases} \tag{2.18}$$

Thus, we have decoupled the system (2.11) into N_y deterministic diffusion problems (2.18) in D. The solution

$$u_{M,\mathbf{r}}(x,y) = \sum_{\mathbf{i}\le\mathbf{r}} u_{M,\mathbf{i}}(x)\psi_{\mathbf{i},\mathbf{r}}(y). \tag{2.19}$$

We can use finite difference method to discretize the left-hand side of (2.17) and obtain the sequence of matrices $A_{M,\mathbf{i}}, \mathbf{i} = 1, \ldots, N_y$.

THEOREM 2.1. *If C_a is piecewise smooth on $D \times D$ and $\xi_j, j = 1, \ldots, M$ are all bounded, then all the matrices $A_{M,\mathbf{i}}$ are spectrally equivalent for M large enough.*

The proof follows from the Proposition 4.3 in [10], which gives the bound: $\frac{\alpha}{2} \leq a_M(x,y) \leq 2\beta$ for all x, y. The constant $C^j_{k,\,r_j}$ in (2.18) has the same upper bound as y_j in (2.10):

$$|C^j_{k,\,r_j}| \leq \int_{\Gamma_j} |y_j| \psi^j_{k,\,r_j}(y_j)^2 dy_j \leq |y_j| \int_{\Gamma_j} \psi^j_{k,\,r_j}(y_j)^2 dy_j = |y_j|.$$

This implies that $a_{M,\mathbf{i}}$ are also bounded in $[\frac{\alpha}{2}, 2\beta]$ as $a_M(x,y)$ for all x, y. Consequently, all the matrices $A_{M,\mathbf{i}}$ are spectrally equivalent.

The statistics of the solution can be found from the approximation solution. For example, the mean of $u(x,y)$ can be approximated by,

$$\langle u_{M,\mathbf{r}} \rangle = \sum_{\mathbf{i} \leq \mathbf{r}} u_{M,\,\mathbf{i}}(x) \int_\Gamma \psi_{\mathbf{i},\,\mathbf{r}}(y) \rho(y) dy = \sum_{\mathbf{i} \leq \mathbf{r}} u_{M,\,\mathbf{i}}(x) \prod_{j=1}^{M} \int_{\Gamma_j} \psi^j_{k,\,r_j}(y_j) \rho_j(y_j) dy_j. \tag{2.20}$$

3. Numerical Method. After the discretization by using the double orthogonal basis, we obtain a sequence of PDEs. For simplicity, we use center difference method to discretize these PDEs. This results in a sequence of independent but similar linear systems like (1.1). In the special case all the matrix are the same, the block method such as block CG [13] and block GMRES [9] can be used. Here, we propose an additive Schwarz preconditioned domain decomposition recycling Krylov space method for these systems. Domain decomposition techniques are powerful iterative methods for solving linear systems in parallel. The recycling Krylov space method [16] reduces some repeated work between the systems. Consequently, it saves a lot of computational time. The additive Schwarz algorithm [17] provides a means of constructing effective preconditioners for many problems.

3.1. Recycling Krylov Space Method. Here, we only present the main idea of this method as well as our modification to the original algorithm. For a detailed description of this method, please refer to [16]. The idea of recycling Krylov space is to retain a Krylov subspace for subsequent restarted GMRES cycles, and also between linear systems. Suppose we solved the ith system with right preconditioned GMRES. We retain k approximate eigenvectors, $\tilde{Y} = [\tilde{y_1}, \tilde{y_2}, \ldots, \tilde{y_k}]$. We maintain matrices U_k, $C_k \in R^{n \times k}$ such that

$$A^{(i+1)} U_k = C_k, \;\; C_k^H C_k = I_k, \tag{3.1}$$

where $C_k = Q$, $U_k = Y_k R^{-1}$ and Q, R is the reduced QR decomposition of $A^{(i+1)} \tilde{Y}_k$.

We find the optimal solution over the subspace $range(U_k)$ as $x = x_0 + U_k C_k^H r_0$, and set $r = r_0 - C_k C_k^H r_0$, and $v_1 = r/\|r\|_2$. We next generate a Krylov space of dimension $m-k+1$ with $(I - C_k C_k^H) A^{(i+1)}$, where m is the maximum number of iteration before restarting, which produces the Arnoldi relation

$$(I - C_k C_k^H) A^{(i+1)} Z_{m-k} = V_{m-k+1} \bar{H}_{m-k}, \tag{3.2}$$

where Z_{m-k} are the preconditioned orthogonal vectors V_m. Each of the Arnoldi vectors $V_{m-k+1} = [v_1, v_2, \ldots, v_{m-k+1}]$ is orthogonal to $range(C_k)$. We can rewrite (2.14) as

$$A[U_k \;\; Z_{m-k}] = [C_k \;\; V_{m-k+1}] \begin{bmatrix} I_k & B_k \\ 0 & \bar{H}_{m-k} \end{bmatrix}, \tag{3.3}$$

where $B_{m-k} = C_k^H A Z_{m-k}$. For numerical reasons, we normalize the column vectors of U_k and replace the identity matrix I_k above with a diagonal matrix D_k, such that $U_k D_k$ has unit columns. We denote the resealed U_k as $\tilde{U}_k$.

We define

$$\hat{V}_m = [\tilde{U}_k \;\; Z_{m-k}], \;\; \hat{W}_{m+1} = [C_k \;\; V_{m-k+1}], \;\; \bar{G}_m = \begin{bmatrix} D_k & B_k \\ 0 & \bar{H}_{m-k} \end{bmatrix} \tag{3.4}$$

and write (3.3) more compactly, as $A\hat{V}_m = \hat{W}_{m+1} \bar{G}_m$. Note that $\bar{G}_m = \hat{W}^H_{m+1} A \hat{V}_m$ is upper Hessenberg, with D diagonal. The columns of $\hat{W}_{m+1}$ are orthogonal, but this is not true for the column of $\hat{V}_m$.

At each cycle, we need to find y to minimize $\|r - A\hat{V}_m y\|_2$. Generally, we solve y by Givens rotation to the Hessenberg matrix $\bar{G}_m$. However, by using the special structure of $\bar{G}_m$, we can solve a much smaller minimization problem over the Hesenberg matrix $\bar{H}_{m-k}$ first, and then construct y.

It turns out that

$$\|r - A\hat{V}_m y\|_2 = \|r - \hat{W}_{m+1}\bar{G}_m y\|_2 = \|\hat{W}_{m+1}^H r - \bar{G}_m y\|_2 = \|e_{k+1}\|r\|_2 - \bar{G}_m y\|_2 \tag{3.5}$$

$$= \left\| \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \|r\|_2 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} - \begin{bmatrix} D_k \bar{y}_k + B_{m-k}\tilde{y}_{m-k} \\ \bar{H}_{m-k}\bar{y}_{m-k+1} \end{bmatrix} \right\|_2, \tag{3.6}$$

where $\bar{y}_k$ denotes the vector formed by the first k elements in vector y, $\bar{y}_{m-k+1}$ denotes the vector formed by the last $m-k+1$ elements in vector y, and $\tilde{y}_{m-k}$ denotes the vector formed by the $(k+1)^{th}$ element to m^{th} element in the vector y.

To find y, we can solve the second part of (3.6); i.e.,

$$min_y \left\| e_1\|r\|_2 - \bar{H}_{m-k}\, \bar{y}_{m-k+1} \right\| \tag{3.7}$$

here, e_1 is a vector in R^{m-k+1}. (3.7) is the standard minimization problem in GMRES, which can be solved by the Givens rotation. Once we find $\bar{y}_{m-k+1}$, $\bar{y}_k = D_k^{-1} B_{m-k}\, \tilde{y}_{m-k}$. Thus, we obtain the full vector y. The residual and solution are updated by

$$r = r - A\hat{V}_m y = r - \hat{W}_{m+1}\bar{G}_m y, \;\; x = x + \hat{V}_m y. \tag{3.8}$$

3.2. Additive Schwartz Method. Let D be a bounded open domain. For simplicity, we assume D is a simple box domain with uniform mesh size h. Let A be the matrix of the linear system to be solved. A typical additive Schwarz preconditioner is constructed as following. First, we partition the domain into rectangular subdomains $D_j, j = 1, \ldots, N_s$. These subdomains can be overlapped. Let N, N_j denotes the number of mesh points on domain D, D_j. For each subdomain D_j, we construct a $N_j \times N$ restriction matrix R_j. The element $(R_j)_{k,l}$ of matrix R_j is either one if the subindices k, l indices indicate a mesh point in D_j or zero if the indicated mesh point is outside of D_j. The transpose of R_j is the so called extension matrix for subdomain D_j. The subdomain matrix A_j of size $N_j \times N_j$ on D_j is defined as

$$A_j = R_j A R_j^T.$$

Let P_j^{-1} be either an inverse of A_j or a preconditioner for A_j. The one-level additive Schwarz preconditioner for matrix A is constructed as

$$P_{asm}^{-1} = \sum_{j=1}^{N_s} R_j^T P_j^{-1} R_j.$$

Moreover, we can define a coarse mesh domain on D, denoted as D_0. The restriction matrix R_0 for the coarse mesh can be constructed by an interpolation method. Let P_0^{-1} be the preconditioner for the coarse mesh matrix A_0, the coarse mesh preconditoner for A is $R_0^T P_0^{-1} R_0$. Combining the coarse mesh preconditioner and the one-level additive Schwarz preconditier, we obtain the two-level additive Schwarz preconditioner

$$P_{asm}^{-1} = \sum_{j=0}^{N_s} R_j^T P_j^{-1} R_j.$$

It is obvious that all the subdomain preconditioning are independent of each other and can therefore be solved in parallel. We also want to emphasize that the size of the overlap between the subdomain $D_1, \ldots, D_{N_s}$ is an important factor for the effectiveness of the additive Schwarz preconditioner. For more description, please see [5, 6, 7].

4. Numerical Experiments. We test the performance of the additive Schwarz preconditioned recycling Krylov space GMRES algorithm for solving the model problem (2.4) on two-dimensional square domain $D = [0,1]^2$. The source term $f(x) = 1$. The mean and covariance function are explicitly given as

$$a_0(x) = 3 + \sin(\pi x_1) \qquad C_a(x, x') = e^{-|x-x'|^2}.$$

For the KL expansion of a, we choose $M = 11$ terms truncation as the approximation. This implies the dimension for the y-space is 11. We also choose the highest order of the polynomials for each of the 11 y_j as a order index $\mathbf{r} = (3, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1)$. This choice of r will produce 9216 systems like (2.18). Readers can also use the algorithm in [10] to select $\mathbf{r}$. At the same time, we assume that the y_j, $j = 1, 2, \ldots, 11$ are uniformly distributed in $\Gamma_j = [-\frac{1}{2}, \frac{1}{2}]$ with the probability density function $\rho_j = 1$. Note that this implies that the variance of y_j are no longer unit. However, we can scale the eigenfunction or eigenvalue to make up this in our computation. We compute the eigenpairs $(\lambda_j, k_j(x))$ of by Matlab. The $k_j(x)$ is saved as a step function together with eigenvalues into a file to be used in the main program. The computation of the orthogonal polynomials (2.14) involves the eigenvalue problem of the type $Ax = \lambda Bx$ with symmetric and positive semidefinite matrices A, B of size $r_j, j = 1, 2, \ldots, 11$. This is also done in Matlab and saved in a file for the main program.

The main program implements the additive Shcwarz preconditioned recycling Krylov space method with flexible GMRES. PETSc are employed for the parallel computation. Both the relative error tolerance and absolute error tolerance for all the tests are set to be $1.0e-10$.

To take the full advantage of the parallel computation, we must be careful about the QR factorization part in the recycling Krylov space method algorithm. Generally, QR is a sequential process. In the implementation, we use the Modified Gram-Schmidt [12] to code the QR factorization. The operations in the algorithm are all vector products. Even this algorithm is sequential, these vectors can be distributed to multiple processors. The vector product thus can be done in parallel.

In recycling the Krylov subspace, we choose the harmonic Ritz vectors corresponding to the harmonic Ritz values of smallest magnitude. Actually, any subset of the Ritz vectors can be recycled. We will discuss this further later.

4.1. Recycling Both the Krylov Subspace And the Preconditioner. Since all these matrices have the same nonzero-pattern, we naturally think of recycling the preconditioner as well as recycling the Krylov subspace. We first test the performance of the one-level additive Schwarz method (ASM) preconditoined recycling flexible GMRES on mesh size 480×480 against the number of processors $1, 4, 9, 16, 25, 36, 64$. The size of overlap is set to be $8\frac{1}{480}$. In this test, we fix the number of basis of the recycled Krylov subspace to be 10. When the number of GMRES iterations is less than 10, we don't recycle any subspace.

By observing the iteration numbers of all the 9216 systems, we find that the iteration numbers of systems 2 to 2304 are almost the half of the iteration number of the system 1. This is due to the recycled Krylov subspace and preconditioner. However, at the system 2305, the iteration number steeply increases, even much higher than the iteration number of the system 1. The iteration numbers of systems 2306 to 9216 are also around the half of or lower than half of the iteration number of the system 2305. In table 4.1, we present the iteration numbers of systems 1 to 5 and systems 2305 to 2309 for illustration. It can be seen that the recycling Krylov subspace and preconditioner method saves about 50% of the iteration numbers. The total running time decreases as the number of the processor increases. This shows the advantage of parallel processing.

We leave the discussion of the strange behavior at system 2305 to section 4.2. Currently, let us see the average number of iterations for the combination of mesh size, number of processors and overlap, see table 4.1. Generally speaking, for fixed number of processors, when the mesh size increases, the number of iterations increases too. However, we double the number of overlap as the mesh size is doubled. This keeps the number of iteration almost the same. Table 4.1 also clearly shows the number of iterations is not scalable with regard to the number of processors. This scalability problem can be corrected by using a two-level ASM preconditioning.

Now we sample the systems 2302 to 2307 as representatives to do the next experiment. We run the two-level additive Schwarz preconditioned solver to solve systems 2302 to 2307. We put the iteration numbers in table 4.3. The overlaps are $2/136, 4/256, 8/496$ for the mesh sizes 136×136, 256×256, and 496×496 respectively.

TABLE 4.1

Efficiency with respect to the number of processors; mesh size 480×480*. LU factorization for all subdomains, overlap* $= 8$*; number of recycled Krylov space basis vector* $= 10$*; The problem (all 9216 systems) is solved with* $1, 4, 9, 16, 25, 36$*, and* 64 *processors. It. denotes the number of iterations in the table; Aver. It. denotes the average number of iterations.*

Iteration numbers of 1-Level AS Preconditioned recycling Krylov space method						
np	t[sec]	It. of sys.1-5	It. of sys. 2305-2309	Largest It.	smallest It.	Aver. It.
1	156700	1 3 3 3 3	43 23 23 23 23	43	1	14.8
4	119700	24 14 12 12 12	89 38 41 37 38	89	12	22.7
9	63870	31 15 15 14 15	75 42 39 36 39	75	13	26.2
16	41220	35 18 15 16 19	86 50 37 34 39	86	15	27.8
25	31420	39 20 18 18 19	95 55 46 43 47	95	17	30.6
36	25480	42 23 19 18 20	105 61 47 46 48	105	18	32.2
64	19410	48 24 21 21 21	115 72 52 49 53	115	20	35.3

TABLE 4.2

Average It. Numbers for One-Level ASM								
mesh	overlap	number of processors						
		1	4	9	16	25	36	64
120×120	2	14.7	22.5	25.5	28.5	30.4	31.6	34.2
240×240	4	14.7	23.1	25.4	28.6	30.6	32.1	34.9
480×480	8	14.8	22.7	26.2	27.8	30.6	32.2	35.3

As we can see from Table 4.3, when the number of processors increases, the iteration numbers of systems 2302 to 2304 with two-level ASM are much more scalable than one-level ASM. The iteration number of system 2305 still have a big steep increase , which consequently affects the scalability of the following systems.

4.2. Recycling the Krylov Subspace Only. From Theorem 2.1, we expect that the iteration numbers of all the systems have the same scale. We have observed that the system 2305 has a much larger iteration number than system 1 when solving all the 9216 systems with the recycled same preconditioner. To see how the recycled preconditoner affect the iteration number of system 2305, we compute the systems 2302 to 2307 on mesh size 256×256. First we solve these 6 systems by the same preconditioner constructed from the starting matrix. Then we solve all the 6 systems with different preconditioner constructed from each matrix itself. We list the iteration numbers in table 4.4.

Compared with the same preconditioner, by using different preconditioners for each matrix, the iteration numbers of the system 2305 and the following systems have greatly decrease. This shows that the same preconditioner is not suitable for system 2305. In a general sense, we need to update the preconditioner once in a while to get a better performance. In addition, in Table 4.4, for the two-level additive schwarz precoditioned method, the iteration number of system 2305 is still larger than the iteration number of the starting system 2302. To study this, we compute systems 2302 to 2307 with different preconditoners and the same input variables as for Table 4.4 except that we restart to construct a new recycled Krylov subspace at system 2305. Thus, the system 2305 doesn't use any recycled Krylov subspace. Systems 2306 and 2307 use the recycled Krylov subspace starting from the system 2305. We compare the results in table 4.5.

When the recycled Krylov subspace is restarted at system 2305, for one-level additive Schwarz preconditioned method, it is preferable to keep using the recycled Krylov subspace, which still helps to reduce the iteration number at the system 2305. For two-level additive Schwarz preconditioned method, the recycled Krylov subspace doesn't help to reduce the iteration number but greatly increases the iteration numbers. More accurately, the *selected* Krylov subspace is not suitable for system 2305. So restart is preferable for two-level additive Schwarz preconditioned method.

TABLE 4.3

Scalability of the Iteration Numbers					
mesh	ASM	number of processors			
		1	4	16	64
136×136	1-level	1 4 4 26 19 19	24 15 10 30 23 23	35 18 12 49 29 29	45 21 18 59 38 38
	2-level	20 17 14 54 43 44	27 20 18 66 46 47	29 22 19 69 49 49	33 21 20 77 58 59
256×256	1-level	1 3 3 26 18 19	24 13 10 32 24 23	36 18 12 49 30 32	47 20 18 64 36 37
	2-level	25 22 20 77 83 75	35 30 25 104 97 88	38 34 27 134 116 115	41 28 25 127 87 94
496×496	1-level	1 3 3 25 16 18	24 14 9 36 26 24	36 18 12 54 31 33	47 20 18 73 41 40
	2-level	32 26 23 160 90 85	43 32 27 123 77 79	46 35 30 121 79 82	51 35 29 148 90 98

TABLE 4.4

The Effect of the Recycled Preconditioner on the Iteration Numbers					
ASM	Preconditioner	number of processors			
		1	4	16	64
1-level	Same	1 3 3 26 18 19	24 13 10 32 24 23	36 18 12 49 30 32	47 20 18 64 36 37
	Different	1 1 1 1 1 1	24 13 10 16 10 11	36 18 12 33 14 15	47 20 18 43 16 18
2-level	Same	25 22 20 77 83 75	35 30 25 104 97 88	38 34 27 134 116 115	41 28 25 127 87 94
	Different	25 21 18 30 24 26	34 27 23 40 30 30	37 27 23 51 32 30	41 28 25 40 30 33

4.3. Adaptively Recycling Krylov Subspace and Preconditioner. From the numerical experiments, we have found that,

1. when we recycle both the Krylov subspace and the one-level additive Schwarz preconditoner, we can update the preconditioner once a while to get better and efficient performance.
2. when we recycle both the Krylov subspace and the two-level additive Schwarz preconditoner, we'd better restart to construct the Krylov subspace and update the preconditioner once a while to get better and efficient performance.

The ideal case is that the recycled Krylov subspace and preconditioner can be updated adaptively as the system changes.

There are also other factors worthy of thinking for better performance. In our implementation, we select the harmonic Ritz vectors corresponding to the harmonic Ritz values of smallest magnitude. Is there a better criteria for selecting the recycled subspace adaptively with the matrices? Moreover, how many of the Ritz vectors to recycle is better? If recycling the preconditioner, do we recycle the whole preconditoner or just part (e.g. coarse level or the LU factorization on the subdomain) of the preconditioner and how?

5. Conclusion. In this paper, we introduced a parallel domain decomposition recycling Krylov space method for the class of elliptic stochastic partial differential equations. We developed a parallel algorithm and greatly decreased the computation time. By recycling the Krylov subspace, the computation cost is saved by about 50%. We can also recycling the preconditioner as well as the Krylov subspace to save more computation. For the one-level additive Schwarz preconditoner, we need to change a new preconditioner adaptively. For two-level additive Schwarz preconditoner, we'd better restart to construct the recycled Krylov subspace and update the preconditioner adaptively. The two-level additive Schwarz preconditioner is scalable with respect to the number of processors.

6. Acknowledgement. The authors wish to thank Dongbin Xiu for many helpful discussions.

REFERENCES

[1] I. Babuska, P. Chatzipantelidis, On solving linear elliptic stochastic partial differential equations, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 4093-4122

[2] I. Babuska, R. Tempone, and G. Zouraris Galerkin finite element approximations of stochastic elliptic partila differential equations, SIAM J. Numer. Anal., 42 (2004), pp. 800-825

[3] I. Babuska, R. Tempone, and G. Zouraris Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation, Comput. Methods Appl. Mech. Engrg., In Press.

TABLE 4.5

In column "Restart", "Yes" denotes the recycled Krylov subspace is restarted to construct a new one from system 2305. "No" means no restart. Different Preconditioners are used.

The Effect of the Recycled Krylov Subspace on the Iteration Numbers					
ASM	Restart	number of processors			
		1	4	16	64
1-level	Yes	1 1 1 1 1 1	24 13 10 25 12 12	36 18 12 35 18 15	47 20 18 45 23 20
	No	1 1 1 1 1 1	24 13 10 16 10 11	36 18 12 33 14 15	47 20 18 43 16 18
2-level	Yes	25 21 18 27 20 21	34 27 23 35 26 26	37 27 23 39 27 27	41 28 25 42 27 27
	No	25 21 18 30 24 26	34 27 23 40 30 30	37 27 23 51 32 30	41 28 25 40 30 33

[4] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, PETSc Users Manual, ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2002.

[5] X.-C. Cai, An additive Schwarz algorithm for nonselfadjoint elliptic equations in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. Chan, R. Glowinski, J. Periaux and O Widlund, eds. SIAM 1990

[6] X.-C. Cai and O. B. Wilbund, Domain decomposition algorithms for indefinite elliptic problems, SIAM J. Sci. Stat. Comp., 13(1992), pp. 243-258.

[7] X.-C. Cai, An optimal two-level overlapping domain decomposition method for elliptic problems in two and three dimensions, SIAM J. Sci. Comput., 14 (1993), pp. 239-247

[8] M.-K. Deb, I. M. Babuska, and J.T. Oden, Solution of stochastic partial differential equations using galerkin tinite element techniques, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 6359-6372

[9] H. Elman, O. Ernst, D. O'Leary, and M. Stewart, Efficient iterative algorithms for the stochastic finite element method with application to acoustic scattering, Comput. Methods Appl. Mech. Engrg., In Press.

[10] P. Frauenfelder, C. Schwab, and R. A. Todor, Finite elements for elliptic problems with stochastic coefficients, Comput. Methods APPl. Mech. Engrg., 194(2005), pp. 205-228

[11] R. G. Ghanem and P. D. Spanos, Stochastic Finite Elements: A Spectral Approach, Revised Edition, Springer-Verlag, 1991.

[12] G. H. Golub, C. F. Van Loan, Matrix Computation, 3rd Edition, The John Hopkings University Press, 1996.

[13] D. P. O'Leary, The block conjugate gradient algorithm and related methods, Linear Algebra and its Applications, 29(1980), pp. 293-322.

[14] M. Loeve, Probability Theory Vol.I, II, Springer, New York, 1978.

[15] H. Matthies and A. Keese, Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations, Comput. Methods Appl. Mech. Engrg., In Press

[16] M. L. Parks, E. D. Sturler, G. Mackey, D. Johnson, and S. Maiti, Recycling Krylov subspaces for sequences of linear systems, Technical Report UIUCDCS-R-2004-2421.

[17] B. Smith, P. Bjorstad, and W. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, Cambridge, MA, 1996.

[18] D. Xiu, G. E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput., In Press.

[19] D. Xiu, G. E. Karniadakis, Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 4927 -4948

[20] N. Wiener, The homogeneous chaos, Amer. J. Math. 60(1930), pp. 897-936.