SAND99-0785J

APPLICATION OF THE FETI METHOD TO ASCI PROBLEMS: SCALABILITY RESULTS ON A THOUSAND-PROCESSORS AND DISCUSSION OF HIGHLY HETEROGENEOUS PROBLEMS

[1]Manoj Bhardwaj, [1]David Day, [2]Charbel Farhat
[2]Michel Lesoinne, [2]Kendall Pierson, and [2]Daniel Rixen
[1] Sandia National Laboratories
Albuquerque, New Mexico 87185, U. S. A.
[2]Department of Aerospace Engineering Sciences
and Center for Aerospace Structures
University of Colorado at Boulder
Boulder, CO 80309-0429, U. S. A.

We report on the application of the one-level FETI method to the solution of a class of structural problems associated with the Department of Energy's Accelerated Strategic Computing Initiative (ASCI). We focus on numerical and parallel scalability issues, and discuss the treatment by FETI of severe structural heterogeneities. We also report on preliminary performance results obtained on the ASCI Option Red supercomputer configured with as many as one thousand processors, for problems with as many as 5 million degrees of freedom.

Keywords: ASCI, FETI, scalability, domain decomposition, structural heterogeneities

1. Introduction

In 1996, the US Department of Energy announced its Accelerated Strategic Computing Initiative (ASCI) aimed at creating predictive simulation and virtual prototyping capabilities, and accelerating the development of high-performance computing far beyond what might be achieved in the absence of a focused initiative. More specifically, ASCI's vision is to shift promptly from test-based methods to computational-based methods of ensuring the safety, reliability, and performance of the US nuclear weapons stockpile. An initial result of this initiative was the installation in 1997 at the Sandia National Laboratories of an Intel 1.8-Teraflops (trillion floating-point operations per second) peak massively parallel system known as the ASCI Option Red supercomputer. Two additional Teraflop systems known as the ASCI Blue Pacific and ASCI Blue Mountain machines were

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

subsequently sited at the Livermore and Los Alamos National Laboratories, respectively. Harnessing the power of these ASCI computers and exploiting their full potential requires the development of scalable numerical algorithms, which for many applications is a significant challenge.

Part of the ASCI initiative is the development at Sandia of Salinas, a massively parallel implicit structural dynamics code aimed at providing a scalable computational workhorse for highly accurate structural dynamic models. Such large-scale finite element models require significant computational effort, but provide important information including, vibrational loads for components within larger systems, design optimization, frequency response information for guidance and space systems, modal data necessary for active vibration control, and characterization data for structural health monitoring.

As in the case of many other ASCI software research and development projects, the success of Salinas hinges on its ability to deliver scalable performance results. However, unlike many other ASCI computational efforts, Salinas is an implicit code and therefore requires, among others, a scalable equation solver in order to meet its objectives. Because all three ASCI machines are massively parallel computers with thousands of processors, our definition of scalability here is the ability of an algorithm implemented on an ASCI system to solve an n-times larger problem using an n-times larger number of processors in a *nearly* constant CPU time. Achieving such a scalability requires not only a parallel hardware with relatively inexpensive interprocessor communication costs, but most importantly an equation solver that is (a) numerically scalable — that is, with an arithmetic complexity that grows *almost* linearly with the problem size, and (b) amenable to a scalable parallel implementation — that is, which can exploit as large a number of processors as possible while incurring relatively small interprocessor communication costs. Such a stringent definition of scalability rules out sparse direct solvers because their arithmetic complexity is a nonlinear function of the problem size. Furthermore, for large-scale three-dimensional structural problems with tens of millions of degrees of freedom (d.o.f.), the memory requirements of sparse direct solvers can overwhelm even the largest of the current ASCI machines. This is rather unfortunate because sparse direct methods offer otherwise a robustness that is not matched by any iterative algorithm. On the other hand, several multilevel [1] iterative schemes such as multigrid [2–4] algorithms and domain decomposition methods with coarse auxiliary problems [5] can be characterized by a nearly linear arithmetic complexity, or an iteration count that depends only weakly on the size of the problem to be solved. Such algorithms are prime candidates for a scalable equation solver. For Salinas, the Finite Element Tearing and Interconnecting (FETI) [6–12] solver was chosen because of its underlying mechanical concepts, as well as its potential for delivering a scalable performance.

FETI is a domain decomposition based iterative method with Lagrange multipliers. In its simplest form, it is also known as the one-level FETI method, and can be described as a two-step preconditioned conjugate gradient (PCG) algorithm where subdomain problems with Dirichlet (displacement) boundary conditions are solved in the preconditioning step, and related subdomain problems with Neumann (traction) boundary conditions are solved in a second step. The one-level FETI method incorporates a relatively small size auxiliary problem that is based on the subdomain rigid body modes. This coarse problem propagates the error globally during the PCG iterations and accelerates convergence.

For second-order elasticity problems discretized by plane stress/strain and/or solid elements, the condition number of the FETI interface problem preconditioned by the Dirichlet preconditioner [8] grows asymptotically as

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

where h denotes the mesh size, and H denotes the subdomain size (Fig. 1). Note that h, H, and h/H are indirect measures of the problem size, the number of subdomains, and the subdomain problem size, respectively. Hence, the condition number estimate (1) establishes the numerical scalability of the FETI method with respect to both the problem size and the number of subdomains. In particular, it proves that in theory, when the mesh discretization is refined, and the number of subdomains is increased as to maintain a constant number of elements per subdomain, the number of FETI iterations required for convergence remains asymptotically constant. This theoretical result has been demonstrated in practice for numerous applications [8–10]. The parallel scalability of the FETI method has also been demonstrated on various parallel computers with a number of processors ranging between 2 and 128 [13,14]. For fourth-order plate and shell problems, the condition number estimate (1) also holds when the rigid body based coarse problem is enriched by the subdomain corner modes [11,12]. In that case, the FETI method is transformed into a genuine two-level algorithm known as the two-level FETI method.

An important issue in multilevel methods that pertains to parallel scalability is the solution of the lowest level problem, which for domain decomposition methods corresponds to the coarse problem. The size of this coarse problem increases with the number of subdomains. Initially, it was advocated to solve the FETI coarse problems iteratively, using a CG algorithm that is optimized for the solution of problems with repeated right hand-sides [20,21]. That approach was motivated by the fact that the CG method requires only matrix-vector products that can be performed in parallel at the subdomain level, and which necessitate only short range communication between neighboring subdomains. For small mesh partitions and therefore small size coarse problems, it was shown that such a strategy is computationally efficient and allows both one-level and two-level FETI solvers to achieve parallel scalability [10,12]. However, the modified CG algorithm described in [20,21] is not numerically scalable with respect to the the size of the coarse problem, and therefore is not suitable for problems involving a large number of subdomains. Given that the most practical way for implementing domain decomposition methods on distributed memory parallel processors is to generate and assign one or several subdomains to each processor, it follows that the specific iterative solution strategy described in [20,21] is not suitable for ASCI computational platforms. When the given problem is partitioned into a large number of subdomains, it was shown in [14] that for shared memory multiprocessors, solving the FETI coarse problems by a direct method is computationally efficient. Hence, a first objective of this paper is to revisit this issue in the context of the Salinas code, ASCI structural problems, and ASCI computational hardware.

Strictly speaking, the condition number estimate (1) holds for uniform mesh discretizations, uniform mesh partitions with a perfect subdomain aspect ratio, and homogeneous problems — for example, structural problems with a single material, or different materials but with similar constitutive properties. However, it was shown in [15] that in practice, the numerical scalability of the FETI method holds for irregular discretizations and arbitrary mesh partitions, as long as the subdomains have reasonable aspect ratios. Algorithms for generating subdomains with reasonable aspect ratios can be found in [15–17]. For heterogeneous problems — for example, structural problems involving materials whose constitutive properties differ by several orders of magnitude — an improved coarse problem was proposed in [9] for preserving the numerical scalability of the FETI method. This alternative coarse problem was further investigated in [18] for model structural applications. More recently, a simple and virtually no-cost extension of the FETI preconditioner was proposed in [19] for addressing highly heterogeneous structural problems, but was also mainly verified on academic applications. Since ASCI structural problems are typically heterogeneous, a second objective of this paper is to construct a general strategy that combines both developments exposed in [9] and [19] for addressing the treatment by FETI of structural heterogeneities, and validate it for a realistic ASCI application problem.

Finally, a third objective of this paper is to report on preliminary performance results obtained on a thousand-processor configuration of the ASCI Option Red supercomputer, by the Salinas code equipped with the FETI solver, for various ASCI benchmark and real problems.

For this purpose, the remainder of this paper is organized as follows. In Section 2, we overview the one-level FETI method as our initial effort focuses on three-dimensional solid structures. In Section 3, we discuss a revised strategy for solving the subdomain rigid body mode based FETI coarse problem on a massively parallel processor. We also report some scalability results of the FETI method on the ASCI Option Red machine configured with as many as one thousand processors, for benchmark problems with as many as 5 million d.o.f. In Section 4, we consider the issue of structural heterogeneities and present a strategy for addressing them when using the FETI solver. In Section 5, we apply the FETI method to the finite element analysis on the ASCI Option Red supercomputer of a reentry vehicle, and in Section 6 we conclude this paper.

2. Overview of the one-level FETI method

Stress analyses, implicit linear and a large class of implicit nonlinear dynamic analyses, and vibration (eigenvalue) analyses lead to the solution of one or several systems of equations of the form

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{2}$$

where $\mathbf{K}$ is in general a symmetric positive definite matrix, $\mathbf{u}$ is a vector of generalized displacements, and $\mathbf{f}$ a vector of generalized forces. In the FETI method, the structure's computational domain Ω is partitioned into N_s *non-overlapping* subdomains $\Omega^{(s)}$, and Lagrange multipliers λ are introduced at the subdomain interfaces to enforce the compatibility of the subdomain generalized displacement field $\mathbf{u}^{(s)}$. Consequently, Eq. (2) above is transformed into the equivalent set of equations

$$\begin{aligned} \mathbf{K}^{(s)}\mathbf{u}^{(s)} &= \mathbf{f}^{(s)} - \mathbf{B}^{(s)^T}\lambda \qquad s = 1, \dots N_s \\ \sum_{s=1}^{N_s} \mathbf{B}^{(s)}\mathbf{u}^{(s)} &= 0 \end{aligned} \tag{3}$$

where for each subdomain $\Omega^{(s)}$, $\mathbf{K}^{(s)}$ denotes its generalized stiffness matrix, $\mathbf{f}^{(s)}$ its vector of generalized forces, and $\mathbf{B}^{(s)}$ the signed Boolean matrix that extracts from a subdomain vector $\mathbf{v}^{(s)}$ its signed ($\pm$) restriction to the subdomain interface boundary. The first of Eqs. (3) expresses the local equilibrium of the subdomains $\Omega^{(s)}$, and the second of Eqs. (3) the continuity of the subdomain generalized displacement fields across the subdomain interfaces.

The general solution of the first of Eqs. (3) can be written as

$$\mathbf{u}^{(s)} = \mathbf{K}^{(s)^+}\left(\mathbf{f}^{(s)} - \mathbf{B}^{(s)^T}\boldsymbol{\lambda}\right) + \mathbf{R}^{(s)}\boldsymbol{\alpha}^{(s)} \tag{4}$$

where $\mathbf{K}^{(s)^+}$ denotes the inverse of $\mathbf{K}^{(s)}$ if $\Omega^{(s)}$ has sufficient Dirichlet boundary conditions to prevent $\mathbf{K}^{(s)}$ from being singular, or a generalized inverse of $\mathbf{K}^{(s)}$ if $\Omega^{(s)}$ is a floating subdomain. In the latter case, the columns of $\mathbf{R}^{(s)}$ represent the rigid body (or more generally the zero energy) modes of $\Omega^{(s)}$, i.e. $\mathbf{R}^{(s)} = \ker \mathbf{K}^{(s)}$, and $\boldsymbol{\alpha}^{(s)}$ is the set of amplitudes that specifies the contribution of the null space $\mathbf{R}^{(s)}$ to the solution $\mathbf{u}^{(s)}$. These coefficients can be determined by requiring that each subdomain problem be mathematically solvable — that is, each floating subdomain be self-equilibrated — which can be written as

$$\mathbf{R}^{(s)^T}\left(\mathbf{f}^{(s)} - \mathbf{B}^{(s)^T}\boldsymbol{\lambda}\right) = 0 \tag{5}$$

Substituting Eq. (4) into the compatibility equation and exploiting the solvability condition (5) transforms problem (3) into the interface problem

$$\begin{bmatrix} \mathbf{F}_I & -\mathbf{G}_I \\ -\mathbf{G}_I{}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ -\mathbf{e} \end{bmatrix} \tag{6}$$

where

$$\begin{aligned} \mathbf{F}_I &= \sum_{s=1}^{N_s} \mathbf{B}^{(s)}\mathbf{K}^{(s)^+}\mathbf{B}^{(s)^T} \\ \mathbf{d} &= \sum_{s=1}^{N_s} \mathbf{B}^{(s)}\mathbf{K}^{(s)^+}\mathbf{f}^{(s)} \\ \mathbf{G}_I &= \left[\mathbf{G}_I^{(1)} \quad ... \quad \mathbf{G}_I^{(N_f)}\right] = \left[\mathbf{B}^{(1)}\mathbf{R}^{(1)} \quad ... \quad \mathbf{B}^{(N_f)}\mathbf{R}^{(N_f)}\right] \\ \boldsymbol{\alpha} &= \left[\boldsymbol{\alpha}^{(1)^T} \quad ... \quad \boldsymbol{\alpha}^{(N_f)^T}\right]^T \\ \mathbf{e} &= \left[\mathbf{f}^{(1)^T}\mathbf{R}^{(1)} \quad ... \quad \mathbf{f}^{(N_s)^T}\mathbf{R}^{(N_f)}\right]^T \end{aligned} \tag{7}$$

and where N_f denotes the number of floating subdomains.

The above indefinite interface problem (6) can be transformed into a semi-definite system of equations as follows. Let $\mathbf{Q}$ be any symmetric matrix for which the product $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ is invertible. The self-equilibrium condition $\mathbf{G}_I^T\boldsymbol{\lambda} = \mathbf{e}$ can be eliminated from Eqs. (6) by introducing the splitting

$$\boldsymbol{\lambda} \;=\; \boldsymbol{\lambda}^0 + \mathbf{P}(\mathbf{Q})\bar{\boldsymbol{\lambda}} \tag{8}$$

where λ^0 is a particular solution of $\mathbf{G}_I^T\lambda = \mathbf{e}$ of the form

$$\lambda^0 = \mathbf{Q}\mathbf{G}_I(\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I)^{-1}\mathbf{e} \tag{9}$$

and $\mathbf{P}(\mathbf{Q})$ is a projector of the form

$$\mathbf{P}(\mathbf{Q}) = \mathbf{I} - \mathbf{Q}\mathbf{G}_I(\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I)^{-1}\mathbf{G}_I^T \tag{10}$$

Note that $\mathbf{P}$ satisfies

$$\mathbf{P}^2 = \mathbf{P} \qquad \mathbf{G}_I^T\mathbf{P} = 0 \tag{11}$$

Substituting Eq. (8) into the first of Eqs. (6) and premultiplying that equation by $\mathbf{P}^T$ transforms the indefinite interface problem (6) into the projected interface problem

$$(\mathbf{P}^T\mathbf{F}_I\mathbf{P})\,\bar{\lambda} = \mathbf{P}^T\left(\mathbf{d} - \mathbf{F}_I\lambda^0\right) \tag{12}$$

which is symmetric positive semi-definite for any given matrix $\mathbf{Q}$.

The one-level FETI method consists in transforming the original global problem (2) into the symmetric positive semi-definite interface problem (12), and solving the latter system of equations by a PCG algorithm. Note that the projector $\mathbf{P}$ contains the matrix $(\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I)^{-1}$, which is symmetric when $\mathbf{Q}$ is symmetric. In general, each subdomain has at most 6 rigid body modes, and therefore the dimension of $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ is at most equal to $6N_s$. Hence, this matrix defines an auxiliary coarse problem that couples all the subdomain computations, and which was proved to propagate the error globally and accelerate convergence [8,22].

Two preconditioners have been proposed in the literature for the FETI method: the mathematically optimal Dirichlet preconditioner $\mathbf{P}^T\overline{\mathbf{F}}_I^{D^{-1}}\mathbf{P}$ introduced in [8], and the computationally economical even if not numerically scalable lumped preconditioner $\mathbf{P}^T\overline{\mathbf{F}}_I^{L^{-1}}\mathbf{P}$ proposed in earlier works [6,7]. If each subdomain generalized stiffness matrix is partitioned as

$$\mathbf{K}^{(s)} = \begin{bmatrix} \mathbf{K}_{ii}^{(s)} & \mathbf{K}_{ib}^{(s)} \\ \mathbf{K}_{ib}^{(s)^T} & \mathbf{K}_{bb}^{(s)} \end{bmatrix} \tag{13}$$

where the subscripts i and b designate the subdomain interior and interface boundary d.o.f., respectively, then the component $\overline{\mathbf{F}}_I^{D^{-1}}$ of the Dirichlet preconditioner can be written as

$$\begin{aligned} \overline{\mathbf{F}}_I^{D^{-1}} &= \sum_{s=1}^{N_s} \mathbf{W}^{(s)}\mathbf{B}^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_{bb}^{(s)} \end{bmatrix} \mathbf{B}^{(s)^T}\mathbf{W}^{(s)} \\ &\text{where} \\ \mathbf{S}_{bb}^{(s)} &= \mathbf{K}_{bb}^{(s)} - \mathbf{K}_{ib}^{(s)^T}\mathbf{K}_{ii}^{(s)^{-1}}\mathbf{K}_{ib}^{(s)} \end{aligned} \tag{14}$$

and the component $\overline{\mathbf{F}}_I^{L^{-1}}$ of the lumped preconditioner can be written as

$$\overline{\mathbf{F}}_I^{L^{-1}} = \sum_{s=1}^{N_s} \mathbf{W}^{(s)}\mathbf{B}^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{K}_{bb}^{(s)} \end{bmatrix} \mathbf{B}^{(s)^T}\mathbf{W}^{(s)} \tag{15}$$

In the above expressions of $\overline{\mathbf{F}}_I^{D^{-1}}$ and $\overline{\mathbf{F}}_I^{L^{-1}}$, $\mathbf{W}^{(s)}$ is a diagonal "scaling" matrix. In its simplest form, $\mathbf{W}^{(s)}$ stores in each of its entries the inverse of the multiplicity of the corresponding interface node — that is, the inverse of the number of subdomains attached to that node [23,9,11]. For example, if the i-th Lagrange multiplier component $\lambda(i)$ acts on an interface node that is shared between 2 subdomains, then $\mathbf{W}^{(s)}(i) = 1/2$; if it acts on an interface node that is shared by m subdomains, then $\mathbf{W}^{(s)}(i) = 1/m$. Such a matrix $\mathbf{W}^{(s)}$ is referred to as the "topological scaling" matrix.

Recently, both Dirichlet and lumped preconditioners have been extended to address more efficiently heterogeneous structural problems [19]. These extensions are simply obtained by redefining appropriately the scaling matrix $\mathbf{W}^{(s)}$.

Finally, we note that for homogeneous problems, the simplest choice $\mathbf{Q} = \mathbf{I}$ is the most computationally efficient. Most FETI computations reported in the literature have been performed with this trivial choice. However, it was shown in [9] that heterogeneous problems call for a matrix $\mathbf{Q}$ that is physically homogeneous to a generalized stiffness. For this reason, two alternative choices for $\mathbf{Q}$ were first proposed in [9]: $\mathbf{Q} = \mathbf{Q}^L = \overline{\mathbf{F}}_I^{L^{-1}}$, and $\mathbf{Q} = \mathbf{Q}^D = \overline{\mathbf{F}}_I^{D^{-1}}$. These choices were further investigated in [18] and shown to be effective for model heterogeneous problems.

3. Scalabilty results on the ASCI Red Option supercomputer

3.1. Implementation of Salinas and FETI on massively parallel distributed memory systems

Like FETI, Salinas is based on substructuring, and relies on the same concept of mesh partitioning. For this reason, interfacing both codes was a relatively straightforward task. Using an automatic mesh decomposer [29,30], a given finite element structural model is first decomposed into $N_s \geq N_p$ subdomains, where N_p denotes the target number of processors. The potential advantages of generating more subdomains than there are processors are discussed in [14,31], among other references. Next, the generated subdomains are re-arranged into N_p clusters containing each one or several subdomains, and each cluster is assigned to one processor. Most if not all Salinas and FETI computations can be performed concurrently at the subdomain level, and necessitate interprocessor communication only between neighboring clusters. As far as FETI is concerned, only the solution at each PCG iteration k of a coarse problem of the form

$$(\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I)\boldsymbol{\alpha}^k = \mathbf{G}_I^T\mathbf{w}^k \tag{16}$$

deserves special attention. Such a coarse problem is associated with a matrix-vector multiplication of the form $\mathbf{P}\mathbf{w}^k$ or $\mathbf{P}^T\mathbf{w}^k$, where $\mathbf{w}^k$ denotes the k-th preconditioned residual generated by the PCG algorithm applied to the solution of the interface problem $(\mathbf{P}^T\mathbf{F}_I\mathbf{P})\bar{\lambda} = \mathbf{P}^T(\mathbf{d}-\mathbf{F}_I\lambda^0)$ (12). Hence, it arises twice at each FETI iteration. Before addressing this issue, we note that

- the system matrix $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ is independent of the iteration number k. Only the right hand side vector $\mathbf{G}_I^T\mathbf{w}^k$ varies throughout the FETI iterations.
- for any $\mathbf{Q}$, the system matrix $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ has the same size which depends on the number of floating subdomains N_f, and the dimensions of the null spaces $\ker\mathbf{K}^{(s)}$, $s = 1, ..., N_f$. In general, the size of $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ is of the order of $6N_s$.
- for $\mathbf{Q} = \mathbf{I}$, $\mathbf{G}_I^T\mathbf{G}_I$ is a sparse symmetric positive matrix. Its sparsity pattern is dictated by the connectivity of the mesh partition, and is identical to the sparsity pattern of any finite element matrix obtained by treating each subdomain as a "superelement". More specifically, $\mathbf{G}_I^{(s)^T}\mathbf{G}_I^{(q)} \neq 0$ if and only if $\Omega^{(s)}$ and $\Omega^{(q)}$ are neighboring subdomains.
- for $\mathbf{Q} = \mathbf{Q}^L = \overline{\mathbf{F}}_I^{L^{-1}}$ and $\mathbf{Q} = \mathbf{Q}^D = \overline{\mathbf{F}}_I^{D^{-1}}$, $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ is also a sparse symmetric positive matrix, and its sparsity pattern is also dictated by the connectivity of the mesh partition. However, in these two cases $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ is slightly more populated than $\mathbf{G}_I^T\mathbf{G}_I$. More specifically, $\mathbf{G}_I^{(s)^T}\mathbf{Q}\mathbf{G}_I^{(q)} \neq 0$ if and only if $\Omega^{(q)}$ is a neighbor of $\Omega^{(s)}$, or the neighbor of a neighbor of $\Omega^{(s)}$.
- for all three choices of $\mathbf{Q}$ specified above, $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ can be assembled in parallel using only subdomain level computations, and a small amount of interprocessor communication between processors mapped onto neighboring clusters of subdomains.

As mentioned in the introduction, we consider here solving the coarse problem (16) by a direct method. Such a strategy improves the robustness of the FETI method, but complicates its implementation on massively parallel distributed memory systems. As stated earlier, parallel sparse direct algorithms do not scale well in the sense defined in this paper, particularly for these small size coarse problems. Furthermore, we note that because the system matrix $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ needs be factored only once, but the coarse problem (16) must be solved twice at each

FETI iteration, it is essential to focus on a strategy that addresses not only the factorization of $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$, but most importantly the subsequent forward and backward substitutions. Indeed, the scalable parallelization of the direct solution of sparse lower and upper triangular systems is even more challenging than that of the factorization of a sparse matrix.

For all of the above reasons, we consider here the following approach for solving the coarse problem (16) on a massively parallel distributed memory system such as the ASCI Option Red supercomputer. For the sake of notational simplicity, but without any loss of generality, we assume in the following algorithmic description that each floating subdomain has exactly 6 rigid body modes.

a) form $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ in parallel and duplicate this relatively small size sparse matrix in each processor.

b) request that each processor factor $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$.

c) compute in parallel a distributed inverse of $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ as follows. For each floating subdomain $\Omega^{(j)}$ assigned to processor p_j, request that p_j performs 6 forward and backward substitutions to solve

$$(\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I)\mathbf{X}_j = \mathbf{I}_j \tag{17}$$

where $\mathbf{I}_j$ contains the 6 columns of the identity matrix $\mathbf{I}$ that are assigned to subdomain $\Omega^{(j)}$ in conjunction with its 6 rigid body modes $\mathbf{R}^{(j)}$.

d) at each FETI iteration k, solve each coarse problem of the form given in (16) by performing a parallel distributed matrix-vector multiplication. Indeed, from Eqs. (7,16,17) it follows that

$$\alpha^k \;=\; \mathbf{X}(\mathbf{G}^T\mathbf{w}^k) \;=\; \sum_{j=1}^{j=N_f} \mathbf{X}_j[\mathbf{G}_I^{(j)^T}\mathbf{w}^k] \tag{18}$$

which shows that the evaluation of α^k can be performed using subdomain-by-subdomain parallel computations and requires only one global range communication.

The strategy outlined above for solving the coarse problem (16) is essentially composed of three sequences of embarrassingly parallel computations.

The first one has two caveats. From a computational viewpoint, requesting that all processors perform the factorization of the same matrix $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ is equivalent to serializing this computation. This serialization does not significantly affect the overall performance of FETI, as long as the cost of the factorization of $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ is negligible compared to the cost of the other FETI operations — that is, as long as the number of subdomains is below a certain critical value N_s^{cr}. However, because of Amdahl's law, there also exists a certain number of processors N_p^{cr} beyond which this serialization will prevent FETI from scaling well on a massively parallel system. Furthermore, given that the size of $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ increases with the number of subdomains N_s, and that N_s increases with the number of processors N_p, there also exists a critical number of subdomains and/or processors beyond which storing $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ in a single processor of a local memory system will not be feasible. However, note that after the $\mathbf{X}_j$ column matrices have been computed, $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ can be deleted, which frees memory for other usage, for example, by Salinas.

The second sequence (c) of embarrassingly parallel computations is an effective one from both computational complexity and parallel scalability viewpoints. The third sequence (d) is also perfectly scalable from a parallel processing viewpoint. It is also computationally efficient if the size of each cluster of subdomains is such that the total number of column matrices $\mathbf{X}_j$ assigned to a processor p_j is comparable to the average number of nonzero entries in a row of the factors of the sparse matrix $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$. In particular, if one and only one subdomain is assigned to each processor ($N_s = N_p$), the embarrassingly parallel steps (c) and (d) are both numerically and parallel-wise scalable.

In summary, one can reasonably expect that the FETI method equipped with the coarse problem solver described above will scale well on massively parallel distributed memory systems, up to a certain problem and/or machine size (number of processors) beyond which the storage scheme and factorization method of the coarse matrix $\mathbf{G}_I^T\mathbf{Q}\mathbf{G}_I$ will need to be revisited. Hence, a first objective of this work is to assess this limit in the context of the solution of three-dimensional second-order elasticity problems on the ASCI Option Red supercomputer.

3.2. The ASCI Option Red supercomputer

The ASCI initiative supports the ASCI Option Red supercomputer, a massively parallel processor with a distributed memory multiple instruction and multiple data architecture, as well as the ASCI Option Blue Mountain and ASCI Option Blue Pacific supercomputers. The ASCI Option Red and Blue Mountain systems run MP LINPACK, one of the computer industry's standard speed tests for large systems, at 1.3 and 1.6 Teraflops respectively [24].

The ASCI Option Red supercomputer, also known as the Intel Teraflops machine, is the first large-scale supercomputer built entirely of commodity, commercial, off-the-shelf components. It has 4,536 compute and 72 service nodes each with 2 Pentium Pro processors, 594 Gbytes of real memory, and two independent

1-Terabyte disk systems. It occupies 1600 sq. ft. of floor-space (Fig. 2). The system's 9,216 Pentium Pro processors are connected by a 38 × 32 × 2 mesh.

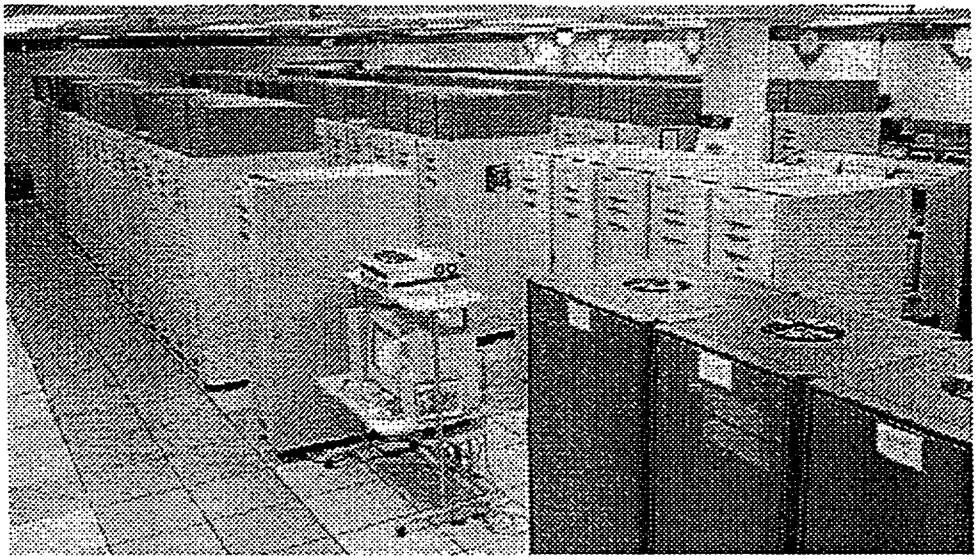

Fig. 2. The ASCI Option Red supercomputer

The Pentium Pro processor runs at 200 MHz and has a peak floating-point rate of 200 Mflops (million floating-point operations per second). It has separate on-chip data and instruction L1 caches of 8 Kbytes each. It has also an L2 cache of 256 Kbytes packaged with the CPU in a single dual-cavity PGA package. All cache lines are 32 bytes wide. The system was delivered with 128 Mbytes of memory per node, but supports up to 256 Mbytes of memory per node. The two processors on each node support two on-board PCI interfaces; each of these interfaces provides 133 Mbytes/sec I/O bandwidth. The memory subsystem is structured as four rows of four independently controlled and sequentially interleaved banks of DRAM to produce up to 533 Mbytes/sec of data throughput. Each memory bank is 72 bits wide. The router supports bi-directional bandwidths of up to 800 Mbytes/sec over each of six ports. As many as four message streams can pass on any given port and at any given time.

Two UNIX-based operating systems collectively called the Teraflops OS run on the ASCI Option Red supercomputer and present a single system image to the user. Compute nodes run an efficient small operating system called Cougar [25–27]. Service nodes run POSIX 1003.1 and XPG3, and AT&T System V.3 and 4.3 BSD Reno VFS [28]. The file system is concentrated on a small set of specialized nodes that process I/O requests. Symbios RM20 Redundant Arrays of Independent Disks (RAIDs) are used for secondary storage. A Symbios RM20 RAID has two bays of ten drives each and two controllers. The disk drives are Seagate 4-GbytesBarracudas with a 3.5" form-factor [28].

3.3. Preliminary scalability results

Assessing the scalability (in the sense defined in this paper) of both the FETI method and its massively parallel implementation described in Section 3.2 requires generating, for a given problem, a sequence of finite element models where the total number of d.o.f. is increased proportionally to an increasing sequence of number of processors, in order to maintain the ratio problem size over machine size constant. Generating such a sequence of finite element models and the corresponding sequence of mesh partitions is in general a tedious task. For this reason, and because the numerical scalability of the FETI method has already been established and repeatedly demonstrated for realistic structural problems [9–14,31], we consider here two simple three-dimensional benchmark problems that are easy to generate and manipulate for scalability studies. Both benchmark problems correspond to homogeneous structures uniformly discretized by 4-noded brick elements, and partitioned into cubic subdomains. For this reason, we set $\mathbf{Q} = \mathbf{I}$ for both problems. In both cases, we generate the sequence of finite element meshes by fixing the size of each cubic subdomain, and increasing the number of subdomains N_s to match the target number of processors N_p. Hence, we consider here only the case $N_s = N_p$, because we consider large values of N_p ranging between 8 and 1000 processors. We fix the subdomain size to 1728 elements ($12 \times 12 \times 12$), which corresponds to the maximum subdomain size affordable by FETI for one of the two benchmark problems on a single processor with 128 Mbytes of memory, after the Salinas memory requirements have been met. The two benchmark problems considered here differ as follows

1) in benchmark problem BP1, the structure has a cubic shape and is partitioned into $n \times n \times n$ subdomains (Fig. 3.). It is clamped at one end, and subjected to a distributed vertical load at the other.

2) in benchmark problem BP2, the structure has the shape of a rectangular parallelepiped and is partitioned into $2 \times 2 \times n$ subdomains (Fig. 4.). It is clamped at one end, and subjected to a distributed vertical load at the other.

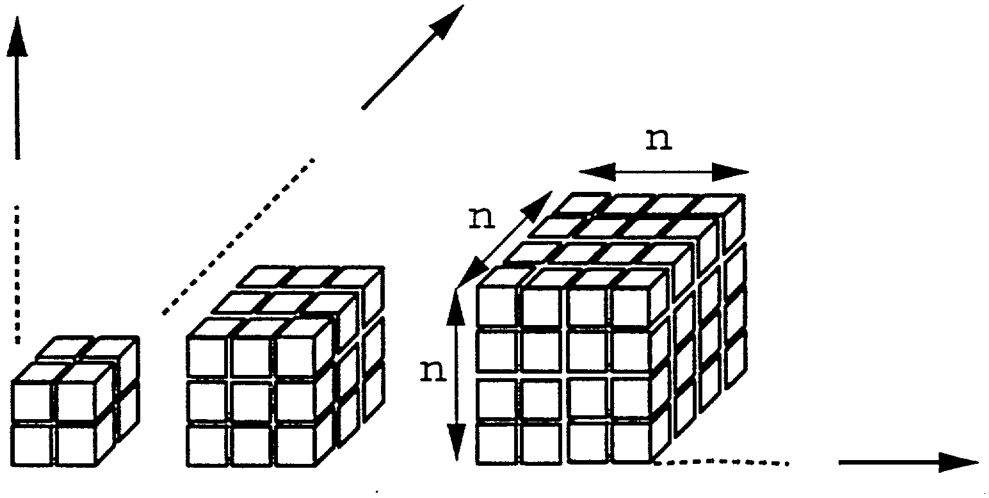

Fig. 3. The benchmark problem BP1

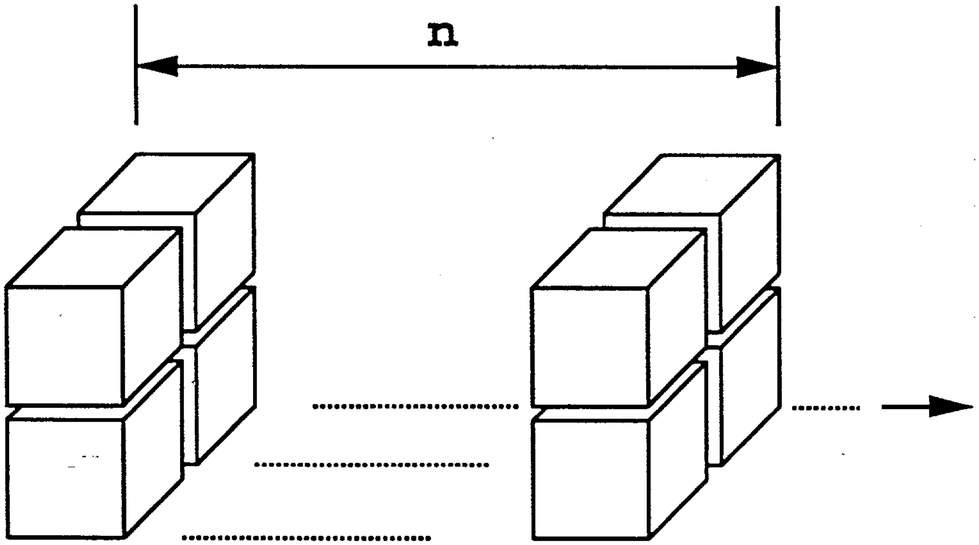

Fig. 4. The benchmark problem BP2

In both cases, the size of the $\mathbf{G}_I^T\mathbf{G}_I$ matrix increases linearly with the number of subdomains, and the amount of fill-in suffered during the factorization of this matrix grows with the number of the subdomains lying in a plane perpendicular to the main axis of the structure. Hence, for benchmark problem BP1, this amount of fill-in increases as $O\ (n^2)$, while for benchmark problem BP2 it remains constant. Since the parallel implementation of the one-level FETI method described in Section 3.2 calls for duplicating the storage of $\mathbf{G}_I^T\mathbf{G}_I$ in every processor, and effectively serializes the factorization of this matrix, it follows that problems BP1 and BP2 provide a worst-case and best case scenarios, respectively, for the memory requirements and parallel scalability of the FETI method. In particular, given that the size of each subdomain is fixed to 1728 elements, that each processor of the target ASCI Red Option supercomputer has 128 Mbytes of memory only, and that Salinas has its own memory requirements that must be accomodated, the $O\ (n^2)$ growth of fill-in for the factorization of the coarse problem associated with the benchmark problem BP1 limits the number of subdomains that can be considered in this investigation to $N_s = 1000$, and therefore limits the number of processors to $N_p = 1000$. Of course, this number of subdomains and/or processors can be increased by decreasing the subdomain problem size. However, because access to more than 1000 processors on the ASCI Red Option machine is also a practical challenge by itself, we limit here our investigation of the performance and scalability of the FETI method on the ASCI Red Option supercomputer to a maximum number of 1000 processors.

We also note that most realistic problems are neither cube-shaped, nor parallelepiped-shaped. The decomposition of their meshes seldom generates perfectly load-balanced subdomains, or subdomains with a perfect aspect ratio. However, based on the arguments presented above, we can reasonably argue that the scalability results of FETI for the benchmark problems BP1 and BP2 provide lower and upper bounds of the scalability results to be expected for the solution by FETI on the ASCI Red Option supercomputer of more realistic problems.

Even though Salinas is primarily a structural dynamics implicit code, we report here on the scalability of this software equipped with the FETI solver for linear static analysis. This is because the complexity of solving a system of equations arising from one step of an implicit structural dynamic (large time-step) analysis, or from a static analysis, is essentially the same. For the optimization of FETI to the solution of repeated systems arising form the linear dynamic analysis or the eigenvalue analysis of a structure, we refer the reader to [10,20,21]. For both benchmark problems, we equip FETI with the topological scaling matrix $\mathbf{W}^{(s)}$. We report in Table 1 and Table 2 the performance results obtained for problem BP1 on a 1000-processor configuration of the ASCI Red Option supercomputer, using the Dirichlet and lumped preconditioners, respectively. Similarly, we report in Table 3 and Table 4 the performance results obtained for problem BP2. In all cases, we use the following stopping criterion

$$||\mathbf{Ku} - \mathbf{f}||_2 < 10^{-6} \times ||\mathbf{f}||_2 \tag{19}$$

Table 1
Solution by FETI equipped with the Dirichlet preconditioner of the benchmark problem BP1 on the ASCI Option Red supercomputer

n	$N_p = N_s$	N_{dof}	N_{itr}	Factor $\mathbf{G}_I^T\mathbf{G}_I$	FETI	Salinas + FETI
2	8	46,875	14	0.001 sec	209 sec	336 sec
3	27	151,959	20	0.006 sec	216 sec	346 sec
4	64	352,947	25	0.05 sec	222 sec	355 sec
5	125	680,943	27	0.3 sec	225 sec	358 sec
6	216	1,167,051	30	1.1 sec	229 sec	365 sec
7	343	1,842,375	31	2.9 sec	235 sec	367 sec
8	512	2,738,019	33	5.8 sec	239 sec	380 sec
9	729	3,885,087	33	14.9 sec	252 sec	405 sec
10	1,000	5,314,683	34	32.4 sec	275 sec	413 sec

Table 2
Solution by FETI equipped with the lumped preconditioner of the benchmark problem BP1 on the ASCI Option Red supercomputer

n	$N_p = N_s$	N_{dof}	N_{itr}	Factor $\mathbf{G}_I^T\mathbf{G}_I$	FETI	Salinas + FETI
2	8	46,875	27	0.001 sec	140 sec	267 sec
3	27	151,959	36	0.006 sec	148 sec	278 sec
4	64	352,947	45	0.05 sec	155 sec	288 sec
5	125	680,943	48	0.3 sec	157 sec	290 sec
6	216	1,167,051	51	1.1 sec	161 sec	297 sec
7	343	1,842,375	55	2.9 sec	167 sec	299 sec
8	512	2,738,019	55	5.8 sec	171 sec	312 sec
9	729	3,885,087	58	14.9 sec	187 sec	340 sec
10	1,000	5,314,683	60	32.4 sec	261 sec	399 sec

Table 3
Solution by FETI equipped with the Dirichlet preconditioner of the benchmark problem BP2 on the ASCI Option Red supercomputer

n	$N_p = N_s$	N_{dof}	N_{itr}	Factor $\mathbf{G}_I^T\mathbf{G}_I$	FETI	Salinas + FETI
2	8	46,875	14	0.001 sec	209 sec	336 sec
7	28	159,375	18	0.007 sec	214 sec	343 sec
16	64	316,875	18	0.018 sec	215 sec	354 sec
31	124	699,375	18	0.038 sec	216 sec	347 sec
54	216	1,216,875	19	0.073 sec	216 sec	350 sec
86	344	1,936,875	19	0.131 sec	217 sec	351 sec
128	512	2,881,875	19	0.203 sec	218 sec	354 sec
182	728	4,096,875	19	0.298 sec	218 sec	355 sec
250	1,000	5,626,875	19	0.414 sec	222 sec	360 sec

Table 4
Solution by FETI equipped with the lumped preconditioner of the benchmark problem BP2 on the ASCI Option Red supercomputer

n	$N_p = N_s$	N_{dof}	N_{itr}	Factor $\mathbf{G}_I^T\mathbf{G}_I$	FETI	Salinas + FETI
2	8	46,875	27	0.001 sec	140 sec	267 sec
7	28	159,375	30	0.007 sec	142 sec	271 sec
16	64	316,875	31	0.018 sec	143 sec	282 sec
31	124	699,375	31	0.038 sec	143 sec	274 sec
54	216	1,216,875	31	0.073 sec	144 sec	278 sec
86	344	1,936,875	31	0.131 sec	144 sec	278 sec
128	512	2,881,875	32	0.203 sec	146 sec	282 sec
182	728	4,096,875	33	0.298 sec	147 sec	284 sec
250	1,000	5,626,875	34	0.414 sec	151 sec	289 sec

The performance results reported in Tables 1–4 show that

- the FETI method equipped with the Dirichlet preconditioner achieves numerical scalability (constant asymptotic iteration count) for both benchmark problems BP1 and BP2.

- when equipped with the lumped preconditioner, the FETI method achieves numerical scalability for problem BP2. It also exhibits a reasonable numerical scalability for problem BP1. For both benchmark problems, the FETI method performs on average 1.6 times more iterations when equipped with the lumped preconditioner than when equipped with the Dirichlet preconditioner. However for problem BP2, the FETI method is on average 1.5 times faster when equipped with the lumped preconditioner than with the Dirichlet one. This demonstrates the computational efficiency of the lumped preconditioner for the solution of second-order elasticity problems by the FETI method. For problem BP1 and $8 \leq N_p \leq 1000$, the FETI method is also faster when equipped with the lumped preconditioner. However, the results reported in Table 1 and Table 2 suggest that for $N_p \geq 1000$, the Dirichlet preconditioner will outperform the lumped preconditioner, which highlights the importance of ensuring numerical scalability for achieving parallel scalability on a very large number of processors.
- for problem BP1, the cost of the factorization of the matrix $\mathbf{G}_I^T \mathbf{G}_I$ — which is the only sequential operation performed by the current implementation of FETI on massively parallel local memory machines — is shown to increase dramatically with $n = N_s^{1/3} = N_p^{1/3}$. This is consistent with the $O\ (n^2)$ growth of the fill-in predicted for the factorization of this matrix for problem BP1. Nevertheless, the results reported in Table 2 show that for the benchmark problem BP1, the FETI method equipped with the lumped preconditioner solves 5,314,683 equations in 261 seconds CPU on a 1000-processor configuration of the ASCI Red Option supercomputer.
- for problem BP2, the CPU time consumed by the sequential factorization of the coarse problem of the FETI method is reported to grow only linearly with the number of subdomains. This is consistent with our analytical prediction that is based on the fact that the size of $\mathbf{G}_I^T \mathbf{G}_I$ grows linearly with the number of subdomains, and the fact that for problem BP2, the fill-in suffered during the factorization of this matrix is independent of the number of subdomains. For the benchmark problem BP2, the FETI method equipped with the lumped preconditioner solves 5,626,875 equations on 1000 processors in 151 seconds.

In order to quantify the scalabilty of the current implementation of the FETI method on the ASCI Red Option supercomputer, we introduce the following definition of the speed-up

$$Sp = \frac{8 \times T_8}{T_{N_p}} \times \frac{N_{dof_{N_s}}}{N_{dof_8}} \qquad (20)$$

where T_8 and T_{N_p} denote respectively the CPU timings corresponding to 8 and N_p processors, and N_{dof_8} and $N_{dof_{N_s}}$ denote respectively the sizes (in d.o.f.) of the global problems corresponding to 8 and N_s subdomains. Here, the case $N_p = 8$ is taken as a reference point. Note that the above definition of the speed-up is a *strict* one: it accounts for both concepts of numerical and parallel scalability. It assesses the combined performances of the given algorithm, its parallel implementation, and the parallel hardware on which this algorithm is executed. For the benchmark problems BP1 and BP2, the speed-ups achieved by the FETI method are reported Fig. 5 for the case of the Dirichlet preconditioner, and in Fig. 6 for the case of the lumped preconditioner.

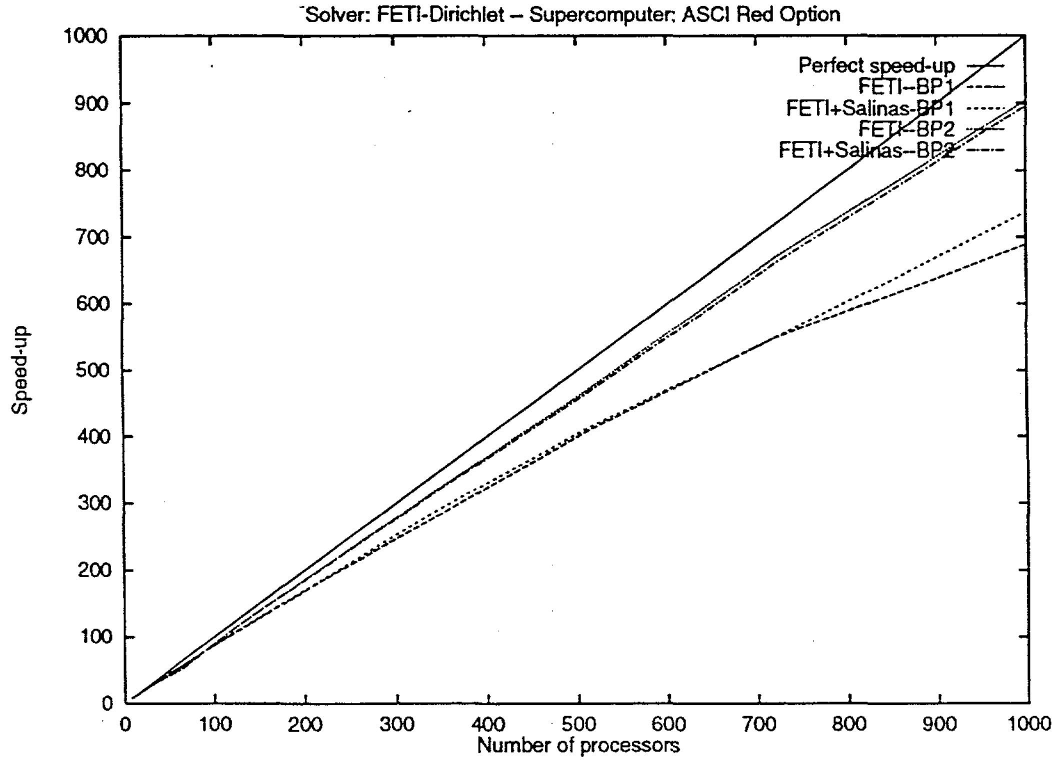

Fig. 5. Scalability results of the FETI method equipped with the Dirichlet preconditioner on the ASCI Red Option supercomputer

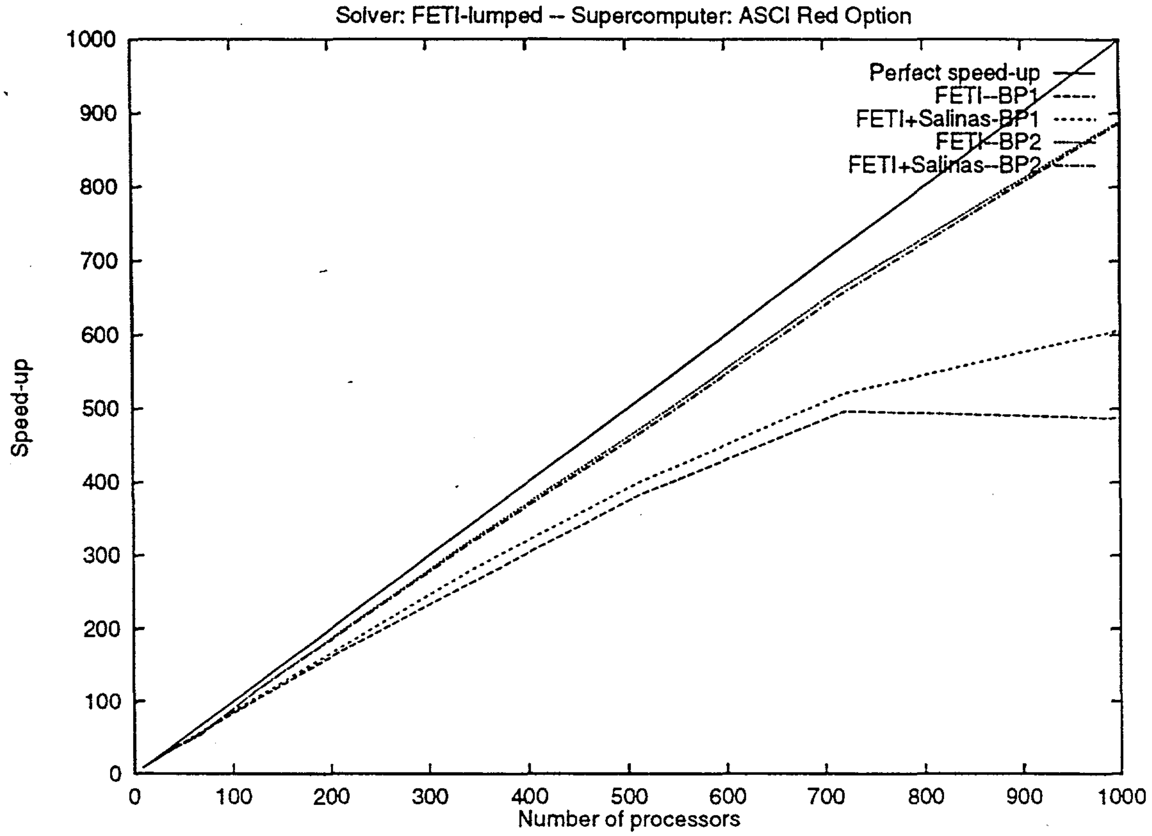

Fig. 6. Scalability results of the FETI method equipped with the lumped preconditioner on the ASCI Red Option supercomputer

From Fig. 5, we conclude that when equipped with the Dirichlet preconditioner, the one-level FETI method implemented on a 1000-processor configuration of the ASCI Red Option machine, as described in Section 3.1 can be expected to achieve for realistic second-order elasticity problems a speed-up in the range of 700 to 900, and therefore an efficiency (speed-up per processor) ranging between 70% and 90%. The lower bound of this trend for efficiency suggests that for $N_p > 1000$, maintaining this level of speed-up will require the parallelization of the factorization of the matrix $\mathbf{G}_I^T\mathbf{G}_I$ of the coarse problem in order to address the effect of Amdahl's law.

4. Highly heterogeneous structural problems

Benchmark problems BP1 and BP2 are homogeneous problems, and their mesh partitions are characterized by perfect subdomain aspect ratios. This explains the relatively low iteration counts reported in Tables 1–4 for their solution by the FETI method.

The structural problems targeted by the ASCI initiative are typically heterogeneous, with some material properties differing by as much as three orders of magnitude. For such problems, FETI delivers the same parallel scalability as for the benchmark problems BP1 and BP2. However, maintaining the numerical scalability of the FETI method for highly heterogeneous structural problems requires equipping it with different choices for $\mathbf{Q}$ and $\mathbf{W}^{(s)}$ than the identity and topological scaling matrices, respectively. For example, it was argued in [9] that for heterogeneous structural problems, $\mathbf{Q}$ must be chosen as a matrix that captures the different stiffnesses of the various subdomains. It was also shown in [9] that the two specific choices $\mathbf{Q}^D = \overline{\mathbf{F}}_I^{D^{-1}}$ and $\mathbf{Q}^L = \overline{\mathbf{F}}_I^{L^{-1}}$ not only meet this requirement, but also offer a computational advantage because $\mathbf{P}(\overline{\mathbf{F}}_I^{D^{-1}})\overline{\mathbf{F}}_I^{D^{-1}}\mathbf{P}^T(\overline{\mathbf{F}}_I^{D^{-1}}) = \overline{\mathbf{F}}_I^{D^{-1}}\mathbf{P}^T(\overline{\mathbf{F}}_I^{D^{-1}})$, and $\mathbf{P}(\overline{\mathbf{F}}_I^{L^{-1}})\overline{\mathbf{F}}_I^{L^{-1}}\mathbf{P}^T(\overline{\mathbf{F}}_I^{L^{-1}}) = \overline{\mathbf{F}}_I^{L^{-1}}\mathbf{P}^T(\overline{\mathbf{F}}_I^{L^{-1}})$. Subsequently, it was verified numerically in [18] that for highly heterogeneous model problems, these two specific choices of $\mathbf{Q}$ maintain indeed the numerical scalability of the FETI method with respect to both the mesh size h and subdomain size H.

A variational approach for tuning FETI to the solution of heterogeneous structural problems was also proposed in [32], then simplified in [19] to provide computational efficiency. This alternative approach does not focus on the projector $\mathbf{P}$, and therefore does not affect the choice of the matrix $\mathbf{Q}$. It focuses on the scaling matrix $\mathbf{W}^{(s)}$, and proposes a "superlumped stiffness" scaling procedure rather than the topological one described in Section 2. The variational theory exposed in [19] suggests that an efficient scheme for accelerating the convergence of the FETI method applied to the solution of highly heterogeneous problems is to construct the diagonal matrix $\mathbf{W}^{(s)}$ as follows. If $\lambda(i)$ is the i-th component of the Lagrange multiplier vector viewed by subdomain $\Omega^{(s)}$ and connecting the interface displacement d.o.f. $\mathbf{u}^{(s)}(j_s)$ in $\Omega^{(s)}$ to the interface displacement d.o.f. $\mathbf{u}^{(q)}(j_q)$ in the neighboring subdomain $\Omega^{(q)}$, then $\mathbf{W}^{(s)}(i)$ is set to

$$\mathbf{W}^{(s)}(i) \;=\; \frac{k^{(q)}_{j_q j_q}}{k^{(s)}_{j_s j_s} + \sum\limits_{\Omega^{(l)} \in \mathcal{N}(\Omega^{(s)})} k^{(l)}_{j_l j_l}} \tag{21}$$

where $\mathcal{N}(\Omega^{(s)})$ denotes the set of neighbors of $\Omega^{(s)}$, $k^{(l)}_{j_l j_l}$ is the diagonal coefficient of the subdomain stiffness matrix $\mathbf{K}^{(l)}$ associated with the displacement d.o.f. $\mathbf{u}^{(l)}(j_l)$, and j_l is such that the displacement d.o.f. $\mathbf{u}^{(l)}(j_l)$ and the displacement d.o.f. $\mathbf{u}^{(s)}(j_s)$ correspond to the same displacement d.o.f. of the global finite element model. Note that if all the subdomains in $\mathcal{N}(\Omega^{(s)})$ have the same material properties as subdomain $\Omega^{(s)}$, then $\mathbf{W}^{(s)}(i) = 1/m$ (see Section 2), which shows that the topological scaling overviewed in Section 2 is a particular case of the superlumped stiffness scaling summarized in Eq. (21). In [19], using a set of model problems and a few realistic ones, it was shown that the FETI method equipped with the stiffness scaling matrix $\mathbf{W}^{(s)}$ specified in Eq. (21) is numerically scalable with respect to both the problem and mesh partition sizes.

The superlumped stiffness scaling (21) does not increase neither the computational complexity nor the storage requirements of the FETI method by any significant amount. Therefore, it can be invoked by default. On the other hand, equipping the FETI method with $\mathbf{Q}^D = \overline{\mathbf{F}}_I^{D^{-1}}$ or $\mathbf{Q}^L = \overline{\mathbf{F}}_I^{L^{-1}}$ increases the computational complexity and storage requirements of the projection steps in FETI by a small percentage. Hence, a first objective of this section is to investigate when and whether equipping the FETI method not only with the stiffness scaling procedure (21) but also with a matrix $\mathbf{Q} \neq \mathbf{I}$ is worthwhile.

Furthermore, since both the $\mathbf{Q}$- and $\mathbf{W}^{(s)}$-approach address in an explicit manner only the structural heterogeneities viewed by the subdomain interfaces, a second objective of this section is to investigate whether the subdomain interfaces should include all the mesh boundaries separating the different materials of a heterogeneous finite element model, which would affect the mesh partitioning strategy.

Finally, a third objective of this section is to devise a general strategy for optimizing the solution of highly heterogeneous structural problems by the FETI method.

4.1. Findings and recommendations

In our numerous experimentations with the solution of heterogeneous problems by the FETI method, we have observed the following

O1) in addition to $\mathbf{Q}^D$ and $\mathbf{Q}^L$, the following matrix should be considered

$$\mathbf{Q}^{SL} = \sum_{s=1}^{N_s} \mathbf{W}^{(s)} \mathbf{B}^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & \text{diag}(\mathbf{K}_{bb}^{(s)}) \end{bmatrix} \mathbf{B}^{(s)^T} \mathbf{W}^{(s)} \qquad (22)$$

This matrix is a diagonal (lumped) approximation of the lumped preconditioner. Hence, we refer to it as the superlumped matrix $\mathbf{Q}^{SL}$, which explains the SL superscript. It can be shown that equipping the projector $\mathbf{P}$ with the superlumped matrix $\mathbf{Q}^{SL}$ can be interpreted as introducing the superlumped stiffness scaling procedure (21) in the solution of the coarse problem. This matrix is inexpensive to compute and store, and is such that $\mathbf{G}_I^T\mathbf{Q}^{SL}\mathbf{G}_I$ has the same sparsity pattern — and therefore the same memory requirements — as $\mathbf{G}_I^T\mathbf{G}_I$.

O2) if *all* the subdomain interfaces separate regions with similar high jumps in the material properties, then the FETI method equipped with the superlumped stiffness scaling procedure (21) performs well, and converges even faster than when the problem is homogeneous. We have observed this behavior of FETI even for mesh partitions with poor subdomain aspect ratios.

O3) if *some* but not all of the subdomain interfaces separate regions with similar high jumps in the material properties, then FETI exhibits a good convergence when equipped with the superlumped stiffness scaling procedure (21) and $\mathbf{Q} \neq \mathbf{I}$, and when the mesh partition has good subdomain aspect ratios. By $\mathbf{Q} \neq \mathbf{I}$, we mean here $\mathbf{Q} = \mathbf{Q}^D$, or $\mathbf{Q} = \mathbf{Q}^L$, or $\mathbf{Q} = \mathbf{Q}^{SL}$.

Based on these observations, some of which are illustrated in the next section, we make the following recommendations for the solution of highly heterogeneous structural problems by the FETI method

R1) by default, use the superlumped stiffness scaling matrix (21), and equip FETI's projector $\mathbf{P}$ with $\mathbf{Q}^{SL}$ (10). This improves the convergence of FETI at almost zero additional storage and computational cost.

R2) if possible, design a mesh partition where all subdomain interfaces are along boundaries between materials with similar jumps in their properties. Unfortunately, this may be possible only for cyclic structures, or academic problems.

R3) in the general case, partition the mesh along the material boundaries, then refine the obtained partition to generate the target number of subdomains N_s. If, because of topological reasons, this process can be expected to create subdomains with poor aspect ratios, modify this strategy as follows. First, re-organize the material groups into a smaller number of clusters each containing materials with relatively similar properties. Then, decompose the mesh along the boundaries of the clusters, and refine the obtained mesh partition to generate the desired number N_s of balanced subdomains. If needed for ensuring good subdomain aspect ratios, include in a cluster a neighboring material even if it has significantly different properties.

4.2. Justifications

In order to highlight the importance of the recommendations formulated above, we consider here the stress analysis of the heterogeneous cantilever structure shown in Fig. 7. This structure has a length $L_x = 4$, a depth $L_y = 1$, a thickness $th = 0.01$, and a Poisson ration $\nu = 0.3$. It is constructed by gluing together 8 slices of 3 different materials M_i in the following sequence: M_1, M_2, M_3, M_2, M_1, M_2, M_3, M_2 (Fig. 7). The Young moduli E_i of the materials M_i are such that

$$E_1 = 1000 \times E_2 \quad \text{and} \quad E_2 = 100 \times E_3 = 2.05 \times 10^{11} \tag{23}$$

Hence, $E^{max}/E^{min} = 10^5$.

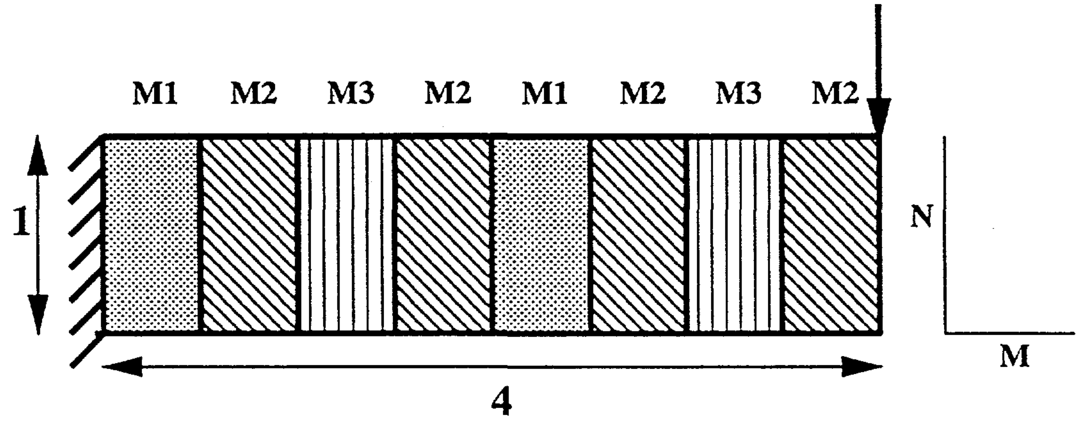

Fig. 7. A heterogeneous cantilever problem

We uniformly discretize this structure by 80×20 plane stress elements, and generate several $M \times N$ mesh partitions wih different characteristics. For each mesh partition, we solve the corresponding system of equations by the FETI method equipped with the lumped and Dirichlet preconditioners, and with $\mathbf{Q} = \mathbf{I}$, $\mathbf{Q} = \mathbf{Q}^D$, $\mathbf{Q} = \mathbf{Q}^L$, and $\mathbf{Q} = \mathbf{Q}^{SL}$. In all cases, we use the superlumped stiffness scaling procedure (21). We adopt the stopping criterion (19) and report the obtained iteration counts in Table 5.

Table 5
Iteration count for FETI applied to the solution of the heterogeneous problem graphically depicted in Fig. 7

		Lumped preconditioner			Dirichlet preconditioner		
N_s	AR	**I**	$\mathbf{Q}^{SL}$	$\mathbf{Q}^{L}$	**I**	$\mathbf{Q}^{SL}$	$\mathbf{Q}^{D}$
4 (4×1)	1	18	17	17	5	5	4
8 (8×1)	1/2	23	23	23	7	7	6
16 (16×1)	1/4	43	42	41	19	17	22
16 (8×2)	1	34	21	19	22	15	15
40 (40×1)	1/10	113	112	112	82	81	81
40 (8×5)	1/2	68	37	35	53	25	27
64 (4×16)	1	66	20	19	52	14	17

From Fig. 7 and the results summarized in Table 5, the reader can check that

- for both 4×1 and 8×1 mesh partitions, all subdomain interfaces separate regions with high jumps of Young's modulus. In both cases, the FETI method performs well, particularly when equipped with the Dirichlet preconditioner, which is in agreement with the general observation O2.
- both 16×1 and 8×2 mesh partitions have the same number of subdomains. In both cases, only half the subdomain interfaces separate regions with high jumps of Young's modulus. However, the subdomains of the 8×2 mesh partition have a better aspect ratio (AR = 1) than the subdomains of the 16×1 decomposition (AR = 1/4). This explains why FETI performs better for the 8×2 mesh partition than for the 16×1 decomposition.
- similarly, both 40×1 and 8×5 mesh partitions have the same number of subdomains, same number of homogeneous interfaces, and same number of heterogeneous interfaces. However, the subdomains of the 8×5 decomposition have a better aspect ratio (AR = 1/2) than those of the 40×1 mesh partition (AR = 1/10). Consequently, FETI performs better for the 8×5 mesh partition than for the 40×1 decomposition.
- the 16×4 mesh partition is another example of a mesh partition with both homogeneous and heterogeneous subdomain interfaces. For this decomposition and $\mathbf{Q} \neq \mathbf{I}$, the FETI method performs as well as for the 8×2 mesh

partition, which is also characterized by a uniformly perfect subdomain aspect ratio.

- in all cases, the FETI method equipped with $\mathbf{Q}^{SL}$ performs almost as well as when equipped with $\mathbf{Q}^{D}$ or $\mathbf{Q}^{L}$, and in some cases it performs even better (because of variations in the initial residual). In a few cases the FETI method equipped with $\mathbf{Q} \neq \mathbf{I}$ performs much better than when equipped with $\mathbf{Q} = \mathbf{I}$. These two observations are consistent with the general observations O1 and O3.
- the Dirichlet preconditioner is needed when the mesh partition has poor subdomain aspect ratios.
- $\mathbf{Q} \neq \mathbf{I}$ is justified and needed when the mesh partition has heterogeneous crosspoints.
- when the recommendations formulated in the previous section are followed, FETI exhibits a reasonable numerical scalability with respect to the number of subdomains.

Next, we consider the case of a realistic ASCI-type heterogeneous structural problem, and illustrate in particular the importance of recommendation R3.

5. Application to the analysis of a mockup reentry vehicle

Here, we report on the performance of the FETI method applied to the finite element analysis of a mockup reentry vehicle (RV) on the ASCI Red Option supercomputer.

An RV can be expected to experience different loadings in normal and hostile environments. Its structural response during vibration is usually predicted by a modal analysis, while its shock response is usually simulated by a direct transient analysis. The predictive computation of responses at component levels requires a detailed finite element model of the full body as well as individual components.

We focus on a large-scale finite element model of a mockup RV with 330,300 elements, and 334,759 nodes. With slightly more than one million d.o.f., this model requires significant computational power, and provides a reasonable benchmark for massively parallel computational platforms. All elements of the mesh are either 8-noded brick or 6-noded wedge elements. Decomposing this mesh into subdomain with good aspect ratios is a difficult task because the RV shown in Fig. 8 has a thin wall tubular-like overall structure. Hence, the finite element model considered herein poses serious computational challenges to substructure-based methods.

There are eight different materials that are scattered within the RV model (Fig. 8), and their Young's moduli vary from 10^2 psi to 3×10^7 psi. Hence, this RV structure is a highly heterogeneous one with $E^{max}/E^{min} = 3 \times 10^5$, and therefore can be expected to challenge any iterative solver. For the same reasons described in Section 3.3, we consider here only the stress analysis of this RV model using Salinas equipped with the FETI solver.

The results of the analysis performed in Section 3.3 (see Tables 1–4) suggest that the solution by FETI of this million d.o.f. problem requires (the memories of) at least 216 processors of the ASCI Red Option supercomputer. For this reason, we consider partitioning the given RV mesh into 250 subdomains and assigning each subdomain to one processor. We also consider partitioning this mesh into 500 subdomains for computations on 500 processors, in order to provide at least one example of the parallel scalability for a fixed problem size of our current massively parallel implementation of the FETI method. More specifically, in order to illustrate recommendation R3, we consider three different mesh decomposition strategies

- partitioning the mesh as is, with particular attention to the subdomain aspect ratio using the optimizers described in [15,16].
- partitioning the mesh along its material boundaries, then refining the obtained mesh partition to generate the requested number of subdomains. In that case, the subdomain aspect ratio optimizer [15,16] is applied locally, within each material group.
- re-organizing all the material groups of the RV finite element model into two clusters and partitioning each cluster independently from the other.

Furthermore, our mesh decomposer [29] automatically post-processes each mesh partition to remove any internal mechanism generated by the partitioning algorithm, in order to allow a robust evaluation of the rigid body modes and generalized inverse of the stiffness matrix of each floating subdomain [33]. For this reason, and because of other issues associated with the clustering process, the number of generated subdomains N_s may differ from the requested number of subdomains N_s^{req}, usually by less than 5 %.

Following recommendation R1, we equip FETI with the superlumped stiffness scaling procedure (21). However, as stated earlier, the topology of the RV model shown in Fig. 8 is such that mesh partitions with good subdomain aspect ratios cannot be reasonably expected. For this reason, we employ the Dirichlet preconditioner, and equip FETI's projector $\mathbf{P}$ with the matrix $\mathbf{Q}^D$. We monitor the convergence of FETI with the stopping criterion (19), and report in Table 6 the obtained performance results.

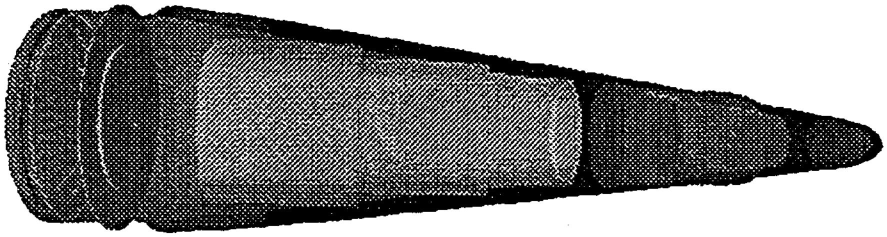

Fig. 8. Mockup reentry vehicle: each color indicates a different material

Table 6
Solution by FETI on the ASCI Red Option supercomputer of the RV problem with 1,004,277 d.o.f. (Fig. 8)

N_s^{req}	$N_p = N_s$	Type	N_{itr}	Factor $\mathbf{G}_I^T \mathbf{Q}^D \mathbf{G}_I$	FETI	Salinas + FETI
250	252	Regular	290	1.8 sec	378 sec	474 sec
250	251	Material	463	1.7 sec	563 sec	657 sec
250	257	Cluster	221	1.5 sec	261 sec	350 sec
500	513	Regular	325	8.8 sec	162 sec	252 sec
500	505	Material	584	10.6 sec	443 sec	539 sec
500	517	Cluster	276	11.6 sec	167 sec	227 sec

For both requested number of subdomains, the following trend can be observed

- the FETI method performs better on the regular mesh partition than on the material based mesh partition. This can be explained as follows. Many of the material interface boundaries run parallel to the longitudinal axis of the RV, within its thin wall structure. Consequently, each material group defines a substructure with a poor aspect ratio, and partitioning this substructure into tens of subdomains generates subdomains with even poorer subdomain aspect ratios.

- on the other hand, the FETI method performs much better for the cluster based decomposition, which highlights the relevance of recommendation R3. Again, we remind the reader that this notion of clustering is motivated here by topological reasons and the objective of generating subdomains with as good an aspect ratio as possible.

For the cluster based mesh decompositions, the performance results reported in Table 6 demonstrate a reasonable numerical scalability of the FETI method for this highly heterogeneous problem. They also show that the CPU time of the FETI method is reduced by a factor equal to 1.6 when the number of processors is increased from 257 to 517. This corresponds to an efficiency of 80%, which demonstrates a good parallel scalability of our implementation of FETI on the ASCI Red Option supercomputer.

Finally, from Table 1 and Table 6, the reader can observe that the performance of the FETI method for this heterogeneous RV problem is consistent with that of the BP1 problem with 216 subdomains.

6. Closure

Conclusions and future work.

Acknowledgements

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under Contract No. DE-AC04-94AL85000. Charbel Farhat acknowledges partial support by the Sandia National Laboratories under Contract No. BD-2435 Michel Lesoinne and Daniel Rixen acknowledge partial support by the Department of Energy under Award No. B347880/W-740-ENG-48. Kendall Pierson acknowledges the support by the National Science Foundation under Grant No. GE49355046. All authors thank their colleagues Ken Alvin, David Martinez, John RedHorse, and Garth Reese at the Sandia National Laboratories for their various contributions to this effort.

References

[1] S. F. McCormick, Multilevel adaptive methods for partial differential equations, Frontiers in Applied Mathematics, SIAM, 1989.

[2] S. F. McCormick, ed., Multigrid methods, Frontiers in Applied Mathematics, SIAM, 1987.

[3] W. Briggs, A multigrid tutorial,SIAM, 1987.

[4] P. Vanek, J. Mandel and M. Brezina, Algebraic multigrid on unstructured meshes, Computing 56, 179-196 (1996).

[5] P. LeTallec, Domain-decomposition methods in computational mechanics, Computational Mechanics Advances 1, 121-220 (1994).

[6] C. Farhat, A Lagrange multiplier based divide and conquer finite element algorithm, J. Comput. Sys. Engrg. 2, 149-156 (1991).

[7] C. Farhat and F. X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, Internat. J. Numer. Meths. Engrg. 32, 1205-1227 (1991).

[8] C. Farhat, J. Mandel and F. X. Roux, Optimal convergence properties of the FETI domain decomposition method, Comput. Meths. Appl. Mech. Engrg. 115, 367-388 (1994).

[9] C. Farhat and F. X. Roux, Implicit parallel processing in structural mechanics, Computational Mechanics Advances 2, 1-124 (1994).

[10] C. Farhat, P. S. Chen and J. Mandel, A scalable Lagrange multiplier based domain decomposition method for implicit time-dependent problems, Internat. J. Numer. Meths. Engrg. 38, 3831-3858 (1995).

[11] C. Farhat and J. Mandel, The two-level FETI method for static and dynamic plate problems - part I: an optimal iterative solver for biharmonic systems, Comput. Meths. Appl. Mech. Engrg. 155, 129-152 (1998).

[12] C. Farhat, P. S. Chen, J. Mandel and F. X. Roux, The two-level FETI method - Part II: extension to shell problems, parallel implementation and performance results, Comput. Meths. Appl. Mech. Engrg. 155, 153-180 (1998).

[13] C. Farhat, P. S. Chen and P. Stern, Towards the ultimate iterative substructuring method: combined numerical and parallel scalability, and multiple load cases, J. Comput. Sys. Engrg. 5, 337-350 (1995).

[14] M. Lesoinne and K. Pierson, An efficient FETI implementation on distributed shared memory machines with independent numbers of subdomains and processors, Contemporary Mathematics 218, 318-324 (1998).

[15] C. Farhat, N. Maman and G. Brown, Mesh partitioning for implicit computations via iterative domain decomposition: impact and optimization of the subdomain aspect ratio, Internat. J. Numer. Meths. Engrg. 38, 989-1000 (1995).

[16] D. Vanderstraeten, C. Farhat, P. S. Chen, R. Keunings and O. Zone, A retrofit and contraction based methodology for the fast generation and optimization of mesh partitions: beyond the minimum interface size criterion, Comput. Meths. Appl. Mech. Engrg. 133, 25-45 (1996).

[17] R. Diekmann, B. Meyer and B. Monien, Parallel decomposition of unstructured FEM-Meshes, Concurrency: Practice and Experience 10, 53-72 (1998).

[18] D. Rixen, Substructuring and dual methods in structural analysis, Ph.D. Thesis, Université de Liège, Belgium, Collection des Publications de la Faculté des Sciences appliquées, n° 175, 1997.

[19] D. Rixen and C. Farhat, A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems, Internat. J. Numer. Meths. Engrg. 44, 489-516 (1999).

[20] C. Farhat and P. S. Chen, Tailoring domain decomposition methods for efficient parallel coarse grid Solution and for systems with many right hand sides, Contemporary Mathematics 180, 401-406 (1994).

[21] C. Farhat, L. Crivelli and F. X. Roux, Extending substructure based iterative solvers to multiple load and repeated analyses, Comput. Meths. Appl. Mech. Engrg. 117, 195-200 (1994).

[22] J. Mandel and R. Tezaur, On the convergence of a substructuring method with Lagrange multipliers, Numerische Mathematik 73, 473-487 (1996).

[23] R. Glowinski, G. H. Golub, G. A. Meurant, and Jacques Périaux, eds., First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA, USA, SIAM, 1988.

[24] ASCI Blue Mountain web site, http://www.lanl.gov/projects/asci/bluemtn.

[25] S. L. Lillevik, The Touchstone 30 Gigaflop DELTA prototype, in Proceedings of the Sixth Distributed Memory Computer Conference, p. 671, IEEE Computer Society Press, 1991.

[26] B. Traversat, B. Nitzberg and S. Fineberg, Experience with SUNMOS on the Paragon XP/S-15, in Proceedings of the Intel Supercomputer Users' Meeting, San Diego, 1994.

[27] L. Shuler, R. Riesen, C. Jong, D. van Dresser, A. Maccabe, L. A. Fisk, and T. M. Stallcup, The Puma operating system for massively parallel computers, in Proceedings of the Intel Supercomputer North America Users' Conference, 1995.

[28] S. Garg, R. Godley, R. Griffths, A. Pfiffer, T. Prickett, D. Robboy, S. Smith, T. M. Stallcup, and S. Zeisset, Achieving large scale parallelism through operating system resource management on the Intel TFLOPS supercomputer, Intel Technology Journal, Q1'98 issue, 1998.

[29] C. Farhat, S. Lanteri and H. D. Simon, TOP/DOMDEC, A software tool for mesh partitioning and parallel processing, J. Comput. Sys. Engrg. 6, 13-26 (1995).

[30] B. Hendrickson and R. Leland, The Chaco user's guide: version 2.0, Tech. Report SAND94–2692, 1994.

[31] D. Rixen, C. Farhat, R. Tezaur and J. Mandel, Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver, Internat, J. Numer. Methds. Engrg., (in press).

[32] C. Farhat and D. Rixen, A new coarsening operator for the optimal preconditioning of the dual and primal domain decomposition methods: application to problems with severe coefficient jumps, in Proceedings of the Seventh Copper Mountain Conference on Multigrid Methods, N. Duane Melson, T. A. Manteuffel, S. F. McCormick and C. C. Douglas, eds., pp. 301-316 (1995)

[33] C. Farhat, K. Pierson and M. Lesoinne, The second generation of FETI methods and their application to the parallel solution of large-scale linear and geometrically nonlinear structural analysis problems," Comput. Meths. Appl. Mech. Engrg., (in press).