A FETI-Preconditioned Congugate Gradient Method for Large-Scale Stochastic Finite Element Problems

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 for analyzing an uncertain system, the uncertainty is represented by a set of random variables and a quantity of interest such as the system response is considered a function of these random variables. Consequently, the underlying Galerkin projection yields a block system of deterministic equations where the blocks are sparse but coupled. The solution of this algebraic system of equations becomes rapidly challenging when the size of the physical system and/or the level of uncertainty is increased. This paper addresses this challenge by presenting a preconditioned conjugate gradient method for such block systems where the preconditioning step is based on the Dual-Primal (DP) Finite Element Tearing and Interconnecting (FETI) method equipped

---

*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

Received

Revised

with a Krylov subspace reusage technique for accelerating the iterative solution of systems with multiple and repeated right hand-sides. Preliminary performance results on a Linux Cluster suggest that the proposed solution method is numerically scalable and demonstrate its potential for making the uncertainty quantification of realistic systems tractable. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: domain decomposition, FETI, polynomial chaos, stochastic finite element, uncertainty

1. INTRODUCTION

The realistic design and analysis of a physical system must take into account uncertainties contributed by various sources such as manufacturing variability, insufficient data, unknown physics and aging. In many probabilistic frameworks, these uncertainties are first modeled as random quantities with assigned probability distributions. Then, the probabilistic nature of the system response to a deterministic or random loading is estimated using a stochastic finite element method (SSFEM). In such a method, the random parameters and external forces are first modeled using square-integrable random variables and processes. The processes are further discretized using a denumerable set of random variables known as the set of *basic* random variables. Next, the system response is represented using a set of polynomials in these basic random variables. When these are Gaussian, the natural choice of a set of orthogonal polynomials in these variables becomes the set of Hermite polynomials. In this case, the resulting representation is called the polynomial chaos expansion (PCE) [35] method. Once the coefficients of this representation — referred to as chaos coefficients — are estimated, any statistical quantity such as the mean, standard deviation and probability density function

(PDF) of the system response can be derived in a straightforward manner. 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 by a finite element model with n degrees of freedom (dof), and the response is represented by a P-term PCE. To estimate the chaos coefficients, a Galerkin projection can be applied to minimize the residual of the governing equation [35, 27, 28]. In Section 2, it is shown that such a procedure transforms the stochastic problem 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

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

The matrix $\mathbf{K}$ is block-sparse. Each of its blocks $\mathbf{K}_{ij}$ is of size $(n \times n)$ and is also sparse. 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. The overall problem size, which is characterized by the product nP, depends on (i) the size of the physical system and the spatial mesh resolution which affect n, and (ii) the level of uncertainty which affects P. The majority of the current literature addresses the solution of Eq. (1) for small- or medium-sized problems. However, as P (and) or n is increased, solving Eq. (1)

becomes challenging from both memory and CPU viewpoints. Therefore, the development of efficient computational techniques for solving problem (1) has emerged as an active research area in recent years [36, 31, 1, 25].

Previous attempts at solving Eq. (1) have relied on iterative techniques such as block Gauss-Jacobi [25], MINRES, or the preconditioned conjugate gradient (PCG) algorithm [36, 31, 1]. In this paper, the latter method is adopted and an incomplete block-diagonal preconditioner is proposed. When the uncertainty in the system is Gaussian in nature, this preconditioner coincides with the block-Jacobi preconditioner used in [36, 31], and differs from that proposed in [1] by a set of scaling factors of the diagonal blocks that significantly enhance the performance of the preconditioner.

The application of a block-diagonal preconditioner to Eq. (1) requires solving repeated linear systems of equations with different right hand-sides. The left hand-side of each of these systems is the mean part of the stiffness matrix and therefore is sparse, symmetric, and positive definite. In [31], such a preconditioner was based on an approximate inversion technique. In [36], it was based on a matrix factorization. In [1], a CG algorithm was used for this purpose. In all of these references, the reported performance results suggest a need for a better preconditioning step.

In this paper, the scalable domain decomposition (DD) based Finite Element Tearing and Interconnecting Dual-Primal (FETI-DP) method [12, 13, 6, 14, 5, 26], which is an enhanced variant of the ubiquitous iterative FETI solver [4, 15, 9, 8, 10, 17, 7, 18, 16, 22, 23, 30, 37, 20, 6], is proposed as an incomplete block-diagonal preconditioning solver for Eq. (1). It is equipped with the Krylov reusage technique first proposed in [15, 9] for accelerating its convergence for systems with multiple and repeated right hand-sides. These typically arise in nested iteration

loops for linear problems. It is noted that an overlapping additive Schwarz DD method with a similar idea of Krylov subspace reusage was recently exploited in [19] for preconditioning stochastic problems where the uncertainty model is a special case of the general model considered in this paper and is such that $\mathbf{K}$ is block-diagonal. To this effect, the remainder of this paper is organized as follows.

In Section 2, the SSFEM of interest is presented and Eq. (1) is derived to keep this paper as self-contained as possible. In Section 3, important issues related to the solution of this potentially large linear system of equations are exposed, and the incomplete block-diagonal preconditioner proposed in this paper is introduced. Section 4 overviews the FETI-DP method which is chosen for computing the preconditioned residuals arising from the chosen PCG based solution strategy. This section also describes the tailoring of FETI-DP to the solution of systems with multiple and/or repeated right hand-sides such as those arising from the target application. Implementational details of the proposed preconditioning step and resulting overall PCG solver are described in Section 5. A three-dimensional, large-scale application problem is discussed in Section 6. The performance results obtained for this problem suggest that the overall PCG solver proposed in this paper is numerically scalable and therefore holds a great potential for enabling the uncertainty quantification of realistic systems. Finally, Section 7 concludes this paper.

2. SPECTRAL STOCHASTIC FINITE ELEMENT METHOD

Let $(\Omega, \mathcal{F}, P)$ denote the underlying probability space of uncertainty, Ω denote the set of elementary events θ, $\mathcal{F}$ denote a σ-algebra on this event set, and μ denote the probability measure. Let also 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 defined in Ω:

$$-\nabla \cdot (a(\boldsymbol{x},\theta)\nabla u(\boldsymbol{x},\theta)) = f(\boldsymbol{x},\theta) \quad \boldsymbol{x} \in \mathcal{D}, \tag{3}$$

$$u(\boldsymbol{x},\theta) = 0 \qquad \boldsymbol{x} \in \partial\mathcal{D},$$

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 and external functions $a(\boldsymbol{x},\theta)$ and $f(\boldsymbol{x},\theta)$, respectively. Here, the process $a(\boldsymbol{x},\theta)$ is assumed to be bounded away from zero and $f(\boldsymbol{x},\theta)$ is assumed to 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 function set in the space of square-integrable random variables $L_2(\Omega)$. For example, when the covariance of a process is know, the Karhunen-Loève (KL) expansion [35] can be used for such a 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 from each other, and their joint distribution may be non-Gaussian. However, these random variables can be transformed into a function of an independent Gaussian vector using various techniques [34]. Therefore, without loss of generality, $\boldsymbol{\xi}$ can be assumed to be a vector of independent standard normal random variables. In the literature, m is often referred to as the stochastic dimension of the problem. Since the response of the system, u, is actually also a function of $\boldsymbol{\xi}$, it 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 $\boldsymbol{\xi}$. In this paper, the integral $\int_{\Omega} \cdot d\mu(\theta)$ or $\int_{\mathbb{R}^m} \cdot p(\boldsymbol{\xi}) d\boldsymbol{\xi}$ is denoted by 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 by $\{N_i(\boldsymbol{x})\}_{i=1}^{i=n}$. The response of the system is next represented in a tensor product Hilbert space as $u(\boldsymbol{x}, \boldsymbol{\xi}) \in H = H_0^1(\mathcal{D}) \otimes L^2(\Omega)$ where the inner product is defined as

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

A set of basis functions 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 basis functions [32, 38, 35] are used for this purpose. Accordingly, any square integrable random variable, vector, or process can be represented using a basis function set $\{\psi_i(\boldsymbol{\xi})\}_{i=0}^{i=\infty}$, where the basis functions are chosen to be Hermite polynomials in the set of orthonormal variables $\boldsymbol{\xi}$. The ψ_i functions have the following properties:

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

where $\delta_{i,j}$ denotes the Kronecker delta function. For computational purpose, only a finite number P of basis functions is used. P depends on the stochastic dimension of the problem, m, and on the highest retained polynomial degree which is also referred to as the order of the expansion. For example, when $\boldsymbol{\xi} = \{\xi_1, \xi_2\}$, the stochastic dimension is $m = 2$, a second-order polynomial chaos expansion yields $P = 6$, and the corresponding polynomials $\psi_i(\xi_1, \xi_2)$ are

$$\psi_0(\xi_1, \xi_2) = 1 , \quad \psi_1(\xi_1, \xi_2) = \xi_1 , \quad \psi_2(\xi_1, \xi_2) = \xi_2 ,$$

$$\psi_3(\xi_1, \xi_2) = \xi_1^2 - 1 , \quad \psi_4(\xi_1, \xi_2) = \xi_1 \xi_2 , \quad \psi_5(\xi_1, \xi_2) = \xi_2^2 - 1 .$$

Besides the Hermite polynomials, other basis functions such as other types of polynomials [24] or wavelet functions [33] are often employed. Using both the deterministic or physical shape functions $\{N_i(\boldsymbol{x})\}_{i=1}^{i=n}$ and the stochastic basis functions $\{\psi_j(\boldsymbol{\xi})\}_{j=0}^{j=P-1}$, the approximation $\hat{u}(\boldsymbol{x}, \boldsymbol{\xi})$ of 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 $u_{(j)}(\boldsymbol{x}) = \sum\limits_{i=1}^{i=n} u_{i,j} N_i(\boldsymbol{x})$ 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

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

The statistical moments of the stress and strain quantities can be computed using one of the numerical techniques discussed in [21].

To evaluate the representation (6), the coefficients $u_{i,j}$ must first be estimated. This can be achieved using the *intrusive* or *stochastic Galerkin finite element method* [35, 27, 28] as follows. First, define a bilinear form $\mathcal{B}(u, v) : H \times H \to \mathbb{R}$ as

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

Then, construct the variational form as

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

where $\mathcal{V}(v) : H \rightarrow \mathbb{R}$ 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 imply the continuity and coercivity of the bilinear form $\mathcal{B}(u, v)$. Under these conditions, the Lax-Milgram lemma [39] guarantees the existence and uniqueness of the solution of the variational problem (9).

Let the process $a(\boldsymbol{x}, \boldsymbol{\xi})$ be represented as

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

Eq. (11) can correspond to, for example, a KL expansion or a PCE. Using Eqs. (8,10,11), the PCE of $u(\boldsymbol{x}, \boldsymbol{\xi})$ (6), and choosing $v(\boldsymbol{x}, \boldsymbol{\xi})$ as $N_k(\boldsymbol{x}) \psi_l(\boldsymbol{\xi})$, the variational formulation (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 and can be 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 $(i, k)^{th}$ element of the matrix $K_{(r)}$ is given by

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

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 re-written in the form of Eq. (1). Although described for a second-order elliptic equation, this equation can be obtained for any elliptic equation of the form $\mathcal{L}(u(\boldsymbol{x}, \theta)) = f(\boldsymbol{x}, \theta)$.

In a computer implementation, the matrices $K_{(i)}$, $i = 0, \ldots, L-1$, are computed by calling the usual finite element stiffness routines L times and changing only the coefficient $a_{(i)}(\boldsymbol{x})$ in each call. Physically, $K_{(0)}$ refers to the mean stiffness and all other $K_{(r)}$ matrices refer to the random fluctuations around the mean stiffness.

Next, the following notation is introduced

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

where the functions $f_k(\boldsymbol{\xi})$ are defined in Eq. (15). As mentioned earlier, $f(\boldsymbol{\xi})$ is also discretized using KL or PCE. 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 $f(\boldsymbol{\xi}) = f$ — only $f_{(0)}$ survives in the above equation and all other $f_{(i)}$ terms vanish.

Using the above notation, $\mathbf{K}$, $\mathbf{u}$ and $\mathbf{f}$ in Eq. (1) can now be written as

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

$$(17)$$

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

From the sparsity and other properties of the triple products $\mathbb{E}\{\psi_i \psi_j \psi_k\}$, it follows that $\mathbf{K}$ is block-sparse and its diagonal blocks are always non-zero. More specifically, the sparsity pattern of $\mathbf{K}$ depends on the representation described in Eq. (11), the choice of basis functions ψ_i, and the order of the expansion. For example, for the linear static problem studied in Section 6 where the random fluctuations of the five Young's modulis of five structural components are expressed as second-degree polynomials of Gaussian random variables and the displacement field is represented by a fourth-order PCE, the sparsity pattern of $\mathbf{K}$ is graphically depicted in Figure 1. Increasing or decreasing the order of the PCE adds or removes branches in $\mathbf{K}$.
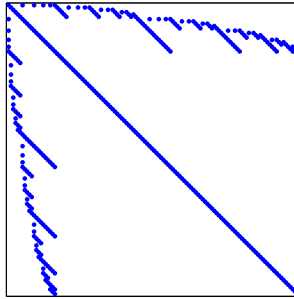

Figure 1. Block-sparsity of the stiffness matrix $\mathbf{K}$ of the example cylinder head problem.

3. NESTED PCG SOLVERS FOR A SYSTEM OF SYSTEMS

Eq. (1) is actually a *system of systems* of equations. As shown by Eq. (17), the $(P \times P)$ blocks of $\mathbf{K}$ of size $(n \times n)$ are linear combinations of the finite element stiffness matrices $K_{(i)}$. When n or P is large, solving Eq. (1) becomes a real challenge.

The matrices $K_{(i)}$ are typical finite element stiffness matrices. Therefore, for the elliptic problems considered in this paper, these matrices are symmetric. Consequently, $\mathbf{K}$ is also symmetric. Here, it is assumed that the chosen uncertainty model is such that $K(\boldsymbol{\xi})$ (13) is positive definite, which implies that $\mathbf{K}$ is also positive definite. In this case, the PCG algorithm is the preferred iterative solver for problem (1) and the issue becomes the construction of a suitable preconditioner. To this end, $\mathbf{K}$ is closely examined below.

The matrix $K_{(0)}$ corresponds to the stiffness matrix of the mean system. Hence, it is positive definite provided the modeled system is properly constrained by sufficient boundary conditions. Using the orthogonality of the chaos polynomials, it can be shown that $K_{(0)}$ contributes only to the diagonal blocks of $\mathbf{K}$. In other words, the off-diagonal block matrices have contributions only from the fluctuations of the system properties. This suggests that the diagonal blocks are dominant and calls for a block-diagonal preconditioner for $\mathbf{K}$.

Furthermore for large-scale systems, factoring any diagonal block of $\mathbf{K}$ can be so computationally prohibitive that a second iterative solver becomes also necessary for constructing the desired block-diagonal preconditioner. This is the context of this work which targets large-scale, large-dimension stochastic problems, and where the second iterative solver is also chosen to be a PCG algorithm. In this case, the overall iterative solution method has two nested PCG iterations: an outer one associated with the main solution of Eq. (1), and an inner one associated with the solution of the auxiliary problem incurred by the block-diagonal preconditioner.

Nested PCG iterations can be computationally inefficient, unless special care is taken for accelerating the convergence of one of the two iterative loops. Here, the focus is set on the inner-PCG iteration. To accelerate its convergence, the following approach is adopted. The

blocks of the block-diagonal preconditioner are chosen to be independent of the index of the outer-PCG iteration, and the performance of the inner-PCG algorithm is optimized for the solution of multiple systems with a constant left hand-side but different right hand-sides. More specifically, the following *incomplete* block-diagonal preconditioner is proposed

$$\mathbf{M} = \begin{bmatrix} \frac{1}{\mathbb{E}\{\psi_0^2\}}K_{(0)}^{-1} & 0 & \cdots & 0 \\ 0 & \frac{1}{\mathbb{E}\{\psi_1^2\}}K_{(0)}^{-1} & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \frac{1}{\mathbb{E}\{\psi_{P-1}^2\}}K_{(0)}^{-1} \end{bmatrix}. \tag{19}$$

The reader can observe that the diagonal blocks of $\mathbf{M}$ differ only by scaling factors, and only the mean part $K_{(0)}$ contributes to them. Because of the latter observation, the $\mathbf{M}$ matrix described in (19) is referred to here as an *incomplete* block-diagonal preconditioner. It can be shown that for a Gaussian model for $K(\boldsymbol{\xi})$, only the term $K_{(0)}\mathbb{E}\{\psi_j^2\}$ survives under the summation operator in the jth diagonal block of $\mathbf{K}$. Hence in this case, the proposed preconditioner $\mathbf{M}$ coincides with the block-Jacobi preconditioner proposed in [36, 31]. It differs from that proposed in [1] by the presence of the scaling factors $\frac{1}{\mathbb{E}\{\psi_j^2\}}$, $j = 1, \cdots, P$ in the diagonal blocks which is due to the presence of the term $\mathbb{E}\{\psi_j^2\}$ as the coefficient of $K_{(0)}$ in the jth diagonal block of $\mathbf{K}$, as described in Eq. (17) (the reader is reminded that $\psi_0 = 1$).

Each i-th application of the preconditioner $\mathbf{M}$ described in (19) requires the solution of P problems of the form

$$K_{(0)}z_j(i) = e_j(i), \qquad j = 1, \cdots, P \tag{20}$$

where for large-scale problems $K_{(0)}$ is sparse and large, and the notation (i) is used to emphasize the dependence of the right hand-side e_j on the outer-PCG iteration i. Hence, Eq. (20) describes a linear system of equations with multiple and repeated right hand-sides:

the multiple right hand-side aspect derives from the stochastic dimension of the problem $P > 1$, and the repeated right hand-side aspect from the outer-PCG iteration associated with the chosen overall solution strategy. Here, it is proposed to solve this linear system by the FETI-DP method equipped with a Krylov subspace re-usage technique for accelerating its convergence in the presence of multiple and/or repeated right hand-sides.

4. FETI-DP AND ITS TAILORING FOR REPEATED RIGHT HAND-SIDES

4.1. FETI-DP: a domain decomposition based iterative solver

During the last two decades, DD methods have emerged as a popular and often efficient category of algorithms for the solution of large-scale systems of equations. These methods rely on partitioning the computational domain into a set of subdomains (Fig. 2), and on applying a divide-and-conquer strategy for solving the associated system of equations. They are usually more amenable to parallel processing than traditional solution methods, which makes them attractive particularly for computations on parallel computers.
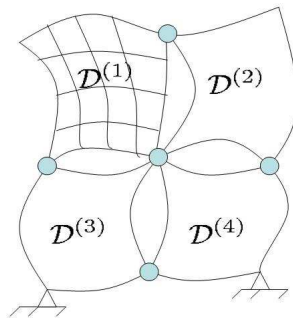
Figure 2. Decomposition of a physical domain $\mathcal{D}$ into subdomains.

FETI-DP [12, 13, 6, 14, 5, 26] is a third-generation FETI-type [4, 15, 9, 8, 10, 17, 7, 18, 16, 22, 23, 30, 37, 20, 6] DD method — that is, a DD method with Lagrange multipliers and in which the interface problem is solved by a suitable preconditioned conjugate gradient (PCG) algorithm — developed for the scalable and fast iterative solution of systems of equations arising from the FE discretization of static, dynamic, second-order, and fourth-order elliptic partial differential equations. When equipped with the Dirichlet preconditioner [8] and applied to plane stress/strain or shell problems, the condition number κ of its interface problem grows asymptotically as [29]

$$\kappa = O\left(1 + \log^2 \frac{H}{h}\right), \tag{21}$$

where H and h denote the subdomain and mesh sizes, respectively. When equipped with the same Dirichlet preconditioner and an auxiliary coarse problem constructed by enforcing some set of optional constraints at the subdomain interfaces [13], the condition number estimate (21) also holds for second-order scalar elliptic problems that model three-dimensional solid mechanics problems [2]. This estimate proves the scalability of the FETI-DP method with respect to all of the problem size, subdomain size, and number of subdomains. More specifically, it suggests that one can expect FETI-DP to solve small-scale and large-scale problems in similar iteration counts, a property often referred to as *numerical* scalability. This in turn suggests that when the FETI-DP method is well-implemented on a parallel processor, it should be capable of solving an n-times larger problem using an n-times larger number of processors in almost a constant CPU time, a property often referred to as *parallel* scalability. This property shared by all FETI methods was demonstrated in practice for many complex structural mechanics problems (for example, see [12, 13, 30] and the references cited therein).

4.2. Acceleration of convergence for problems with multiple right hand-sides

Let

$$\mathbf{K}\mathbf{u}_j = \mathbf{b}_j \qquad j = 1, \ ..., \ N_{rhs} \tag{22}$$

denote a series of N_{rhs} successive problems with different right hand-sides. Solving these problems by the FETI-DP method requires first transforming them into the interface problems (for example, see [12, 13, 6, 14, 5, 26])

$$\mathbf{F}_I \lambda_j = \mathbf{g}_j \qquad j = 1, \ ..., \ N_{rhs} \ , \tag{23}$$

where $\mathbf{F}_I$ is a symmetric positive definite matrix of size n_I equal to the total number of degrees of freedom on the global interface of the mesh decomposition (designated here by I), and λ is the vector of Lagrange multipliers introduced at the subdomain interfaces to enforce there the continuity of the solution, then computing the solutions of the above interface problems by a PCG algorithm. After the vector of Lagrange multiplier unknowns λ_j is computed, the primal solution $\mathbf{u}_j$ is obtained by straightforward and parallel local (subdomain) substitutions.

Solving problems (23) is equivalent to solving the following minimization problems

$$\min_{\lambda \in \mathbb{R}_I^n} \Phi_j(\lambda) = \frac{1}{2} \ \lambda^T \mathbf{F}_I \lambda - \mathbf{g}_j^T \lambda \qquad j = 1, \ ..., \ N_{rhs} \ . \tag{24}$$

Let

$$V_j^{r_j} = \{\mathbf{p}_j^1, \ \mathbf{p}_j^2, \ ..., \ \mathbf{p}_j^k, \ ..., \ \mathbf{p}_j^{r_j}\} \qquad j = 1, \ ..., \ N_{rhs} \tag{25}$$

denote the Krylov space consisting of the search directions $\mathbf{p}_j^k$ generated during the solution of the j-th of problems (24) by r_j FETI-DP iterations, and let $\mathbf{V}_j \in \mathbb{R}^{n_I \times r_j}$ denote the rectangular matrix associated with $V_j^{r_j}$. From the orthogonality properties of the conjugate gradient method, it follows that

$$\mathbf{V}_j^T \mathbf{F}_I \mathbf{V}_j = \mathbf{D}_j \qquad\qquad j = 1, \ ..., \ N_{rhs} \tag{26}$$

where $\mathbf{D}_j$ is the diagonal matrix

$$\mathbf{D}_j = \begin{bmatrix} d_{j_1} & d_{j_2} & \cdots & d_{j_{r_j}} \end{bmatrix} . \tag{27}$$

Suppose that during the solution by FETI-DP of the first problem $\mathbf{F}_I \lambda_1 = \mathbf{g}_1$, the matrices $\mathbf{V}_1$ and $\mathbf{F}_I \mathbf{V}_1$ are stored. Consider next the second problem $\mathbf{F}_I \lambda_2 = \mathbf{g}_2$, which can be also written as

$$\min_{\lambda \in \mathbb{R}^{n_I}} \Phi_2(\lambda) = \frac{1}{2} \lambda^{\mathbf{T}} \mathbf{F}_{\mathbf{I}} \lambda - \mathbf{g_2^T} \lambda . \tag{28}$$

If $\mathbb{R}^{n_I}$ is decomposed as follows

$$\mathbb{R}^{n_I} = V_1^{r_1} \oplus V_1^{r_1^*}, \quad dim\,(V_1^{r_1^*}) = n_I - r_1, \quad V_1^{r_1} \text{ and } V_1^{r_1^*} \text{ are } \mathbf{F}_I - \text{orthogonal} \tag{29}$$

then λ_2 can be searched for in the following form

$$\lambda_2 = \lambda_2^0 + \mu_2, \quad \lambda_2^0 \in V_1^{r_1}, \ \mu_{\mathbf{2}} \in V_1^{r_1^*}, \ \text{ and } \ \lambda_2^{0^T} \mathbf{F}_I \mu_2 = \mu_2^T \mathbf{F}_I \lambda_2^0 = 0 . \tag{30}$$

Substituting the above expression of λ_2 into (28), and exploiting the orthogonality conditions expressed in (30) reveals that λ_2^0 is the solution of the uncoupled minimization problem

$$\min_{\lambda \in V_1^{r_1}} \Phi_2(\lambda) = \frac{1}{2} \lambda^T \mathbf{F}_I \lambda - \mathbf{g}_2^T \lambda , \tag{31}$$

and μ_2 is the solution of the uncoupled minimization problem

$$\min_{\mu \in V_1^{r_1^*}} \Phi_2(\mu) = \frac{1}{2} \mu^T \mathbf{F}_I \mu - \mathbf{g}_2^T \mu . \tag{32}$$

Since $\lambda_2^0 \in V_1^{r_1}$, there exists a $\theta_2^0 \in \mathbb{R}^{r_1}$ such that

$$\lambda_2^0 = \mathbf{V}_1 \theta_2^0 . \tag{33}$$

The vector θ_2^0 is determined by substituting the above equation into Eq. (31) and solving the resulting minimization problem, which gives

$$(\mathbf{V}_1^T \mathbf{F}_I \mathbf{V}_1)\theta_2^0 = \mathbf{V}_1^T \mathbf{g}_2 . \tag{34}$$

From Eq. (26), it follows that θ_2^0 is given by

$$\mathbf{D}_1\theta_2^0 = \mathbf{V}_1^T\mathbf{g}_2 \quad \Rightarrow \quad \theta_{2j}^0 = \frac{[\mathbf{V}_1^T\mathbf{g}_2]_j}{d_{1_j}} \qquad j = 1, \ ..., \ r_1, \tag{35}$$

and from Eqs. (33,34) it is concluded that

$$\lambda_2^0 = \mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{V}_1^T\mathbf{g}_2 \ . \tag{36}$$

Here, it is noted that the evaluation of θ_2^0 requires only r_1 floating point operations, and that of λ_2^0 a single matrix-vector product.

Next, attention is turned to the solution of problem (32) by the FETI-DP method. Since the decomposition (30) requires μ_2 to be $\mathbf{F}_I$-orthogonal to λ_2^0, it follows that at each PCG iteration k, the search directions $\mathbf{p}_2^k$ must be explicitly $\mathbf{F}_I$-orthogonalized to $\mathbf{V}_1$. This entails modifying the PCG-loop of the FETI-DP solver to compute the following "enriched" search directions $\tilde{\mathbf{p}}_2^k$

$$\tilde{\mathbf{p}}_2^k = \mathbf{p}_2^k + \sum_{q=1}^{q=r_1} \eta_q\mathbf{p}_1^q, \quad \eta_q = -\frac{\mathbf{p}_1^{q^T}\mathbf{F}_I\mathbf{p}_2^k}{\mathbf{p}_1^{q^T}\mathbf{F}_I\mathbf{p}_1^q} = -\frac{\mathbf{p}_2^{k^T}\mathbf{F}_I\mathbf{p}_1^q}{\mathbf{p}_1^{q^T}\mathbf{F}_I\mathbf{p}_1^q} \tag{37}$$

instead of the usual search directions $\mathbf{p}_2^k$. The right hand-sides of Eq. (34) and Eq. (37) explain why it was assumed that $\mathbf{V}_1$ and $\mathbf{F}_I\mathbf{V}_1$ are stored during the solution by FETI-DP of the first problem $\mathbf{F}_I\lambda_1 = \mathbf{g}_1$.

Since $V_1^{r_1^*}$ is only a subspace of $\mathbb{R}^{n_I}$, it follows that the FETI-DP method equipped with the starting value given by Eq. (36) and the orthogonalization procedure implied by Eq. (37) can be expected to converge faster for the second problem than for the first one. This was extensively demonstrated in [15] where this approach was first proposed, and in [9, 30] for various applications. The extension of this methodology to $N_{rhs} > 2$ right hand-sides is straightforward and can be found in [15, 9]. Essentially, the matrices of search directions $\mathbf{V}_j$ and products $\mathbf{F}_I\mathbf{V}_j$ generated and performed during the solution of the previous problems are accumulated in two

matrices $\mathbf{V}$ and $\mathbf{W} = \mathbf{F}_I \mathbf{V}$, repectively, and the scope of the orthogonalization procedure (37) is extended to all accumulated and stored search directions. For many applications, convergence was observed to be continuously accelerated from one right hand-side to the next one. This is because for each additional right hand-side, the number of components of the new solution that are captured by the starting value of the form of λ_2^0 increases, and the dimension of the supplementary space of the form of $V_1^{r_1^*}$ in which the FETI-DP method is forced to iterate decreases.

The computational methodology outlined above is applicable to any Krylov based iterative solver. However, it is feasible only for DD based iterative solvers such as FETI-DP. Indeed, iterative DD methods iterate only on subdomain interface unknowns whereas global iterative methods iterate on all unknowns of a given problem. Hence, only in the case of a DD method the storage of $\mathbf{V}$ and $\mathbf{W} = \mathbf{F}_I \mathbf{V}$ is feasible, and only in this case the computational overhead incurred by the inner products associated with the orthogonalization procedure (37) is affordable.

5. DISTRIBUTED IMPLEMENTATION

The methodology described in the previous sections for solving Eq. (1) can be implemented in a totally distributed fashion. Therefore, it is particularly suitable for parallel computing on distributed memory systems including Linux clusters. First, a given FE mesh is partitioned into subdomains using a mesh decomposer such as that described in [3, 11] or any other similar software tool. Multiple subdomains are assigned to a given processor depending on the number of generated subdomains and number of available processors. Each non-zero stiffness matrix $K_{(i)} \in \mathbb{R}^{n \times n}$, $i = 0, \ldots, P - 1$ and associated displacement and force vectors are formed

and assembled (if needed) on a subdomain-by-subdomain basis only. Each vector $\mathbf{v} \in \mathbb{R}^{\mathbf{nP}}$ is

divided into P number of n-dimensional blocks as follows

$$
\mathbf{v} = \left\{ \begin{array}{c} v_{(0)} \\ v_{(1)} \\ . \\ v_{(P-1)} \end{array} \right\} , \qquad v_{(i)} \in \mathbb{R}^n , \tag{38}
$$

and the blocks $v_{(i)}$ are distributed among the processors according to the distribution of the

mesh subdomains.

5.1. Implementation of the outer-PCG algorithm

The solution of problem (1) is initialized with $\mathbf{u}^0 = 0$. Convergence is monitored at the outer-

level and in this work, is declared when

$$
\|\mathbf{Ku} - \mathbf{f}\|_2 \leq 10^{-8} \times \|\mathbf{f}\|_2 . \tag{39}
$$

At each outer-PCG iteration, a matrix-vector product of the form $\mathbf{Kv}$ is performed at the

block-level using the block partitioning outlined in Eq. (38), as follows

$$
\mathbf{Kv} = \left\{ \begin{array}{c} \sum_{i=0}^{L-1} \sum_{j=0}^{P-1} K_{(i)} \mathbb{E}\{\psi_j \psi_j \psi_0\} v_{(j)} \\ \sum_{i=0}^{L-1} \sum_{j=0}^{P-1} K_{(i)} \mathbb{E}\{\psi_j \psi_j \psi_1\} v_{(j)} \\ . \\ \sum_{i=0}^{L-1} \sum_{j=0}^{P-1} K_{(i)} \mathbb{E}\{\psi_j \psi_j \psi_{P-1}\} v_{(j)} \end{array} \right\} , \qquad v_{(j)} \in \mathbb{R}^n . \tag{40}
$$

Each block-level matrix-vector product $K_{(i)} v_{(j)}$ $\big(i = 0, \ \dots, \ L-1, \ K_{(i)} \neq 0, \ j = 0, \ \dots, \ P-1$

) is performed in parallel on a subdomain-by-subomain basis. The triple products $\mathbb{E}\{\psi_j \psi_j \psi_k\}$,

$0 \leq i, j, k \leq P - 1$, involving the random variables are computed only once and stored in a

dense three-dimensional array that is replicated in each processor.

5.2. Implementation of the inner-PCG algorithm

The FETI-DP method for computing the block-diagonal preconditioned residuals is invoked only when a non-zero right hand-side $e_j(i)$ $\big($see Eq. (20)$\big)$ is encountered. This DD method, which incorporates its own PCG solver, is equipped in this work with the Dirichlet preconditioner [8] and the auxiliary coarse problem described in [12]. Within this DD method, all local subdomain problems as well as the coarse problem are solved by a direct sparse algorithm.

Unless otherwise mentioned, a maximum of $1,000$ search directions are accumulated and stored throughout the calls to the FETI-DP solver in order to accelerate its convergence for subsequent right hand-sides as described in Section 4.2. For any given problem, the convergence of FETI-DP is declared when

$$\|\mathbf{F}_I \lambda_j - \mathbf{g}_j\|_2 \leq 10^{-6} \times \|\mathbf{g}_j\|_2 \ . \tag{41}$$

6. APPLICATION

The non deterministic static analysis on a Linux Cluster of the cylinder head of a car engine (see Figure 3) with uncertainties in the material properties is considered here. Given a deterministic static load, the objective is to estimate the statistics of the structural response using the SSFEM outlined above. For this purpose, three finite element models of the cylinder head with three different mesh sizes are considered: model CH1 with $54,198$ dof, model CH2 with $335,508$ dof, and model CH3 with $2,290,437$ dof. All three models are constructed using three different types of elements: eight-noded brick elements with three dof per node, three-noded shell elements with six dof per node, and six-noded pentahedral elements with three dof per

node. The cylinder head is assumed to be made of five different pieces whose Young moduli E_i are assumed to be independent random variables that can be expressed as

$$E_i = \bar{E}_i + \frac{\sigma_{E_i}}{\sqrt{2}}(\xi_i^2 - 1) \qquad i = 1, \ldots, 5 \ . \tag{42}$$

In the above equation, $\bar{E}_i$ denotes the mean value of E_i, σ_{E_i} its standard deviation, and the ξ_is are independent standard normal random variables. To ensure the positivity of E_i, it is assumed that $\forall i, \frac{\sigma_{E_i}}{\sqrt{2}} < \bar{E}_i$. Here, σ_{E_i} is assumed to be equal to 20% of $\bar{E}_i$ for all i, and therefore the aforementioned positivity constraint is satisfied. Furthermore, the structure is sufficiently constrained to remove all possible rigid body modes. The displacement field is represented by a fourth-order polynomial chaos expansion. The total number of chaos polynomials in this expansion is $P = 126$. These are estimated by solving Eq. (1) by the PCG method equipped with the incomplete block-diagonal preconditioner proposed in this paper.
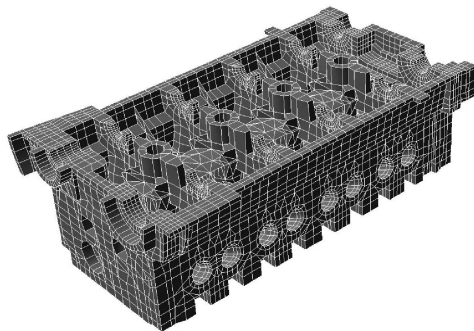

Figure 3. Finite element discretization of a cylinder head.

First, Eq. (1) is solved for model CH1 using eight processors and four different mesh decompositions with 22, 44, 90 and 223 subdomains, respectively. In each case, the CPU time incurred by the preconditioning step is reported in Table I. As for any DD based iterative

solver, there exists a range of number of subdomains for which the performance of FETI-DP

is optimal. In general, this range depends on many factors including the problem size. In this

case, this range appears to be in the neighborhood of 44 to 90. Figure 4 reports the history

of the number of FETI-DP iterations performed to achieve convergence for problem (20), as a

function of the j-th instance of its application to the solution of a problem of the form given

in (20). The reader can observe that after a few initial calls to the FETI-DP solver, the number

of iterations for convergence of this DD iterative solver drops significantly. This demonstrates

the effectiveness of the technique described in Section 4.2 for accelerating the iterative solution

of a system with multiple and/or repeated right hand-sides.

N_s	FETI-DP CPU Time
22	100 sec.
44	82 sec.
90	94 sec.
223	122 sec.

Table I. Model CH1: performance of the FETI-DP solver for one problem of the form (20) using eight

processors.

Similarly, it is found that for models CH2 and CH3, the performance of the FETI-DP

iterative solver is optimal for numbers of subdomains in neighborhoods centered around 229

and 944, respectively. For this reason, all subsequent performance results are discussed for the

partitionings of models CH1, CH2 and CH3 into 44, 229, and 944 subdomains, respectively.

Next, Eq. (1) is solved for all three partitioned models CH1, CH2, and CH3 using an

increasing number of processors. The obtained performance results are reported in in Tables II
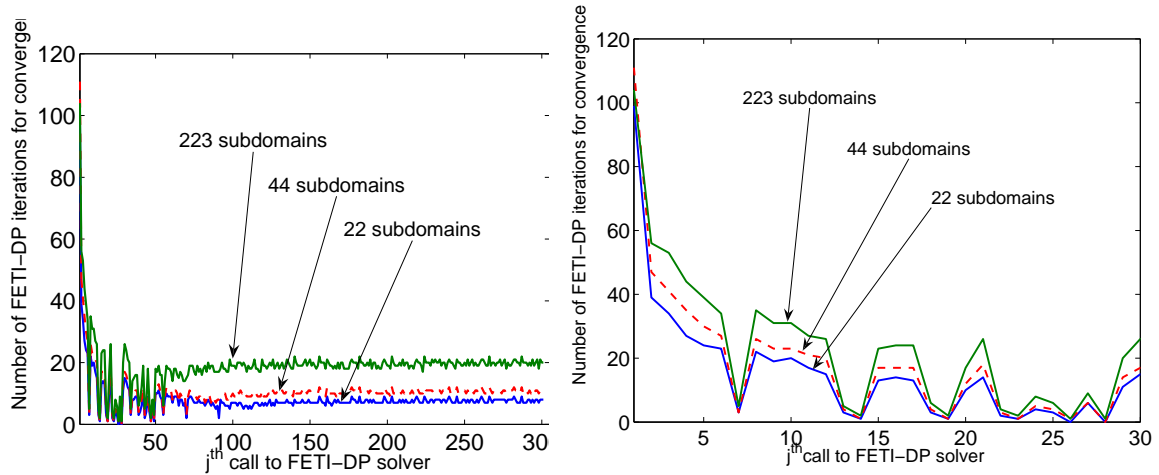
Figure 4. Effectiveness of the Krylov subspace reusage technique for accelerating the convergence of

FETI-DP for problems with multiple and repeated right hand-sides.

and III, respectively, where N_s and N_p denote the number of subdomains and processors, respectively, N_{itr}^{PCG} denotes the number of iterations for convergence of the overall PCG method (outer-loop), and $N_{call}^{FETI-DP}$ and $N_{itr}^{FETI-DP}$ denote the total number of calls to the FETI-DP solver and the accumulated number of performed FETI-DP iterations, respectively. These results reveal XXXXX

Model and size	N_s	N_p	N_{itr}^{PCG}	$N_{call}^{FETI-DP}$	$N_{itr}^{FETI-DP}$	FETI-DP CPU time	Total CPU time
CH1	44	8	16	301	2,573	82 sec.	109 sec.
54,198 dof	44	4	16	301	2,573	158 sec.	212 sec.
CH2	229	16	17	322	4,520	512 sec.	601 sec.
335,508 dof	229	8	17	322	4,530	912 sec.	1072 sec.

Table II. Performance results for models CH1 and CH2.

Model and size	N_p	N_{itr}^{PCG}	$N_{call}^{FETI-DP}$	$N_{itr}^{FETI-DP}$	FETI-DP CPU time	Total CPU time
CH2 335,508 dof	60	17	322	4,531	285 sec.	326 sec.
CH3 2,290,437 dof	60	16	301	6,432	2,548 sec.	2,863 sec.

Table III. Performance results for models CH2 and CH3.

From the results reported in Table II and Table III, the following observations can be made:

- It is found that for the CH1 model, the solution on four processors of one preconditioning problem of the form given in (20) consumes 5.5 seconds CPU. The solution on the same four processors of 301 of such problems during the solution of the global problem (1) consumes 158 seconds CPU — that is, 0.52 seconds on average per preconditioning problem. This CPU efficiency of FETI-DP is not only due to the acceleration technique described in Section 4.2, but also to the fact that the local matrices governing the subdomain-by-subdomain version of problems (20) are factored only once, during the first call to the FETI-DP solver.

- Model CH2 has 6.2 times more dof than model CH1. On eight processors, the total CPU time consumed by FETI-DP during the solution of Eq. (1) for model CH2 is 11.2 times larger than that consumed by FETI-DP during the solution of Eq. (1) for model CH1. This illustrates the numerical scalability property of FETI-DP highlighted in Section 4.1.

- Similarly on eight processors, the total CPU time consumed by the overall PCG algorithm for the solution of Eq. (1) associated with model CH2 is 9.8 times larger than that

consumed for the solution of the similar equation associated with model CH1. This suggests that the overall PCG algorithm proposed in this paper for the solution of Eq. (1) is also numerically scalable.

- Model CH3 is 6.8 times larger than model CH2, and using 60 processors, the total CPU time elapsed in the solution by FETI-DP of all of the preconditioning problems (1) is 9.1 times larger for CH3 than for CH2. Again, this illustrates FETI-DP's numerical scalability.

- Similarly on 60 processors, the total CPU time consumed by the overall PCG algorithm for the solution of Eq. (1) is 8.78 times larger for model CH3 than for model CH2. Once again, this suggests that the overall PCG algorithm proposed in this paper is numerically scalable.

- Even though $P = 126$ and the proposed overall PCG algorithm converges in one (outer-) iteration less for model CH3 than model CH2, the number of calls to the FETI-DP solver is reported to be lower by 21 calls only for the case of model CH3. This is because only 21 of the 126 diagonal blocks in the preconditioner (19) turn out to be non zero.

7. CONCLUSIONS

An incomplete block-diagonal preconditioner and its FETI-DP solver tailored for systems with multiple and repeated right hand-sides are proposed in this paper for the solution by an outer PCG algorithm of large-scale block systems of deterministic equations arising from the finite element stochastic analysis of structural problems with uncertainties. Performance results obtained for a three-dimensional problem from the automotive industry suggest that the proposed solution strategy is numerically scalable. These and similar performance results

Int. J. Numer. Meth. Engng 2008; **00**:1–6

obtained for other stochastic problems also suggest that the proposed algebraic solver has the potential for making the uncertainty quantification of realistic systems tractable.

REFERENCES

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

2. Klawonn A, Widlund O, and Dryja M. Dual-primal feti methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM Journal on Numerical Analysis*, 40:159–179, 2002.

3. Farhat C. A simple and efficient automatic fem domain decomposer. *Computer and Structures*, 28:579–602, 1988.

4. 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.

5. Farhat C and Li J. An iterative domain decomposition method for the solution of a class of indefinite problems in computational structural dynamics. *IMACS Journal of Applied Numerical Mathematics*, 54:150–166, 2005.

6. 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.

7. 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*, 155:129–152, 1998.

8. 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.

9. 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.

10. Farhat C, Crivelli L, and Roux FX. A transient feti methodology for large-scale parallel implicit computations in structural mechanics. *International Journal for Numerical Methods in Engineering*, 37:1945–1975, 1994.

11. Farhat C and Lesoinne M. Automatic partitioning of unstructured meshes for the parallel solution of

problems in computational mechanics. *International Journal for Numerical Methods in Engineering*, 36:745–764, 1993.

12. Farhat C, Lesoinne M, and Pierson K. A scalable dual-primal domain decomposition method. *Numerical Linear Algebra with Applications*, 7:687–714, 2000.

13. Farhat C, Lesoinne M, LeTallec P, Pierson K, and Rixen D. Feti-dp: A dual-primal unified feti method - part i: a faster alternative to the two-level feti method. *International Journal for Numerical Methods in Engineering*, 50:1523–1544, 2001.

14. Farhat C, Avery P, Tezaur R, and Li J. Feti-dph: a dual-primal domain decomposition method for acoustic scattering. *Journal of Computational Acoustics*, 13:499–524, 2005.

15. Farhat C and Chen PS. Tailoring domain decomposition methods for efficient parallel coarse grid solution and for systems with many right hand sides. *Contemporary Mathematics*, 180:401–406, 1994.

16. Farhat C, Chen PS, Risler F, and Roux FX. A unified framework for accelerating the convergence of iterative substructuring methods with lagrange multipliers. *International Journal for Numerical Methods in Engineering*, 42:257–288, 1998.

17. Farhat C, Chen PS, and Mandel J. A scalable lagrange multiplier based domain decomposition method for implicit time-dependent problems. *International Journal for Numerical Methods in Engineering*, 38:3831–3854, 1995.

18. Farhat C, Chen PS, and Mandel J. A two-level FETI method for static and dynamic plate problems Part II: Extension to shell problems, parallel implementation and performance results. *Computer Methods in Applied Mechanics and Engineering*, 155:153–180, 1998.

19. 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.

20. Dureisseix D and Farhat C. A numerically scalable domain decomposition method for the solution of frictionless contact problems. *International Journal for Numerical Methods in Engineering*, 50:2643–2666, 2001.

21. 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.

22. Rixen D and Farhat C. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *International Journal for Numerical Methods in Engineering*, 44:489–516, 1999.

23. Rixen D, Farhat C, Tezaur R, and Mandel J. Theoretical comparison of the feti and algebraically

partitioned feti methods, and performance comparisons with a direct sparse solver. *International Journal for Numerical Methods in Engineering*, 46:501–534, 1999.

24. Xiu D and Karniadakis G. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, 2003.

25. 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.

26. Bavestrello H, Avery P, and Farhat C. Incorporation of linear multipoint constraints in domain-decomposition-based iterative solvers - part ii: Blending feti-dp and mortar methods and assembling floating substructures. *Computer Methods in Applied Mechanics and Engineering*, 86:1347–1368, 2007.

27. 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.

28. 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.

29. Mandel J and Tezaur R. On the convergence of a dual-primal substructuring method. *Numerische Mathematik*, 88:543–558, 2001.

30. Bhardwaj M, Day D, Farhat C, Lesoinne M, Pierson K, and Rixen D. Application of the FETI method to ASCI problems - scalability results on 1000 processors and discussion of highly heterogeneo us problems. *International Journal for Numerical Methods in Engineering*, 47:513–535, 2000.

31. 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.

32. Wiener N. Homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

33. 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.

34. 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.

35. Ghanem R and Spanos PD. *Stochastic Finite Elements: A Spectral Approach*. Revised Edition, Dover Publications, 2003.

36. Ghanem R and Kruger RM. Numerical solution of spectral stochastic finite element systems. *Computer*

Methods in Applied Mechanics and Engineering, 129:289–303, 1996.

37. 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.

38. 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.

39. Brenner SC and Scott LR. *The Mathematical Theory of Finite Element Methods*. Second Edition, Springer, 2002.