INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING

Int. J. Numer. Meth. Engng 2008; **00**:1–6 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

A method to solve spectral stochastic finite element problems for large-scale systems

Debraj Ghosh*, Philip Avery and Charbel Farhat

Department of Mechanical Engineering and Institute for Computational and Mathematical Engineering, Stanford University, Mail Code 3035, Stanford, CA 94305, U.S.A.

SUMMARY

In the spectral stochastic finite element method of analyzing an uncertain system, first the underlying uncertainty is represented using a set of random variables and then the quantities of interest such as response are represented as functions of these random variables. A Galerkin projection then yields a block system of deterministic equations with coupling among various blocks. Solving this coupled system is the main computational burden of the whole method. Computation becomes more challenging as the size of the physical system and the level of uncertainty grow, and a solution to this challenge is addressed in this paper. Usually an iterative solver such as conjugate gradient is used to solve this large system of equations. To accelerate the convergence of the iterative solver, an

*Correspondence to: D. Ghosh, Department of Mechanical Engineering, Building 500, 488 Escondido Mall, Mail Code 3035, Stanford University, Stanford, CA 94305, U.S.A.

Email: debrajg@stanford.edu

Contract/grant sponsor: Financial support from the Advanced Simulation and Computing Program of the Department of Energy is gratefully acknowledged.; contract/grant number:

Received

Revised

incomplete block-diagonal preconditioner is used here. The preconditioner is applied using a domain decomposition based solver called finite element tearing and interconnecting - dual primal (FETI-DP). Application of the preconditioner requires solving one particular system of equations for multiple right hand sides. To accelerate the computation associated of this repeated solving, a particular variant of the FETI-DP solver is used which is adapted for solving a given system for multiple right hand sides efficiently by re-using a Krylov subspace. It is observed through a numerical study that the proposed method is scalable in a parallel computing environment — a property inherited from the FETI-DP. Furthermore, re-usage of the Krylov subspace greatly reduces the overall computational cost. Therefore the proposed method has the promise to make the uncertainty quantification of realistic systems tractable.

KEY WORDS: polynomial chaos; stochastic finite element; domain decomposition; uncertainty; large system; preconditioner; CG; FETI

1. INTRODUCTION

A realistic analysis and design of physical systems must take into account uncertainties contributed by various sources such as manufacturing variability, insufficient data, unknown physics and aging. In a probabilistic framework these uncertainties are first modeled as random quantities with assigned probability distributions. The probabilistic nature of the response of the uncertain system under deterministic or random loading is then estimated using various available methods such as spectral stochastic finite element methods (SSFEM). In SSFEM the random parameters and external random forces are first modeled using square-integrable random variables and processes. The processes are further discretized using a denumerable set of random variables. The set of all of these random variables is called the set of basic

random variables. The response is next represented using a set of polynomials in these basic random variables. When the basic random variables are Gaussian, the natural choice of a set of orthogonal polynomials in these variables becomes the set of Hermite polynomials, and the resulting representation is called the polynomial chaos expansion (PCE) [20]. Once the coefficients of this representation — referred to as chaos coefficients — are estimated, any statistical quantity such as mean, standard deviation and probability density function (PDF) of the response can be derived in a straightforward way. When the physical domain of the problem is also discretized using, for example, deterministic finite element bases, the approximation space for the entire problem naturally becomes a tensor product space defined on the cartesian product of the physical and random domains.

Consider a linear elliptic equation with uncertain parameters where the physical domain is discretized using an n degrees of freedom (dof) finite element model, and the response is represented using a P-term PCE. For this problem, in order to estimate the chaos coefficients when a Galerkin projection is used to minimize the residual of the governing equation [20, 13, 14], the stochastic problem is translated into the following system of linear deterministic equations

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \quad \mathbf{K} \in \mathbb{R}^{nP \times nP}, \quad \mathbf{u}, \mathbf{f} \in \mathbb{R}^{nP} \ , \tag{1}$$

where the matrix $\mathbf{K}$ has the form

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \dots & \mathbf{K}_{1P} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \dots & \mathbf{K}_{2P} \\ \dots & \dots & \dots & \dots \\ \mathbf{K}_{P1} & \mathbf{K}_{P2} & \dots & \mathbf{K}_{PP} \end{bmatrix} \ , \quad \mathbf{K_{ij}} \in \mathbb{R}^{\mathbf{n} \times \mathbf{n}} \tag{2}$$

A detailed derivation of this equation is presented in the following section. The matrix $\mathbf{K}$ is block-sparse, where each of the blocks $\mathbf{K}_{ij}$ is of size $(n \times n)$ and is also sparse itself. The block-sparsity results from the properties of the polynomial chaos bases, while the sparsity within the individual blocks results from the deterministic finite element discretization. Clearly the problem size — which can be characterized by the number nP — depends on (i) the size of the physical system and spatial mesh resolution — that affects n, and (ii) the level of uncertainty — that affects P. The majority of the current literature addresses solvability of Eq. (1) for small or medium problems. However, as the problem size grows, either through P or n, solving this system becomes more challenging, as the memory and computational time requirement tend to a prohibitive range. Development of efficient computational techniques to solve this problem has therefore emerged as an active area of research in recent years [21, 16, 1, 12].

The system (1) can be efficiently solved using an iterative technique such as the block Gauss-Jacobi [12], MINRES, or conjugate gradient (CG) algorithm [21, 16, 1] — the solver used in this paper. A closer inspection of the block-sparse structure of the matrix $\mathbf{K}$ suggests that a block-diagonal preconditioner could be used to accelerate the convergence of CG, thus resulting a preconditioned conjugate gradient (PCG) algorithm. It has been observed that such a preconditioner indeed helps improving the convergence [21, 16, 1]. In this paper an incomplete block-diagonal preconditioner is used. When the underlying uncertainty in the system is Gaussian in nature, this preconditioner coincides with the block-Jacobi preconditioner used in [21, 16], and differs by a set of scaling factors to the diagonal blocks that is absent in [1]. It is argued later in this paper that these scaling factors are important in improving the preconditioner.

The application of this preconditioner requires solving an $(n \times n)$ linear system of equations

repeatedly, for different right hand sides. The coefficient matrix in this linear system corresponds to the mean part of the stiffness matrix, therefore sparse, symmetric, and positive definite. The main goal of this paper is to accelerate the application of the preconditioner. In [16] the preconditioner was applied using an approximate inversion technique, in [21] it was applied using matrix factorization, and in [1] a CG algorithm was used for this purpose. In all of these papers, the need of a better method for applying the preconditioner is pronounced.

In this paper a domain decomposition based solver called finite element tearing and interconnecting (FETI) is used to apply the preconditioner. FETI can be viewed a class of methods rather than a single method because there exist several versions of it [2, 5, 4, 7]. In a FETI formalism, the physical domain is divided into a number of subdomains, with one or a few subdomains assigned to each processor in a parallel computing environment. The main computational burden is then reduced to solving a number of subdomain-level problems and a problem involving a set of Lagrange variables defined on the subdomain interfaces. The latter problem is solved using an iterative method, where the equilibrium at each subdomain is enforced at each iteration and the continuity across subdomain boundaries is satisfied at convergence. It has been implemented successfully in various applications of deterministic analysis and is found to have good scalability when properly implemented [5]. In this paper, a particular version of FETI solver named as FETI-Dual Primal or FETI-DP [7] is used.

Since the application of the preconditioner requires solving the same linear system repeatedly with different right hand sides, a multiple right hand sides version of FETI-DP [8, 6] is used here that accelerates the convergence of the PCG method. In this version, the whole or a part of the Krylov subspace generated during the first FETI-DP solve is stored, and this subspace is used for choosing the initial iterate for the subsequent applications of FETI-DP. It is noted

that an overlapping additive Schwarz domain decomposition method with the similar Krylov subspace re-usage technique is used to apply a preconditioner in a recent paper [9] where the uncertainty model is special in the sense that it allows selection of a doubly orthogonal random polynomial basis set such that the matrix $\mathbf{K}$ becomes block diagonal. Our work treats a more general model of uncertainty and the model used in [9] is a special case of this general model.

In the following section the ingredients of the SSFEM are presented and Eq. (1) is derived. In section 3 issues related to solving this large linear system are presented. The incomplete block-diagonal preconditioner that is used in this paper is also defined in this section. In section 4 the domain decomposition method FETI-DP and its adaptation to multiple right hand sides are introduced. The implementation details of the PCG solver and the preconditioner are described in section 5. A numerical study is next conducted and presented in section 6. From the study it is observed that the proposed FETI-DP based preconditioner works as a great accelerator for solving the coupled system. This acceleration is further improved by re-using the Krylov subspace in the multiple right hand sides version of FETI-DP. The paper ends with and a few concluding remarks.

2. SPECTRAL STOCHASTIC FINITE ELEMENT METHODS

Let $(\Omega, \mathcal{F}, P)$ denote the underlying probability space of uncertainty, where Ω denotes the set of elementary events θ, $\mathcal{F}$ denotes a σ-algebra on that event set, and P denotes the probability measure. Let the physical domain $\mathcal{D}$ be a closed interval in the space $\mathbb{R}^d$, where d is 1, 2 or 3 and $\boldsymbol{x}$ be a point in this domain. Consider the second-order elliptic partial differential equation (pde) to hold almost everywhere (a.e.) in Ω :

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{x},\theta)\nabla u(\boldsymbol{x},\theta)) &= f(\boldsymbol{x},\theta) \quad \boldsymbol{x} \in \mathcal{D}, \\ u(\boldsymbol{x},\theta) &= 0 \qquad\quad \boldsymbol{x} \in \partial\mathcal{D}\ , \end{aligned} \tag{3}$$

where $u(\boldsymbol{x},\theta), f(\boldsymbol{x},\theta) : \mathcal{D}\times\Omega \to \mathbb{R}$. The uncertain parameters in this equation are embedded in the coefficient $a(\boldsymbol{x},\theta)$ and in the external forcing function. Here the process $a(\boldsymbol{x},\theta)$ is bounded away from zero, and f must satisfy the square integrability condition

$$\int_\Omega \int_\mathcal{D} f^2(\boldsymbol{x},\theta) d\boldsymbol{x} dP(\theta) < \infty. \tag{4}$$

Since the random processes $a(\boldsymbol{x},\theta)$ and $f(\boldsymbol{x},\theta)$ are infinite dimensional objects, for computational purpose they are further discretized using a suitable basis set in the space of square-integrable random variables $L_2(\Omega)$. For example, when the covariance of a process is known then the Karhunen-Loève (KL) expansion [20] can be used for such discretization. A finite-dimensional representation of these processes yields the random vector $\boldsymbol{\xi} = \{\xi_i(\theta)\}_{i=1}^{i=m}$, which completely characterizes the uncertainty in the underlying system. In general these random variables may not be completely independent of each other, and their joint distribution will be non-Gaussian in general. This set can be transformed to a function of an independent Gaussian vector using various techniques [19, 24]. Therefore, without loss of generality, $\boldsymbol{\xi}$ is assumed to be a vector of independent standard normal random variables. In literature, m is often referred to as the stochastic dimension of the problem. Since u, the response of the system, will also be a function of $\boldsymbol{\xi}$, therefore $u(\boldsymbol{x},\theta)$ can now be denoted as $u(\boldsymbol{x},\boldsymbol{\xi})$.

Let the function $p(\boldsymbol{\xi})$ denote the joint probability density function of the random vector $\boldsymbol{\xi}$. The integration $\int_\Omega \cdot dP(\theta)$ or $\int_{\mathbb{R}^m} \cdot p(\boldsymbol{\xi}) d\boldsymbol{\xi}$ is mentioned here as the expectation operator $\mathbb{E}\{\cdot\}$. Let the physical approximation space be a deterministic finite element space $H_0^1(\mathcal{D})$ with the

shape functions denoted as $\{N_i(\boldsymbol{x})\}_{i=1}^{i=n}$. The response of the system is next represented in a tensor product Hilbert space as $u(\boldsymbol{x},\boldsymbol{\xi}) \in H = H_0^1(\mathcal{D}) \otimes L^2(\Omega)$, with the inner product defined as

$$(u,v)_{H_0^1(\mathcal{D})\otimes L^2(\Omega)} = \int_{\mathbb{R}^m} \left(\int_{\mathcal{D}} \nabla u(\boldsymbol{x},\boldsymbol{\xi}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{\xi}) \; d\boldsymbol{x} \right) p(\boldsymbol{\xi}) d\boldsymbol{\xi} = \mathbb{E}\left\{ \int_{\mathcal{D}} \nabla u(\boldsymbol{x},\boldsymbol{\xi}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{\xi}) \; d\boldsymbol{x} \right\} . \tag{5}$$

A set of bases should be chosen in the random space $L^2(\Omega)$ to represent the stochastic counterpart of the response $u(\boldsymbol{x},\boldsymbol{\xi})$. In this paper the polynomial chaos bases [17, 23, 20] are used. Accordingly, any square integrable random variable, vector or process can be represented using a basis set $\{\psi_i(\boldsymbol{\xi})\}_{i=0}^{i=\infty}$, where the bases are chosen to be Hermite polynomials in the set of orthonormal variables $\boldsymbol{\xi}$. The bases ψ_i have the following properties :

$$\psi_0 \equiv 1 \; , \quad \mathbb{E}\{\psi_i\} = 0 \;\; \text{for} \;\; i > 0 \; , \qquad \text{and} \;\; \mathbb{E}\{\psi_i \psi_j\} = \delta_{i,j} \mathbb{E}\{\psi_i^2\} \; ,$$

where $\delta_{i,j}$ denotes the Kronecker delta function. For computational purpose, only a finite number of bases — P are used. P depends upon m, the stochastic dimension of the problem and the highest degree of polynomials retained — also referred to as the order of expansion. For example, when $\boldsymbol{\xi} = \{\xi_1, \xi_2\}$, that is, the stochastic dimension is two, a second-order polynomial chaos expansion corresponds to $P = 6$, and the corresponding polynomials $\psi_i(\xi_1,\xi_2)$ are [20]

$$\begin{aligned} \psi_0(\xi_1,\xi_2) &= 1 \; , \qquad \psi_1(\xi_1,\xi_2) = \xi_1 \; , \qquad \psi_2(\xi_1,\xi_2) = \xi_2 \; , \\ \psi_3(\xi_1,\xi_2) &= \xi_1^2 - 1 \; , \quad \psi_4(\xi_1,\xi_2) = \xi_1\xi_2 \; , \quad \psi_5(\xi_1,\xi_2) = \xi_2^2 - 1 \; . \end{aligned}$$

Besides the Hermite polynomials, other bases such as other types of polynomials [11] or wavelet bases [18] are also often used. Although in this paper only the Hermite polynomials are used

throughout, the framework is equally valid for such other bases. Using both the deterministic or physical bases $\{N_i(\boldsymbol{x})\}_{i=1}^{i=n}$ and the stochastic bases $\{\psi_j(\boldsymbol{\xi})\}_{j=0}^{j=P-1}$ the approximation $\hat{u}(\boldsymbol{x},\boldsymbol{\xi})$ to the response $u(\boldsymbol{x},\boldsymbol{\xi})$ can be represented as

$$\hat{u}(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{i=1}^{i=n}\sum_{j=0}^{j=P-1} u_{i,j} N_i(\boldsymbol{x})\psi_j(\boldsymbol{\xi}), \quad N_i(\boldsymbol{x}) \in V(\mathcal{D}),\ \psi_j(\boldsymbol{\xi}) \in L^2(\Omega)\ . \tag{6}$$

The chaos coefficients $u_{(j)}(\boldsymbol{x}) := \sum_{i=1}^{i=n} u_{i,j} N_i(\boldsymbol{x})$ completely capture the probabilistic description of the random quantities involved. For example, the mean and variance of the response at a physical location $\boldsymbol{x}$ can be readily computed from the above expansion as

$$\text{Mean} = \bar{u}(\boldsymbol{x}) = \sum_{i=1}^{i=n} u_{i,0} N_i(\boldsymbol{x}), \quad \text{Variance} = Var(\hat{u}(\boldsymbol{x})) = \sum_{i=1}^{i=n}\sum_{j=1}^{j=P-1} \left(u_{i,j} N_i(\boldsymbol{x})\right)^2 \mathbb{E}\{\psi_j^2\}\ . \tag{7}$$

The statistical moments of the stress and strain quantities can also be computed, often using a numerical technique [10].

To obtain the representation (6) the coefficients $u_{i,j}$ are needed to be estimated, which is achieved using the *intrusive* or *Stochastic Galerkin finite element method* [20, 13, 14]. First define a bilinear form $\mathcal{B}(u,v) : H \times H \to \mathbb{R}$ as

$$\mathcal{B}(u,v) = \int_{\mathbb{R}^m} \left(\int_{\mathcal{D}} a(\boldsymbol{x},\boldsymbol{\xi}) \nabla u(\boldsymbol{x},\boldsymbol{\xi}) \cdot \nabla v(\boldsymbol{x},\boldsymbol{\xi}) d\boldsymbol{x} \right) p(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad \forall\ u, v \in H\ . \tag{8}$$

The variational form is then constructed as

$$\mathcal{B}(u,v) = \mathcal{V}(v) \quad \forall\ v \in H \tag{9}$$

where $\mathcal{V}(v) : H \to \mathbb{R}$ is a bounded linear functional defined as

$$\mathcal{V}(v) = \int_{\mathbb{R}^m} \left(\int_{\mathcal{D}} f(\boldsymbol{x},\boldsymbol{\xi}) v(\boldsymbol{x},\boldsymbol{\xi}) d\boldsymbol{x} \right) p(\boldsymbol{\xi}) d\boldsymbol{\xi}\ . \tag{10}$$

The positivity and boundedness of the coefficient $a(\boldsymbol{x},\boldsymbol{\xi})$ almost everywhere implies continuity and coercivity of the bilinear form $\mathcal{B}(u,v)$. Under these conditions the Lax-Milgram lemma [25] guarantees the existence and uniqueness of the solution of the variational formulation (9). Let the process $a(\boldsymbol{x},\boldsymbol{\xi})$ be represented as

$$a(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{i=0}^{L-1} a_{(i)}(\boldsymbol{x})\psi_i(\boldsymbol{\xi}) \ , \tag{11}$$

which can be, for example, KL expansion or PCE. Using Eqs. (8), (10) and (11), using the PCE representation of $u(\boldsymbol{x},\boldsymbol{\xi})$ as in Eq. (6), and choosing $v(\boldsymbol{x},\boldsymbol{\xi})$ to be $N_k(\boldsymbol{x})\psi_l(\boldsymbol{\xi})$, the variational form (9) can be written as

$$\sum_{i=1}^{i=n}\sum_{j=0}^{j=P-1} u_{i,j}\int_{\mathbb{R}^m}[K(\boldsymbol{\xi})]_{k,i}\psi_j(\boldsymbol{\xi})\psi_l(\boldsymbol{\xi})p(\boldsymbol{\xi})d\boldsymbol{\xi} = \int_{\mathbb{R}^m} f_k(\boldsymbol{\xi})\psi_l(\boldsymbol{\xi})p(\boldsymbol{\xi})d\boldsymbol{\xi} \tag{12}$$
$$\forall k = 1\ldots n, \quad l = 0\ldots P-1, \quad \text{and } \psi_l(\boldsymbol{\xi}) \in L^2(\Omega)$$

where $K(\boldsymbol{\xi})$ is an $(n\times n)$ matrix whose elements are random variables, expressed as

$$K(\boldsymbol{\xi}) = \sum_{r=0}^{L-1} K_{(r)}\psi_i(\boldsymbol{\xi}) \ , \qquad K_{(r)} \in \mathbb{R}^{n\times n} \tag{13}$$

where the $(k,i)^{th}$ element of the matrix $K_{(r)}$ is

$$\int_{\mathcal{D}} a_{(r)}(\boldsymbol{x})\nabla N_i(\boldsymbol{x})\cdot\nabla N_k(\boldsymbol{x})d\boldsymbol{x} \ . \tag{14}$$

and

$$f_k(\boldsymbol{\xi}) = \int_{\mathcal{D}} f(\boldsymbol{x},\boldsymbol{\xi})N_k(\boldsymbol{x})d\boldsymbol{x} \ . \tag{15}$$

Eq. (12) can be written as Eq. (1). Although described for an second order elliptic equation, this expression is obtained for general elliptic equations of the form $\mathcal{L}(u(\boldsymbol{x},\theta)) = f(\boldsymbol{x},\theta)$.

In a computer implementation, the matrices $K_{(i)}$ for $i = 0,\ldots,L-1$ are computed by calling the usual finite element stiffness computation routine L times, only changing the coefficient

$a_{(i)}(\boldsymbol{x})$ in each call. Physically, $K_{(0)}$ refers to the mean stiffness and other $K_{(i)}$-s refer to the random fluctuation around the mean. Next let us introduce the notations

$$u_{(i)} = \left\{ \begin{array}{c} u_{1,i} \\ u_{2,i} \\ . \\ u_{n,i} \end{array} \right\} \in \mathbb{R}^n \ , \quad f_{(i)} = \left\{ \begin{array}{c} \mathbb{E}\{f_1(\boldsymbol{\xi})\psi_i\} \\ \mathbb{E}\{f_2(\boldsymbol{\xi})\psi_i\} \\ . \\ \mathbb{E}\{f_n(\boldsymbol{\xi})\psi_i\} \end{array} \right\} \in \mathbb{R}^n \ , \quad i = 0, \ldots, P-1, \tag{16}$$

where $f_k(\boldsymbol{\xi})$ are defined in Eq. (15). As mentioned earlier, $f(\boldsymbol{\xi})$ is also discretized using KL or PCE, and the expectation operations in the above equation can be evaluated using the orthogonality of the random bases. As a special example, when $f(\boldsymbol{\xi})$ is deterministic, say $f(\boldsymbol{\xi}) = f$, then in the above equation only $f_{(0)}$ survives and the other $f_{(i)}$s vanish. Using the notations introduced, the terms $\mathbf{K}$, $\mathbf{u}$ and $\mathbf{f}$ in Eq. (1) can be written as

$$\mathbf{K} = \left[\begin{array}{cccc} \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_0\psi_0\} & \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_1\psi_0\} & \ldots & \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_{P-1}\psi_0\} \\ \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_0\psi_1\} & \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_1\psi_1\} & \ldots & \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_{P-1}\psi_1\} \\ \ldots & \ldots & \ldots & \ldots \\ \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_0\psi_{P-1}\} & \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_1\psi_{P-1}\} & \ldots & \sum_{i=0}^{L-1} K_{(i)}\mathbb{E}\{\psi_i\psi_{P-1}\psi_{P-1}\} \end{array} \right], \tag{17}$$

$$\mathbf{u} = \left\{u_{(0)}, u_{(1)}, \ldots, u_{(P-1)}\right\}^T \ , \quad \mathbf{f} = \left\{f_{(0)}, f_{(1)}, \ldots, f_{(P-1)}\right\}^T \tag{18}$$

The matrix $\mathbf{K}$ is block-sparse which follows from the sparsity of the triple products $\mathbb{E}\{\psi_i\psi_j\psi_k\}$; the diagonal blocks are always non-zero. The sparsity depends upon the representation Eq. (11), the types of bases ψ_i used, and the order of expansion. A general and concise recipe of this sparsity cannot be suggested. For a known type of representation of the uncertain system parameters and response, the sparsity can be found through computation. For example, in a

linear statics problem — used in the numerical study here — when the random fluctuation of the Young's moduli of five structural components are expressed as second degree polynomials of Gaussian random variables and the displacement field is represented in fourth order PCE, the resulting sparsity pattern is shown in Figure 2. Increasing or decreasing the order of PCE will augment or remove some of the branches in this matrix.

3. SOLUTION OF THE LINEAR SYSTEM

The resulting system in Eq. (1) is actually a *system of systems* of equations. As mentioned in Eq. (13) and the discussion following it, the $(n \times n)$ blocks are linear combinations of the finite element stiffness matrices $K_{(i)}$. When n is large — which is the case when a fine mesh resolution is needed in the physical domain, or when P is large — which is the case when the system has many random parameters and the levels of variation of these parameters are high, then solving this system becomes challenging. As any other large-scale problems, an attempt to solve this coupled system naturally brings forth a number of issues related to solution methodology and implementation. Specific implementation details are discussed in section 5. The solution methods and the preconditioning aspects are discussed first.

The matrices $K_{(i)}$ are the finite element stiffness matrices, therefore symmetric. Therefore the matrix $\mathbf{K}$ is also symmetric. To efficiently use the symmetry and sparsity while solving Eq. (1) iteratively, two good candidates are the MINRES algorithm and CG — if the system is positive definite. In this paper the assumed uncertainty model of the parameters ensures that the matrix $K(\boldsymbol{\xi})$ is positive definite almost everywhere, it implies that the matrix $\mathbf{K}$ is also positive definite. In this case CG becomes a strong choice as the solver.

For solving large scale linear systems the preconditioners have been playing an important

role [15]. Using a preconditioning matrix $\mathbf{M}$ the system (1) is transformed as

$$\text{solve} \quad \mathbf{MKu} = \mathbf{Mf} \ , \tag{19}$$

here $\mathbf{M}$ should be chosen such that (i) $\mathbf{MK}$ is well-conditioned, if possible $\mathbf{M} \approx \mathbf{K}^{-1}$, (ii) $\mathbf{M}$ should be easily computable. To this end, the matrix $\mathbf{K}$ needs to be examined closely.

The matrix $K_{(0)}$ corresponds to the stiffness matrix of the mean system, thus it is positive definite, provided the system is properly constrained by sufficient boundary conditions. Using the orthogonality of the chaos polynomials it can be shown that the matrix $K_{(0)}$ contributes only toward the diagonal blocks of the matrix $\mathbf{K}$ and not to the off-diagonal blocks. This can be interpreted as the diagonal blocks have a contribution from the mean system properties, and often also from the fluctuation part of these properties — depending upon the expansion of $K(\boldsymbol{\xi})$ in Eq. (13), whereas the off-diagonal blocks have contributions only from the fluctuation part. For such matrices it is often expected and has been observed that a block-diagonal preconditioner helps to accelerate the convergence. In this paper an *incomplete block-diagonal* preconditioner is used, which is defined as

$$\mathbf{M} = \begin{bmatrix} \frac{1}{\mathbb{E}\{\psi_0^2\}} K_{(0)}^{-1} & 0 & \dots & 0 \\ 0 & \frac{1}{\mathbb{E}\{\psi_1^2\}} K_{(0)}^{-1} & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{\mathbb{E}\{\psi_{P-1}^2\}} K_{(0)}^{-1} \end{bmatrix} . \tag{20}$$

The reason for deviating from the block-Jacobi preconditioner is as follows. In general, the diagonal blocks of the matrix $\mathbf{K}$ are not equal, therefore in the block-Jacobi preconditioner, the diagonal blocks in $\mathbf{M}$ also will be different. However, in our construction of the preconditioner, all the diagonal blocks in $\mathbf{M}$ are kept as same — except the scaling factors, which allows using

a multiple right hand sides version of a linear solver. Since only the mean part $K_{(0)}$ of the diagonal blocks is retained, the preconditioner is called as *incomplete* block-diagonal. It can be shown that for a Gaussian model of $K(\boldsymbol{\xi})$, that is, when in Eq. (13) the only non-zero coefficients $K_{(i)}$-s are $K_{(0)}$ and the ones corresponding to ξ_i-s, the preconditioner $\mathbf{M}$ coincides with the block-Jacobi preconditioner [21, 16]. In [1], a similar preconditioner as above was used, except there was no $\frac{1}{\mathbb{E}\{\psi_i^2\}}$ factor in the diagonal blocks. The reason we chose the factor is that in the i^{th} diagonal block of $\mathbf{K}$, the coefficient of $K_{(0)}$ is $\mathbb{E}\{\psi_i^2\}$. It was also substantiated by numerical experiments, although the results are not mentioned in this paper, that this factor helps improving the convergence.

Each application of the preconditioner $\mathbf{M}$ requires the same matrix $K_{(0)}$ to be inverted P times. Note that this is the count for each PCG iteration. However, in the PCG algorithm the action of the preconditioner is implemented as a system solving approach rather than requiring the access to $K_{(0)}^{-1}$. Thus, a system

$$K_{(0)}z = d \tag{21}$$

needs to be solved instead of computing $K_{(0)}^{-1}$ explicitly, where $z, d \in \mathbb{R}^n$, and d varies. The matrix $K_{(0)}$ is sparse, and large for large-scale problems. Thus an iterative solver can be used to this end. Since this system needs to be solved repeatedly for different right hand sides, a solution method that can reduce the computational cost in successive solving is greatly desired.

Here a domain decomposition based solver named FETI-DP (with multiple right hand sides) is used to solve $K_{(0)}z = d$.

4. DOMAIN DECOMPOSITION

With the progress of parallel computing technology, the domain decomposition techniques have emerged as efficient choice for solving large-scale systems. As the name suggests, in these techniques the computational domain is decomposed into a set of subdomains, as shown in Fig. 3, and a divide-and-conquer strategy is developed that permits using multiple processors more efficiently than the traditional single domain approaches. Here a particular family of domain decomposition methods called FETI is considered. FETI methods are iterative methods where typically one or more subdomains are assigned to one processor and the subdomains communicate with each other through a set of Lagrange multipliers $\lambda \in \mathbb{R}^{N_\lambda}, \quad N_\lambda << n$ defined at subdomain boundaries. The equilibrium of each subdomain is satisfied at each iteration whereas the continuity of the displacement field is achieved at convergence. The computational burden of solving $K_{(0)}z = d$ reduces down to solving a much smaller interface problem as

$$A\lambda = g \ , \quad A \in \mathbb{R}^{N_\lambda \times N_\lambda}, \quad \lambda, g \in \mathbb{R}^{N_\lambda} \ , \tag{22}$$

actual expressions for A and g can be found in [7]. This problem is again solved using PCG algorithm with a suitable preconditioner. In this paper this PCG will be referred to as the inner-PCG to distinguish it from the PCG used to solve Eq. (19), which will be referred to as the outer-PCG.

The FETI methods have gone through, and are still going through a series of modifications aimed at improving the performance and adapting with various applications [2, 5, 4, 7]. Here only the FETI-DP version [7] is discussed. This method was developed to achieve scalability for fourth-order plate and shell problems. Detailed description of the method can be found in

[7].

... YYY A FEW COMMENTS on role of preconditioning on scaling issues. ... Also the following formula on the condition number of the interface problem is to be presented in a proper context

$$\kappa = \mathcal{O}\left(1 + \log\left(\frac{H}{h}\right)\right)^m \quad m \leq 3 \ , \tag{23}$$

where h characterizes the mesh size and H characterizes the subdomain size. YYY NEEDS more discussion on convergence ? such as what happens when the number of elements are increased keeping the number of subdomains fixed etc.

As can be seen from Eq. (20), application of the incomplete block-diagonal preconditioner requires solving the same system of equations with different right hand sides. When a FETI-DP solver is used to implement this preconditioner, effectively the problem reduces to solving repeatedly Eq. (22) for different right hand sides, which can be written as

$$A\lambda_j = g_j \ , \quad A \in \mathbb{R}^{N_\lambda \times N_\lambda}, \quad \lambda_j, g_j \in \mathbb{R}^{N_\lambda} \ , \quad j = 1, \ldots, n_{rhs} \ , \tag{24}$$

where n_{rhs} denotes the number of right and sides for which the system is to be solved. Computational cost of this repetitive solving can be greatly reduced by re-using the Krylov subspaces used by the PCG algorithm used to solve Eq. (24). For FETI-DP, a strategy of re-using the Krylov subspace, which will be referred to here as FETI-DP-mrhs, was demonstrated in [8, 6] and further used in various applications such as acoustic scattering [22]. Outline of the idea is as follows. Consider for $j = 1$ the system $A\lambda_1 = g_1$ is solved, the generated Krylov subspace $\mathcal{S}_1$ is stored. For $j = 2$ onward the initial iterate λ_j^0 will be chosen from subspace $\mathcal{S}_1$ and the successive iterates will be selected from the space A-orthogonal to $\mathcal{S}_1$.

In FETI-DP the storage requirement for the Krylov subspace and the overall computational cost corresponds to λ, a vector defined only on the interface dof and not on the entire domain, which implies $N_\lambda << n$. Clearly this is a significant saving compared to many other solvers where the required resource corresponds to n.

At this point, the essential ingredients of implementing the incomplete block-diagonal preconditioner (20) to solve the linear system of systems (1) using PCG algorithm are ready. For implementing the preconditioner, the physical domain $\mathcal{D}$ is decomposed into a number of subdomains, and the FETI-DP-mrhs is used. The implementation details are presented in the following section.

5. IMPLEMENTATION

Key points of the computer implementation of the outer-PCG solver and the FETI-DP-mrhs preconditioner are presented in this section. Issues related to storage, matrix-vector multiplication, load balancing, and a few fine details about implementing the FETI-DP are addressed here. To facilitate the discussion different levels of computation are defined and outlined in Figure 1.

A distributed memory system is used in computation; only the memory is used to store the matrices and vectors. Let the whole physical domain be decomposed into N_s subdomains. A few subdomains are assigned to each processor. That means each processor is responsible for constructing and storing the matrices and vectors and performing computations corresponding to a fixed set of subdomains. Each of the non-zero matrices $K_{(i)} \in \mathbb{R}^{n\times n},\ i = 0, \ldots, P-1$ are constructed accordingly. One important factor should be considered in this assignment of subdomains among the processors is load balancing. Any vector $\boldsymbol{v} \in \mathbb{R}^{nP}$ is first divided into

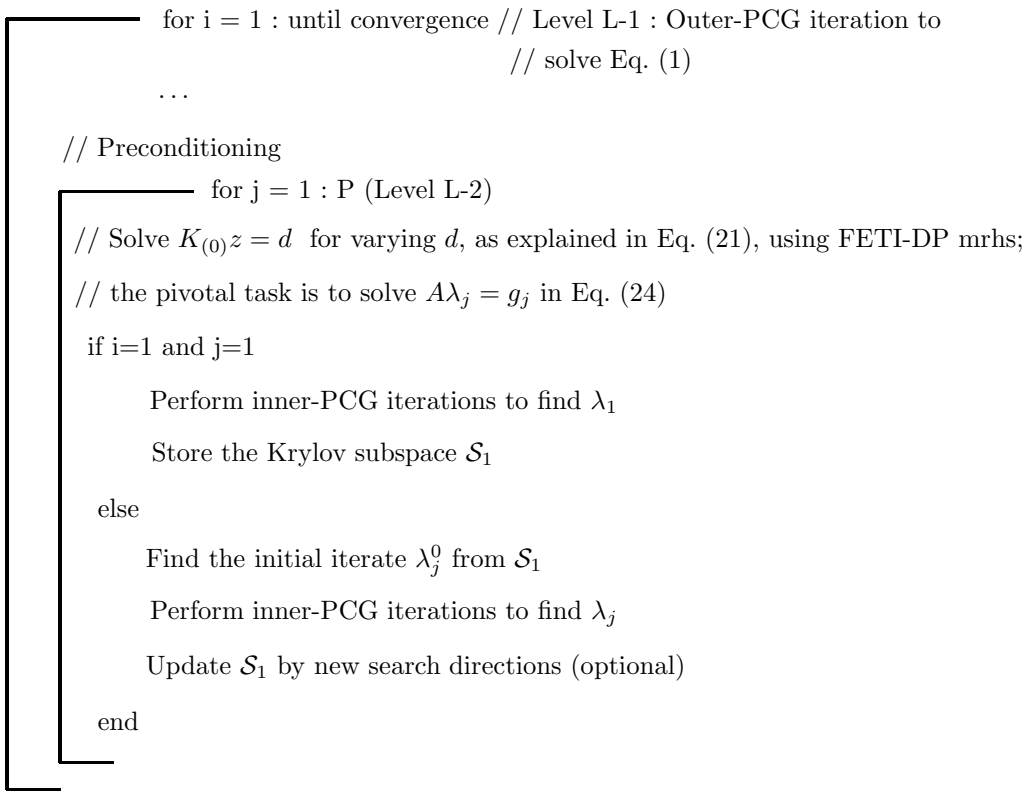

Figure 1. Outline of computation

P number of n-dimensional blocks as

$$\mathbf{v} = \left\{ \begin{array}{c} v_{(0)} \\ v_{(1)} \\ . \\ v_{(P-1)} \end{array} \right\} , \qquad v_{(i)} \in \mathbb{R}^n \ . \tag{25}$$

Computation and storage of each of the vectors $v_{(i)}$ are distributed among the processors in accordance with the distribution of the subdomains among the processors.

5.1. Computational implementation details at various levels

5.1.1. Level L-1 : The outer-PCG iterations The initial iterates for the outer-PCG iterations are taken as zero vectors, and the stopping criteria is set as the relative residual — defined as the ratio of the 2-norms of the residual and the right hand side — to be less than 10^{-8}. Each iteration requires computation of a matrix-vector (mat-vec) product $\mathbf{Kv}$ for a given vector $\mathbf{v} \in \mathbb{R}^{\mathbf{nP}}$. To minimize computational burden this mat-vec product is implemented at the block level [16]. To this effect, first the vector $\boldsymbol{v}$ is divided into P number of n-dimensional blocks as described in Eq. (25). From Eqs. (17) and (25)

$$\mathbf{Kv} = \left\{ \begin{array}{c} \sum_{i=0}^{L-1}\sum_{j=0}^{P-1} K_{(i)}\mathbb{E}\{\psi_i\psi_j\psi_0\}v_{(j)} \\ \sum_{i=0}^{L-1}\sum_{j=0}^{P-1} K_{(i)}\mathbb{E}\{\psi_i\psi_j\psi_1\}v_{(j)} \\ \vdots \\ \sum_{i=0}^{L-1}\sum_{j=0}^{P-1} K_{(i)}\mathbb{E}\{\psi_i\psi_j\psi_{P-1}\}v_{(j)} \end{array} \right\} , \qquad v_{(j)} \in \mathbb{R}^n \ . \tag{26}$$

The block level mat-vec products $K_{(i)}v_{(j)}$ for $i = 0, \ldots, L-1,\ K_{(i)} \neq 0,\ \ j = 0, \ldots, P-1$ are computed first and the resulting products are stored. Finally, the mat-vec product $\mathbf{Kv}$ is computed using linear combinations of these $K_{(i)}v_{(j)}$, as expressed in Eq. (26). Here the triple products of the random variables $\mathbb{E}\{\psi_i\psi_j\psi_k\}$, $0 \leq i, j, k \leq P-1$ are computed only once and are stored as a three-dimensional dense array.

5.1.2. Level L-2 : Preconditioning and FETI-DP-mrhs The FETI-DP preconditioner routine is called only when a non-zero right hand side (rhs) is encountered. The Dirichlet preconditioner [7] is used for the FETI solver. The coarse problem within FETI-DP [7] is augmented as described in [7, 3]. Sparse direct solvers are used to solve the subdomain level problems and

the coarse problem.

Unless otherwise mentioned, at most 1000 search directions in the Krylov subspace $\mathcal{S}_1$ are stored at any stage the FETI-DP-mrhs.

6. NUMERICAL STUDY

To conduct the numerical study a linear statics problem is considered. A cylinder head of a car engine is considered as the structure which has uncertain material properties. For a deterministic loading, the goal is to estimate the statistics of the response using SSFEM. Three finite element models of the cylinder head with three different mesh sizes are considered. These three models will be referred to here as CH1, CH2 and CH3, a typical model is shown in Figure 4. The model CH1 has 54 198 dof, CH2 has 335 508 dof and CH3 has 2 290 437 dof. All the models are built using three types of elements : 3-D 8-node brick elements with 3dof/node, 3-D 3-AQR shell elements with 6dof/node, and 3-D 6-node pentahedral elements with 3dof/node. The cylinder head is made with five different components. The Young's modulus E_i of the materials of these five components are assumed to be independent random variables, expressed as

$$E_i = \bar{E}_i + \frac{\sigma_{E_i}}{\sqrt{2}}(\xi_i^2 - 1) \qquad i = 1, \ldots, 5 \ . \tag{27}$$

Here $\bar{E}_i$ are the mean values, σ_{E_i} are the standard deviations, and ξ_i are independent standard normal random variables. To ensure positivity of the Young's modulus a constraint is imposed as $\frac{\sigma_{E_i}}{\sqrt{2}} < \bar{E}_i$ for all i. In the present study σ_{E_i} is assumed to be 20% of $\bar{E}_i$ for all i, which satisfies this constraint. An arbitrarily chosen static force is applied to the structure. The rigid body modes are removed from the structure by properly constraining at the boundary.

The displacement field is represented using fourth-order polynomial chaos expansion, the total number of chaos polynomials in this expansion becomes $P = 126$. The chaos coefficients are estimated by solving Eq. (1) by the PCG method, and using an incomplete block-diagonal preconditioner as defined in Eq. (20).

In the first step of the numerical study Eq. (1) is solved for the model CH1 using 8 processors, and using four different domain decompositions: dividing the entire domain into 22, 44, 90 and 223 subdomains, respectively. The computational time required only at the preconditioning stage to solve Eq. (1) is presented in Table I. It is observed from this table that in terms of computational time, there exists an optimal number of subdomains, which is 44 in this case (also 90 is a closer candidate), among the four decompositions considered. In Figure 5 the number of PCG iterations within the FETI-DP solver — also referred to here as the inner-PCG iterations — is plotted against the sequence of the FETI-DP solver is called. It is observed here that compared to the first few solves, the number of iterations dropped significantly afterward, that is, the mrhs version of FETI-DP helped significantly in reducing the computational cost of preconditioning. It is further observed in this figure that the trend of reduction of the number of inner-PCG iterations is almost similar for all three domain decompositions. Similar study to find the optimal domain decomposition is also carried out for models CH2 and CH3 and the results are presented in Table II and III. The optimal number of subdomains for CH2 is 229 and for CH3 it is 944 (also 320 is a closer candidate).

In the next step of the numerical study, the models CH1, CH2 and CH3 are decomposed into 44, 229 and 944 subdomains, respectively. These numbers are chosen considering the optimality study as presented in the previous paragraph. For three models and their corresponding domain decompositions Eq. (1) is solved using various number of processors, and the computational

time and iteration counts are presented in Tables IV and V. The main observations from these tables are as follows.

The number of outer-PCG iterations, denoted here by $N_{iter,cg}$, does not change considerably with the mesh refinement of the models. However, the number of inner-PCG iterations, $N_{iter,FETI}$, increases noticeably as the mesh resolution increases.

Before moving to further observations an apparent contradiction is clarified here. In this example there are $P = 126$ diagonal blocks in the preconditioner $\mathbf{M}$ defined in Eq. (20). Therefore it is expected that for each outer-PCG iteration — that is, for each implementation of the preconditioner $\mathbf{M}$ — the FETI-DP solver to solve $K_{(0)}z = d$ will be called $P = 126$ times. However, in these tables, when the outer-PCG iteration count $N_{iter,cg}$ changes from 16 to 17 (third column), then the number of FETI-DP calls $N_{call,FETI}$ changes from 301 to 322 (fourth column), that is, increases by 21, and not by 126. To resolve this contradiction, a closer look at the final solution $\mathbf{u}$ revealed that among 126 number of $u_{(i)}$-s in Eq. (18), only 21 turned out to be non-zero. Therefore, the FETI-DP solver is called only 21 times in the later stage of outer-PCG iterations.

To study the scalability properties, variation of the preconditioning time T_{prec} with respect to mesh resolution and number of processors should be observed. From these two tables it is observed that for a fixed number of processors, solving a 6 times bigger problem — measured in terms of dof — takes between $5 - 7$ times CPU time.

In both of the Tables IV and V it is found that irrespective of the model, number of processors or number of subdomains, preconditioning time T_{prec} dominates the total solve time T_{sol}. This observation justifies the emphasis of this paper on improving the preconditioning stage as it significantly improves the overall computational cost.

Finally, the sensitivity of the total number of inner-PCG iterations with respect to the dimension of the stored Krylov subspace in FETI-DP-mrhs is studied. For the model CH2, three different storages, 1000, 2000, and 4000 vectors are considered, and the corresponding inner-PCG iteration counts are computed and presented in Table VI. This study helps in deciding the computational budget considering the trade-off between the memory and CPU time requirement.

7. CONCLUDING REMARKS

The numerical study demonstrated the success of the proposed methodology of analyzing uncertainty in large-scale problems. Using the incomplete block-diagonal type preconditioner, the preconditioned conjugate gradient PCG method could solve large problems within a few iterations. The domain decomposition based solver called finite element tearing and interconnecting - dual primal (FETI-DP) worked successfully in implementing this preconditioner. The multiple right hand sides (mrhs) version of FETI-DP helped significantly in reducing the total computational time. YYY SCALABILITY

The domain decomposition-based approach is a major step toward uncertainty quantification of real engineering systems through efficient usage of parallel computation. Future research directions on this area will include applications to various other kind of engineering problems.

REFERENCES

1. Keese A and Matthies HG. Hierarchical parallelisation for the solution of stochastic finite element equations. *Computers and Structures*, 83:1033–1047, 2005.

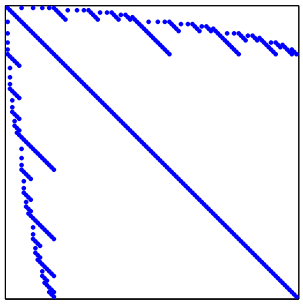

Figure 2. Block-sparsity of the matrix $\mathbf{K}$ of the example problem. As the sparsity is independent of the mesh resolution, all the models CH1, CH2 and CH3 have same sparsity. Each dot indicates an $(n \times n)$ sparse matrix.

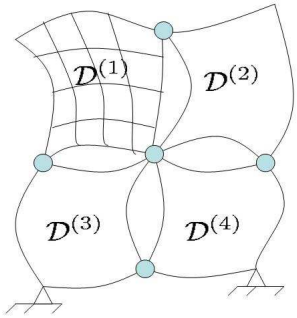

Figure 3. Decomposition of the physical domain $\mathcal{D}$

2. Farhat C and Roux FX. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205–1227, 1991.
3. Farhat C, Li J, and Avery P. A FETI-DP method for parallel iterative solution of indefinite and complex-valued solid and shell vibration problems. *International Journal for Numerical Methods in Engineering*, 63:398–427, 2005.
4. Farhat C and Mandel J. A two-level FETI method for static and dynamic plate problems Part I : An optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering*, 155:129–151, 1998.

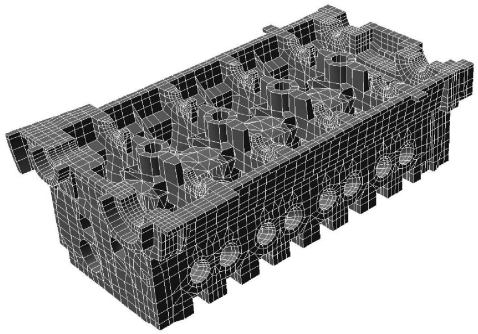

Figure 4. A finite element model of the cylinder head

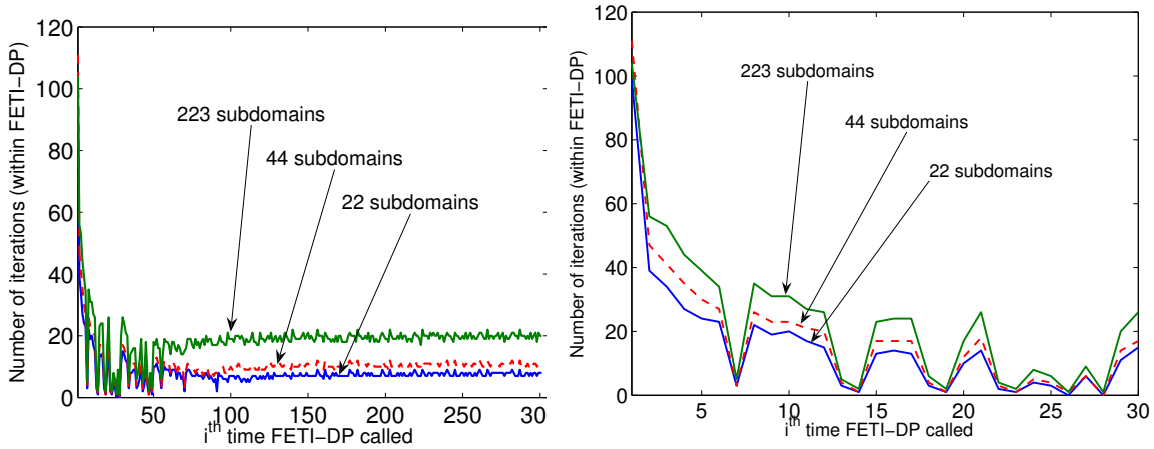

Figure 5. Efficiency of FETI-DP with multiple right hand sides : Here the number of inner-PCG iterations — PCG iterations within the FETI-DP solver, noted as $N_{iter,FETI}$ in Tables IV and V — is plotted as successive times this solver is called in the preconditioning stage of the large system arising from the stochastic Galerkin scheme. Model CH1, the output displacement field is represented in fourth-order chaos expansion. Re-usage of the Krylov subspace reduced the computational cost significantly after a first few FETI-DP solves. The figure on the right is a magnified part of the figure on the left. In this figure it is observed that the cost reduction patterns are similar for all three domain decompositions.

Number of subdomains	Time taken in preconditioning (sec.)
22	275
44	225
90	230
223	273

Table I. CPU time required in the preconditioning stage while solving for Model CH1, the output displacement field is represented in fourth-order chaos expansion, total 8 processors are used.

Number of subdomains	Time taken in preconditioning (sec.)
64	1182
122	1142
229	1085
274	1156

Table II. CPU time required in the preconditioning stage while solving for Model CH2, the output displacement field is represented in fourth-order chaos expansion, total 12 processors are used.

5. Farhat C, Mandel J, and Roux FX. Optimal convergence properties of the FETI domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 115:365–385, 1994.
6. Farhat C, Crivelli L, and Roux FX. Extending substructure based iterative solvers to multiple load and repeated analyses. *Computer Methods in Applied Mechanics and Engineering*, 117:195–209, 1994.
7. Farhat C, Lesoinne M, and Pierson K. A scalable dual-primal domain decomposition method. *Numerical Linear Algebra with Applications*, 7:687–714, 2000.
8. Farhat C and Chen PS. Tailoring domain decomposition methods for efficient parallel coarse grid solution

Number of subdomains	Time taken in preconditioning (sec.)
117	6634
320	4069
944	3940
2159	4405

Table III. CPU time required in the preconditioning stage while solving for Model CH3, the output displacement field is represented in fourth-order chaos expansion, total 60 processors are used.

Model	N_{proc}	$N_{iter,cg}$	$N_{call,FETI}$	$N_{iter,FETI}$	T_{sol} (sec.)	T_{prec} (sec.)
CH1	12	16	301	2573	252	181
	8			2573	277	207
	4			2574	322	256
CH2	12	17	322	4520	1076	881
	8			4530	1399	1144

Table IV. Computational performance details for models CH1 and CH2, with different number of processors. The output displacement field is represented in fourth-order chaos expansion. N_{proc} : number of processors, $N_{iter,cg}$: total number of outer-PCG iterations, $N_{call,FETI}$: number of times the FETI-DP-mrhs solver is called, $N_{iter,FETI}$: total number of inner-PCG iterations (within FETI-DP-mrhs), T_{sol} : total solve time, T_{prec} : time taken at the preconditioning stage, here T_{sol} includes T_{prec}. The 8 processors are a subset of the 12 processors, the 4 processors are further a subset of these 8 processors. The model CH1 is divided into 44 subdomains and CH2 into 229 subdomains.

Model	N_{proc}	$N_{iter,cg}$	$N_{call,FETI}$	$N_{iter,FETI}$	T_{sol} (sec.)	T_{prec} (sec.)
CH2	60	17	322	4531	748	631
CH3	60	16	301	6432	4165	3565

Table V. Computational performance details for models CH2 and CH3. The output displacement field is represented in fourth-order chaos expansion. N_{proc} : number of processors, $N_{iter,cg}$: total number of outer-PCG iterations, $N_{call,FETI}$: number of times the FETI-DP-mrhs solver is called, $N_{iter,FETI}$: total number of inner-PCG iterations (within FETI-DP-mrhs), T_{sol} : total solve time, T_{prec} : time taken at the preconditioning stage, here T_{sol} includes T_{prec}. The model CH2 is divided into 229 subdomains and CH3 into 944 subdomains.

$\max_i(s_i)$	$N_{iter,FETI}$
1000	4466
2000	3184
4000	2838

Table VI. Effect of the maximum dimension of the stored Krylov subspace stored on the FETI-DP-mrhs acceleration. $\max_i(s_i)$: maximum number of orthogonal bases stored from any Krylov subspace $\mathcal{S}_i$, and $N_{iter,FETI}$: total number of inner-PCG iterations (within FETI-DP-mrhs). Model CH2.

and for systems with many right hand sides. *Contemporary Mathematics*, 180:401–406, 1994.

9. Jin C, Cai X-C, and Li C. Parallel domain decomposition methods for stochastic elliptic equations. *SIAM Journal on Scientific Computing*, 29(5):2096–2114, 2007.
10. Ghosh D and Farhat C. Strain and stress computation in stochastic finite element methods. *International Journal for Numerical Methods in Engineering*, 74(8):1219–1239, 2008.
11. Xiu D and Karniadakis G. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, 2003.

12. Chung DB, Gutiérrez MA, Graham-Brady LL, and Lingen F-J. Efficient numerical strategies for spectral stochastic finite element models. *International Journal for Numerical Methods in Engineering*, 64:1334–1349, 2005.
13. Babŭska I, Tempone R, and Zouraris GE. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004.
14. Babŭska I, Tempone R, and Zouraris GE. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Computer Methods in Applied Mechanics and Engineering*, 194:1251–1294, 2005.
15. Benzi M. Preconditioning techniques for large linear systems : a survey. *Journal of Computational Physics*, 182:418–477, 2002.
16. Pellissetti M and Ghanem R. Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Advances in Engineering Software*, 31:607–616, 2000.
17. Wiener N. Homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.
18. Le Maître OP, Najm H, Ghanem R, and Knio O. Multi-resolution analysis of wiener-type uncertainty propagation schemes. *Journal of Computational Physics*, 197(2):502–531, 2004.
19. Ghanem R. and Doostan A. On the construction and analysis of stochastic predictive models: Characterization and propagation of the errors associated with limited data. *Journal of Computational Physics*, 217(1):63–81, 2006.
20. Ghanem R and Spanos PD. *Stochastic Finite Elements: A Spectral Approach*. Revised Edition, Dover Publications, 2003.
21. Ghanem R and Kruger RM. Numerical solution of spectral stochastic finite element systems. *Computer Methods in Applied Mechanics and Engineering*, 129:289–303, 1996.
22. Tezaur R, Macedo A, and Farhat C. Iterative solution of large-scale acoustic scattering problems with multiple right hand-sides by a domain decomposition method with lagrange multipliers. *International Journal for Numerical Methods in Engineering*, 51(10):1175–1193, 2001.
23. Cameron RH and Martin WT. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, 48(2):385–392, 1947.
24. Das S., Ghanem R., and Spall J. Asymptotic sampling distribution for polynomial chaos representation of data: A maximum entropy and fisher information approach. San Diego, CA, 2006.
25. Brenner SC and Scott LR. *The Mathematical Theory of Finite Element Methods*. Second Edition, Springer, 2002.

