Domain decomposition method of stochastic PDEs: a two-level scalable preconditioner

Domain decomposition method of stochastic PDEs: a two-level scalable preconditioner

Waad Subber[1] **and Abhijit Sarkar**[2]

Department of Civil and Environmental Engineering, Carleton University, Ottawa, Ontario K1S5B6, Canada

E-mail: [1]wsubber@connect.carleton.ca, [2]abhijit_sarkar@carleton.ca

Abstract.

For uncertainty quantification in many practical engineering problems, the stochastic finite element method (SFEM) may be computationally challenging. In SFEM, the size of the algebraic linear system grows rapidly with the spatial mesh resolution and the order of the stochastic dimension. In this paper, we describe a non-overlapping domain decomposition method, namely the iterative substructuring method to tackle the large-scale linear system arising in the SFEM. The SFEM is based on domain decomposition in the geometric space and a polynomial chaos expansion in the probabilistic space. In particular, a two-level scalable preconditioner is proposed for the iterative solver of the interface problem for the stochastic systems. The preconditioner is equipped with a coarse problem which globally connects the subdomains both in the geometric and probabilistic spaces via their corner nodes. This coarse problem propagates the information quickly across the subdomains leading to a scalable preconditioner. For numerical illustrations, a two-dimensional stochastic elliptic partial differential equation (SPDE) with spatially varying non-Gaussian random coefficients is considered. The numerical scalability of the the preconditioner is investigated with respect to the mesh size, subdomain size, fixed problem size per subdomain and order of polynomial chaos expansion. The numerical experiments are performed on a Linux cluster using MPI and PETSc parallel libraries.

1. Introduction

In modeling numerous engineering and physical systems, it is necessary to consider the random heterogeneity of the model parameters for more realistic computer simulation. When sufficient statistical information is available, the probability theory offers a rich mathematical framework to represent uncertainty in terms of random variables and stochastic processes(e.g. [1, 2]). The spectral stochastic finite element method [1] is a popular computational tool for uncertainty quantification of SPDEs. For large-scale engineering problems of practical interest, the application of SFEM becomes computationally challenging as the size of the algebraic linear system grows rapidly with the spatial mesh resolution and the order of the stochastic dimension. To address this issue, a non-overlapping domain decomposition approach (namely a *substructuring* method) of SPDEs is introduced in [2] in the framework of SFEM. The methodology is based on Schur complement based geometric decomposition and an orthogonal decomposition and projection of the stochastic processes using the polynomial chaos expansion [1]. To tackle the interface problem of the stochastic system using the substructuring method, a parallel preconditioned conjugate gradient method (PCGM) adopted in [3] using a lumped preconditioner.

Despite being superior to the direct solver (in terms of memory requirement and floating point operations), the PCGM based iterative solvers equipped with the lumped preconditioner exhibit performance degradation as the number of subdomains increases. Consequently, a one-level Neumann-Neumann domain decomposition preconditioner is introduced in [4] which demonstrates a fairly good scalability for moderate number of subdomains. As already demonstrated for deterministic PDEs, the one-level preconditioners do not scale well for large number of subdomains as their convergence rates depend on the subdomain size and thus a number of two-level preconditioners containing *coarse* grids are proposed (see, e.g. [5, 6, 7, 8, 9]). A two-level preconditioner consists of local (fine) and global (coarse) components leading to the convergence rate of the iterative solver independent of the problem size, subdomain size and fixed problem size per subdomain(e.g. [5, 6, 7, 8, 9]). In this paper, a two-level scalable preconditioner is proposed for stochastic PDEs.

Previously, the authors have reported a two-level preconditioner for SPDEs[10] based on the domain decomposition method [2]. As a continuation of our previous work, we propose a new two-level preconditioner for iterative substructuring methods for SPDEs. In the context of SPDEs, the proposed preconditioner may be viewed as an extension of the Balancing Domain Decomposition by Constraints (BDDC) [9, 7] in the context of stochastic PDEs. The preconditioner involves construction of a stochastic coarse problem that enforces the global exchange of information across the subdomains . At each iteration of the PCGM solver, the local problems are solved on each subdomain in parallel which collectively construct the fine level of the preconditioner. Additionally the coarse grid accomplishes the global exchange of information which makes the preconditioner scalable. The parallel performance of the algorithm is studied for two-dimensional stochastic elliptic PDE with spatially varying non-Gaussian random coefficients. PETSc [11] and MPI [12] parallel libraries are used for efficient parallel implementation of the proposed algorithm. The graph partitioning tool METIS [13] is used for optimal decomposition of the finite element mesh. The numerical experiments are performed on a Linux cluster consists of 22 nodes with an InfiniBand interconnect (2 Quad-Core 3.0 GHz Intel Xeon processors and 32 GB of memory per node).

2. Uncertainty representation by stochastic processes

This section provides a brief review of the theories of stochastic processes relevant to the subsequent developments of the paper [1, 2]. We assume the data induces a representation of the model parameters as random variables and processes which span the Hilbert space $\mathcal{H}_G$. A set of basis functions $\{\xi_i\}$ is identified to characterize this space using the Karhunen-Loeve expansion. The state of the system resides in the Hilbert space $\mathcal{H}_L$ with basis functions $\{\Psi_i\}$ being identified with the Polynomial Chaos expansion (PC). The Karhunen-Loeve expansion of a stochastic process $\alpha(x,\theta)$ is based on the spectral expansion of its covariance function $R_{\alpha\alpha}(x,y)$. The expansion takes the following form

$$\alpha(x,\theta) = \bar{\alpha}(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\xi_i(\theta)\phi_i(x), \tag{1}$$

where $\bar{\alpha}(x)$ is the mean of the stochastic process, θ represents the random dimension, and $\xi_i(\theta)$ is a set of uncorrelated (but generally not independent) random variables, $\phi_i(x)$ are the eigenfunctions and λ_i are the eigenvalues of the covariance kernel which can be obtained as the solution to the following integral equation

$$\int_{\mathcal{D}} R_{\alpha\alpha}(x,y)\phi_i(y)dy = \lambda_i\phi_i(x). \tag{2}$$

where $\mathcal{D}$ denotes the spatial dimension over which the process is defined. The covariance function of the solution process is not known *a priori*. Hence the Karhunen-Loeve expansion cannot be used to represent it. Therefore, a generic basis, that is complete in the space of all second-order random variables will be identified and used in the approximation process. Since the solution process is a function of the material properties, nodal solution variables, denoted by $u(\theta)$ can be formally expressed as some nonlinear functional of the set $\xi_i(\theta)$ used to represent the material stochasticity. It has been shown that this functional dependence can be expanded in terms of polynomials in Gaussian random variables, namely Polynomial Chaos [1, 14] as

$$\begin{aligned} u(\theta) &= a_0\Gamma_0 + \sum_{i_1=1}^{\infty} u_{i_1}\Gamma_1\left(\xi_{i_1}(\theta)\right) \\ &+ \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} u_{i_1 i_2}\Gamma_2\left(\xi_{i_1}(\theta),\xi_{i_2}(\theta)\right) + \ldots . \end{aligned} \tag{3}$$

In this equation, the symbol $\Gamma_n(\xi_{i_1},\ldots,\xi_{i_n})$ denotes the polynomial chaos ([15, 1]) of order n in the variables $(\xi_{i_1},\ldots,\xi_{i_n})$. Introducing a one-to-one mapping to a set with ordered indices denoted by $\{\Psi_i(\theta)\}$ and truncating the polynomial chaos expansion after the N^{th} term, Eq.(3) can be rewritten as,

$$\mathbf{u}(\theta) = \sum_{j=0}^{N} \Psi_j(\theta)\mathbf{u}_j. \tag{4}$$

These polynomials are orthogonal in the sense that their inner product $\langle\Psi_j\Psi_k\rangle$, defined by the statistical average of their product, is equal to zero for $j \neq k$.

3. Review of Schur complement based domain decomposition method of SPDEs

In this section, we provide a brief review of the domain decomposition method for SPDEs based on [2, 3, 4, 10]. For an elliptic SPDE defined on a domain Ω with a prescribed boundary conditions on $\partial\Omega$, the finite element discretization leads to the following linear system

$$\mathbf{A}(\theta)\mathbf{u}(\theta) = \mathbf{f}, \tag{5}$$

where $\mathbf{A}(\theta)$ is the stiffness matrix with random coefficients, $\mathbf{u}(\theta)$ is the random vector representing the response vector and $\mathbf{f}$ is the external force. For large-scale system, Eq.(5) can be solved efficiently using domain decomposition methods [2, 3, 4, 10].

In the non-overlapping domain decomposition method (for example, see [5, 6, 7, 8, 9]), the spatial domain Ω is partitioned into n_s non-overlapping subdomains $\{\Omega_s,\ 1 \leqslant s \leqslant n_s\}$ such that $\Omega = \bigcup_{s=1}^{n_s}\Omega_s, \quad \Omega_s \bigcap \Omega_r = 0, \quad s \neq r$ and $\Gamma = \bigcup_{s=1}\Gamma_s$ where $\Gamma_s = \partial\Omega_s \backslash \partial\Omega$. For a typical subdomain Ω_s the nodal vector $\mathbf{u}^s(\theta)$ is decomposed into the interior unknowns $\mathbf{u}_I^s(\theta)$ associated with the nodes in the interior of Ω_s and the interface unknowns $\mathbf{u}_\Gamma^s(\theta)$ corresponding to the nodes shared among two or more subdomains. This decomposition leads to the following equilibrium equation for the subdomain

$$\begin{bmatrix} \mathbf{A}_{II}^s(\theta) & \mathbf{A}_{I\Gamma}^s(\theta) \\ \mathbf{A}_{\Gamma I}^s(\theta) & \mathbf{A}_{\Gamma\Gamma}^s(\theta) \end{bmatrix} \begin{Bmatrix} \mathbf{u}_I^s(\theta) \\ \mathbf{u}_\Gamma^s(\theta) \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_I^s \\ \mathbf{f}_\Gamma^s \end{Bmatrix} . \tag{6}$$

Enforcing the transmission conditions (compatibility and equilibrium) along the subdomain interfaces and expanding the solution vector by the polynomial chaos (as in Eq.(4)) and then performing Galerkin projection, the following block linear systems of equations express the global equilibrium equation of the stochastic system ([2, 3, 4, 10]):

$$\langle \sum_{i=0}^{L} \Psi_i(\theta) \begin{bmatrix} \mathbf{A}^1_{II,i} & \dots & 0 & \mathbf{A}^1_{I\Gamma,i}\mathbf{R}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mathbf{A}^{n_s}_{II,i} & \mathbf{A}^{n_s}_{I\Gamma,i}\mathbf{R}_{n_s} \\ \mathbf{R}_1^T\mathbf{A}^1_{\Gamma I,i} & \dots & \mathbf{R}^T_{n_s}\mathbf{A}^{n_s}_{\Gamma I,i} & \sum_{s=1}^{n_s}\mathbf{R}_s^T\mathbf{A}^s_{\Gamma\Gamma,i}\mathbf{R}_s \end{bmatrix} \sum_{j=0}^{N} \Psi_j(\theta) \begin{Bmatrix} \mathbf{u}^1_{I,j} \\ \vdots \\ \mathbf{u}^{n_s}_{I,j} \\ \mathbf{u}_{\Gamma,j} \end{Bmatrix} \Psi_k(\theta)\rangle$$

$$= \langle \begin{Bmatrix} \mathbf{f}^1_I \\ \vdots \\ \mathbf{f}^{n_s}_I \\ \sum_{s=1}^{n_s}\mathbf{R}_s^T\mathbf{f}^s_\Gamma \end{Bmatrix} \Psi_k(\theta)\rangle, \quad k = 0, ..., N. \tag{7}$$

The restriction operator $\mathbf{R}_s$ is a Boolean rectangular matrix of size $(n^s_\Gamma \times n_\Gamma)$ that maps the global interface vector $\mathbf{u}_\Gamma(\theta)$ to the local interface vector $\mathbf{u}^s_\Gamma(\theta)$ as $\mathbf{u}^s_\Gamma(\theta) = \mathbf{R}_s\mathbf{u}_\Gamma(\theta)$. In parallel implementation, $\mathbf{R}_s$ represents a *scatter* operator while $\mathbf{R}^T_s$ acts as a *gather* operator. Compactly, Eq.(7) can be expressed as

$$\begin{bmatrix} \mathcal{A}^1_{II} & \dots & 0 & \mathcal{A}^1_{I\Gamma}\mathcal{R}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mathcal{A}^{n_s}_{II} & \mathcal{A}^{n_s}_{I\Gamma}\mathcal{R}_{n_s} \\ \mathcal{R}_1^T\mathcal{A}^1_{\Gamma I} & \dots & \mathcal{R}^T_{n_s}\mathcal{A}^{n_s}_{\Gamma I} & \sum_{s=1}^{n_s}\mathcal{R}_s^T\mathcal{A}^s_{\Gamma\Gamma}\mathcal{R}_s \end{bmatrix} \begin{Bmatrix} \mathcal{U}^1_I \\ \vdots \\ \mathcal{U}^{n_s}_I \\ \mathcal{U}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \mathcal{F}^1_I \\ \vdots \\ \mathcal{F}^{n_s}_I \\ \sum_{s=1}^{n_s}\mathcal{R}_s^T\mathcal{F}^s_\Gamma \end{Bmatrix}, \tag{8}$$

where

$$[\mathcal{A}^s_{\alpha\beta}]_{jk} = \sum_{i=0}^{L}\langle\Psi_i\Psi_j\Psi_k\rangle\mathbf{A}^s_{\alpha\beta,i} \quad , \quad \mathcal{F}^s_{\alpha,k} = \langle\Psi_k\mathbf{f}^s_\alpha\rangle,$$
$$\mathcal{U}^m_I = (\mathbf{u}^m_{I,0}, \dots, \mathbf{u}^m_{I,N})^T, \quad \mathcal{U}_\Gamma = (\mathbf{u}_{\Gamma,0}, \dots, \mathbf{u}_{\Gamma,N})^T.$$

The subscripts α and β represent the index I and Γ. The restriction operator $\mathcal{R}_s$ for the stochastic problem in Eq.(8) takes the following form: $\mathcal{R}_s = blockdiag(\mathbf{R}^0_s, \dots, \mathbf{R}^N_s)$, where $\{\mathbf{R}^j_s,\ j = 1, .., N\}$ are the restriction operators that map the polynomial coefficients of the global interface vector to those of the local interface vector. Performing a Gaussian elimination in Eq.(8), we obtain the global *extended* Schur complement system for the interface variable $\mathcal{U}_\Gamma$ as

$$\mathcal{S}\,\mathcal{U}_\Gamma = \mathcal{G}_\Gamma, \tag{9}$$

where the global extended Schur complement matrix $\mathcal{S}$ is given by

$$\mathcal{S} = \sum_{s=1}^{n_s}\mathcal{R}_s^T[\mathcal{A}^s_{\Gamma\Gamma} - \mathcal{A}^s_{\Gamma I}(\mathcal{A}^s_{II})^{-1}\mathcal{A}^s_{I\Gamma}]\mathcal{R}_s, \tag{10}$$

and the corresponding right hand side vector $\mathcal{G}_\Gamma$ is defined by

$$\mathcal{G}_\Gamma = \sum_{s=1}^{n_s} \mathcal{R}_s^T [\mathcal{F}_\Gamma^s - \mathcal{A}_{\Gamma I}^s (\mathcal{A}_{II}^s)^{-1} \mathcal{F}_I^s]. \tag{11}$$

For clarity, the global interface solution $\mathcal{U}_\Gamma$ of Eq.(9) relates to the interface nodes shown schematically in Fig(1).

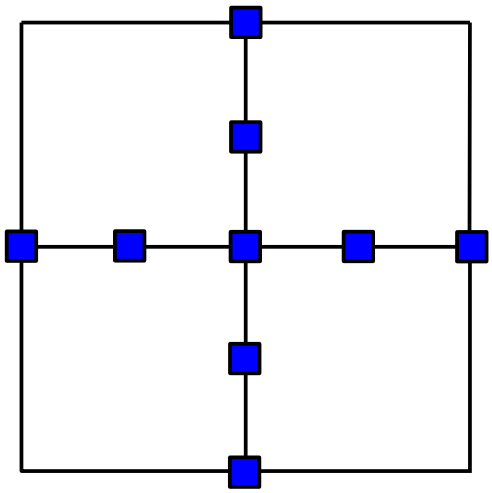


Figure 1. The interface boundary (■).

4. Iterative Solutions of the interface problem

For a large-scale system, the solution of the interface problem in Eq.(9) using direct solvers becomes impractical due to overwhleming memory requirements and poor scalability to large number of processors and thus iterative solution techniques are generally adopted [16]. In domain decomposition approach the Schur complement system is typically solved using preconditioned Krylov subspace method such as preconditioned conjugate gradient method (PCGM). The preconditioned Krylov subspace methods circumvents the need to construct the Schur complement matrix explicitly. Instead only the action of a vector on the Schur complement matrix is needed while using preconditioned Krylov subspace methods. Such matrix vector product can be obtained by solving Dirichlet problem in each subdomain and gathering the resulting vectors at each iteration (e.g. [5, 6]). Non-overlapping domain decomposition method or *iterative substructuring* can be viewed as a preconditioned iterative method to solve the Schur complement system with a parallel preconditioner[5, 6]

$$\mathcal{S}\, \mathcal{U}_\Gamma = \mathcal{G}_\Gamma. \tag{12}$$

For symmetric positive-definite system such as Schur complement system, the Conjugate Gradient Method (CGM) is generally used. The performance of CGM mainly depends on the spectrum of the coefficient matrix. However, the rate of convergence of the iterative method can generally be improved by transforming the original system into an equivalent system that has better spectral properties (i.e. lower condition number $\kappa(\mathcal{S})$) of the coefficient matrix. This transformation is called preconditioning and the matrix used in the transformation is called the preconditioner. In other words, the transformed linear system becomes:

$$\mathcal{M}^{-1}\mathcal{S}\, \mathcal{U}_\Gamma = \mathcal{M}^{-1}\mathcal{G}_\Gamma, \tag{13}$$

where $\mathcal{M}^{-1}$, the preconditioner, is a symmetric and invertible matrix that approximates $\mathcal{S}^{-1}$, the coefficient matrix, in the sense that the condition number of the preconditioned system $\kappa(\mathcal{M}^{-1}\mathcal{S})$ is much smaller than that of the original system $\kappa(\mathcal{S})$ and the eigenvalues of the preconditioned system $\mathcal{M}^{-1}\mathcal{S}$ are clustered near one. This procedure known as Preconditioned Conjugate Gradient Method (PCGM). In practice, the explicit construction of $\mathcal{M}^{-1}$ is not needed. Instead, for a given vector r_Γ, a system of the the following form is solved

$$\mathcal{M}\mathcal{Z} = r_\Gamma. \tag{14}$$

The combination of a good parallel preconditioning technique and the CGM solver makes the iterative solver well-suited to tackle large-scale systems in parallel computers [16].

5. Preconditioners For Stochastic PDEs

As mentioned previously, an efficient scalable preconditioner is required to enhance the convergence rate, reliability and performance of the PCGM iterative solver used to tackle the stochastic Schur complement system. Next we delineate the procedure to construct such preconditioners.

5.1. One-Level Stochastic Neumann-Neumann Preconditioners

A one-level Neumann-Neumann domain decomposition preconditioner is presented in [4] in the context of stochastic PDEs. This preconditioner approximates the inverse of the global extended Schur complement matrix by a weighted sum of the inverse of the local extended Schur complement matrices [5, 17, 6]. The implementation of the algorithm requires a local solve of a stochastic Dirichlet problem followed by a local solve of a stochastic Neumann problem in each iteration of the PCGM solver. The stochastic one-level Neumann-Neumann preconditioner formally takes the following form

$$\mathcal{M}_{NN}^{-1} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{D}_s^T [\mathcal{S}^s]^{-1} \mathcal{D}_s \mathcal{R}_s, \tag{15}$$

where

$$\mathcal{D}_s = blockdiag(\mathbf{D}_s^0, \cdots, \mathbf{D}_s^N), \tag{16}$$

and $\mathbf{D}_s^j$ represents a diagonal scaling matrix that derives from the partition of unity as

$$\sum_{s=1}^{n_s} \mathbf{R}_s^T \mathbf{D}_s^j \mathbf{R}_s = \mathbf{I}. \tag{17}$$

The diagonal entries of $\mathbf{D}_s^j$ are the reciprocal of the number of subdomains that share the boundary nodes [5, 17].

In practice, the local extended Schur complement matrix $\mathcal{S}^s$ is not constructed explicitly, and therefore the inverse of the matrix $\mathcal{S}^s$ is not available. Instead the effect of the inverse of the Schur complement matrix on the residual vector is computed by solving the following subdomain level stochastic Neumann problem

$$\begin{bmatrix} \mathcal{A}_{II}^s & \mathcal{A}_{I\Gamma}^s \\ \mathcal{A}_{\Gamma I}^s & \mathcal{A}_{\Gamma\Gamma}^s \end{bmatrix} \begin{Bmatrix} \mathcal{X}^s \\ \mathcal{U}_\Gamma^s \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{r}_\Gamma^s \end{Bmatrix}. \tag{18}$$

For the case of floating subdomains (subdomains without enough essential boundary conditions), the solution to the local stochastic Neumann problem defined in Eq.(18) exists only

if the right hand side vector satisfies the compatibility condition (i.e. the right hand side vector is orthogonal to the null space of the coefficient matrix). A computationally expensive Moore-Penrose pseudo-inverse can be used to obtain a solution to this singular system. Alternatively a small value can be added to the diagonal entries of the local stochastic stiffness matrix to ensure the solvability of this singular system. Such regularization does not change the problem, but only alters the preconditioner.

5.2. Two-Level Stochastic Neumann-Neumann Preconditioner

For a moderate range of subdomains the stochastic Neumann-Neumann preconditioner demonstrates a fairly good scalability [4]. However, the convergence rate of the one-level Neumann-Neumann preconditioner degrades with increasing the number of subdomains. This performance degradation can be attributed to the lack of global exchange of information. For the one-level preconditioner, the information is transferred only among the neighboring subdomains. In the two-level preconditioner, the coarse problem provides a mechanism for global propagation of information across the subdomains to effectively enhance the convergence rate of the PCGM solver.

In this section, we formulate a novel two-level preconditioner for stochastic PDEs. The preconditioner is equipped with a coarse problem constructed using a collection of corner nodes. The coarse problem connects the subdomains globally via the corner nodes and thus provides a mechanism to propagate the information quickly across the subdomains leading to a scalable preconditioner. Following a version of the Balancing Domain Decomposition by Constraints (BDDC) [9, 7], we formulate a two-level preconditioner for stochastic PDEs. The methodology is detailed next.

In the one-level Neumann-Neumann preconditioner in Eq.(15), a stochastic Neumann problem (see Eq.(18)) is solved to calculate the preconditioned residual for each iteration of PCGM solver. To ensure the solvability of the Neumann problem (defined in Eq.(18)), the artificial Dirichlet boundary conditions are imposed to remove the singularity from the local stiffness matrix of the floating subdomains.

Similar to the dual-primal domain decomposition methods [9, 8, 10], the boundary nodes can be partitioned into two subsets: remaining nodes (boundary nodes shared only between two adjacent subdomains) and corner nodes (boundary nodes shared between more than two subdomains plus the nodes on the ends of interface edges). The corner nodes serve the following purposes: (a) they impose artificial Dirichlet boundary conditions to remove the singularity from the local Schur complement matrix, (b) they introduce a coarse grid that provides a mechanism to propagate information globally.

For a typical subdomain Ω_s, the local unknown vector consists of the interior $\mathcal{X}_i^s$, remaining (fine) $\mathcal{U}_r^s$ and corner (coarse) $\mathcal{U}_c^s$ unknown vectors as schematically shown in Fig.(2). Therefore the local interface unknown vector $\mathcal{U}_\Gamma^s$ is given by

$$\mathcal{U}_\Gamma^s = \left\{ \begin{array}{c} \mathcal{U}_r^s \\ \mathcal{U}_c^s \end{array} \right\}, \tag{19}$$

where

$$\mathcal{U}_r^s = \mathcal{R}_s^r \mathcal{U}_\Gamma^s, \tag{20}$$

$$\mathcal{U}_c^s = \mathcal{R}_s^c \mathcal{U}_\Gamma^s. \tag{21}$$

The stochastic Boolean splitting operators $\mathcal{R}_s^r$ and $\mathcal{R}_s^c$ are defined by

$$\mathcal{R}_s^r = blockdiag(\mathbf{R}_s^{r,0}, \ldots, \mathbf{R}_s^{r,N}),$$
$$\mathcal{R}_s^c = blockdiag(\mathbf{R}_s^{c,0}, \ldots, \mathbf{R}_s^{c,N}),$$

where N is the number of the polynomial chaos basis, and $\mathbf{R}_s^{r,j}$ and $\mathbf{R}_s^{c,j}$ are defined by

$$\mathbf{u}_{r,j}^s = \mathbf{R}_s^{r,j}\mathbf{u}_{\Gamma,j}^s,$$
$$\mathbf{u}_{c,j}^s = \mathbf{R}_s^{c,j}\mathbf{u}_{\Gamma,j}^s.$$

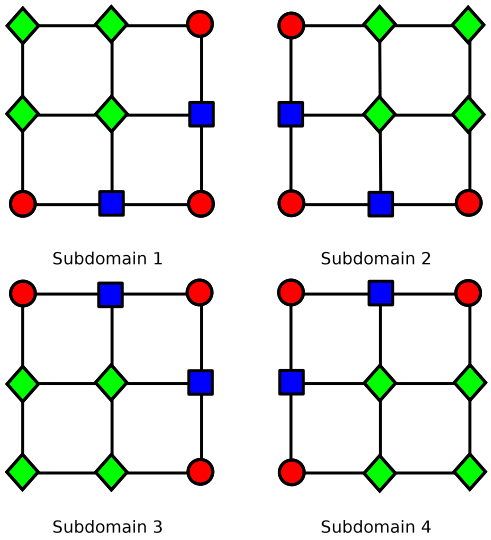


Figure 2. Partitioning the subdomain nodes into: interior (♦), remaining (■) and corner(●)

Performing partial assembly, the following linear system is obtained

$$\begin{bmatrix} \mathcal{A}_{ii}^s & \mathcal{A}_{ir}^s & \mathcal{A}_{ic}^s\mathcal{B}_c^s \\ \mathcal{A}_{ri}^s & \mathcal{A}_{rr}^s & \mathcal{A}_{rc}^s\mathcal{B}_c^s \\ \sum_{s=1}^{n_s}\mathcal{B}_c^{sT}\mathcal{A}_{ci}^s & \sum_{s=1}^{n_s}\mathcal{B}_c^{sT}\mathcal{A}_{cr}^s & \sum_{s=1}^{n_s}\mathcal{B}_c^{sT}\mathcal{A}_{cc}^s\mathcal{B}_c^s \end{bmatrix} \begin{Bmatrix} \mathcal{X}_i^s \\ \mathcal{U}_r^s \\ \mathcal{U}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathcal{F}_r^s \\ \sum_{s=1}^{n_s}\mathcal{B}_c^{sT}\mathcal{F}_c^s \end{Bmatrix}, \tag{22}$$

where

$$\mathcal{F}_r^s = \mathcal{R}_s^r r_\Gamma^s,$$
$$\mathcal{F}_c^s = \mathcal{R}_s^c r_\Gamma^s.$$

The restriction operator $\mathcal{B}_c^s$ in Eq.(22) is a Boolean rectangular matrix that maps the global coarse (corner) unknown $\mathcal{U}_c$ into the local coarse unknown $\mathcal{U}_c^s$ as

$$\mathcal{U}_c^s = \mathcal{B}_c^s\mathcal{U}_c,$$

where

$$\mathcal{B}_c^s = blockdiag(\mathbf{B}_{c,0}^s, \ldots, \mathbf{B}_{c,N}^s),$$

and $\mathbf{B}_{c,j}^s$ is defined by

$$\mathbf{u}_{c,j}^s = \mathbf{B}_{c,j}^s \mathbf{u}_{c,j}.$$

Consequently we obtain the following extended Schur complement system

$$\begin{bmatrix} \mathcal{S}_{rr}^s & \mathcal{S}_{rc}^s \mathcal{B}_c^s \\ \sum_{s=1}^{n_s} \mathcal{B}_c^{sT} \mathcal{S}_{cr}^s & \sum_{s=1}^{n_s} \mathcal{B}_c^{sT} \mathcal{S}_{cc}^s \mathcal{B}_c^s \end{bmatrix} \begin{Bmatrix} \mathcal{U}_r^s \\ \mathcal{U}_c \end{Bmatrix} = \begin{Bmatrix} \mathcal{F}_r^s \\ \sum_{s=1}^{n_s} \mathcal{B}_c^{sT} \mathcal{F}_c^s \end{Bmatrix}, \tag{23}$$

where

$$\mathcal{S}_{\alpha\beta}^s = \mathcal{A}_{\alpha\beta}^s - \mathcal{A}_{\alpha i}^s [\mathcal{A}_{ii}^s]^{-1} \mathcal{A}_{i\beta}^s,$$

the subscripts α and β represent the index r and c.

Due to the artificial Dirichlet boundary conditions imposed by corner nodes, the local Schur complement matrix $\mathcal{S}_{rr}^s$ in Eq.(23) is positive-definite and invertible for the floating subdomains. One could concurrently obtain $\mathcal{U}_r^s$ for each subdomain as

$$\mathcal{S}_{rr}^s \mathcal{U}_r^s = \mathcal{F}_r^s - \mathcal{S}_{rc}^s \mathcal{B}_c^s \mathcal{U}_c. \tag{24}$$

For the global corner node unknown $\mathcal{U}_c$, the following extended Schur complement system is obtained

$$F_{cc} \mathcal{U}_c = d_c, \tag{25}$$

where

$$F_{cc} = \sum_{s=1}^{n_s} \mathcal{B}_c^{sT} (\mathcal{S}_{cc}^s - \mathcal{S}_{cr}^s [\mathcal{S}_{rr}^s]^{-1} \mathcal{S}_{rc}^s) \mathcal{B}_c^s,$$
$$d_c = \sum_{s=1}^{n_s} \mathcal{B}_c^{sT} (\mathcal{F}_c^s - \mathcal{S}_{cr}^s [\mathcal{S}_{rr}^s]^{-1} \mathcal{F}_r^s).$$

The coarse problem defined in Eq.(25) couples the subdomains by the corner nodes in order to transfer information globally. Although the size of the coarse problem is relatively small compared to that of the fine problem, it is still computationally expensive to construct the coarse extended Schur complement system explicitly. Therefore we solve the coarse problem iteratively in parallel without constructing the coarse extended Schur complement matrix explicitly.

Next, the local interface unknowns can be recovered as

$$\mathcal{U}_\Gamma^s = \mathcal{R}_s^{r^T} \mathcal{U}_r^s + \mathcal{R}_s^{c^T} \mathcal{U}_c^s. \tag{26}$$

After some algebraic manipulations, the two-level Neumann-Neumann preconditioner for the stochastic system can be obtained as

$$\mathcal{M}_{CNN}^{-1} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{D}_s^T (\mathcal{R}_s^{r^T} [\mathcal{S}_{rr}^s]^{-1} \mathcal{R}_s^r) \mathcal{D}_s \mathcal{R}_s + R_0^T [F_{cc}]^{-1} R_0, \tag{27}$$

where the fine to coarse space restriction operator is defined by

$$R_0 = \sum_{s=1}^{n_s} \mathcal{B}_c^{sT} (\mathcal{R}_s^c - \mathcal{S}_{cr}^s [\mathcal{S}_{rr}^s]^{-1} \mathcal{R}_s^r) \mathcal{D}_s \mathcal{R}_s. \tag{28}$$

Clearly, the two-level stochastic Neumann-Neumann preconditioner expressed in Eq.(27) consists of local problems $[\mathcal{S}_{rr}^s]^{-1}$ in order to construct a subdomain level preconditioner, and a coarse problem $[F_{cc}]^{-1}$ for global information propagation. This global exchange of information leads to a scalable preconditioner.

5.3. Parallel Implementation

The parallel PCGM iterative solver is used to tackle the extended Schur complement system defined by

$$\mathcal{M}_{CNN}^{-1} \mathcal{S}\, \mathcal{U}_\Gamma = \mathcal{M}_{CNN}^{-1} \mathcal{G}_\Gamma. \tag{29}$$

The procedure is listed in Algorithm(1) [10].

Algorithm 1: The Parallel PCGM Procedure

input : $(\mathcal{U}_{\Gamma_0})$

`compute` : $\mathbf{r}_{\Gamma_0} = \mathcal{G}_\Gamma - \sum_{s=1}^{s=n_s} \mathcal{R}_s^T \mathcal{S}_s \mathcal{R}_s \mathcal{U}_{\Gamma_0}$

`precondition` : $\mathcal{Z}_0 = \mathcal{M}_{CNN}^{-1} \mathbf{r}_{\Gamma_0}$

`compute` : $\mathcal{P}_0 = \mathcal{Z}_0$

`compute` : $\rho_0 = (\mathbf{r}_{\Gamma_0}, \mathcal{Z}_0)$

for $j = 1, 2, \cdots$, **until convergence***:*

do

  `compute` : $\mathcal{Q}_j = \sum_{s=1}^{s=n_s} \mathcal{R}_s^T \mathcal{S}_s \mathcal{R}_s \mathcal{P}_j$

  `compute` : $\rho_{tmp_j} = (\mathcal{Q}_j, \mathcal{P}_j)$

  `compute` : $\alpha_j = \rho_j / \rho_{tmp_j}$

  `update` : $\mathcal{U}_{\Gamma_{j+1}} = \mathcal{U}_{\Gamma_j} + \alpha_j \mathcal{P}_j$

  `update` : $\mathbf{r}_{\Gamma_{j+1}} = \mathbf{r}_{\Gamma_j} - \alpha_j \mathcal{Q}_j$

  `precondition` : $\mathcal{Z}_{j+1} = \mathcal{M}_{CNN}^{-1} \mathbf{r}_{\Gamma_{j+1}}$

  `compute` : $\rho_{j+1} = (r_{\Gamma_{j+1}}, \mathcal{Z}_{j+1})$

  `compute` : $\beta_j = \rho_{j+1} / \rho_j$

  `update` : $\mathcal{P}_{j+1} = \mathcal{Z}_{j+1} + \beta_j \mathcal{P}_j$

output : $(\mathcal{U}_\Gamma)$

The vector $\mathcal{Q}_j$ in steps 1 and 7 of Algorithm (1), which represents the action of the extended Schur complement matrix on a vector, is computed in parallel by three matrix vector products and one direct solve of a local Dirichlet problem (with non-homogeneous boundary condition) on each subdomain [10] as shown in Algorithm (2).

Algorithm 2: Dirichlet-Solver Procedure

input : $(\mathcal{P})$
1 **for** $s = 1, 2, \cdots, n_s$ **in parallel**:
2 **do**
3 `scatter` : $\mathcal{P}^s = \mathcal{R}_s \mathcal{P}$
4 `compute` : $v_1 = \mathcal{A}^s_{I\Gamma} \mathcal{P}^s$
5 `solve` : $\mathcal{A}^s_{II} v_2 = v_1$
6 `compute` : $v_3 = \mathcal{A}^s_{\Gamma I} v_2$
7 `compute` : $v_4 = \mathcal{A}^s_{\Gamma\Gamma} \mathcal{P}^s$
8 `compute` : $\mathcal{Q}^s = v_4 - v_3$
9 `gather` : $\mathcal{Q} = \sum_{s=1}^{n_s} \mathcal{R}_s^T \mathcal{Q}^s$
output : $(\mathcal{Q})$

The two-level preconditioner effect on the residual vector in steps 2 and 12 of Algorithm (1) defined by

$$\mathcal{Z} = \mathcal{M}^{-1}_{CNN} \mathbf{r}_\Gamma, \tag{30}$$

can be implemented in parallel as outlined in Algorithm(3)

Algorithm 3: Two-Level Neumann-Neumann Preconditioner Effect Procedure

input : $(\mathbf{r}_\Gamma)$
1 **for** $s = 1, 2, \cdots, n_s$ **in parallel**:
2 **do**
3 `scatter` : $\mathbf{r}^s_\Gamma = \mathcal{D}_s \mathcal{R}_s \mathbf{r}_\Gamma$
4 `compute` : $\mathcal{F}^s_r = \mathcal{R}^r_s \mathbf{r}^s_\Gamma$
5 `compute` : $\mathcal{F}^s_c = \mathcal{R}^c_s \mathbf{r}^s_\Gamma$
6 `solve` : $\mathcal{S}^s_{rr} v^s_1 = \mathcal{F}^s_r$
7 `update` : $d^s_c = \mathcal{F}^s_{\Gamma_c} - \mathcal{S}^s_{cr} v^s_1$
8 `gather` : $d_c = \sum_{s=1}^{n_s} \mathcal{B}^{sT}_c d^s_c$
9 `solve` : $F_{cc} \mathcal{Z}_c = d_c$
10 **for** $s = 1, 2, \cdots, n_s$ **in parallel**:
11 **do**
12 `scatter` : $\mathcal{Z}^s_c = \mathcal{B}^s_c \mathcal{Z}_c$
13 `update` : $v^s_2 = \mathcal{F}^s_{\Gamma_r} - \mathcal{S}^s_{rc} \mathcal{Z}^s_c$
14 `solve` : $\mathcal{S}^s_{rr} \mathcal{Z}^s_f = v^s_2$
15 `update` : $\mathcal{Z}^s = \mathcal{R}^{r^T}_s \mathcal{Z}^s_f + \mathcal{R}^{c^T}_s \mathcal{Z}^s_c$
16 `gather` : $\mathcal{Z} = \sum_{s=1}^{n_s} \mathcal{R}^T_s \mathcal{D}_s \mathcal{Z}^s$
output : $(\mathcal{Z})$

The local solve in steps 6 and 14 of Algorithm (3) constitutes a subdomain level stochastic Neumann problem which can be tackled using Algorithm (4) as follows

Algorithm 4: Neumann-Solver Procedure

input : $(\mathcal{F}^s_r)$
1 `solve` : $\begin{bmatrix} \mathcal{A}^s_{ii} & \mathcal{A}^s_{ir} \\ \mathcal{A}^s_{ri} & \mathcal{A}^s_{rr} \end{bmatrix} \begin{Bmatrix} \mathcal{X}^s_i \\ \mathcal{U}^s_r \end{Bmatrix} = \begin{Bmatrix} 0 \\ \mathcal{F}^s_r \end{Bmatrix}$
output : $(\mathcal{U}^s_r)$

The coarse problem in step 9 of of Algorithm (3) is solved iteratively in parallel using PCGM solver equipped with a parallel lumped preconditioner as

$$\mathcal{M}^{-1}_c F_{cc} \mathcal{Z}_c = \mathcal{M}^{-1}_c d_c, \tag{31}$$

where

$$\mathcal{M}^{-1}_c = \sum_{s=1}^{n_s} \mathcal{B}^{sT}_c \mathcal{A}^s{}_{\Gamma\Gamma} \mathcal{B}^s_c. \tag{32}$$

6. Numerical Results

For numerical illustration, we consider a two dimensional Poisson PDE with randomly heterogeneous permeability coefficient defined by

$$\nabla \cdot (\kappa(\mathbf{x},\theta)\nabla u(\mathbf{x},\theta)) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{33}$$

$$u(\mathbf{x},\theta)) = 0, \quad \mathbf{x} \in \partial\Omega. \tag{34}$$

The coefficient $\kappa(\mathbf{x},\theta)$ is modeled as a lognormal stochastic process obtained from the underlying Gaussian process with an exponential covariance function given by

$$C_{\alpha\alpha}(\mathbf{x},\mathbf{y}) = \sigma^2 \exp\left(-\frac{|x_1 - y_1|}{b_1} - \frac{|x_2 - y_2|}{b_2}\right). \tag{35}$$

The lognormal process is approximated using the four-dimensional third order polynomial chaos expansion ($L = 34$). The maximum size of the finite element mesh considered consists of $494,322$ elements and $247,159$ nodes. The non-Gaussian response is expressed using the third order polynomial chaos expansion ($N = 34$) leading to a linear system of order $8,650,565$.

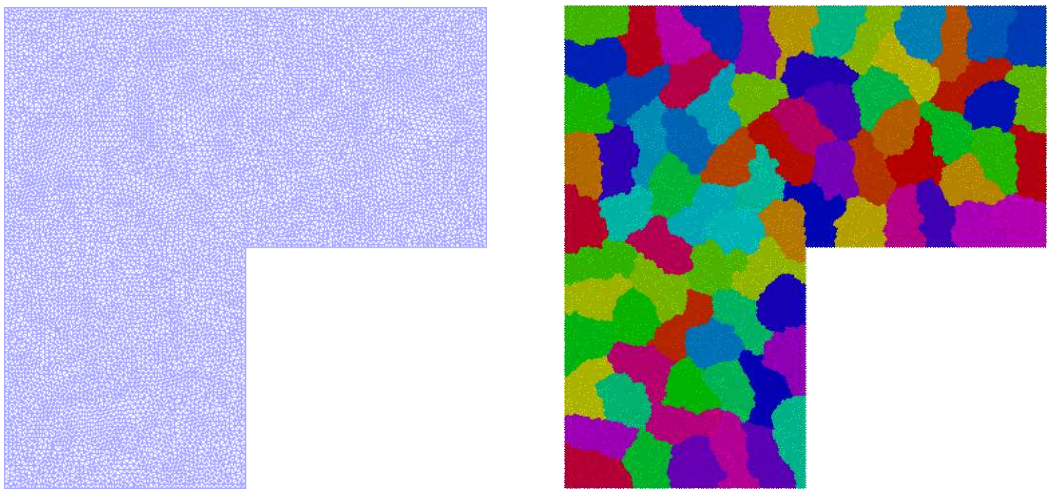
In PCGM implementation, the forcing term is taken to be the initial residual. The iterations are terminated when the ratio of L_2 norms of the current and the initial residual is less than 10^{-6} defined by

$$\frac{\|\mathcal{G}_\Gamma^k - \mathcal{S}\mathcal{U}_\Gamma^k\|_2}{\|\mathcal{G}_\Gamma^0\|_2} \leqslant 10^{-6}. \tag{36}$$

Numerical experiments are performed in a Linux cluster having 22 nodes (2 Quad-Core 3.0 GHz Intel Xeon processors and 32 GB of memory per node) with InfiniBand interconnect using Message Passing Interface (MPI) [12] and PETSc [11] parallel libraries.

6.1. Stochastic features

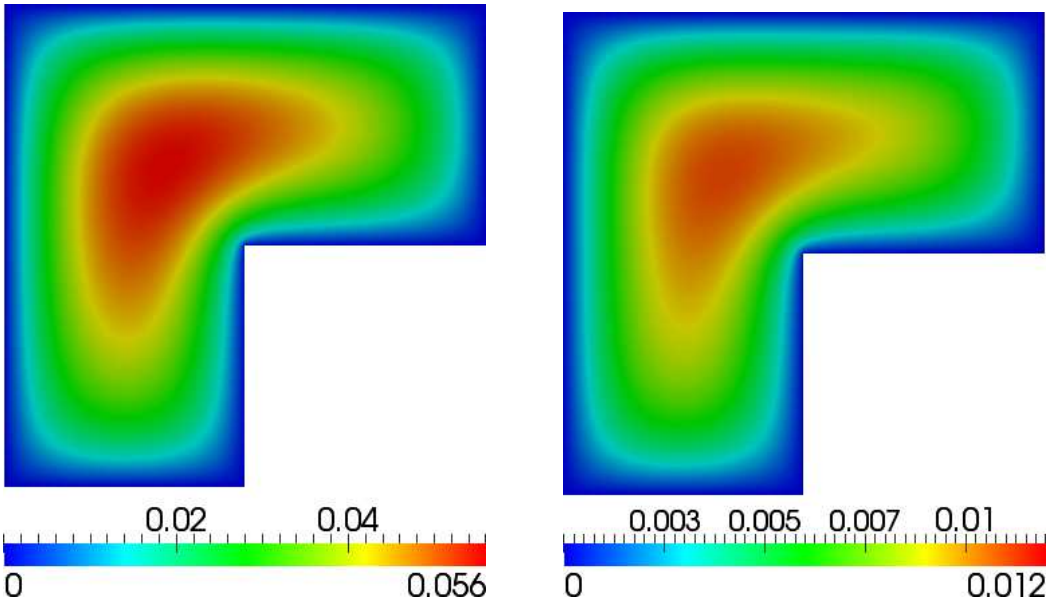
Fig.(3-a) shows the physical domain and Fig.(3-b) represents a typical partitioned mesh using METIS graph partitioner [13]. METIS performs mesh decomposition using the criteria of load balancing among the CPUs and minimum interface boundary among subdomains.



(a) FEM mesh (b) Decomposed mesh

Figure 3. FEM mesh and typical mesh decomposition

The mean and standard deviation of the non-Gaussian solution process are shown in Fig.(4-a) and Fig.(4-b) respectively. The maximum value of the coefficient of variation (CoV) of the solution field is about 21% highlighting the effect of uncertainty.

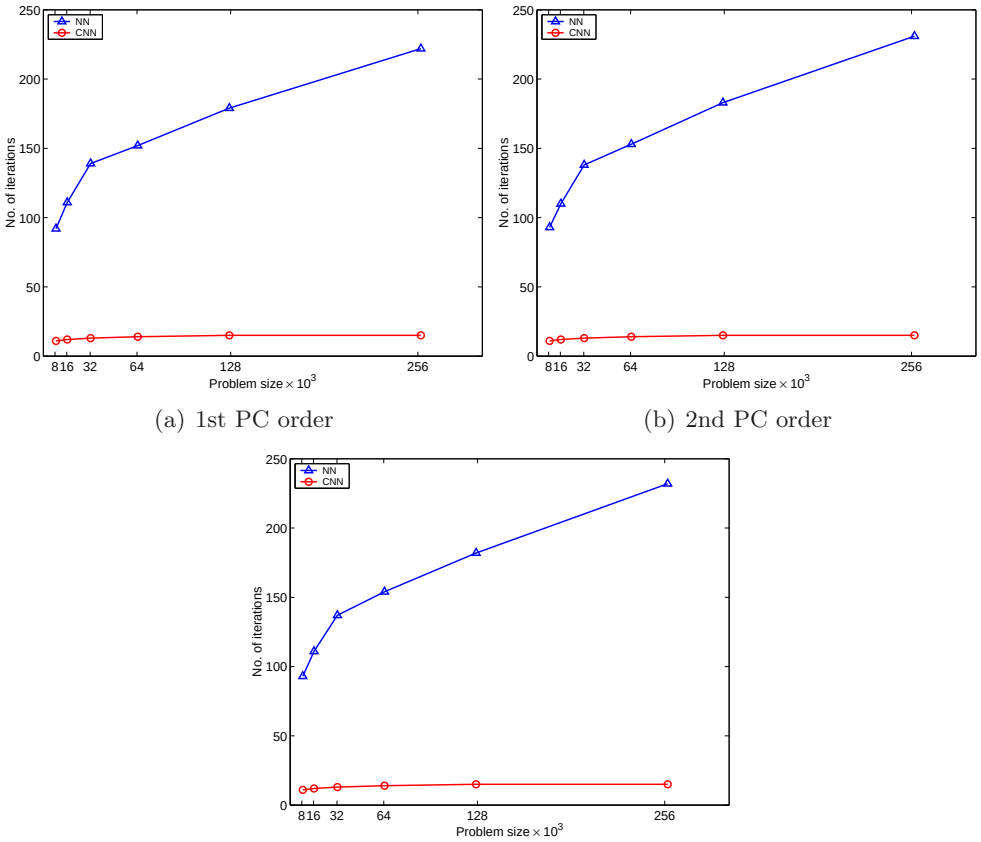


(a) Mean (b) Standard deviation

Figure 4. The mean and standard deviation of the solution process

6.2. Scalability study
Next we investigate the numerical scalability of the two-level Neumann-Neumann preconditioner (CNN) with respect to mesh size (h), subdomain size (H) and fixed problem size per subdomain (h/H) as discussed in the following subsections. In order to demonstrate the usefulness of the coarse grid, we contrast the performance of the one-level (NN) and two-level Neumann-Neumann (CNN) preconditioner.

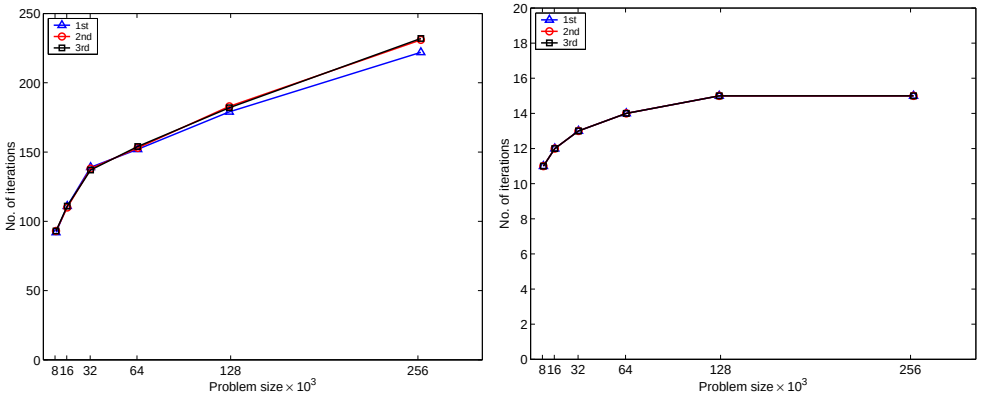
6.2.1. Scalability with respect to mesh size: For this scalability test we fix the number of subdomains to 128 while increasing the mesh resolution in the geometric dimension. Fig.(5) shows the results for both NN and CNN preconditioners for the first, second and third polynomial chaos (PC) expansions while Fig.(6) shows the same results for NN and CNN preconditioners separately for various order of PC expansions. Unlike NN preconditioner, increasing mesh resolution does not deteriorate the performance of the CNN preconditioner as the iteration counts of PCGM solver remains nearly constant. For NN preconditioner the iteration counts increases as the problem size grows indicating the dependency of the NN preconditioner on the problem size. While increasing the order of PC expansions does not affect the performance of CNN preconditioner, it degrades the performance of NN preconditioner. For a given spatial problem size (n), the first and third order PC expansions lead to a total problem size of $(5 \times n)$ and $(35 \times n)$ respectively. These performance results suggest that the CNN preconditioner is numerically scalable with respect to the problem size and order of PC expansion.



(a) 1st PC order (b) 2nd PC order

(c) 3rd PC order

Figure 5. Iteration counts for fixed number of subdomains

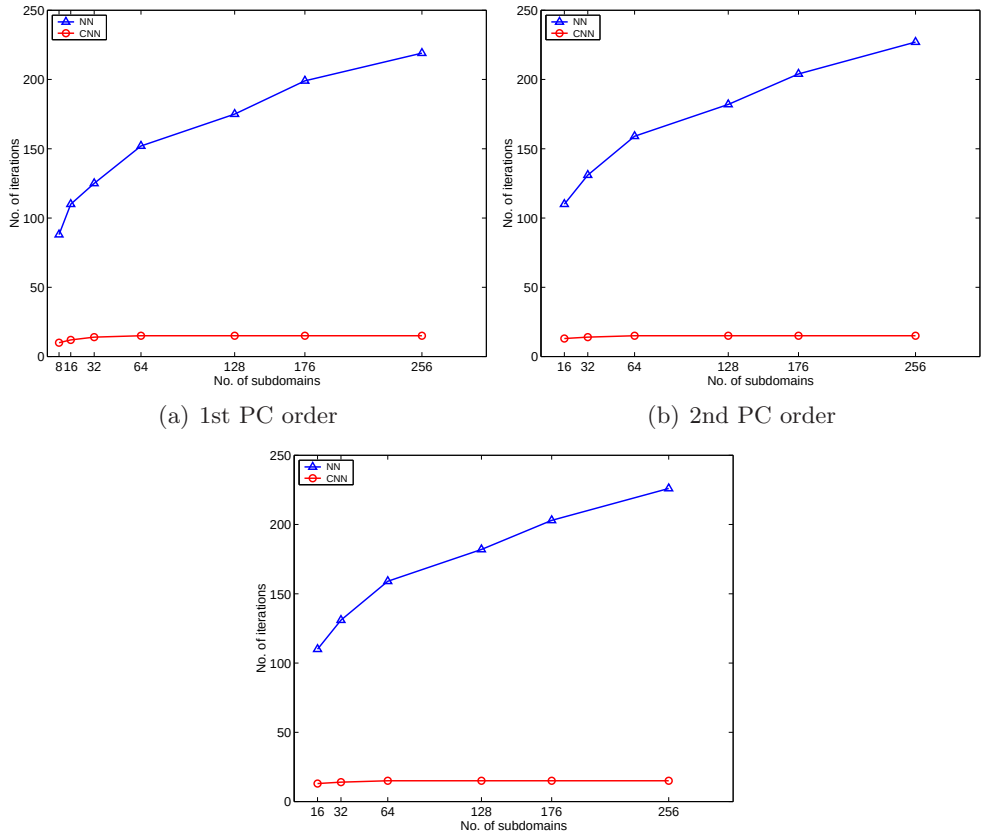


(a) NN preconditioner (b) CNN preconditioner

Figure 6. Iteration counts for fixed number of subdomains

6.2.2. Scalability with respect to subdomain size: Next we fix the problem size in the spatial dimension to $151,179$ and increase number of subdomains to solve the problem. In Fig.(7) and Fig.(8), the results are reported, respectively, for the first, second and third order PC expansions for both NN and CNN preconditioners. These performance results suggest that the

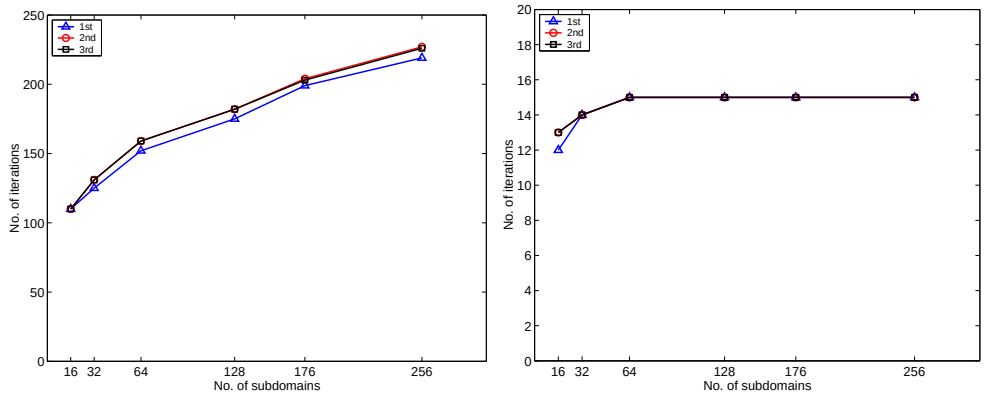
CNN preconditioner is numerically scalable with respect to number of subdomains. Clearly, NN preconditioner requires much more number of iterations to converge compared to CNN preconditioner.



(a) 1st PC order (b) 2nd PC order

(c) 3rd PC order

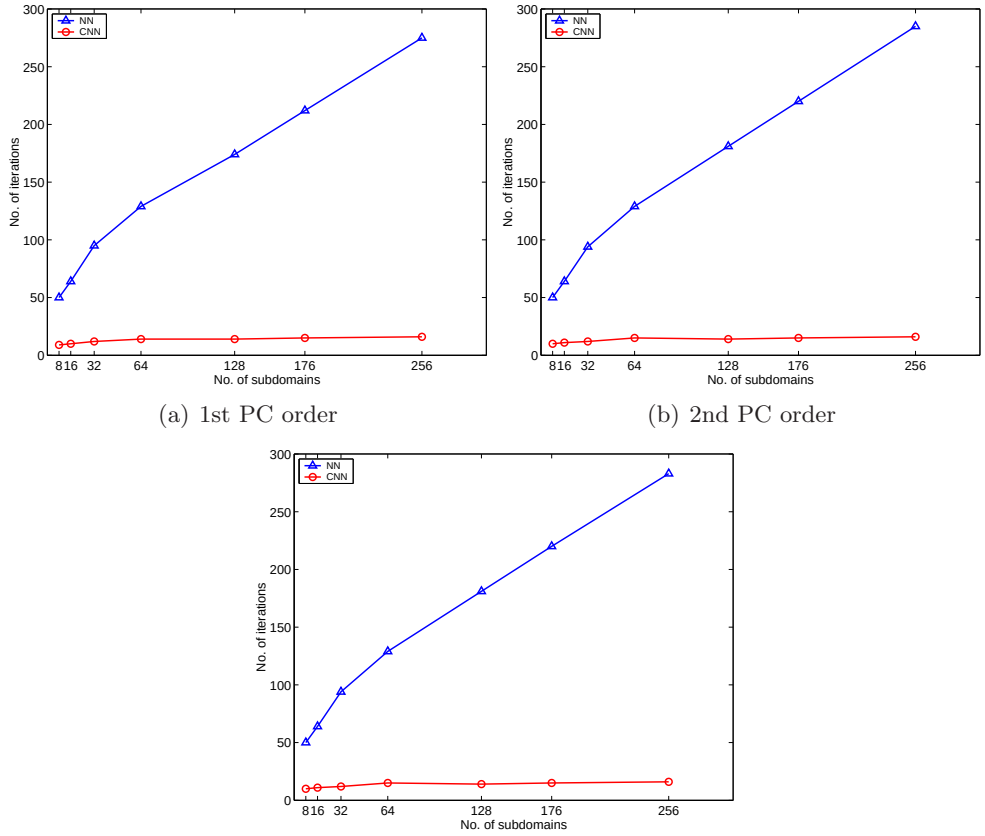
Figure 7. Iteration counts for fixed problem size



(a) NN preconditioner (b) CNN preconditioner

Figure 8. Iteration counts for fixed problem size

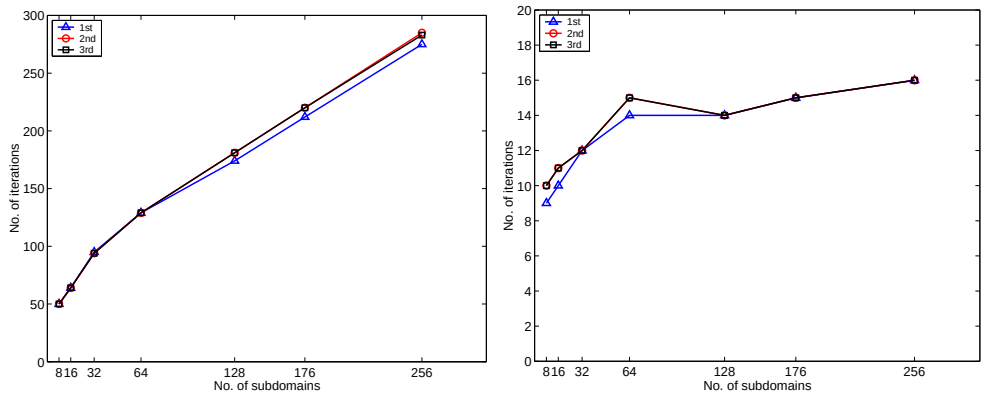
6.2.3. Scalability with respect to the fixed problem size per subdomain: Fig.(9) and Fig.(10) demonstrate the performance of NN and CNN preconditioners for the fixed problem size per subdomain while increasing the overall problem size by adding more subdomains. The results are shown for the first, second and third order PC expansions for both NN and CNN preconditioners. In contrast to NN preconditioner, these performance results suggest that CNN preconditioner is scalable with respect to the fixed problem size per subdomain.



(a) 1st PC order (b) 2nd PC order

(c) 3rd PC order

Figure 9. Iteration counts for fixed problem size per subdomain



(a) NN preconditioner (b) CNN preconditioner

Figure 10. Iteration counts for fixed problem size per subdomain

7. Conclusions

A novel two-level scalable domain decomposition preconditioner is described for the iterative solutions of large-scale linear systems arising from the stochastic finite element method. The preconditioner is equipped with a coarse grid to propagate information globally across the subdomains both in geometric and stochastic dimensions. The proposed preconditioner may be viewed as an extension of the Balancing Domain Decomposition by Constraints (BDDC) method for stochastic PDEs. For the specific illustrations considered in the paper, the numerical experiment demonstrates that the preconditioner is numerically scalable with respect to the problem size, subdomain size and fixed problem size per subdomain. Furthermore, the preconditioner shows a convergence rate nearly independent of the order of the polynomial chaos expansion (namely the order of stochastic dimension).

References

[1] Ghanem R and Spanos P 1991 *Stochastic Finite Element: A Spectral Approach* (New York: Springer-Verlag)
[2] Sarkar A, Benabbou N and Ghanem R 2009 *International Journal for Numerical Methods in Engineering* **77** 689–701
[3] Subber W, Monajemi H, Khalil M and Sarkar A 2008 *International Symposium on Uncertainties in Hydrologic and Hydraulic*
[4] Subber W and Sarkar A 2010 *Proceddings of High Performance Computing Systems and Applications* **5976** 251–268
[5] Toselli A and Widlund O 2005 *Domain Decomposition Methods - Algorithms and Theory* (*Springer Series in Computational Mathematics* vol 34) (Berlin: Springer)
[6] Mathew T 2008 *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations* (*Lecture Notes in Computational Science and Engineering* no 61) (Berlin: Springer)
[7] Sousedik B and Mandel J 2008 *Electronic Transactions in Numerical Analysis* **31** 384–402
[8] Farhat C, Lesoinne M and Pierson K 2000 *Numerical Linear Algebra with Applications* **7**(7-8) 687–714
[9] Cros J 2003 *Fourteenth International Conference on Domain Decomposition Methods* 373–380
[10] Subber W and Sarkar A 2010 *Journal of Physics: Conference Series* **256**
[11] Balay S, Buschelman K, Gropp W, Kaushik D, Knepley M, McInnes L, Smith B and Zhang H 2009 PETSc Web page http://www.mcs.anl.gov/petsc
[12] Message passing interface forum http://www.mpi-forum.org
[13] Karypis G and Kumar V 1995 METIS unstructured graph partitioning and sparse matrix ordering system
[14] Cameron R H and Martin W T 1947 *Ann. Math* **48** 385 – 392
[15] Wiener N 1938 *Amer. J. Math.* **60** 897 – 936
[16] Saad Y 2003 *Iterative methods for sparse linear systems* 2nd ed (Philadelphia: SIAM)
[17] Smith B, Bjorstad P and Gropp W 1996 *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations* (New York: Cambridge University Press)