DOMAIN DECOMPOSITION OF STOCHASTIC PDEs AND ITS PARALLEL IMPLEMENTATION

Abhijit Sarkar, *Carleton University, Ottawa, ON, Canada*
Nabil Benabbou, *HPCVL, University of Ottawa, ON, Canada*
Roger Ghanem, *University of Southern California, California, USA*

Abstract

A parallel algorithm is developed for the domain decomposition of uncertain dynamical systems defined by stochastic partial differential equations. The methodology is particularly amenable to parallel processing for effective exploitation of the computational and storage capability of currently available multiprocessing computational environment. The formulation is tailored towards efficient memory usage and minimum inter-processor communication for shared memory parallelism on cluster of symmetric multiprocessor (SMP) machines. To this end, Message Passing Interface (MPI) is used in conjunction with openMP-based explicit Multi-threading as a second level parallelization to enhance performance of the stochastic domain decomposition method. MPI is used to dynamically decompose and process each subdomain among compute nodes while OpenMP directives achieves the second level parallelism involving loop-level iteration necessary for processing each substructure assigned to the given node. The MPI-OpenMP based hybrid code is designed to be compiled just serially (without requiring the use of MPI or openMP library) and with any combination of MPI and OpenMP enabled. This approach permits a systematic study on the performance improvement of the multi-level parallelism of the domain decomposition method for the stochastically uncertain dynamical systems.

1 INTRODUCTION

Firstly, a parallel algorithm is developed for the domain decomposition of uncertain dynamical systems defined by stochastic partial differential equations. The methodology is particularly amenable to parallel processing for effective exploitation of the computational and storage capability of currently available multiprocessing computational environment. Secondly, the formulation is tailored towards efficient memory usage and minimum inter-processor communication for shared memory parallelism on cluster of symmetric multiprocessor (SMP) machines. To this end, Message Passing Interface (MPI) is used in conjunction with openMP-based explicit Multi-threading as a second level parallelization to enhance performance of the stochastic domain decomposition method. MPI is used to dynamically decompose and process each subdomain among compute nodes while OpenMP directives achieves the second level parallelism involving loop-level iteration necessary for processing each substructure assigned to the given node. The MPI-OpenMP based hybrid code is designed to be compiled just serially (without requiring the use of MPI or openMP library) and with any combination of MPI and OpenMP enabled. This approach permits a systematic study on the performance improvement of the multi-level parallelism of the domain decomposition method for the stochastically uncertain dynamical systems

2 DISCRETE REPRESENTATIONS OF PARTIAL DIFFERENTIAL EQUATIONS

Consider the wave equation defined on a region Ω with given Dirichlet, Neumann or mixed (Robin) data on the boundary $\partial\Omega$ as shown in Figure 1. The frequency domain finite element approximation of the solution in an n-dimensional subspace is given by,

$$\mathbf{A}_n\mathbf{U} = \mathbf{F}_n \tag{1}$$

where the dynamic stiffness matrix, $\mathbf{A}_n$ is given by,

$$\mathbf{A}_n = -\omega^2\mathbf{M}_n + i\omega\mathbf{D}_n + \mathbf{K}_n, \tag{2}$$

with $\mathbf{M}_n \in \mathbb{R}^{n\times n}$, $\mathbf{D}_n \in \mathbb{R}^{n\times n}$, $\mathbf{K}_n \in \mathbb{R}^{n\times n}$ and $\mathbf{F}_n \in \mathbb{C}^n$ being the mass, damping, stiffness matrices and force vector, respectively. For uncertain dynamical system whose parameters are modeled as stochastic processes, the elements of $\mathbf{M}_n$, $\mathbf{D}_n$, and $\mathbf{K}_n$ are random variables satisfying certain correlation properties among themselves.

Note that only a fine finite element mesh resolution can effectively resolve the spatial fluctuations due to short wavelength vibration in the high-frequency range and thereby resulting a large-scale stiffness matrix $\mathbf{A}_n$. Futhermore, presence of imperfections, discontinuities and incomplete or imprecise information on the system may significantly diminish the confidence in the predicted response even with the detailed *deterministic* finite element model. Therefore, probabilistic models are usually adopted to quantify confidence on the predicted response and thereby increasing the computational overhead further. It is thereforeworthwhile to exploit the multi-processing capability of high performance computingplatform to tackle such problems.

In the preceding paragraphs, the formulation is described in the context of entire structural system. For a large-s

Proceedings of the 20th International Symposium on High-Performance Computing in an Advanced Collaborative Environment (HPCS'06)
0-7695-2582-2/06 $20.00 © 2006 IEEE

system, it becomes often convenient and necessary to decompose the total system into a number of subsystems and to construct the model of each subsystem separately and concurrently. In Figure 1, we assume that Ω is partitioned into two nonoverlapping subdomains Ω_1 and Ω_2 such that $\Omega = \Omega_1 \cup \Omega_2$, $\Omega_1 \cap \Omega_2 = \emptyset$ and $\Gamma = \partial\Omega_I \cap \partial\Omega_2$ where Γ defines the interface of Ω_1 and Ω_2. Under suitable regularity assumption on the boundaries (Lipschitz continuous) and forcing (typically square-summable), the dynamic equilibrium equation of a typical subdomain (identified with the superscript s), having m_s degrees of freedom(dof), can be expressed as,

$$\begin{bmatrix} \mathbf{A}^s_{\Gamma\Gamma}(\omega) & \mathbf{A}^s_{\Gamma I}(\omega) \\ \mathbf{A}^s_{I\Gamma}(\omega) & \mathbf{A}^s_{II}(\omega) \end{bmatrix} \begin{Bmatrix} \mathbf{u}^s_\Gamma \\ \mathbf{u}^s_I \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^s_\Gamma \\ \mathbf{f}^s_I \end{Bmatrix}. \tag{3}$$

The partitions Γ and I relate to the boundary dofs m_Γ and interior dofs m_I of the substructure with $m_s = m_\Gamma + m_I$. This partition conveniently allows one to perform the assembly of all the substructure within a finite elements framework.

3 DETERMINISTIC DOMAIN DECOMPOSITION METHOD

Numerical solutions of partial differential equations (PDEs) arising in the physical problems of practical interest ofen lead to large scale algebraic equation. In this context, the domain decomposition method (Toselli and Widlund 2004, Quarteroni and Valli 1999, Smith *et al.* 1996) refers to splitting the computational domain into number of (overlapping or nonoverlapping) subdomains. The original problem involving the numerical solution of the PDEs is reformulated into each subproblems of reduced size. Evidently, the subproblems defined over multiple domains are coupled through their interface boundary conditions to be matched among all adjacent subdomains. Note that the term domain decompostion may be used in different contexts (e.g. process of distributing data from computational model among the processors of distributed memory machines, see Smith *et al.* 1996). In the present context, we will strictly refer to it as the process of decomposing a large-scale algebraic system into a number of sub-problems of reduced size to arrive at the solution with minimal computational costs.

In addition to the usefulness of dividing the computational tasks for solving a large-scale problem, domain decomposition method is most effective in devising parallel algorithms, as mandatory for large-scale numerical problems of practical interest, which conveniently exploits currently available multi-processing computing environment.

Furthermore, domain decomposition method is particularly suitable for tackling multi-physics problem whereby different interacting energy media (e.g. solid and fluid interaction problems) are coupled through common interfaces; and the physics of different media are governed by different types of partial differential equations which are solved by different numerical methods (e.g., finite element method, computational fluid dynamics (CFD) approach etc.).

For the sake of elementary exposition, we consider again the wave equation on region Ω partitioned into two nonoverlapping subdomains Ω_1 and Ω_2 as in Figure 1. Once the FEM models of each subdomain is constructed as described by Equation (3), the equilibrium equation of the system is described in the matrix form as,

$$\begin{bmatrix} [\mathbf{A}^1_{II}]_{m_1\times m_1} & 0 & [\mathbf{A}^1_{I\Gamma}]_{m_1\times m_\Gamma} \\ 0 & [\mathbf{A}^2_{II}]_{m_2\times m_2} & [\mathbf{A}^2_{I\Gamma}]_{m2\times m_\Gamma} \\ [\mathbf{A}^1_{\Gamma I}]_{m_\Gamma\times m_1} & [\mathbf{A}^2_{\Gamma I}]_{m_\Gamma\times m_2} & [\mathbf{A}^1_{\Gamma\Gamma} + \mathbf{A}^2_{\Gamma\Gamma}]_{m_2\times m_2} \end{bmatrix} \times \tag{4}$$
$$\begin{Bmatrix} \mathbf{u}^1_I \\ \mathbf{u}^2_I \\ \mathbf{u}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^1_I \\ \mathbf{f}^2_I \\ \mathbf{f}^1_\Gamma + \mathbf{f}^2_\Gamma \end{Bmatrix}$$

The above equilibrium equation can be partitioned and rearranged into following explicit form:

$$[\mathbf{A}^1_{II}]\{\mathbf{u}^1_I\} = \{\mathbf{f}^1_I\} - [\mathbf{A}^1_{I\Gamma}]\{\mathbf{u}_\Gamma\} \tag{5}$$

$$[\mathbf{A}^2_{II}]\{\mathbf{u}^2_I\} = \{\mathbf{f}^2_I\} - [\mathbf{A}^2_{I\Gamma}]\{\mathbf{u}_\Gamma\} \tag{6}$$

$$\underbrace{[[\mathbf{A}^1_{\Gamma\Gamma}] - [\mathbf{A}^1_{\Gamma I}][\mathbf{A}^1_{II}]^{-1}[\mathbf{A}^1_{I\Gamma}]}_{S_1} + \underbrace{[\mathbf{A}^2_{\Gamma\Gamma}] - [\mathbf{A}^2_{\Gamma I}][\mathbf{A}^2_{II}]^{-1}[\mathbf{A}^2_{I\Gamma}]]}_{S_2}\{\mathbf{u}_\Gamma\} \tag{7}$$
$$= \underbrace{[\{\mathbf{f}^1_\Gamma\} - [\mathbf{A}^1_{\Gamma I}][\mathbf{A}^1_{II}]^{-1}\{\mathbf{f}^1_I\}]}_{F_1} + \underbrace{[\{\mathbf{f}^2_\Gamma\} - [\mathbf{A}^2_{\Gamma I}][\mathbf{A}^2_{II}]^{-1}\{\mathbf{f}^2_I\}]}_{F_2}$$

Equation (7) represents the so-called *interface* problem. The coefficient matrix $S = S_1 + S_2$ are known as *Schur complement* matrix. Note that S_1 and S_2 can be constructed in parallel for each subdomains. The size of S is normally much smaller (but more dense) compared to the coefficient matrix in Equation (4). The interface problem defined in Equation (7) defines the coupling between the subdomains. Once the interface problem is solved to obtain the interface unknown vector $\mathbf{u}_\Gamma$, Equations (5) and (6) can be solved to obtain the internal unknown vectors $\mathbf{u}^1_I$ and $\mathbf{u}^2_I$. It is interesting to note that solutions for $\mathbf{u}^1_I$ and $\mathbf{u}^2_I$ are perfectly parallelizable once the interface unknown vector $\mathbf{u}_\Gamma$ and the size of these problems depends on the granularity of the domain decomposition scheme.

When Schur complements are explicitly constructed, the domain decomposition method is referred to as *direct Schur method or substructuring* which is a common practice is the structural engineering community. When the order $\mathbf{u}_\Gamma$ is large, explicit construction of Schur complement matrix S is not practicable as matrix S is dense and computationally expensive to construct. In such cases, iterative methods (e.g. conjugate gradient method) is preferred (Toselli and Widlund 2004, Quarteroni and Valli 1999, Smith *et al.* 1996) and domain decomposition method in this cased is referred to as *iterative Schur complement method)*.

4 DOMAIN DECOMPOSITION METHOD OF STOCHASTIC SYSTEM

A fine resolution of finite element mesh is mandatory to capture small wave-length vibration features in the high-frequency range which yields very large-scale computational model. Furthermore, confidence level in the numerical predictions diminishes even with such detailed determin-
istic model arising from extreme sensitivity of the respo

due to small perturbation in the system parameters as well as due to *unmodelled* dynamics of the system. Consequently, probabilistic approaches are adopted in order to quantify the confidence in numerical predictions which significantly burdens the computational requirement (see Ref 1, 5-7).

We start with the assumption that the uncertainty in the systems parameters can be represented by stochastic processes. Once their correlation properties are known, these original random processes and their associated transformations can be effectively represented by a suitably chosen basis functions, namely *polynomial chaos* basis Ψ_i defined over the space of all second-order random variables with finite variance (Ghanem and Spanos 1991). The equilibrium equation of the stochastic systems can be written (analogous to its deterministic counterpart in Equation (4)

$$\left[\sum_{i=0}^{L} \Psi_i \mathbf{A}_i\right] \hat{\mathbf{u}} = \mathbf{f} \tag{8}$$

where L is the number terms retained in the random series expansion of the random stiffness parameter. Note that $\mathbf{A}_0$ refers to the mean reduced dynamic stiffness matrix as given by the left-hand side matrix in Equation (4). The additional stiffness terms arising due to stochastic fluctuation in the system parameters are given by

$$\mathbf{A}_i = \begin{bmatrix} [\mathbf{A}^1_{II,i}]_{m_1\times m_1} & 0 & [\mathbf{A}^1_{I\Gamma,i}]_{m_1\times m_\Gamma} \\ 0 & [\mathbf{A}^2_{II,i}]_{m_2\times m_2} & [\mathbf{A}^2_{I\Gamma,i}]_{m_2\times m_\Gamma} \\ [\mathbf{A}^1_{\Gamma I,i}]_{m_\Gamma\times m_1} & [\mathbf{A}^2_{\Gamma I,i}]_{m_\Gamma\times m_2} & [\mathbf{A}^1_{\Gamma\Gamma,i} + \mathbf{A}^2_{\Gamma\Gamma,i}]_{m_2\times m_2} \end{bmatrix} \tag{9}$$

Expanding the solution vector $\hat{\mathbf{u}}$ using the same basis functions of order N

$$\hat{\mathbf{u}} = \sum_{j=0}^{N} \Psi_j \hat{\mathbf{u}}_\mathbf{j} \,, \tag{10}$$

and then substituting this expression in equilibrium equation (Eq. 8) and performing error minimization leads to (see Ref. 1 for details)

$$\sum_{j=0}^{N}\sum_{i=0}^{L} \langle \Psi_i \Psi_j \Psi_k \rangle \mathbf{A}_i \hat{\mathbf{u}}_j = \langle \Psi_k \mathbf{f} \rangle, \quad k = 0, \ldots, N \tag{11}$$

We now define the following new variables as follows

$$\mathcal{A}_{jk} = \sum_{i=0}^{L} \langle \Psi_i \Psi_j \Psi_k \rangle \mathbf{A}_i, \quad \mathcal{F}_k = \langle \Psi_k \mathbf{f} \rangle \tag{12}$$

$$\hat{\mathbf{u}}_j = \begin{Bmatrix} \hat{\mathbf{u}}^1_{I,j} \\ \hat{\mathbf{u}}^2_{I,j} \\ \hat{\mathbf{u}}_{\Gamma,j} \end{Bmatrix} \tag{13}$$

$$\mathcal{F}_k = \begin{Bmatrix} \mathcal{F}^1_{I,k} \\ \mathcal{F}^2_{I,k} \\ \mathcal{F}^1_{\Gamma,k} + \mathcal{F}^2_{\Gamma,k} \end{Bmatrix}, \quad k = 0, \ldots, N \tag{14}$$

Consequently, Eq. (11) can be re-written as

$$\sum_{j=0}^{N}\sum_{i=0}^{L} \langle \Psi_i \Psi_j \Psi_k \rangle \begin{bmatrix} [\mathbf{A}^1_{II,i}] & 0 & [\mathbf{A}^1_{I\Gamma,i}] \\ 0 & [\mathbf{A}^2_{II,i}] & [\mathbf{A}^2_{I\Gamma,i}] \\ [\mathbf{A}^1_{\Gamma I,i}] & [\mathbf{A}^2_{\Gamma I,i}] & [\mathbf{A}^1_{\Gamma\Gamma,i} + \mathbf{A}^2_{\Gamma\Gamma,i}] \end{bmatrix} \times \tag{15}$$
$$\begin{Bmatrix} \hat{\mathbf{u}}^1_{I,j} \\ \hat{\mathbf{u}}^2_{I,j} \\ \hat{\mathbf{u}}_{\Gamma,j} \end{Bmatrix} = \begin{Bmatrix} \mathcal{F}^1_{I,k} \\ \mathcal{F}^2_{I,k} \\ \mathcal{F}^1_{\Gamma,k} + \mathcal{F}^2_{\Gamma,k} \end{Bmatrix}, \quad k = 0, \ldots, N$$

Further rearrangement of of Eq. (15) leads to,

$$\begin{bmatrix} [\mathcal{A}^1_{II}]_{m_1N\times m_1N} & 0 & [\mathcal{A}_{I\Gamma}]_{m_1N\times m_\Gamma N} \\ 0 & [\mathcal{A}^2_{II}]_{m_2N\times m_2N} & [\mathcal{A}^2_{I\Gamma}]_{m_2N\times m_\Gamma N} \\ [\mathcal{A}^1_{\Gamma I}]_{m_\Gamma N\times m_1N} & [\mathcal{A}^2_{\Gamma I}]_{m_\Gamma N\times m_2N} & [\mathcal{A}^1_{\Gamma\Gamma} + \mathcal{A}^2_{\Gamma\Gamma}]_{m_\Gamma N\times m_\Gamma N} \end{bmatrix} \times \tag{16}$$
$$\begin{Bmatrix} \hat{\mathbf{u}}^1_I \\ \hat{\mathbf{u}}^2_I \\ \hat{\mathbf{u}}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \mathcal{F}^1_I \\ \mathcal{F}^2_I \\ \mathcal{F}_\Gamma \end{Bmatrix}$$

where

$$\begin{Bmatrix} \hat{\mathbf{u}}^1_I \\ \hat{\mathbf{u}}^2_I \\ \hat{\mathbf{u}}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \begin{Bmatrix} \hat{\mathbf{u}}^1_{I,0} \\ \hat{\mathbf{u}}^1_{I,1} \\ \ldots \\ \hat{\mathbf{u}}^1_{I,N} \end{Bmatrix} \\ \begin{Bmatrix} \hat{\mathbf{u}}^2_{I,0} \\ \hat{\mathbf{u}}^2_{I,1} \\ \ldots \\ \hat{\mathbf{u}}^2_{I,N} \end{Bmatrix} \\ \begin{Bmatrix} \hat{\mathbf{u}}_{\Gamma,0} \\ \hat{\mathbf{u}}_{\Gamma,1} \\ .. \\ \hat{\mathbf{u}}_{\Gamma,N} \end{Bmatrix} \end{Bmatrix}, \quad \begin{Bmatrix} \mathcal{F}^1_I \\ \mathcal{F}^2_I \\ \mathcal{F}_\Gamma \end{Bmatrix} = \begin{Bmatrix} \begin{Bmatrix} \mathcal{F}^1_{I,0} \\ \mathcal{F}^1_{I,1} \\ \ldots \\ \mathcal{F}^1_{I,N} \end{Bmatrix} \\ \begin{Bmatrix} \mathcal{F}^2_{I,0} \\ \mathcal{F}^2_{I,1} \\ \ldots \\ \mathcal{F}^2_{I,N} \end{Bmatrix} \\ \begin{Bmatrix} \mathcal{F}_{\Gamma,0} \\ \mathcal{F}_{\Gamma,1} \\ .. \\ \mathcal{F}_{\Gamma,N} \end{Bmatrix} \end{Bmatrix}$$

Equation (16) can also be expressed as following set of equations:

$$\sum_{j=0}^{N} [\mathcal{A}^1_{II}]_{jk} \{\hat{\mathbf{u}}^1_{I,j}\} + \sum_{j=0}^{N} [\mathcal{A}^1_{I\Gamma}]_{jk} \{\hat{\mathbf{u}}_{\Gamma,j}\} = \{\mathcal{F}^1_{I,k}\}, \quad k = 0, \ldots, N \tag{17}$$

$$\sum_{j=0}^{N} [\mathcal{A}^2_{II}]_{jk} \{\hat{\mathbf{u}}^2_{I,j}\} + \sum_{j=0}^{N} [\mathcal{A}^2_{I\Gamma}]_{jk} \{\hat{\mathbf{u}}_{\Gamma,j}\} = \{\mathcal{F}^2_{I,k}\}, \quad k = 0, \ldots, N \tag{18}$$

$$\sum_{j=0}^{N} [\mathcal{A}^1_{\Gamma I}]_{jk} \{\hat{\mathbf{u}}^1_{I,j}\} + \sum_{j=0}^{N} [\mathcal{A}^2_{\Gamma I}]_{jk} \{\hat{\mathbf{u}}^2_{I,j}\} + \sum_{j=0}^{N} [[\mathcal{A}^1_{\Gamma\Gamma}]_{jk} + [\mathcal{A}^2_{\Gamma\Gamma}]_{jk}] \{\hat{\mathbf{u}}_{\Gamma,j}\} \tag{19}$$
$$= \{\mathcal{F}^1_{I,k}\} + \{\mathcal{F}^2_{I,k}\}, \quad k = 0, \ldots, N$$

Partitioning and rearrangement of Eqs. (17-19) leads to the following three equations,

$$[\mathcal{A}^1_{II}]\{\hat{\mathbf{u}}^1_I\} = \{\mathcal{F}^1_I\} - [\mathcal{A}^1_{I\Gamma}]\{\hat{\mathbf{u}}_\Gamma\} \tag{20}$$

$$[\mathcal{A}^2_{II}]\{\hat{\mathbf{u}}^2_I\} = \{\mathcal{F}^2_I\} - [\mathcal{A}^2_{I\Gamma}]\{\hat{\mathbf{u}}_\Gamma\} \tag{21}$$

$$\underbrace{[[\mathcal{A}^1_{\Gamma\Gamma}] - [\mathcal{A}^1_{\Gamma I}][\mathcal{A}^1_{II}]^{-1}[\mathcal{A}^1_{I\Gamma}]}_{S^{stoc}_1} + \underbrace{[\mathcal{A}^2_{\Gamma\Gamma}] - [\mathcal{A}^2_{\Gamma I}][\mathcal{A}^2_{II}]^{-1}[\mathcal{A}^2_{I\Gamma}]]}_{S^{stoc}_2} \{\hat{\mathbf{u}}_\Gamma\}$$
$$= \underbrace{[\{\mathcal{F}^1_I\} - [\mathcal{A}^1_{\Gamma I}][\mathcal{A}^1_{II}]^{-1}\{\mathcal{F}^1_I\}]}_{F^{stoc}_1} + \underbrace{[\{\mathcal{F}^2_I\} - [\mathcal{A}^2_{\Gamma I}][\mathcal{A}^2_{II}]^{-1}\{\mathcal{F}^2_I\}]}_{F^{stoc}_2} \tag{22}$$

Note that we can draw an analogy of Eqs. 20-22 to Eqs. 5-7. In the contexts of stochastic systems, the matrix $S^{stoc} = S_1^{stoc} + S_2^{stoc}$ in Eq. (22) can be construed to be the stochastic counterpart of Schur complement matrix in analogy to Eq. (7) and hence will be termed as *stochastic Schur complement* matrix. In analogy with Eqs. 5-6, the *internal* unknown vectors $\hat{\mathbf{u}}_I^1$ and $\hat{\mathbf{u}}_I^2$ in Eqs (17,18) can be solved *concurrently* once the interface unknown $\hat{\mathbf{u}}_\Gamma$ are solved in Eq. (19). Similar to the deterministic case, S_1^{stoc} and S_2^{stoc} can be constructed in parallel and of course, the order of the interface problem in Eq. (19) is much smaller compared to the complete system in Eq. (16). This fact substantially minimizes the computational overhead in tackling the solution of the stochastic system.

At this stage, it is worthwhile to point out that the ability to perform a *functional decomposition* of the stochastic solution vector in the probabilistic space, namely in the space of Polynomial Chaos permits a direct comparison of the domain decomposition method in the cases of deterministic and stochastic systems. Therefore, the above approach which simultaneously capitalizes on a domain decomposition method in the physical space and a functional decomposition method in the probabilistic space can elegantly exploit the numerical techniques developed for domain decomposition of deterministic systems with minor modifications.

5 THE CASE OF MANY SUBDOMAINS

In the preceding sections, the methodology introduced was restricted to the case of two subdomains in order to provide detailed exposition of the mathematical formulation. In this section, a straightforward generalized of the methodology is presented to the case of many subdomains. Let $[\hat{R}]_i$ be the restriction matrix consisting of zeros and ones for each subdomain Ω_i which relates the Polynomial Chaos coefficients of global interface variable $\{\hat{\mathbf{u}}_{\Gamma,j}\}$ to that of local interface variable of subdomain Ω_i, namely $\{\hat{\mathbf{u}}^i_{\Gamma,j}\}$ as stated below

$$\{\hat{\mathbf{u}}_{\Gamma,j}\} = \sum_{i=1}^{nsb} [\hat{R}]_i^T \{\hat{\mathbf{u}}^i_{\Gamma,j}\}, \quad k = 0, \ldots, N \tag{23}$$

The above equation can be presented in the following concise form as

$$\{\hat{\mathbf{u}}_\Gamma\} = \sum_{i=1}^{nsb} [\mathcal{R}]_i^T \{\hat{\mathbf{u}}^i_\Gamma\}, \tag{24}$$

where the restriction matrix $[\mathcal{R}]_i$ consists of N block submatrices with each block submatrix being $[\hat{R}]_i$. Following Ref[2,3], it can be shown that the global interface problem (involving reduced subdomain Schur complement matrices) for the stochastic case takes the following form:

$$\sum_{i=1}^{nsb} [\mathcal{R}]_i^T [S_i^{stoc}][\mathcal{R}]_i \{\hat{\mathbf{u}}_\Gamma\} = \sum_{i=1}^{nsb} [\mathcal{R}]_i^T [F_i^{stoc}] \tag{25}$$

Once the global interface unknown $\{\hat{\mathbf{u}}_\Gamma\}$ is obtained by solving the above equation, the interior unknowns for each subdomain Ω_i can be calculated in parallel by solving

$$[\mathcal{A}^i_{II}]\{\hat{\mathbf{u}}^i_I\} = \{\mathcal{F}^i_I\} - [\mathcal{A}^i_{I\Gamma}]\{\hat{\mathbf{u}}^i_\Gamma\} \tag{26}$$

6 A HYBRID PARALLEL IMPLEMENTATION USING MESSAGE PASSING INTERFACE AND EXPLICIT MULTITHREADING

A two-level parallel implementation of the theoretical formulation is addressed in this section. While often used for shared memory parallelism on symmetric multiprocessor (SMP) machines, OpenMP (see Ref. 9 for details) can be used in conjunction with *Message Passing Interface* or MPI (see Ref. 8 for details) to provide a second level parallelism for improved performance on a clusters having SMP compute nodes. A parallel code which mixes both MPI and OpenMP referred to as *hybrid parallel code*. Both MPI and openMP are portable across a wide variety of computing platform. Loosely speaking, MPI codes dynamically decomposes a problem or data among MPI processes which run their tasks on different processors usually having their own (distributed) memory. On the otherhand, openMP codes have directives that describes how parallel sections or loop iterations are to be split among threads. A carefully designed hybrid code may be compiled as serial program or may be compiled with any combintation of MPI and openMP enables which conveniently permits the performance assessment of the multi-level approach.

In a parallel program, communication cost should be subdominant to computation cost. Decomposing the problem into many subdomain enhances the communication cost. The explicit multithreading using openMP many help in limiting the number of subdomains while maintaining decent computation time when a hybrid parallelization approach is adopted using both MPI and openMP.

A problem involving mid-frequency dynamics of a coupled rod system (having 3000 degree-of-freedom) is considered here as described in detail in Ref. 1. For the sake of brevity, we omit the numerical details of the model. The algorithm is currently being implemented on a cluster of SUN SMP machines. The nodes are equiped with at least 24 processors up 144 processors. The numerical experiments focuses on the measure of performance for the SUN compilers using SUN Performace Library(SUN optimized version of LAPACK). to perform all time-consuming matrix-matrix and matrix-vector computations.

The parallelization has been achieved so far at 2 levels. Firstly, a domain decomposition based on *stochastic Schur complement method* as delineated previously. The second level parallelization is achieved at the loop level for all the linear systems involved in the discretized problem. Explicit parallelization has been implemented with OpenMP for all matrix-based operations. Calculation were performed for different numbers of CPUs and contrasted with the results for the serial version of the code. A good performance enhancement is observed in the code as evidenced by the numerics in Table 1. A comprehensive study is in progres

order to study scalability of the proposed algorithm in terms of medium and large-scale problem.

7 CONCLUSIONS

A numerical solution technique is developed for linear stochastic PDEs which capitalizes on (1) a domain decomposition method in the physical space and (2) a functional decomposition technique in the probabilistic space. A direct parallel can be drawn for this methodology for tackling stochastic system with traditional domain decomposition method developed to solve large-scale deterministic PDEs. A multi-level parallel implementation of this methodology has been addressed using Message Passing (MPI) and explicit multi-threading using openMP in a cluster of SMPs.

8 ACKNOWLEDGMENTS

The first author acknowledges the support of a Discovery Grant from National Sciences and Engineering Research Council of Canada and the Canada Research Chair Program.The use of HPCVL computing resources are also greatly appreciated. The third author gratefully acknowledges the support of the Office of Naval Research, the Air Force Office of Scientific Research and Sandia National Laboratories.

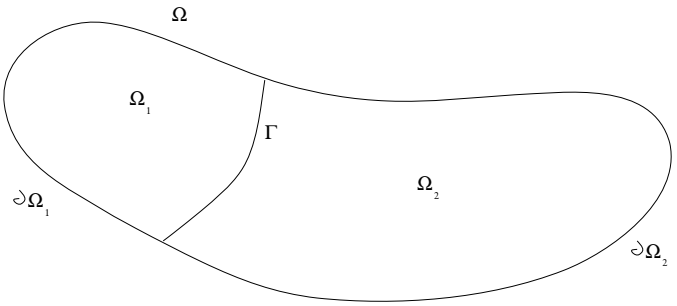

Figure 1. A typical domain partitioned into two nonoverlapping subdomains

Number of Processors	1	2	4	6	8
User CPU Time	71333.637	79138.488	95212.164	100471.154	126788.254
Execution Time	75305.064	41279.574	24192.353	17786.575	16729.985

Table 1. Program performance versus processor counts

9 REFERENCES

1. Sarkar, A. and Ghanem, R., Substructure Approach for the Mid-frequency Vibration of Stochastic Systems, Journal of the Acoustical Society of America, 2003, Vol. 113, Issue 4, pp. 1922-193.

2. Toselli, A and Widlund, O., Domain Decomposition Methods - Algorithms and Theory, Springer Series in Computational Mathematics, Vol. 34, 2004.

3. Smith, B., Bjorstad, P., and Gropp, W., Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press 1996.

4. Quarteroni, A., and Valli, A., Domain Decomposition Methods for Partial Differential Equations, Oxford Science Publications 1999.

5. Ghanem, R. and Spanos, P. 1991. Stochastic Finite Elements: A Spectral Approach. Springer Verlag.

6. Langley, R. S. and Bremner, P., 1998. A hybrid method for the vibration analysis of complex structural-acoustic systems. Journal of the Acoustical Society of America.105(3): 1657-1671.

7. Ohayon, R. and Soize, C. 1998. Structural Acoustics and Vibration. Academic: San Diego.

8. http://www.mpi-forum.org/

9. http://www.openmp.org/