



ELSEVIER

Comput. Methods Appl. Mech. Engrg. 115 (1994) 365-385

Computer methods
in applied
mechanics and
engineering

English. They
of references
ptance of an
emination of

Optimal convergence properties of the FETI domain decomposition method

Charbel Farhat^{a,*}, Jan Mandel^b, Francois Xavier Roux^c

^a Department of Aerospace Engineering Sciences and Center for Space Structures and Controls, University of Colorado at Boulder, Boulder, CO 80309-0429, USA

^b Center for Computational Mathematics, University of Colorado at Denver, Denver, CO 80217-3364, USA

^c O.N.E.R.A., Groupe Calcul Parallèle, 29 Av. de la Division Leclerc, BP 72, 92322 Chatillon Cedex, France

he paper.

laser printer.
ction by the
talics, where
ze of a figure
priate places
rge if, in the
prepared to

occur for the

es should be

Editors and

Abstract

The Finite Element Tearing and Interconnecting (FETI) method is a practical and efficient domain decomposition (DD) algorithm for the solution of self-adjoint elliptic partial differential equations. For large-scale structural problems discretized with shell and beam elements, this method was found to outperform popular iterative algorithms and direct solvers on both serial and parallel computers, and to compare favorably with leading DD methods. In this paper, we discuss some numerical properties of the FETI method that were not addressed before. In particular, we show that the mathematical treatment of the floating subdomains and the specific conjugate projected gradient algorithm that characterize the FETI method are equivalent to the construction and solution of a coarse problem that propagates the error globally, accelerates convergence, and ensures a performance that is independent of the number of subdomains. We also show that when the interface problem is optimally preconditioned and the mesh is partitioned into well structured subdomains with good aspect ratios, the performance of the FETI method is also independent of the mesh size. However, we also argue that the FETI and other leading DD methods for unstructured problems lose in practice these scalability properties when the mesh contains junctures with rotational degrees of freedom, or the decomposition is irregular and characterized by arbitrary subdomain aspect ratios. Finally, we report that for realistic problems, optimal preconditioners are not necessarily computationally efficient and can be outperformed by non-optimal ones.

ectronic
oded in
on how

or. The
with no

number) in
files to

vier.nl.

ies) a year.
ptions are
s where air
nal p.p.h.),
f the world
ion date of

1. Introduction

The Finite Element Tearing and Interconnecting (FETI) method introduced by Farhat and Roux [1] is a practical and efficient domain decomposition (DD) method for the parallel solution of self-adjoint elliptic partial differential equations. A given spatial domain is partitioned into *non-overlapping* subdomains where an incomplete solution for the primary field is first evaluated using a direct solver. Next, intersubdomain field continuity is enforced via Lagrange multipliers applied at the subdomain interfaces. This "gluing" phase generates a smaller size symmetric *dual* problem where the unknowns are the Lagrange multipliers, and which is best solved with a preconditioned conjugate gradient (PCG) algorithm. Each iteration of the PCG algorithm requires the solution of independent subdomain problems. For static structural analysis, every floating subdomain—that is, a subdomain with Neumann boundary conditions only—is associated with a singular stiffness matrix and generates a set of interface constraints. Consequently, the system of equations governing the dual interface problem is in general indefinite. The FETI algorithm deals with both issues by incorporating

1000 AE

will ensure

nal use, or
ated in the
ropriate fee
AA 01970,
y from the
loss of the
promotion

* Corresponding author.

in the solution the contribution of the subdomain rigid body modes and by solving the indefinite interface problem with a preconditioned conjugate projected gradient algorithm (PCPG).

So far, we have applied the FETI method to the solution of three-dimensional structural problems discretized essentially with beam and shell elements. For such problems, we have shown [2] that the FETI algorithm compares favorably with leading DD methods such as the Neumann–Neumann algorithm [3,4]. We have also shown that even when the global stiffness matrix can be assembled and stored in real memory, the FETI method often outperforms optimized direct solvers on both serial and parallel/vector processors [1]. However, whereas theoretical results related to the condition number of the interface problem have been proved for many DD methods—often for a model problem described by the Laplace operator—we have not conducted in the past any theoretical or numerical investigation of the behavior of the condition number of the dual interface problem associated with the FETI method. Such an analysis is usually performed to establish the scalability or non-scalability of a DD algorithm. The presence or absence of a coarse grid problem is one of several fundamental criteria to classify DD methods [5]. Algorithms lacking a coarse problem are popular because they are perfectly parallel when load balanced. However, such algorithms lack a mechanism for exchanging information between all subdomains in the preconditioning step and thus for preventing the growth of the condition number with the number of subdomains. Therefore, such methods do not scale well in the fine granularity regime targeted by emerging parallel processors. For example, it was observed that the condition number of the Neumann–Neumann algorithm without a coarse problem deteriorates with the number of subdomains, and that 16 subdomains is a practical limit [6] for this algorithm. In other DD methods, the needed error propagation was accomplished by solving in each PCG iteration a coarse problem with few degrees of freedom (d.o.f.) per subdomain [7,8]. Such methods are close in spirit to multigrid methods and especially to two-level methods such as those presented in [9–11]. Recently, Mandel [6] has proposed a computational procedure for constructing a coarse problem for the Neumann–Neumann method which, unlike a previous procedure [12], does not require to know what are the faces, edges, or vertices of the subdomains. He has called the resulting DD algorithm the Balancing DD method and has shown that its convergence is independent of the number of subdomains.

The main ideas presented by Mandel in [6] are in many ways similar to those of the FETI method. For example, the coarse problem proposed in [6] is derived from the same solvability conditions as those introduced in [1] for floating subdomains, and its representative system matrix is constructed using the same building blocks as those of the projection operator first developed in [1]. These observations have motivated us to further analyze the FETI method and establish that: (a) the projection step in the PCPG algorithm used in the FETI method leads to a “natural” coarse problem, and (b) the FETI method with the Dirichlet preconditioner is asymptotically optimal—that is, the condition number of the preconditioned dual interface problem is independent of the number of subdomains and grows only slowly when the mesh size $h \rightarrow 0$.

The remainder of the paper is organized as follows. In Section 2, we cast the FETI formulation into an elegant saddle-point problem, and derive in a new way the solution algorithm that was first proposed in [1]; we also establish that the FETI method contains inherently a coarse problem that propagates the error globally and accelerates convergence. In Section 3 we prove new bounds on the convergence of the FETI algorithm without preconditioning, and in Section 4 we show numerically that it is asymptotically scalable when the interface problem is optimally preconditioned and the mesh partitions have good subdomain aspect ratios. In Section 5, we show that scalability is lost in practice for problems with junctures, and/or mesh partitions with arbitrary subdomain aspect ratios. We also argue that the same conclusion applies to other optimal DD algorithms that are applicable to unstructured meshes. In Section 6, we address some computational issues and conclude that the optimal Dirichlet preconditioner is not necessarily computationally efficient. In Section 7, we discuss the superconvergence properties of the FETI algorithm and justify the use of the non-optimal local preconditioner first advocated in [1]. Finally in Section 8 we offer some concluding remarks.

For a background on the FETI method, we refer the reader to Refs. [1,2,13].

2. A saddle-point problem and its solution algorithm

Let Ω denote the volume of the structure to be analyzed, K its stiffness matrix, and f the vector of prescribed forces. The problem to be solved is:

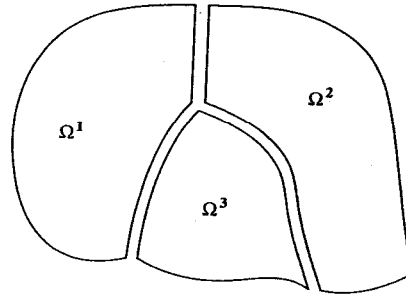


Fig. 1. Domain partitioning.

$$Ku = f, \tag{1}$$

where u is the displacement of the structure caused by the forces f . We partition Ω into N_s substructures or subdomains $\{\Omega^s\}_{s=1, \dots, N_s}$ (Fig. 1), and we denote by K^s , u^s , f^s and B^s , respectively the subdomain stiffness matrix, the subdomain displacement and force vectors, and the signed matrix with entries $-1, 0, +1$ describing the subdomain interconnectivity.

The Lagrangian associated with the partitioned problem is:

$$\begin{aligned} \mathcal{L}(v, \mu) &= \frac{1}{2} \sum_{s=1}^{N_s} v^{sT} K^s v^s - \sum_{s=1}^{N_s} v^{sT} f^s + \mu^T \sum_{s=1}^{N_s} B^s v^s \\ &= \frac{1}{2} \sum_{s=1}^{N_s} v^{sT} K^s v^s - \sum_{s=1}^{N_s} v^{sT} (f^s - B^{sT} \mu), \end{aligned} \tag{2}$$

where $v = (v^1, v^2, \dots, v^{N_s})$ is a vector of subdomain displacements, μ is a vector of Lagrange multipliers introduced at the subdomain interfaces, $u = (u^1, u^2, \dots, u^{N_s})$, and $f = (f^1, f^2, \dots, f^{N_s})$. We define two energy functionals $\mathcal{E}(v)$ and $\mathcal{C}(\mu)$ as follows:

$$\mathcal{E}(v) = \sup_{\mu} \mathcal{L}(v, \mu), \quad \mathcal{C}(\mu) = \inf_v \mathcal{L}(v, \mu). \tag{3}$$

It is well-known [14] that solving (1) is equivalent to solving the saddle-point problem:

$$\text{Find } (u, \lambda) \text{ such that } \mathcal{L}(u, \lambda) = \inf_v \sup_{\mu} \mathcal{L}(v, \mu) = \inf_v \mathcal{E}(v) \tag{4}$$

or its dual problem:

$$\text{Find } (u, \lambda) \text{ such that } \mathcal{L}(u, \lambda) = \sup_{\mu} \inf_v \mathcal{L}(v, \mu) = \sup_{\mu} \mathcal{C}(\mu). \tag{5}$$

Therefore, the solution (u, λ) minimizes $\mathcal{E}(v)$, maximizes $\mathcal{C}(\mu)$, and is such that $\mathcal{E}(u) = \mathcal{C}(\lambda)$.

In problem (4), the displacement v is admissible if and only if $\mathcal{E}(v)$ is finite, and in problem (5), the Lagrange multipliers μ are admissible if and only if $\mathcal{C}(\mu)$ is finite. Moreover, it can be easily shown that:

$$\mathcal{E}(v) < +\infty \Leftrightarrow \sum_{s=1}^{N_s} B^s v^s = 0, \tag{6}$$

$$\mathcal{C}(\mu) > -\infty \Leftrightarrow f^s - B^{sT} \mu \perp \text{Ker}(K^s). \tag{7}$$

The right hand side of the equivalence relation (6) expresses the continuity of the displacement field at the subdomain interfaces. The right hand side of the equivalence relation (7) is a necessary condition for the existence of a displacement solution in a floating subdomain where the stiffness matrix is singular. Note that if Ω^s is a non-floating subdomain—that is, with enough Dirichlet boundary conditions to eliminate its rigid body motions— $\text{Ker}(K^s) = \{0\}$ and condition (7) is automatically satisfied.

In summary, the solution of the saddle-point problem (4) can be written as follows:

$$\boxed{\begin{aligned} \frac{\partial \mathcal{L}(v, \mu)}{\partial v^s} \Big|_{(u^s, \lambda)} &= 0 \Leftrightarrow K^s u^s + B^{sT} \lambda = f^s \\ \frac{\partial \mathcal{L}(v, \mu)}{\partial \mu} \Big|_{(u^s, \lambda)} &= 0 \Leftrightarrow \sum_{s=1}^{N_s} B^s u^s = 0. \end{aligned}} \quad (8)$$

REMARK 2.1. The first of Eqs. (8) implies that $f^s - B^{sT} \lambda \perp \text{Ker}(K^s)$.

The FETI method is now derived as the conjugate gradient method for the maximization of the energy functional $\mathcal{C}(\mu)$ on the admissible set:

$$\mathcal{A} = \{\mu : \mathcal{C}(\mu) > -\infty\}. \quad (9)$$

From (7) it follows that if $\mathcal{C}(\mu) > -\infty$, then $\mathcal{C}(\mu + \delta) > -\infty$ if and only if $\delta \in \mathcal{V}$, where:

$$\mathcal{V} = \{\delta : B^s \delta \perp \text{Ker}(K^s), s = 1, \dots, N_s\}. \quad (10)$$

For an admissible $\mu \in \mathcal{A}$ and a direction $\delta \in \mathcal{V}$, the directional derivative $D(\mathcal{C}(\mu); \delta)$ can be computed using Eqs. (8) as follows:

$$D(\mathcal{C}(\mu); \delta) = \frac{\partial \mathcal{C}}{\partial t}(\mu + t\delta) \Big|_{t=0} = \delta^T \sum_{s=1}^{N_s} B^s K^{s+} (f^s - B^{sT} \mu), \quad (11)$$

where K^{s+} is a generalized inverse of K^s , and which does not need to be explicitly computed [1, Appendix I]. Clearly, if Ω^s is not a floating subdomain, $K^{s+} = K^{s-1}$.

The directional derivative is represented by the inner product of an increment δ and a gradient vector $\nabla \mathcal{C}(\mu)$ in \mathcal{V} as follows:

$$\forall \delta \in \mathcal{V}, \quad D(\mathcal{C}(\mu), \delta) = \delta^T \nabla \mathcal{C}(\mu). \quad (12)$$

Using Eqs. (8) and imposing that the gradient vector be in \mathcal{V} leads to expressing $\nabla \mathcal{C}(\mu)$ as the orthogonal projection onto \mathcal{V} of the sum in Eq. (11):

$$\nabla \mathcal{C}(\mu) = P \sum_{s=1}^{N_s} B^s K^{s+} (f^s - B^{sT} \mu) = P(d - F_1 \mu), \quad (13)$$

where:

$$P = I - G_1 (G_1^T G_1)^{-1} G_1^T, \quad G_1 = [B^1 R^1 \dots B^{N_f} R^{N_f}],$$

$$d = \sum_{s=1}^{N_s} B^s K^{s+} f^s, \quad F_1 = \sum_{s=1}^{N_s} B^s K^{s+} B^s,$$

and where R^s is an arbitrary matrix with linearly independent columns that span $\text{Ker}(K^s)$, and N_f is the total number of floating subdomains. Clearly, the introduction of R^s is necessary only if Ω^s is a floating subdomain, in which case the columns of R^s are linear combinations of the subdomain rigid body modes.

REMARK 2.2. There are at most 6 rigid body modes per floating subdomain and therefore G_1 contains at most $6 \times N_f$ columns.

REMARK 2.3. Unlike other approaches [4] where the subdomain rigid body modes are ignored in the preconditioning step, the FETI method incorporates these modes in the formulation of the interface problem itself. This enables a global interchange of information with the coarse space:

$$\text{Im}(G_1) = \left\{ \xi : \xi = \sum_{s=1}^{N_f} B^s R^s z^s, z^s \in \mathbb{R}^{\dim(\text{Ker}(K^s))} \right\}. \quad (14)$$

Since \mathcal{V} and $\text{Im}(G_1)$ are complementary spaces, it follows from Eqs. (7) and (10) that:

$$\forall \mu \in \mathcal{A}, \forall \xi \neq 0 \text{ and } \xi \in \text{Im}(G_1), \quad \mathcal{C}(\mu) > \mathcal{C}(\mu + \xi) = -\infty, \quad (15)$$

which implies that all $\mu \in \mathcal{A}$ and all search directions $\nabla \mathcal{C}(\mu) \in \mathcal{V}$ are optimal with respect to increments in the space $\text{Im}(G_1)$.

The generalized inverses K^{s+} can be chosen to be symmetric without changing the value of $\nabla \mathcal{C}(\mu)$, and therefore F_1 can be assumed to be symmetric without any loss of generality. Moreover, for all $\delta \in \mathcal{V}$ and $\eta \in \mathcal{V}$ we have:

$$\delta^T P F_1 \eta = \delta^T F_1 \eta = \eta^T F_1 \delta = \eta^T P F_1 \delta, \quad (16)$$

which proves that $P F_1$ is symmetric on \mathcal{V} .

Similarly, it can be shown that if \tilde{F}_1^{-1} is a symmetric preconditioner, then $P \tilde{F}_1^{-1}$ is also symmetric on \mathcal{V} . Hence, the preconditioned conjugate projected gradient algorithm proposed in [1] can be obtained simply by choosing an initial approximation of λ in \mathcal{A} and projecting all residuals and search direction onto \mathcal{V} as follows:

<p>1. Initialize:</p> $\lambda^{(0)} = G_1(G_1^T G_1)^{-1} [f^1 R^1 \dots f^{N_f} R^{N_f}]^T,$ $r^{(0)} = d - F_1 \lambda^{(0)};$ <p>2. Iterate $k = 1, 2, \dots$ until convergence:</p> <p>Project:</p> $w^{(k-1)} = [I - G_1(G_1^T G_1)^{-1} G_1^T] r^{(k-1)};$ <p>Precondition:</p> $z^{(k-1)} = \tilde{F}_1^{-1} w^{(k-1)};$ <p>Re-project:</p> $y^{(k-1)} = [I - G_1(G_1^T G_1)^{-1} G_1^T] z^{(k-1)},$ $\zeta^{(k)} = y^{(k-1)T} w^{(k-1)} / y^{(k-2)T} w^{(k-2)} \quad (\zeta^{(1)} = 0),$ $p^{(k)} = y^{(k-1)} + \zeta^{(k)} p^{(k-1)} \quad (p^{(1)} = y^{(0)}),$ $\nu^{(k)} = y^{(k-1)T} w^{(k-1)} / p^{(k)T} F_1 p^{(k)},$ $\lambda^{(k)} = \lambda^{(k-1)} + \nu^{(k)} p^{(k)},$ $r^{(k)} = r^{(k-1)} - \nu^{(k)} F_1 p^{(k)}.$	(17)
--	------

In [13], it is shown that $G_1^T G_1$ couples all of the floating subdomains but retains a banded structure. Moreover, $G_1^T G_1$ is at most $6N_f \times 6N_f$ and therefore can be easily stored in memory. Clearly, at each iteration of the above PCPG algorithm, the *Project* and *Re-project* steps disseminate numerical information among all of the subdomains, and therefore accelerate convergence.

3. Convergence analysis and scalability

The PCPG algorithm (17) is identical to the standard PCG algorithm but applied to the system operator $P F_1 : \mathcal{V} \rightarrow \mathcal{V}$ and the preconditioner $P \tilde{F}_1^{-1} : \mathcal{V} \rightarrow \mathcal{V}$. Therefore, it admits the standard error reduction bound [15]:

$$\|e^{(k)}\|_{F_1} \leq \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2k} \|e^{(0)}\|_{F_1}, \quad (18)$$

where $e^{(k)}$ is the error at step k , $\|\mu\|_{F_1}^2 = \mu^T F_1 \mu$, $\mu \in \mathcal{V}$, is the energy norm on the space of Lagrange multipliers \mathcal{V} , and κ is the condition number of $P \tilde{F}_1^{-1} P F_1$. Here, we show that for $\tilde{F}_1^{-1} = I$ (unpreconditioned case) this condition number can be bounded independently of the number of subdomains N_s .

THEOREM 3.1. Let c_0 denote the maximum number of subdomains that share a degree of freedom. If there exist two constants $c_1 > 0$ and c_2 such that for all v^s satisfying $v^s \in \text{Im}(B^{sT})$ and $v^s \perp \text{Ker}(K^s)$ we have:

$$c_1 v^{sT} v^s \leq v^{sT} K^{s+} v^s \leq c_2 v^{sT} v^s, \quad (19)$$

then:

$$\kappa \leq \frac{c_0 c_2}{c_1}. \quad (20)$$

PROOF. By summation over all subdomains Ω^s we have:

$$\forall \mu \in \mathcal{V}, \quad c_1 \mu^T \sum_{s=1}^{N_s} B^s B^{sT} \mu \leq \mu^T F_1 \mu \leq c_2 \mu^T \sum_{s=1}^{N_s} B^s B^{sT} \mu. \quad (21)$$

On the other hand, from the definition of B^s and the symmetry of $\sum_s B^{(s)} B^{(s)T}$ it follows that:

$$I \leq \sum_{s=1}^{N_s} B^s B^{sT} \leq c_0 I. \quad (22)$$

Finally, from (21) and (3) we deduce that:

$$\forall \mu \in \mathcal{V}, \quad c_1 \mu^T \mu \leq \mu^T F_1 \mu \leq c_0 c_2 \mu^T \mu, \quad (23)$$

which shows that the Rayleigh quotient $\mu^T F_1 \mu / \mu^T \mu$ is bounded from below by c_1 and from above by $c_0 c_2$, and therefore the condition number κ is bounded by $c_0 c_2 / c_1$. \square

Next, we partition the subdomain unknowns into internal degrees of freedom designated by the subscript i , and interface boundary degrees of freedom designated by the subscript b . Hence, a subdomain vector v^s can be written as:

$$v^s = \begin{bmatrix} v_i^s \\ v_b^s \end{bmatrix}. \quad (24)$$

Note that if $v^s \in \text{Im}(B^{sT})$, then $v_i^s = 0$. The evaluation of the matrix vector product $q^s = K^{s+} v^s$ can be performed via the solution of the following system of equations:

$$\begin{bmatrix} K_{ii}^s & K_{ib}^s \\ K_{ib}^{sT} & K_{bb}^s \end{bmatrix} \begin{bmatrix} q_i^s \\ q_b^s \end{bmatrix} = \begin{bmatrix} 0 \\ v_b^s \end{bmatrix}, \quad (25)$$

which leads to the reduced system:

$$S_{bb}^s q_b^s = v_b^s, \quad (26)$$

where S_{bb}^s is the Schur complement matrix given by:

$$S_{bb}^s = K_{bb}^s - K_{ib}^{sT} K_{ii}^{s-1} K_{ib}^s. \quad (27)$$

From Eqs. (25)–(28) it follows that:

$$v^{sT} K^{s+} v^s = q_b^{sT} v_b^s = v_b^{sT} S_{bb}^{s+} v_b^s \quad (28)$$

and therefore the inequalities (19) are equivalent to:

$$\forall v_b^s \perp \text{Ker}(S_{bb}^s), \quad \frac{1}{c_2} v_b^{sT} v_b^s \leq v_b^{sT} S_{bb}^s v_b^s \leq \frac{1}{c_1} v_b^{sT} v_b^s. \quad (29)$$

Let H and h denote respectively the characteristic subdomain and element sizes. Decreasing H corresponds to increasing the number of subdomains—for example, to increase the degree of parallelism and the number of processors—and decreasing h corresponds to refining the finite element mesh. Since for Poisson and elasticity problems the quadratic form $v_b^{sT} S_{bb}^s v_b^s$ is equivalent to the square of the trace norm [7,16], we have that the inequalities (29) are satisfied with:

$$\frac{c_2}{c_1} \leq \text{const} \frac{H}{h}. \quad (30)$$

Thus, from (29) and (30), we obtain the following theorem:

THEOREM 3.2. *If the elements and the subdomains have regular shape and size, then the FETI method without a preconditioner applied to the Poisson or elasticity problems has a condition number that is bounded by:*

$$\kappa \leq \text{const} \frac{H}{h}. \quad (31)$$

REMARK 3.3. The above estimate of the condition number can be expected to hold also for the preconditioned FETI method, for example, when the preconditioner is spectrally equivalent to a multiple of the identity matrix. The “lumping” preconditioner introduced in Section 6 verifies this property.

Theorem 3.2 establishes the scalability of the unpreconditioned FETI domain decomposition method. When the mesh size is fixed and the number of subdomains is increased, H and the condition number of the interface problem decrease. When the mesh is refined and the number of subdomains is increased such that H/h is kept constant, the condition number of the interface problem remains constant. This is unlike several other unpreconditioned domain decomposition methods where the number of iterations for convergence grows with the number of subdomains.

4. Numerical investigation of asymptotic optimality

4.1. Preconditioning with primal subdomain operators

The mechanical interpretation of the FETI method is straightforward. At each iteration k of the conjugate projected gradient algorithm (17), the matrix–vector product:

$$\bar{p}_b^{(k)} = F_1 p_b^{(k)} = \sum_{s=1}^{N_s} B^s K^{sT} B^{sT} p_b^{(k)} \quad (32)$$

can be assembled from the following subdomain level results:

$$\bar{p}^{s(k)} = K^{sT} B^{sT} p_b^{s(k)}. \quad (33)$$

Each of the above local matrix–vector products (32) can be re-formulated as a subdomain problem with Neumann boundary conditions (indicated here between braces $\{ \}$) as follows:

$$\begin{bmatrix} K_{ii}^s & K_{ib}^s \\ K_{ib}^{sT} & K_{bb}^s \end{bmatrix} \begin{bmatrix} \bar{p}_i^{s(k)} \\ \bar{p}_b^{s(k)} \end{bmatrix} = B^{sT} p_b^{s(k)} = \begin{bmatrix} 0 \\ \{q_b^{s(k)}\} \end{bmatrix}. \quad (34)$$

Hence, at each iteration of the FETI method, traction forces are applied at the subdomain interfaces and a corresponding jump in the displacement field is evaluated. From a mechanical viewpoint, the inverse problem

consists in imposing this jump at the subdomain interfaces and computing the corresponding interface traction forces. At the subdomain level, this leads to a primal Dirichlet problem that can be written as:

$$\begin{bmatrix} K_{ii}^s & K_{ib}^s \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{\omega}_i^{s(k)} \\ \{\bar{\omega}_b^{s(k)}\} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{\omega}_b^{s(k)} \end{bmatrix} \quad (35)$$

and whose solution is given by:

$$\bar{\omega}_i^{s(k)} = -K_{ii}^{s-1} K_{ib}^s \bar{\omega}_b^{s(k)}. \quad (36)$$

The corresponding traction forces are then obtained via the following matrix–vector multiplication:

$$\begin{bmatrix} K_{ii}^s & K_{ib}^s \\ K_{ib}^{sT} & K_{bb}^s \end{bmatrix} \begin{bmatrix} \bar{\omega}_i^{s(k)} \\ \{\bar{\omega}_b^{s(k)}\} \end{bmatrix} = \begin{bmatrix} 0 \\ (K_{bb}^s - K_{ib}^{sT} K_{ii}^{s-1} K_{ib}^s) \{\bar{\omega}_b^{s(k)}\} \end{bmatrix}. \quad (37)$$

From the above mechanical interpretation, it follows that a good preconditioner for the dual interface problem can be constructed by assembling the primal subdomain operators as follows:

$$D_1^{-1} = \sum_{s=1}^{N_s} B^s \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^s - K_{ib}^{sT} K_{ii}^{s-1} K_{ib}^s \end{bmatrix} B^{sT} = \sum_{s=1}^{N_s} B^s \begin{bmatrix} 0 & 0 \\ 0 & S_{bb}^s \end{bmatrix} B^{sT}. \quad (38)$$

By analogy with the theoretical results for the closely related Balancing DD method [6], one can expect that the condition number of the FETI method with the Dirichlet preconditioner will be bounded by $(1 + \log H/h)^2$, where H/h is again the ratio of the subdomain and element sizes. Hence, one can reasonably expect that the performance of this optimal preconditioner will be weakly dependent on h .

REMARK 4.1. The FETI interface operator can also be written as:

$$F_1 = \sum_{s=1}^{N_s} B^s \begin{bmatrix} 0 & 0 \\ 0 & S_{bb}^{s+} \end{bmatrix} B^{sT}, \quad (39)$$

where S_{bb}^{s+} is a generalized inverse of S_{bb} . In the absence of cross-points—that is, points where more than two subdomains intersect— $B^s B^{sT} = I$ and the Dirichlet preconditioner becomes exactly the sum of the generalized inverses of the local FETI operators. In that case, the FETI and Neumann–Neumann [4] methods become dual methods, except for some sign differences in the definition of the B^s matrices.

Next, we confirm numerically that the condition number of the FETI interface operator with the optimal Dirichlet preconditioner grows only slowly as $h \rightarrow 0$, and that it is bounded independently of the number of subdomains.

4.2. The Poisson model problem

The Laplace operator appears in many technologically important problems such as heat transfer conduction and flow through porous media. Even though it is not representative of all structural problems, it is often used to illustrate the salient features of domain decomposition algorithms for elasticity problems.

Here, the problem to be solved is the Poisson problem:

$$\Delta \phi(x, y) = b, \quad \phi(0, y) = 0, \quad (40)$$

where $\phi(x, y)$ is some scalar field defined over a two-dimensional rectangular domain Ω , and b is a constant. The finite element discretization of Eq. (40) leads to an algebraic problem similar to that of Eq. (1). Usually, the convergence properties of a DD algorithm are discussed in terms of the subdomain size H and the element size within a subdomain h . Therefore, we consider various $M \times N$ box-wise decompositions where every subdomain is a square with a unit side length and is uniformly discretized with $1/h \times 1/h$ elements (Fig. 2).

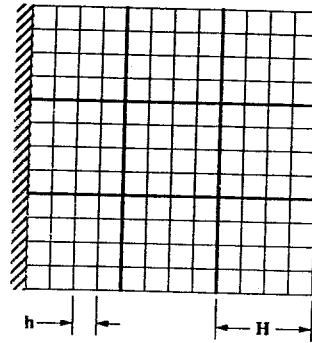


Fig. 2. Uniform mesh partitions for the Poisson model problem.

For all cases and throughout this paper we use the following global stopping criterion:

$$\frac{\|Ku - f\|_2}{\|f\|_2} < 10^{-6}. \quad (41)$$

First, we set $h = 1/10$ then $h = 1/20$, and report in Tables 1 and 2 the condition number of the preconditioned dual interface problem and the number of iterations to achieve convergence (between parenthesis) for different values of H obtained by varying M and N . The results show that for a fixed value of h and a decreasing value of H , the condition number and the number of iterations stabilize even for a number of substructures as large as 1,024 ($M = N = 32$). Clearly, these results highlight the presence of a coarse grid and demonstrate the numerical scalability of the FETI method for this model problem.

Next, we set $H = 1/4$ then $H = 1/8$ and report in Tables 3 and 4 the condition number and the number of iterations for different values of h . The results validate the claim made in Section 4.1: for a fixed value of H , the performance of the FETI method with the Dirichlet preconditioner is found to be weakly dependent on h .

Finally, we fix the size of the global problem to 102,400 elements (320×320) then 409,600 elements (640×640) and monitor the performance of the FETI method for various numbers of substructures (Tables 5 and 6). For structural engineering applications, this test is often considered to be the most meaningful one as

Table 1
FETI method with \widetilde{F}_1^{D-1} preconditioner, asymptotic scalability—Two-dimensional Poisson problem, $h = 1/10$

N	M										
	1	2	4	8	16	32	1	2	4	8	
1	—	2.0 (5)	2.1 (7)	2.1 (9)	2.1 (9)	2.9 (9)	2.8 (4)	5.0 (10)	5.5 (14)	5.8 (18)	5.5 (18)
2	2.8 (4)	5.0 (10)	5.5 (14)	5.8 (18)	5.8 (18)	5.5 (18)	3.5 (6)	5.5 (14)	5.8 (19)	5.9 (18)	5.5 (18)
4	3.5 (6)	5.5 (14)	5.8 (19)	6.0 (19)	5.9 (18)	5.5 (18)	3.6 (10)	5.9 (18)	6.0 (20)	6.0 (19)	5.7 (18)
8	3.6 (10)	5.9 (18)	6.0 (20)	6.0 (19)	5.7 (18)	5.5 (18)	3.7 (13)	6.0 (18)	6.2 (20)	6.1 (19)	5.5 (18)
16	3.7 (13)	6.0 (18)	6.2 (20)	6.1 (19)	5.6 (18)	5.5 (18)	3.7 (13)	6.1 (18)	6.2 (19)	6.2 (18)	5.6 (18)
32	3.7 (13)	6.1 (18)	6.2 (19)	6.2 (18)	6.7 (18)	5.6 (18)					

Table 2
FETI method with \widetilde{F}_1^{D-1} preconditioner, asymptotic scalability—Two-dimensional Poisson problem, $h = 1/20$

N	M										
	1	2	4	8	16	32	1	2	4	8	
1	—	2.6 (5)	2.8 (8)	2.8 (11)	2.8 (11)	2.9 (11)	3.8 (4)	5.9 (11)	6.4 (16)	6.6 (21)	6.3 (21)
2	3.8 (4)	5.9 (11)	6.4 (16)	6.6 (21)	6.7 (21)	6.3 (21)	4.5 (8)	6.5 (17)	6.8 (21)	7.0 (22)	6.3 (21)
4	4.5 (8)	6.5 (17)	6.8 (21)	7.0 (22)	6.9 (21)	6.3 (21)	4.7 (12)	6.9 (20)	7.0 (22)	7.0 (21)	6.4 (21)
8	4.7 (12)	6.9 (20)	7.0 (22)	7.0 (21)	6.4 (21)	6.3 (20)	4.8 (15)	7.2 (21)	7.2 (21)	7.2 (21)	6.4 (20)
16	4.8 (15)	7.2 (21)	7.2 (21)	7.2 (21)	6.4 (20)	6.4 (20)	4.8 (15)	7.3 (21)	7.2 (21)	7.2 (21)	6.4 (20)
32	4.8 (15)	7.3 (21)	7.2 (21)	7.2 (21)	6.4 (20)	6.4 (20)					

Table 3

FETI method with \tilde{F}_1^{D-1} preconditioner, weak dependence on h —Poisson problem on a square domain, $H = 1/4$ ($M = N = 4$, i.e. 16 substructures)

h	Condition number	Number of iterations
1/10	5.8	19
1/20	6.8	21
1/40	8.0	22
1/80	9.7	25
1/160	9.7	25

Table 4

FETI method with \tilde{F}_1^{D-1} preconditioner, weak dependence on h —Poisson problem on a square domain, $H = 1/8$ ($M = N = 8$, i.e. 64 substructures)

h	Condition number	Number of iterations
1/10	6.0	19
1/20	7.1	21
1/40	8.4	23
1/80	9.7	25
1/160	9.7	25

Table 5

FETI method with \tilde{F}_1^{D-1} preconditioner—Effect of the inherent coarse grid problem—Poisson problem on a square domain, 102,400 elements

H	N_s	Condition number	Number of iterations
1/4	16	9.7	25
1/8	64	8.4	23
1/16	256	6.4	20
1/32	1024	5.7	18

Table 6

FETI method with \tilde{F}_1^{D-1} preconditioner—Effect of the inherent coarse grid problem—Poisson problem on a square domain, 409,600 elements

H	N_s	Condition number	Number of iterations
1/4	16	9.7	25
1/8	64	8.4	23
1/16	256	6.4	20
1/32	1024	5.7	18

it represents the case where an attempt is made at increasing the amount of parallelism in the solution process without artificially inflating the number of equations.

When the global discretization is kept constant and the number of substructures is increased, additive substructuring methods lacking a coarse problem converge towards the Jacobi algorithm. Therefore, the performance of such methods deteriorates when H is decreased. However, the results reported in Tables 5 and 6 show that for a fixed global problem, the number of iterations performed by the FETI method decreases when the number of substructures is increased. These results clearly demonstrate that when H is decreased, the FETI method converges towards a direct solver and therefore is inherently numerically scalable.

REMARK 4.2. Similar results have been obtained for the three-dimensional Poisson problem.

4.2.1. Plane stress or strain problems

Two-dimensional problems in elasticity are in general either plane stress or plane strain problems and can be written as:

$$\begin{aligned}
 G \left(\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) \right) + \beta G \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x}(x, y) + \frac{\partial v}{\partial y}(x, y) \right) + X(x, y) &= 0, \\
 G \left(\frac{\partial^2 v}{\partial x^2}(x, y) + \frac{\partial^2 v}{\partial y^2}(x, y) \right) + \beta G \frac{\partial}{\partial y} \left(\frac{\partial u}{\partial x}(x, y) + \frac{\partial v}{\partial y}(x, y) \right) + Y(x, y) &= 0,
 \end{aligned} \tag{42}$$

where $u(x, y)$, $v(x, y)$, $X(x, y)$ and $Y(x, y)$ denote respectively the horizontal and vertical displacement fields and distributed body forces, ν denotes Poisson's ratio, G is Lamé's constant, and $\beta = (1 + \nu)/(1 - \nu)$ for plane state of stress and $\beta = 1/(1 - 2\nu)$ for plane state of strain [17]. Given that the first term of each of Eqs. (42) is a Laplacian operator and that the second term involves only second order partial derivatives of $u(x, y)$ and $v(x, y)$, one can reasonably expect that the behavior of an iterative algorithm will be similar for both the plane stress or strain problem and the model Poisson problem, when similar shape functions are used for both of them.

A sample plane stress problem defined on a square domain of side L and thickness t is depicted in Fig. 3. If for this problem h is kept constant and H is decreased (the number of substructures is increased), the resulting

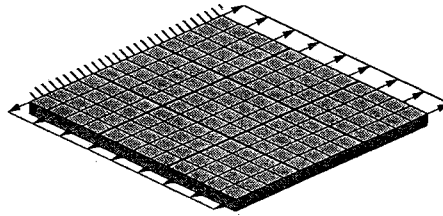


Fig. 3. A plane stress problem.

Table 7

FETI method with \tilde{F}_1^{D-1} preconditioner—Effect of the inherent coarse grid problem—Plane stress problem on a square domain, 4,704 equations

H	N_s	Condition number	Number of iterations
1/2	4	18.8	13
1/4	8	20.1	23
1/8	64	15.7	24
1/16	256	10.8	24

mesh partitions will have different t/L ratios. This is not an issue for plane stress or strain problems but it is a major one for shell problems that we investigate next using the same domain as that shown in Fig. 3 and a combined membrane/bending load. For this reason, and because we are mainly interested in highlighting the numerical scalability of the FETI method, we consider only the case of a fixed global problem and an increasing number of substructures. We set $L = 1.0$, $t = 0.01$, $G = 7.84 \cdot 10^6$, and $\nu = 0.34$. The square domain is uniformly discretized with 48×48 four-node elements and 4,704 equations are generated. The performance of the FETI method for this problem is reported in Table 7 for 4, 16, 64 and 256 substructures.

Here again, the results reported in Table 7 highlight the coarse grid effect of the projector $P = [I - G_1(G_1^T G_1)^{-1} G_1^T]$ and the asymptotic numerical scalability of the FETI method. Note that the fastest convergence is achieved for $H = 1/2$, for which the condition number is higher than for $H = 1/8$ and $H = 1/16$. In Section 7, we show that such a phenomenon corresponds to a superconvergent behavior of the FETI method that is usually triggered when the number of substructures is relatively small.

4.2.2. Smooth shell problems

Here we consider a smooth shell structure that has the same geometry and material properties as those of the plane stress problem previously discussed, and that is subject to a combined membrane/bending load (Fig. 4). The structure is uniformly discretized with 48×48 four-node shell elements with 5 degrees of freedom per node. The corresponding number of equations is 14,112. As for the plane stress problem, we partition the mesh into 4, 16, 64 and 256 substructures and report in Table 8 the condition number of the preconditioned dual interface problem, and the number of iterations to achieve convergence.

Clearly, the condition number of the preconditioned dual interface problem is shown to decrease when the number of substructures is increased, which demonstrates the asymptotic numerical scalability of the FETI method. Superconvergence is observed for $H = 1/2$ and $H = 1/4$. Note however that, as expected, the shell problem leads to significantly larger condition numbers than the Poisson and plane stress problems. In particular,

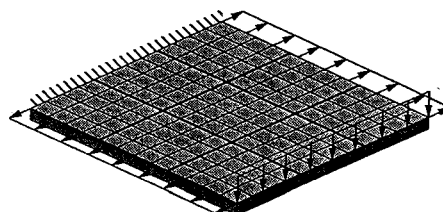


Fig. 4. A smooth shell problem.

Table 8
FETI method with \tilde{F}_1^{D-1} preconditioner—Effect of the inherent coarse grid problem—Smooth shell problem on a square domain, 14,112 equations

H	N_s	Condition number	Number of iterations
1/2	4	61,244	40
1/4	16	13,870	124
1/8	64	3,837	169
1/16	256	3,084	138

the results depicted in Table 8 suggest that for shell problems, the condition number of the interface problem grows faster than $O(1 + \log^2(H/h))$.

5. Effects of the subdomain aspect ratio and the presence of junctures

Ironically, the authors of the FETI method were not aware of its optimal convergence properties until Mandel's results [6] became available. This is essentially because the FETI method was developed to solve realistic aircraft structural models for which optimal convergence was not observed. For such problems, it was found that the FETI method is capable of outperforming popular direct and iterative solvers, but its rate of convergence was also found to deteriorate for a number of subdomains larger than 32. Here, we comment on why complex structural problems represent a bigger challenge to optimal DD solvers than the Poisson, plane stress, and smooth shell problems.

For large number of subdomains, conditioning of the global problem alone cannot explain the discrepancy between the convergence rates observed for the previous academic problems and those observed for more realistic structural problems. We believe that a key factor in the convergence rate of the FETI method and several other DD algorithms is the subdomain aspect ratio. In order to highlight the effect of this important factor, we consider again the plane stress and smooth shell problems each discretized with 48×48 elements. We keep the total number of elements constant and construct various $M \times N$ mesh decompositions. We report in Tables 9 and 10 the corresponding subdomain aspect ratio $A_{M,N}$, the condition number of the preconditioned interface problem, and the number of iterations to achieve convergence (between parenthesis).

The reader should note that the performance results reported in Tables 9 and 10 are not symmetric even though $A_{N,M} = A_{M,N}$. This is because the boundary conditions of both problems are not symmetric with respect to the M and N partitions. Clearly, these results demonstrate that for a fixed number of subdomains, convergence is faster when the subdomain aspect ratio is closer to 1. For the smooth shell problem, the 8×8 mesh partition generates

Table 9
FETI method with \tilde{F}_1^{D-1} preconditioner—Effect of the substructure aspect ratio—Plane stress problem on a square domain (4,704 equations)

N	M				
	1	2	4	8	16
1	—	$A_{2,1} = 0.5$ 4.1 (9)	$A_{4,1} = 0.25$ 19.2 (15)	$A_{8,1} = 0.125$ 237.0 (34)	$A_{16,1} = 0.0625$ 3,713.0 (86)
2	$A_{1,2} = 0.5$ 2.6 (6)	$A_{2,2} = 1.0$ 18.8 (13)	$A_{4,2} = 0.5$ 9.4 (19)	$A_{8,2} = 0.25$ 20.3 (30)	$A_{16,2} = 0.125$ 173.0 (63)
4	$A_{1,4} = 0.25$ 147.0 (14)	$A_{2,4} = 0.5$ 94.3 (24)	$A_{4,4} = 1.0$ 20.1 (23)	$A_{8,4} = 0.5$ 8.8 (23)	$A_{16,4} = 0.25$ 22.2 (35)
8	$A_{1,8} = 0.125$ 3,714.0 (36)	$A_{2,8} = 0.25$ 677.0 (40)	$A_{4,8} = 0.5$ 79.3 (30)	$A_{8,8} = 1.0$ 15.7 (24)	$A_{16,8} = 0.5$ 8.6 (22)
16	$A_{1,16} = 0.0625$ 63,985.0 (85)	$A_{2,16} = 0.125$ 8,861.0 (84)	$A_{4,16} = 0.25$ 566.0 (53)	$A_{8,16} = 0.5$ 57.4 (36)	$A_{16,16} = 1.0$ 10.8 (24)

Table 10
 FETI method with \tilde{F}_1^{D-1} preconditioner—Effect of the substructure aspect ratio—Smooth shell problem on a square domain (14,112 equations)

N	M				
	1	2	4	8	16
1	—	$A_{2,1} = 0.5$ 57.0 (24)	$A_{4,1} = 0.25$ 292.0 (40)	$A_{8,1} = 0.125$ 665.0 (83)	$A_{16,1} = 0.0625$ 4,333.0 (181)
2	$A_{1,2} = 0.5$ 4.4 (15)	$A_{2,2} = 1.0$ 61,244.0 (40)	$A_{4,2} = 0.5$ 12,465.0 (78)	$A_{8,2} = 0.25$ 11,408.0 (145)	$A_{16,2} = 0.125$ 51,899.0 (280)
4	$A_{1,4} = 0.25$ 162.0 (28)	$A_{2,4} = 0.5$ 20,2321.0 (76)	$A_{4,4} = 1.0$ 13,870.0 (124)	$A_{8,4} = 0.5$ 3,720.0 (167)	$A_{16,4} = 0.25$ 9,604.0 (271)
8	$A_{1,8} = 0.125$ 3,799.0 (66)	$A_{2,8} = 0.25$ 1,042,579.0 (146)	$A_{4,8} = 0.5$ 2,6782.0 (192)	$A_{8,8} = 1.0$ 3,837.0 (169)	$A_{16,8} = 0.5$ 4,035.0 (163)
16	$A_{1,16} = 0.0625$ N.A.	$A_{2,16} = 0.125$ 4,892,620.0 (325)	$A_{4,16} = 0.25$ 78,409.0 (317)	$A_{8,16} = 0.5$ 8,331.0 (202)	$A_{16,16} = 1.0$ 3,084.0 (138)

64 subdomains where the subdomain aspect ratio is equal to 1, the condition number of the preconditioned interface problem is equal to 3837.0, and the number of iterations to achieve convergence is equal to 169. For the same problem, the 4×16 mesh partition also generates 64 subdomains, but produces a subdomain aspect ratio equal to 0.25, a condition number of the preconditioned interface problem equal to 78409.0, and requires 317 iterations for convergence. The results reported in Tables 9 and 10 also indicate that the subdomain aspect ratio is more important for scalability than the number of subdomains. For example, Table 10 shows that the 16×16 mesh partition that generates 256 subdomains with perfect subdomain aspect ratios, converges in only 138 iterations, while the 2×16 mesh decomposition that generates 32 subdomains only, converges in 325 iterations. This influence of the mesh partition on the convergence rate of a DD algorithm is not limited to the FETI method: we have run several experiments with other DD algorithms and have observed the same trend. Moreover, Le Tallec and Vidrascu have recently communicated to us that the Neumann-Neumann [4] and the Balancing domain decomposition [6] methods are both sensitive to the topology of the mesh partition, and that the performance of each of these two methods can differ by up to 300% when different mesh decompositions are used [18].

Each of the academic problems previously discussed has a simple geometry and a regular discretization that allow a uniform $M \times M$ mesh decomposition. However, complex problems involving complicated geometries and unstructured meshes often require the use of an automatic mesh decomposition algorithm to generate the desired subdomains. Such partitioning algorithms are now available in the literature [19-21] and have one common characteristic: they do not pay any special attention to the subdomain aspect ratio. In general, these algorithms focus on minimizing interprocessor communication costs and produce irregular mesh partitions with arbitrary shapes and arbitrary subdomain aspect ratios. Hence, the results in Tables 9 and 10 suggest that the performance of a DD algorithm applied to a realistic structural problem with arbitrary mesh partitions will be less impressive than that observed for an academic problem with perfect subdomain aspect ratios.

As an example, we consider the stress analysis of a High Speed Civil Transport (HSCT) aircraft under a gust load. The structural model includes the skin, the fuselage, the ribs and the stiffeners (Fig. 5). The finite element mesh contains 69120 triangular shell elements with 6 d.o.f. per node and generates 197442 equations. After renumbering, the average column height of the corresponding system of equations is 1496. The finite element mesh is decomposed into 4, 8, 16, and 32 subdomains using the Greedy algorithm [21]. The reader can observe (Fig. 6) that the overall shape of the subdomains degrades when their number is increased. For this problem, the performance of the FETI method is reported in Table 11 and contrasted with the performances of a highly vectorized direct solver [22] (DIR) with an out-of-core capability, and a vectorized Jacobi preconditioned (diagonal scaling) conjugate gradient algorithm (JPCG). All computations are carried out on a CRAY-2 processor. In order to ensure a fair comparison between the FETI method and the direct solver, all subdomain solutions are computed using DIR.

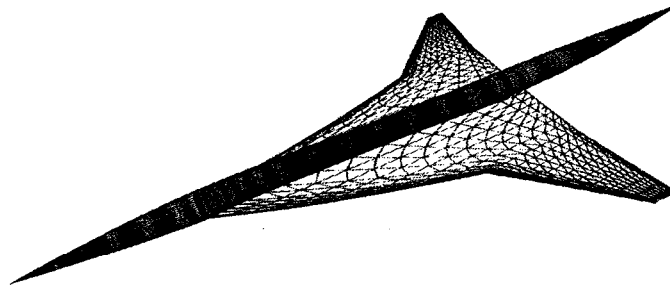


Fig. 5. The HSCT structural model.

Table 11
FETI method with \tilde{F}_1^{D-1} preconditioner—Effect of arbitrary mesh partitions and performance comparison with DIR and JPCG—HSCT static analysis (197,442 equations)

Method	N_s	Number of iterations	CPU
FETI	4	440	2,241.0 s
	8	889	2,479.0 s
	16	1,074	2,644.0 s
	32	1,640	3,980.0 s
DIR	1	1	4,505.0 s
JPCG	1	> 32,000	> 9,295.0 s

Clearly, scalability is lost for the HSCT problem. However, the FETI method is still reported to outperform the direct solver. After 32000 iterations, the conjugate gradient algorithm with diagonal scaling still does not converge.

Geometrical discontinuities such as those encountered at the junctures between the fuselage, the skin, and the stiffeners, are other likely important sources of performance degradation, especially when they coincide with the subdomain interfaces. We remind the reader that for the Poisson problem with over 400,000 unknowns and as many as 1024 subdomains ($h = 1/20$, $M = N = 32$), the FETI method was found to converge in 20 iterations only (Table 2). However, even when the number of subdomains is kept as small as $N_s = 4$ —which usually entails reasonable subdomain aspect ratios—the FETI algorithm requires more than 400 iterations to achieve convergence for the HSCT problem which has less than 200,000 d.o.f. but several material and geometrical heterogeneities.

6. Optimal vs. computationally efficient domain decomposition preconditioners

The optimal Dirichlet (or primal) preconditioner:

$$D_1^{-1} = \sum_{s=1}^{N_s} B^s \begin{bmatrix} 0 & 0 \\ 0 & S_{bb}^s \end{bmatrix} B^{sT} = \sum_{s=1}^{N_s} B^s \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^s - K_{ib}^{sT} K_{ii}^{s-1} K_{ib}^s \end{bmatrix} B^{sT} \quad (43)$$

introduces some memory and computational burdens. Even though its building blocks K_{ii}^s are embedded in the building blocks K^s of the dual interface operator $F_1 = \sum_s B^s K^{sT} B^{sT}$, these blocks must be stored and factored twice. The reason is that for a given Ω^s , K_{ii}^s and K^s have two different optimal numberings for the internal degrees of freedom that lead to two different matrices. Moreover, each iteration with the Dirichlet preconditioner requires an additional pair of forward/backward substitutions on the internal subdomain degrees of freedom, which doubles the cost of the unpreconditioned CG algorithm.

Because of the storage and computational burdens associated with the Dirichlet preconditioner, we have advocated in [1] preconditioning the dual interface problem with the following “lumping” operator:

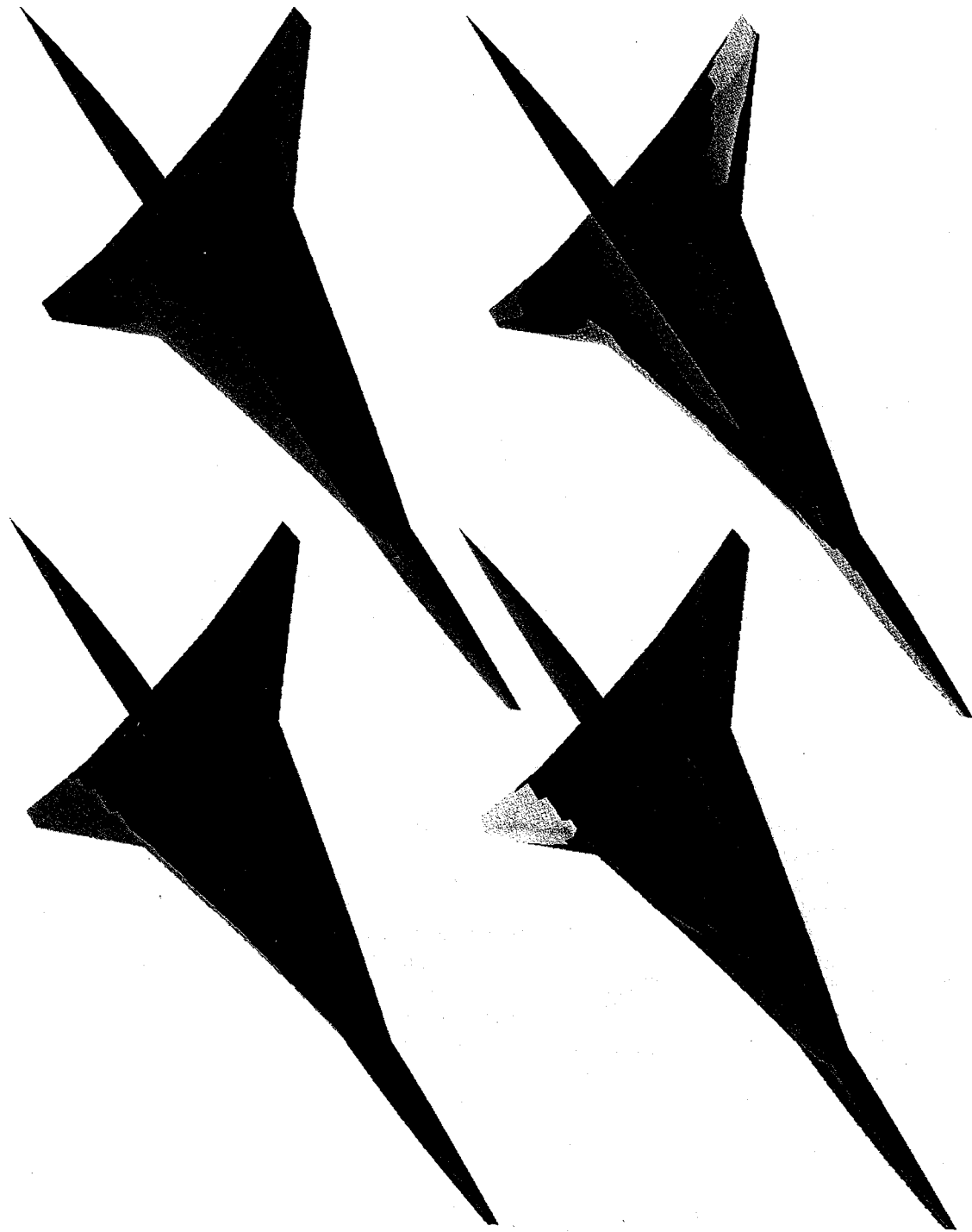


Fig. 6. Decomposition of the HSCT mesh into 4, 8, 16 and 32 subdomains.

Table 12

FETI method with L_1^{-1} preconditioner—Dependence on h —Poisson problem on a square domain, $H = 1/4$ ($M = N = 4$, i.e. 16 substructures)

h	Condition number	Number of iterations
1/10	12.9	23
1/20	25.3	29
1/40	50.4	40
1/80	101.0	52
1/160	206.0	52

Table 13

FETI method with L_1^{-1} preconditioner—Dependence on h —Poisson problem on a square domain, $H = 1/8$ ($M = N = 8$, i.e. 64 substructures)

h	Condition number	Number of iterations
1/10	13.6	26
1/20	26.7	34
1/40	53.1	49
1/80	105.0	54
1/160	208.0	54

$$L_1^{-1} = \sum_{s=1}^{N_s} B^s \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^s \end{bmatrix} B^{sT}. \quad (44)$$

The L_1^{-1} preconditioner is economical because it does not require any additional storage and involves only matrix-vector products of sizes equal to the subdomain interfaces. From a mechanical viewpoint, each preconditioning step using L_1^{-1} corresponds to finding a set of “lumped” interface forces that can reproduce the displacement jumps at the subdomain interfaces when only the interface nodes are allowed to displace.

The lumped preconditioner L_1^{-1} is not mathematically optimal because the condition number of $L_1^{-1}F_1$ grows as $O(H/h)$ for the Poisson problem, as illustrated in Tables 12 and 13. However, the results reported in these tables also show that the number of iterations to achieve convergence initially varies sublinearly with $1/h$ and asymptotically becomes independent of h . Therefore in practice, the performance of the FETI method with the lumped preconditioner can be considered as weakly dependent on h . The exposed difference in behavior between the condition number and the number of iterations when h is decreased is explained in Section 7 with a superconvergence theory.

The performance results reported in Table 14 for the Poisson model problem also show that the lumped preconditioner L_1^{-1} does not perturb the scalability effect of the coarse grid problem. Quite the opposite, the condition number of $L_1^{-1}F_1$ and the number of iterations to achieve convergence decrease with a growing number of subdomains since then H/h decreases.

The FETI method with the Dirichlet preconditioner performs less iterations than with the lumped preconditioner (see Tables 3–5 and Tables 12–14). However, an iteration with D_1^{-1} is more expensive than an iteration with L_1^{-1} , and therefore it is interesting to assess which preconditioner is in practice more efficient. For this purpose, we apply both preconditioners to the solution of the HSCT problem introduced in Section 5 (Table 15).

Clearly, the Dirichlet preconditioner appears to perform at best 30% less iterations than the lumped preconditioner. However for this realistic problem, the FETI method with L_1^{-1} is found to be at least 25% faster than with D_1^{-1} . We have observed similar results for a wide range of complex structural problems. In particular, we have found that for problems with severe random heterogeneities, the Dirichlet preconditioner performs almost as many iterations as the lumped preconditioner and therefore becomes unjustifiable.

Table 14

FETI method with L_1^{-1} preconditioner—Effect of the inherent coarse grid problem—Poisson problem on a square domain, 102,400 elements

H	N_s	Condition number	Number of iterations
1/4	16	101.0	52
1/8	64	53.1	49
1/16	256	27.0	38
1/32	1024	13.8	24

Table 15
FETI method— D_1^{-1} preconditioner vs. L_1^{-1} preconditioner—HSCT static analysis (197,442 equations)

N_s	FETI - D_1^{-1}		FETI - L_1^{-1}	
	Number of iterations	CPU time (s)	Number of iterations	CPU time (s)
4	440	2241.0	629	1860.0
8	489	2479.0	689	1958.0
16	674	2544.0	923	2032.0
32	1040	3580.0	1237	2398.0

7. Superconvergence of the FETI method

The interface matrix $F_1 = \sum_{s=1}^{N_s} B^s K^{s^*} B^{s^T}$ corresponds to the discretization of a *compact* operator that maps the normal components of the stress tensor along the subdomain interfaces onto the traces on these interfaces of the subdomain displacement fields [23]. Because of this compactness, the eigenvalues of the matrix $\sum_{s=1}^{N_s} B^s K^{s^*} B^{s^T}$ accumulate towards zero when $h \rightarrow 0$, and the high end of their spectrum is less populated than the low end. The objectives of this section are: (a) to numerically illustrate the spectral properties of the dual interface operator, and (b) to highlight the impact of these spectral properties on the convergence rate of the CG algorithm.

As an example, we consider the three-dimensional two-subdomain cantilever problem depicted in Fig. 7. The beam has a square cross section, and a length-over-height aspect ratio equal to 10. Two finite element meshes are constructed using 8-node brick elements. The first mesh, M_1 , contains 2,400 internal d.o.f. and 192 interface d.o.f. The second mesh, M_2 , is finer: it contains 16,320 internal d.o.f. and 672 interface d.o.f. However, M_2 has better element aspect ratios than M_1 and therefore generates a better conditioned interface problem, even though it is finer than M_1 .

The distribution of the eigenvalues of the dual interface operator associated with the above problem is depicted in Fig. 8 for mesh M_1 , and in Fig. 9 for mesh M_2 . The x axis of the bar diagrams depicted in Figs. 8 and 9 represents the eigenvalues scaled by the smallest eigenvalue, and the y axis the number of eigenvalues per interval. For both meshes, F_1 is shown to have a few large eigenvalues that are well separated from the small ones. Fig. 9 suggests that this separation amplifies when the mesh size $h \rightarrow 0$. Indeed, the large eigenvalues of F_1 stabilize when the mesh size decreases, while its small eigenvalues accumulate towards zero. Essentially, this is because F_1 involves the inverses of K^s and therefore its high modes correspond to physical modes while its low modes correspond to mesh modes.

The spectral distribution of the interface problem has important consequences on the convergence rate of the CG algorithm. During the first iterations, the CG algorithm captures mostly the eigenvectors associated with the large eigenvalues. Since F_1 has only a few relatively high eigenvalues that correspond to the low physical modes of the structure, the CG algorithm applied to the solution of the interface problem gives quickly a good approximation of the displacement field. Moreover, after the few high modes of the interface operator are captured, the “effective” condition number of F_1 —that is the ratio of its largest uncaptured eigenvalue over its smallest one—becomes significantly smaller than the original condition number of the interface problem, which triggers a superconvergence behavior of the CG algorithm. This superconvergence behavior is often observed for operators with well separated eigenvalues and is analyzed in [24].

The impact of the spectral distribution on the convergence rate of the CG algorithm is highlighted in Table 16

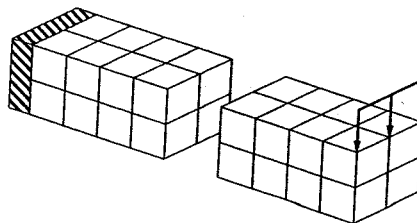


Fig. 7. A three-dimensional two-subdomain cantilever problem.

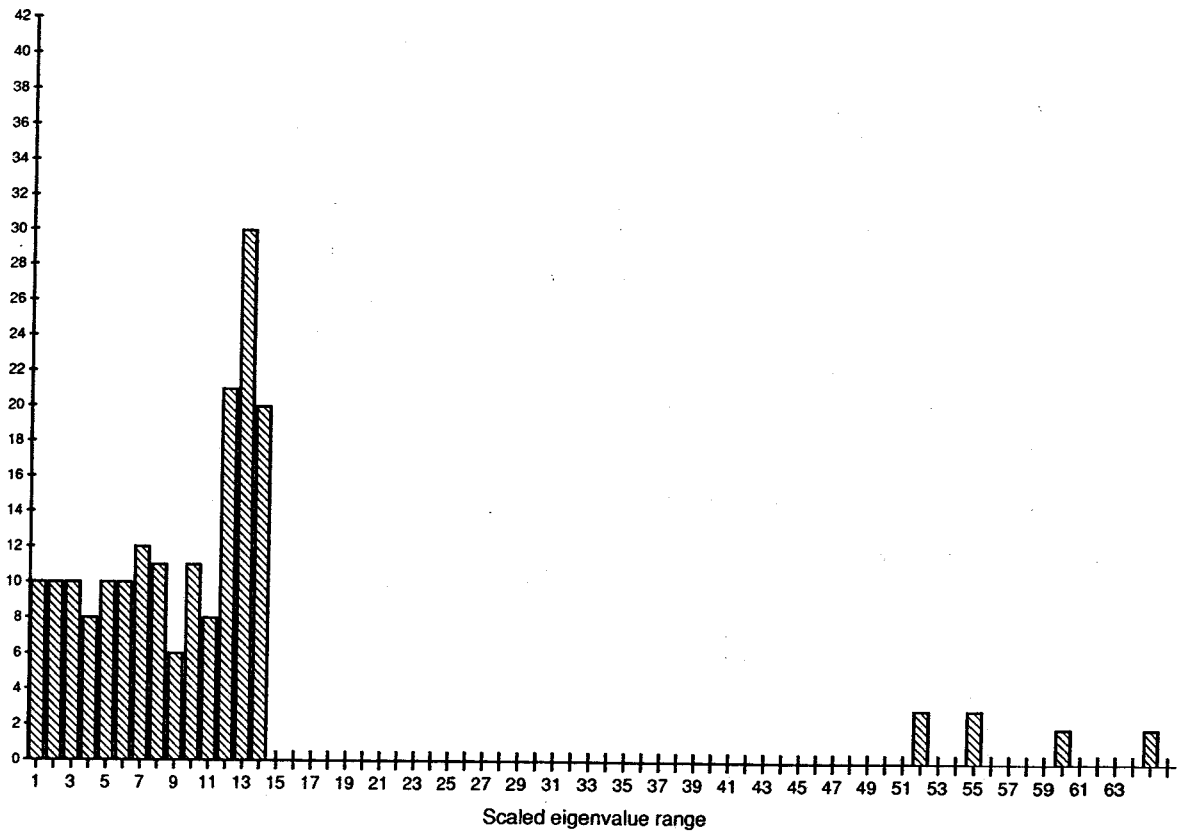


Fig. 8. Three-dimensional cantilever problem—Spectral density of F_1 for mesh M_1 .

which reports the number of iterations to achieve convergence for both the FETI and classical Schur complement methods. The interface operator associated with the Schur method is denoted by S_I and happens to have the same condition number as F_I . However, the CG algorithm applied to F_I is shown to converge faster than when applied to S_I .

For the Poisson problem, we have shown that the condition number of $L_1^{-1}F_1$ varies as $O(H/h)$, but the number of iterations to achieve convergence initially varies sublinearly with $1/h$ and may even become independent of h (Tables 12 and 13). Therefore, we have concluded that in practice, the performance of the FETI method with the lumped preconditioner can be considered as weakly dependent on h . Here, we show that the results reported in Tables 12 and 13 can also be explained with a superconvergence theory. For this purpose, we consider again the two-subdomain cantilever problem depicted in Fig. 7. For this problem and mesh M_2 , the distribution of the eigenvalues of $L_1^{-1}F_1$ is depicted in Fig. 10. A comparison of Figs. 9 and 10 shows that the lumped preconditioner L_1^{-1} essentially amplifies the separation between the clusters of small and large eigenvalues of the dual interface problem and therefore accelerates the convergence of the CG algorithm. The superconvergence effect introduced by L_1^{-1} is highlighted in Table 17 for the three-dimensional two-subdomain cantilever problem.

Table 16

Three-dimensional two-substructure cantilever problem—FETI method (F_I) vs. Schur complement method (S_I)—Effect of the spectral distribution

Mesh	Internal d.o.f.	Interface d.o.f.	κ_2 (F_I)	κ_2 (S_I)	Number of iterations (F_I)	Number of iterations (S_I)
M_1	2,400	192	64.0	64.0	23	34
M_2	16,320	672	56.0	56.0	24	34

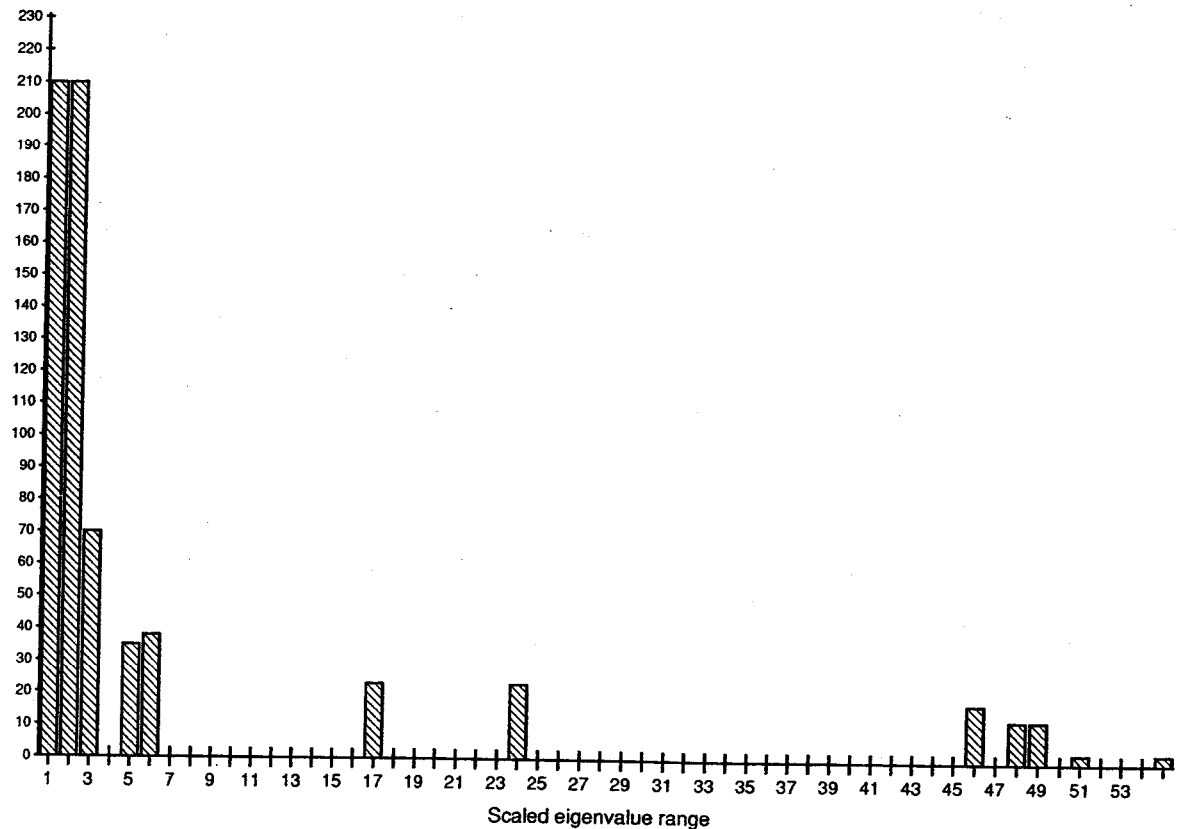


Fig. 9. Three-dimensional cantilever problem—Spectral density of F_1 for mesh M_2 .

Finally, we point out that matrices with the spectral pattern illustrated in Figs. 8 and 9 have been shown to cause a rapid loss of orthogonality between the search direction vectors [25]. In order to avoid the disastrous consequence on the convergence rate of the PCPG algorithm, we always use a reorthogonalization procedure that does not compromise the solution speed [23].

8. Conclusions

The Finite Element Tearing and Interconnecting (FETI) method is a practical and efficient domain decomposition (DD) algorithm for the solution of self-adjoint elliptic partial differential equations. For large-scale structural problems discretized with shell and beam elements, this method was found to outperform direct solvers on both serial and parallel computers, and to compare favorably with leading DD methods. In this paper, we have discussed some numerical properties of the FETI method that were not addressed before. We have shown that the mathematical treatment of the floating subdomains and the specific conjugate projected gradient algorithm that characterize the FETI method are equivalent to the construction and solution of a coarse problem that propagates the error globally, accelerates convergence, and ensures scalability. We have also shown that when the interface problem is optimally preconditioned and the mesh is decomposed into well structured

Table 17

Three-dimensional two-substructure cantilever problem—FETI method with L_1^{-1} preconditioner—Performance improvement

Mesh	Internal d.o.f.	Interface d.o.f.	Number of iterations (F_1)	Number of iterations ($L_1^{-1}F_1$)
M_1	2,400	192	23	10
M_2	16,320	672	24	11

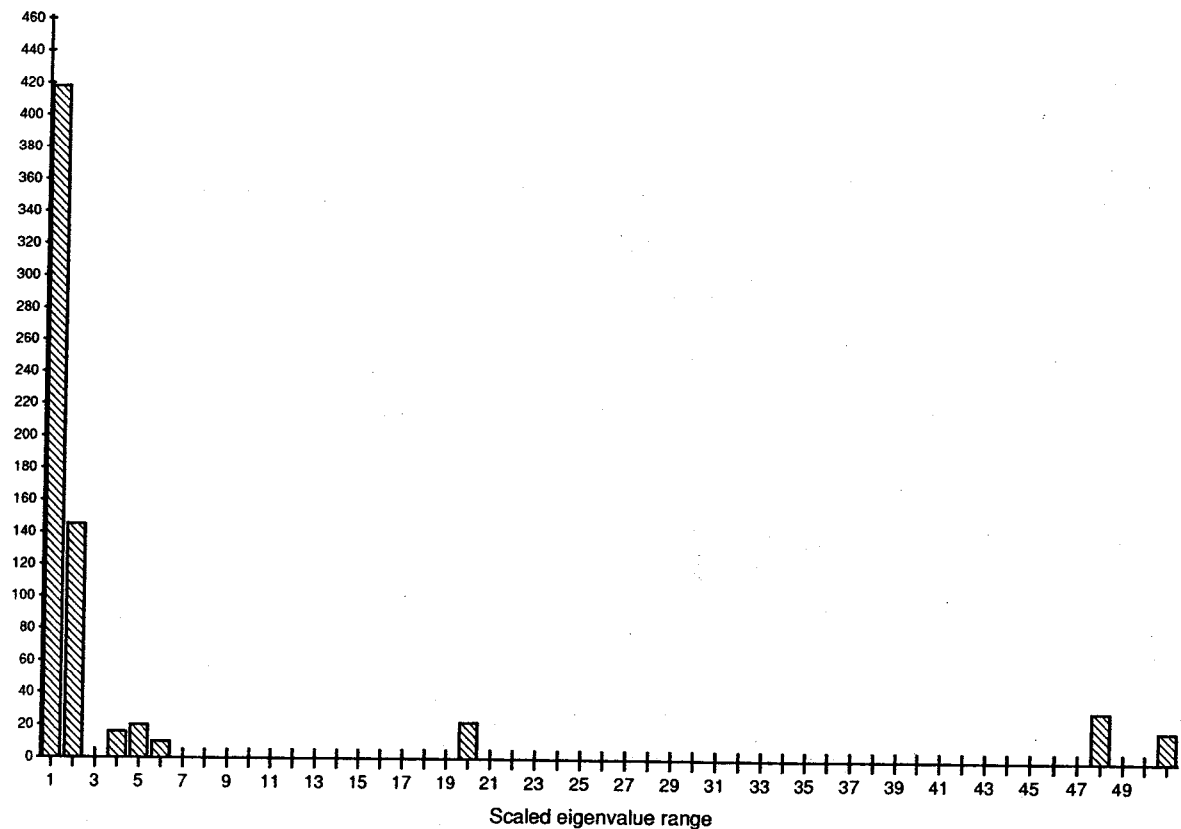


Fig. 10. Three-dimensional cantilever problem—Spectral density of $L_1^{-1} F_1$ for mesh M_2 .

subdomains with good aspect ratios, the performance of the FETI method becomes independent of both the number of subdomains and the mesh size. However, we have also argued that the FETI and other leading DD methods for unstructured problems lose these scalability properties when the mesh contains junctures and/or is partitioned into irregular subdomains with arbitrary aspect ratios. Therefore, it remains to design mesh partitioning algorithms that can always generate perfect subdomain aspect ratios, and/or preconditioning schemes that are insensitive to subdomain aspect ratios. Both objectives are challenging.

Acknowledgments

The first and third authors acknowledge partial support by the NASA Langley Research Center under Grant NAG-1536427. The second author acknowledges the support of the National Science Foundation under Grants ASC-9121431 and ASC-9217394.

References

- [1] C. Farhat and F.X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, *Internat. J. Numer. Methods Engrg.* 32 (1991) 1205–1227.
- [2] C. Farhat and F.X. Roux, An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems, *SIAM J. Sci. Stat. Comput.* 13 (1992) 379–396.
- [3] P.E. Bjordstad and O.B. Widlund, Iterative methods for solving elliptic problems on regions partitioned into substructures, *SIAM J. Numer. Anal.* 23 (1986) 1097–1120.
- [4] P. Le Tallec, Y.H. De Roeck and M. Vidrascu, Domain decomposition methods for large linearly elliptic three dimensional problems, *J. Comput. Appl. Math.* 34 (1991) 93–117.
- [5] D. Keyes, Domain decomposition: a bridge between nature and parallel computers, in: A.K. Noor, ed., *Proc. Symposium on Adaptive Multilevel and Hierarchical Computational Strategies* (ASME, 1992) 293–334.

- [6] J. Mandel, Balancing domain decomposition, *Commun. Appl. Numer. Methods* 9 (1993) 233–241.
- [7] J.H. Bramble, J.E. Pasciak and A.H. Schatz, The construction of preconditioners for elliptic problems by substructuring, *Math. Comput.* 47 (1986) 103–134.
- [8] B.F. Smith, An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems, *SIAM J. Sci. Stat. Comput.* 13 (1992) 364–378.
- [9] O. Axelsson and I. Gustafsson, Preconditioning and two-level multigrid methods of arbitrary degree of approximation, *Math. Comput.* 40 (1983) 219–242.
- [10] M. Kočvara and J. Mandel, A multigrid method for three-dimensional elasticity and algebraic convergence estimates, *Appl. Math. Comput.* 23 (1987) 121–135.
- [11] C. Farhat and N. Sobh, A coarse/fine preconditioner for very ill-conditioned finite element problems, *Internat. J. Numer. Methods Engrg.* 28 (1989) 1715–1723.
- [12] J. H. Bramble, J.E. Pasciak and A.H. Schatz, The construction of preconditioners for elliptic problems by substructuring, I, *Math. Comput.* 47 (1986) 103–134.
- [13] C. Farhat, A saddle-point principle domain decomposition method for the solution of solid mechanics problems, in: D.E. Keyes, T.F. Chan, G.A. Meurant, J.S. Scroggs and R.G. Voigt, eds., *Proc. Fifth SIAM Conference on Domain Decomposition Methods for Partial Differential Equations* (SIAM, 1991) 271–292.
- [14] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, (Springer-Verlag, Berlin, 1991) 26–27.
- [15] G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins, Baltimore, MD, 1983).
- [16] J. Mandel and M. Brezina, Balancing domain decomposition: theory and computations in two and three dimensions, (submitted for publication).
- [17] Y.C. Fung, *Foundations of Solid Mechanics* (Prentice-Hall, Englewood Cliffs, NJ, 1965) 233–234.
- [18] P. Le Tallec and M. Vidrascu (private communication).
- [19] C. Farhat, A simple and efficient automatic FEM domain decomposer, *Comput. & Struct.* 28 (1988) 579–602.
- [20] A. Pothen, H. Simon and K.P. Liou, Partitioning sparse matrices with eigenvectors of graphs, *SIAM J. Math. Anal. Appl.* 11 (1990) 430–452.
- [21] C. Farhat and M. Lesoinne, Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics, *Internat. J. Numer. Methods Engrg.* 36 (1993) 745–764.
- [22] C. Farhat, Redesigning the skyline solver for parallel/vector supercomputers, *Internat. J. High Speed Comput.* 2 (1990) 223–238.
- [23] F.X. Roux, Dual and spectral properties of Schur and saddle point domain decomposition methods, in: D.E. Keyes, T.F. Chan, G.A. Meurant, J.S. Scroggs and R.G. Voigt, eds., *Proc. Fifth SIAM Conference on Domain Decomposition Methods for Partial Differential Equations* (SIAM, 1991) 73–90.
- [24] A. van der Sluis and A. van der Vorst, The rate of convergence of conjugate gradients, *Numerische Mathematik* 48 (1986) 543–560.
- [25] B. N. Parlett, *The Symmetric Eigenvalue Problem* (Prentice Hall, Englewood Cliffs, NJ, 1980).

