

# FETI-DP: a dual–primal unified FETI method—part I: A faster alternative to the two-level FETI method

Charbel Farhat,<sup>1,\*</sup>† Michel Lesoinne,<sup>1</sup> Patrick LeTallec,<sup>2</sup>  
Kendall Pierson<sup>1</sup> and Daniel Rixen<sup>3</sup>

<sup>1</sup>*Department of Aerospace Engineering Sciences and Center for Aerospace Structures, University of Colorado at Boulder, Boulder, CO 80309-0429, U.S.A.*

<sup>2</sup>*Université de Paris Dauphine and Ecole Polytechnique, 91 128 Palaiseau Cedex, France*

<sup>3</sup>*LTAS, Université de Liège, Rue Ernest Solvay, 21, B-4000 Liège, Belgium*

## SUMMARY

The FETI method and its two-level extension (FETI-2) are two *numerically scalable* domain decomposition methods with Lagrange multipliers for the iterative solution of second-order solid mechanics and fourth-order beam, plate and shell structural problems, respectively. The FETI-2 method distinguishes itself from the basic or one-level FETI method by a second set of Lagrange multipliers that are introduced at the subdomain cross-points to enforce at each iteration the exact continuity of a subset of the displacement field at these specific locations. In this paper, we present a dual–primal formulation of the FETI-2 concept that eliminates the need for that second set of Lagrange multipliers, and unifies all previously developed one-level and two-level FETI algorithms into a single dual–primal FETI-DP method. We show that this new FETI-DP method is numerically scalable for both second-order and fourth-order problems. We also show that it is more robust and more computationally efficient than existing FETI solvers, particularly when the number of subdomains and/or processors is very large. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: domain decomposition; numerical scalability; iterative methods

## 1. BACKGROUND

The finite element tearing and interconnecting (FETI) methods are a family of domain decomposition (DD) algorithms with Lagrange multipliers that have been developed during the last decade for the fast sequential and parallel iterative solution of large-scale systems of equations arising from the finite element discretization of partial differential equations. From a mechanical viewpoint, a FETI method can be viewed as an iterative substructuring method where Lagrange multipliers are introduced at the substructure interfaces to enforce the continuity of the displacement field. By

---

\*Correspondence to: Charbel Farhat, Department of Aerospace Engineering and Center for Aerospace Structures, University of Colorado at Boulder, Boulder, CO 80309-0429, U.S.A.

†E-mail: charbel@boulder.colorado.edu

Contract/grant sponsor: Sandia National Laboratories; contract/grant number: BD-2435

Contract/grant sponsor: Department of Energy; contract/grant number: B347880/W-740-ENG-48

Contract/grant sponsor: INRIA

Received 16 August 1999

Revised 10 May 2000

construction, equilibrium is satisfied at each iteration in each substructure. However, the continuity condition is met only at convergence. From a mathematical viewpoint, each FETI method can be viewed as a two-step preconditioned conjugate gradient (PCG) algorithm where subdomain problems with Dirichlet boundary conditions are solved in the preconditioning step, and related subdomain problems with Neumann boundary conditions are solved in a second step to compute residuals.

The basic FETI method, also known as the one-level FETI method, or simply the FETI method, was developed in References [1–5] together with its ‘lumped’ and ‘Dirichlet’ preconditioners, and extended in Reference [4] to substructure problems with non-matching interfaces. Its optimal convergence properties for second-order elliptic problems (i.e. thermal, structural, and solid mechanics problems discretized by plane stress/strain and/or brick elements) were first exposed numerically in Reference [5], then mathematically established in Reference [6]. More specifically, it was proved in Reference [6] that when the basic FETI method is equipped with the Dirichlet preconditioner [5] and applied to second-order elliptic problems, the condition number  $\kappa$  of its interface problem grows asymptotically as

$$\kappa = O(1 + \log^m(H/h)), \quad m \leq 3 \quad (1)$$

where  $h$  and  $H$  denote, respectively, the mesh and subdomain sizes. The conditioning result (1), which was later proved to hold also with  $m = 2$  [7, 8], demonstrates that the FETI method is *numerically scalable* with respect to both the problem size and number of subdomains. Indeed, from Equation (1) it can be concluded that

- P1. If for a fixed number of subdomains  $N_s$  ( $N_s = O(1/H^2)$  in two dimensions and  $N_s = O(1/H^3)$  in three dimensions) the problem size is increased ( $O(1/h^2)$  in two dimensions and  $O(1/h^3)$  in three dimensions), the condition number of the FETI method grows asymptotically only as  $\log^2 1/h$ . In other words, one can expect the FETI method to solve large-scale problems in a similar number of iterations as small-scale problems.
- P2. If for a fixed problem size the number of subdomains is increased—for example, for parallel processing purposes or in order to reduce the operation count of each iteration—the iteration count of the FETI method can be expected to decrease.
- P3. If the size of the subdomain problem, which can be characterized by  $H/h$ , is kept constant and the total size of the problem is increased by increasing the number of subdomains—as is often done when the number of processors of a given parallel computing platform is increased—the condition number of the FETI method remains constant. This property makes it possible for a well-implemented FETI method to solve an  $n$ -times larger problem on an  $n$ -times larger machine in a constant amount of CPU time.

The three numerical scalability properties highlighted above are also verified in practice for complex problems (for example, see References [5, 9, 10] and the references cited therein). For a sufficiently large problem, the *parallel scalability* of the FETI method—that is, its ability to deliver a speed-up that grows reasonably well with the number of processors (almost linearly)—depends on the parallel implementation of some key computational kernels of this DD method, particularly the solution of the associated ‘coarse’ problems [10], and some key characteristics of the target parallel hardware such as memory bandwidth and cache management. The parallel scalability of the FETI method, and its ability to solve an  $n$ -times larger problem on an  $n$ -times larger machine in a constant amount of CPU time have been recently demonstrated on a 1000-processor configuration of the ASCI Option Red massively parallel machine installed at the Sandia National Laboratories.

The FETI method has been successfully extended to problems with multiple and/or repeated right-hand sides [11, 12], component mode synthesis [13], transient response analysis [14], non-linear solution strategies [15–17], free-vibration analysis [18], problems with multipoint constraints [19], problems with severe heterogeneities such as large discontinuities in material properties [20], and most recently acoustic scattering problems [21, 22]. It has also inspired many variants, and other extensions and applications, among which we note those described in [23–28]. All these variants share the same lumped and Dirichlet preconditioners and the same coarse problem constructed with the rigid body modes of the floating subdomains, all of which were originally developed for the FETI method.

A distinctive feature of the FETI method is its rigid-body-based auxiliary problem that is naturally derived from the subdomain equations of equilibrium [1–3]. This auxiliary problem must be solved twice at each PCG iteration. Because its size is in general as small as  $3N_s$  in two dimensions and  $6N_s$  in three dimensions, this problem is referred to as a ‘coarse’ problem. In References [5, 6], it was shown that for second-order partial differential equations, this coarse problem is the main reason why the FETI method is numerically scalable with respect to the number of subdomains. The inexpensive solution at each PCG iteration of this coarse problem propagates the error globally and accelerates convergence.

For transient dynamics applications, a floating subdomain has a singular static stiffness matrix but a non-singular dynamic stiffness matrix. Hence, the subdomain equations of dynamic equilibrium do not naturally lead to any coarse problem. For this reason, a new coarsening procedure was designed in Reference [14] for addressing the solution of second-order transient dynamic problems by a numerically scalable FETI method. This procedure had led to a new FETI coarse problem for time-dependent applications that is also based on the null spaces of the subdomain static stiffness matrices, but which has a different expression than the FETI coarse problem for static applications and therefore requires a different computer implementation.

For fourth-order plate and shell problems, the basic FETI method is not numerically scalable, and therefore is not as performing as for second-order problems. However, it was shown in References [29–31] that if at each PCG iteration the exact continuity of the (transverse) displacement field is enforced at the subdomain crosspoints, the FETI method becomes numerically scalable also for fourth-order (plate) and shell problems. Such a condition can be satisfied by introducing a set of additional Lagrange multipliers at the subdomain crosspoints, and determining their values at each PCG iteration by solving another coarse problem that is based on both the subdomain rigid body and ‘corner’ modes. Because this new coarse problem can be expressed as the projection on a subspace of the interface problem associated with the basic FETI method, the resulting DD method was called in References [30, 31] the two-level FETI method (FETI-2). In Reference [32], it was proved that for fourth-order plate problems, the optimal conditioning result (1) holds also for the interface problem of the FETI-2 method equipped with the Dirichlet preconditioner. In Reference [33], the FETI-2 method developed in References [30, 31] was further expanded into a mathematical framework for unifying all previously developed FETI coarse problems, and paving the way for new ones. Nevertheless, for the same reason as stated above, the two-level FETI framework cannot unify the static and dynamic FETI methods.

For a given uniform mesh partition with  $N_s$  subdomains, the size of the coarse problem of the FETI-2 method, which contains both the subdomain rigid body and corner modes, is of the order of  $6N_s$  for plate bending problems, and  $15N_s$  for shell problems. Hence, this coarse problem is at least twice as large as that of the basic FETI method, but the latter method is optimal only for second-order problems. For this and other reasons, while the basic FETI method applied to

the solution of three-dimensional problems discretized by solid elements outperforms sparse direct solvers on both sequential and parallel machines by a factor typically ranging between three and a full-order of magnitude, the FETI-2 method equipped with the corner modes and applied to the solution of shell problems is twice slower than a sparse direct solver on sequential machines, and only 25 per cent faster on parallel processors [17, 34].

From the above summary of the history of the development of the FETI methodology, it follows that improving the computational performance of the FETI-2 algorithm for plate and shell problems and unifying static and dynamic FETI solvers remain two desirable objectives. The purpose of this paper is to report on recent progress towards achieving these goals. The paper is organized as follows.

In Section 2, we present a new substructuring approach that is based on a dual–primal formulation of the FETI-2 concept, and which replaces the one- and two-level static and transient FETI algorithms by a single dual–primal FETI-DP method. We point out a few interesting features of this new DD method and discuss their impact on robustness and computational complexity. In Section 3, we demonstrate numerically the scalability of the FETI-DP method for both second- and fourth-order problems. In Section 4, we highlight its superior CPU performance for realistic structural applications. In Section 5, we point out an outstanding issue, and in Section 6 we conclude this paper.

## 2. A DUAL-PRIMAL DOMAIN DECOMPOSITION METHOD

### 2.1. Formulation

Let  $\Omega$  denote the computational support of a structural problem, and  $\{\Omega^s\}_{s=1}^{s=N_s}$  its decomposition into  $N_s$  subdomains with matching interfaces. We denote by  $K^s, u^s$ , and  $f^s$  the stiffness matrix, and displacement and force vectors associated with subdomain  $\Omega^s$ , respectively. These quantities can be partitioned as follows:

$$K^s = \begin{bmatrix} K_{ii}^s & K_{ib}^s \\ K_{ib}^{sT} & K_{bb}^s \end{bmatrix}, \quad u^s = \begin{bmatrix} u_i^s \\ u_b^s \end{bmatrix}, \quad f^s = \begin{bmatrix} f_i^s \\ f_b^s \end{bmatrix} \quad (2)$$

where the subscripts  $i$  and  $b$  designate the subdomain internal and interface boundary degrees of freedom (d.o.f.), and the superscript T designates the transpose operation. We furthermore partition the component  $u_b^s$  as follows:

$$u_b^s = \begin{bmatrix} u_{b_r}^s \\ u_{b_c}^s \end{bmatrix} \quad (3)$$

where the additional subscript  $c$  designates the d.o.f. attached to some ‘corners’ of the mesh decomposition, and the additional subscript  $r$  designates the remainder of the interface boundary d.o.f. Here, we define the corners of a mesh partition either as

- D1. its crosspoints—that is, the points belonging to more than two subdomains (Figure 1), or
- D2. the set of nodes located at the beginning and end of each edge of each subdomain (Figure 2).

Definition D1 is valid for both two- and three-dimensional meshes, but produces a large number of corner points in three dimensions. Definition D2 holds only for two- and two-and-a-half

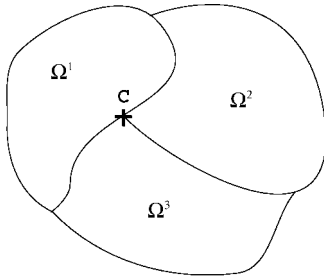


Figure 1. Definition D1 of a corner point.

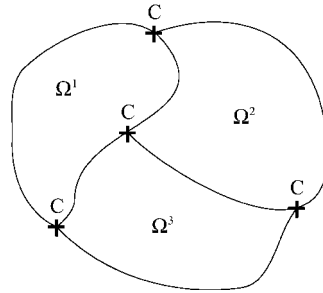


Figure 2. Definition D2 of a corner point.

dimensional problems. When definition D2 is adopted, a significant number of corner points can belong to two and only two subdomains.

The continuity of the displacement field at the subdomain interfaces can be written as

$$u_b^s - u_b^q = 0 \quad \text{on } \Omega^s \cap \Omega^q$$

or

$$\sum_{s=1}^{s=N_s} B^s u^s = 0 \tag{4}$$

where  $B^s$  is a signed boolean matrix defined by

$$B^s u^s = \pm u_b^s \tag{5}$$

and the sign of this equality is determined by a suitable convention. Following the FETI methodology, we seek to introduce Lagrange multipliers  $\lambda$  for enforcing the continuity condition (4), and employ a PCG algorithm for determining the values of these multipliers.

More specifically, we are interested in developing a FETI-2-like method where at each PCG iteration, the continuity of some or all components of the displacement field at the corner nodes is exactly satisfied. In References [29–32], this requirement was shown to play a crucial role in ensuring the numerical scalability of the FETI method for fourth-order plate and shell problems, and was enforced by correcting at each PCG iteration the Lagrange multipliers to achieve this objective.

In this paper, we adopt an alternative approach for ensuring that at each PCG iteration some or all components of the displacement field are continuous at the corner nodes. Basically, we formulate a DD method where these unknowns are defined only at the global level, whereas all other generalized displacement unknowns are defined at the subdomain level.

For this purpose, we re-partition  $K^s$ ,  $u^s$  and  $f^s$  as

$$K^s = \begin{bmatrix} K_{rr}^s & K_{rc}^s \\ K_{rc}^{sT} & K_{cc}^s \end{bmatrix}, \quad u^s = \begin{bmatrix} u_r^s \\ u_{b_c}^s \end{bmatrix}, \quad f^s = \begin{bmatrix} f_r^s \\ f_{b_c}^s \end{bmatrix} \tag{6}$$

where

$$u_r^s = \begin{bmatrix} u_i^s \\ u_{b_r}^s \end{bmatrix} \quad \text{and} \quad f_r^s = \begin{bmatrix} f_i^s \\ f_{b_r}^s \end{bmatrix}$$

and introduce the global vector of corner d.o.f.

$$u_c = \begin{bmatrix} u_c^1 \\ \vdots \\ u_c^j \\ \vdots \\ u_c^{N_c} \end{bmatrix} \tag{7}$$

where  $u_c^j$  denotes a subset or all of the displacement d.o.f. attached to the  $j$ th global node that is also a corner node of the mesh decomposition, and  $N_c$  denotes the total number of selected corner nodes. For each subdomain  $\Omega^s$ , we also define two additional boolean matrices  $B_r^s$  and  $B_c^s$  by

$$B_r^s u_r^s = \pm u_{b_r}^s \quad \text{and} \quad B_c^s u_c = u_{b_c}^s \tag{8}$$

Using the notation of Equations (6)–(8), the subdomain equations of equilibrium can be written as

$$\begin{aligned} K_{rr}^s u_r^s + K_{rc}^s B_c^s u_c &= f_r^s - B_r^{sT} \lambda \\ \sum_{s=1}^{s=N_s} B_c^{sT} K_{rc}^{sT} u_r^s + \sum_{s=1}^{s=N_s} B_c^{sT} K_{cc}^s B_c^s u_c &= \sum_{s=1}^{s=N_s} B_c^{sT} f_{b_c}^s = f_c \end{aligned} \tag{9}$$

and the interface continuity condition (4) can be re-written as

$$\sum_{s=1}^{s=N_s} B_r^s u_r^s = 0 \tag{10}$$

In this work, we propose to include in  $u_c^j$  all the displacement and rotational d.o.f. that are attached to the  $j$ th global node that is also a corner node of the mesh decomposition. For restrained plate and shell as well as plane stress/strain problems, this guarantees that the submatrix  $K_{rr}^s$  is non-singular, whether Definition D1 or Definition D2 is adopted for identifying the corner nodes. The case of three-dimensional problems discretized by solid elements is discussed in Section 5. In all cases, the matrix

$$K_{cc} = \sum_{s=1}^{s=N_s} B_c^{sT} K_{cc}^s B_c^s \tag{11}$$

is also non-singular. Hence, from Equations (9) and (11) it follows that

$$u_r^s = K_{rr}^{s-1} \left( f_r^s - B_r^{sT} \lambda - K_{rc}^s B_c^s u_c \right) \tag{12}$$

Substituting Equation (12) into Equation (10) leads after some algebraic transformations to

$$\begin{bmatrix} F_{I_{rr}} & F_{I_{rc}} \\ F_{I_{rc}}^T & -K_{cc}^* \end{bmatrix} \begin{bmatrix} \lambda \\ u_c \end{bmatrix} = \begin{bmatrix} d_r \\ f_c^* \end{bmatrix}$$

where

$$\begin{aligned}
 F_{I_{rr}} &= \sum_{s=1}^{s=N_s} B_r^s K_{rr}^{s-1} B_r^{sT} \\
 F_{I_{rc}} &= \sum_{s=1}^{s=N_s} B_r^s K_{rr}^{s-1} K_{rc}^s B_c^s \\
 K_{cc}^* &= K_{cc} - \sum_{s=1}^{s=N_s} (K_{rc}^s B_c^s)^T K_{rr}^{s-1} (K_{rc}^s B_c^s) \\
 d_r &= \sum_{s=1}^{s=N_s} B_r^s K_{rr}^{s-1} f_r^s \\
 f_c^* &= f_c - \sum_{s=1}^{s=N_s} B_c^{sT} K_{rc}^{sT} K_{rr}^{s-1} f_r^s
 \end{aligned} \tag{13}$$

The above problem is a dual-primal problem as it relates the dual Lagrange multiplier unknowns  $\lambda$  to the primal displacement d.o.f.  $u_c$ . By eliminating  $u_c$ , it can be transformed however into the following *symmetric positive* definite dual interface problem

$$\left( F_{I_{rr}} + F_{I_{rc}} K_{cc}^{*-1} F_{I_{rc}}^T \right) \lambda = d_r - F_{I_{rc}} K_{cc}^{*-1} f_c^* \tag{14}$$

which is closely related to the original FETI interface problem [1–6]. Indeed,  $F_{I_{rr}}$  and  $d_r$  are the restrictions to the ‘ $r$ ’ d.o.f. of the complete  $F_I$  operator and  $d$  vector introduced in References [1–6], respectively.

### 2.2. A unified FETI-DP method and its coarse problem

We now define the FETI-DP method as (a) the transformation of a given problem of the form  $Ku = f$  into the dual interface problem (14), using the primal-dual DD method described in Section 2.1, and (b) the solution of that interface problem by a PCG algorithm.

At each PCG iteration  $k$ , the residual must be computed by a matrix-vector product of the form  $\left( F_{I_{rr}} + F_{I_{rc}} K_{cc}^{*-1} F_{I_{rc}}^T \right) \lambda^k$ , which can be evaluated in two steps as follows:

- S1:  $\delta^k = F_{I_{rr}} \lambda^k = \sum_{s=1}^{s=N_s} B_r^s K_{rr}^{s-1} B_r^{sT} \lambda^k$
- S2:  $\delta^k = \delta^k + F_{I_{rc}} K_{cc}^{*-1} F_{I_{rc}}^T \lambda^k$

Step S1 is similar to the main step of the original FETI method ( $F_I \lambda^k$ ) and can be implemented using practically the same segments of code. It is easily parallelizable because it involves only subdomain level computations—essentially, local solves—and requires communication only between neighbouring subdomains.

Step 2 can be split into the following 3 substeps:

$$\text{S2-1: } y^k = F_{rc}^\top \lambda^k = \sum_{s=1}^{s=N_s} B_c^{s\top} K_{rc}^{s\top} K_{rr}^{s-1} B_r^{s\top} \lambda^k$$

$$\text{S2-2: Solve } K_{cc}^* x^k = y^k$$

$$\text{S2-3: } z^k = F_{rc} x^k = \sum_{s=1}^{s=N_s} B_r^s K_{rr}^{s-1} K_{rc}^s B_c^s x^k$$

Steps S2-1 and S2-3 involve only local computations that can be parallelized at the subdomain level. The product  $K_{rr}^{s-1} B_r^{s\top} \lambda^k$  has already been evaluated in step S1, and the product  $K_{rr}^{s-1} K_{rc}^s$  is performed only once. Step S2-2 can be interpreted as the solution of an auxiliary problem whose size is smaller or equal to  $6N_c$ . Hence, this auxiliary problem is a coarse problem that we refer to as the FETI-DP coarse problem.

Using Equations (11) and (13), step S2-2 can be re-written as

$$\text{Solve } \left[ \sum_{s=1}^{s=N_s} B_c^{s\top} K_{cc}^s B_c^s - (K_{rc}^s B_c^s)^\top K_{rr}^{s-1} (K_{rc}^s B_c^s) \right] x^k = y^k \tag{15}$$

which shows that the solution of the FETI-DP coarse problem couples the subdomain computations, and therefore propagates the error globally at each PCG iteration.

From Equation (15), it also follows that the FETI-DP coarse problem can be constructed in parallel using subdomain-by-subdomain computations. Its governing matrix  $K_{cc}^*$  is a sparse matrix whose pattern is that of a stiffness matrix obtained by considering only the superelements defined by the corner nodes. In this work, we factor  $K_{cc}^*$  by a parallel algorithm [9], and solve at each PCG iteration the coarse problem (15) by forward and backward substitutions.

We propose the FETI-DP method as an iterative solver for both second- and fourth-order problems—that is for all of plane stress/strain, three-dimensional solid, and beam, plate and shell problems. The FETI-DP coarse problem is the same in all cases. Only its size changes. For plane stress/strain problem, its size is equal to  $2N_c$ , and for three-dimensional solid problems it is equal to  $3N_c$ . For beam, plate, and shell problems, the rotational d.o.f. are included in the definition of  $u_c$ , which increases the size of the FETI-DP coarse problem to  $6N_c$ .

An important observation is that unlike in the original one- and two-level FETI methods, the subdomain problems associated with the FETI-DP method described in this paper are always non-singular. For this reason, the FETI-DP coarse problem (14) does not rely on the null spaces of the subdomain problem matrices, and therefore the same FETI-DP solver can be equally applied to the solution of static and transient dynamics problems. This is in sharp contrast with the original FETI methods that have different coarse problems for static and transient dynamics applications [14, 33].

In summary, we propose the FETI-DP method as a single FETI solver for static and dynamic second- and fourth-order structural problems.

### 2.3. Preconditioners

In this work, we do not develop a truly new preconditioner for the FETI-DP method. Rather, we extend the well-known lumped [1–4] and Dirichlet [5] preconditioners to the dual-primal DD formulation described in Section 2.1.

Hence, we define the FETI-DP Dirichlet preconditioner as

$$\bar{F}_{rr}^{D-1} = \sum_{s=1}^{N_s} W^s B_r^s \begin{bmatrix} 0 & 0 \\ 0 & S_{b_r b_r}^s \end{bmatrix} B_r^{s\top} W^s \tag{16}$$



where

$$S_{b_r, b_r}^s = K_{b_r, b_r}^s - K_{i b_r}^{s \top} K_{i i}^{s-1} K_{i b_r}^s$$

and the FETI-DP lumped preconditioner as

$$\bar{F}_{I_r}^{L-1} = \sum_{s=1}^{N_s} W^s B_r^s \begin{bmatrix} 0 & 0 \\ 0 & K_{b_r, b_r}^s \end{bmatrix} B_r^{s \top} W^s \quad (17)$$

where the subscripts  $r$ ,  $b_r$  and  $i$  designate the d.o.f. partitionings implied by Equations (6), (3) and (2), respectively, and  $W^s$  is a scaling diagonal subdomain matrix that accounts for eventual subdomain heterogeneities [20]. Note that  $K_{i i}^s$  is non-singular because it is the stiffness matrix of  $\Omega^s$  with all interface boundary d.o.f. fixed, and thus  $K_{i i}^{s-1}$  exists.

The reader can observe that the lumped preconditioner is obtained by simplifying the primal Schur complement  $S_{b_r, b_r}^s$  of the Dirichlet preconditioner to its leading term  $K_{b_r, b_r}^s$ . This simplification reduces the arithmetic complexity of the preconditioning step.

We also remind the reader that in the context of the original FETI methods, the Dirichlet preconditioner is mathematically optimal. It is more computationally expensive than the lumped preconditioner, but is more computationally efficient for plate and shell problems. On the other hand, the lumped preconditioner is not mathematically optimal, but is more computationally efficient than the Dirichlet preconditioner for second-order problems.

#### 2.4. Some comments on complexity and robustness

Consider a two-dimensional uniform mesh partitioned into  $1/H \times 1/H$  subdomains ( $N_s = 1/H^2$ ), and a three-dimensional uniform mesh partitioned into  $1/H \times 1/H \times 1/H$  subdomains ( $N_s = 1/H^3$ ).

For second-order problems, the size of the rigid-body-mode-based coarse problem of the one-level FETI method is in general equal to  $3N_s = 3/H^2$  in the two-dimensional case, and to  $6N_s = 6/H^3$  in the three-dimensional one. On the other hand, the size of the coarse problem of the FETI-DP method is equal to  $2N_c \approx 2/H^2$  in the two-dimensional case, and  $3N_c \approx 3/H^3$  in the three-dimensional one. Hence, for second-order problems, the coarse problem of the FETI-DP method is 1.5 times smaller than that of the one-level FETI method in two dimensions, and 2 times smaller in the three dimensions. Both coarse problems are sparse. However, at each PCG iteration, the coarse problem of the one-level FETI method must be solved twice for symmetry reasons, whereas the coarse problem of the FETI-DP method needs to be solved only once.

For fourth-order problems defined on the two-dimensional mesh considered above (or a similar two-and-a-half dimensional mesh), the size of the corner mode based coarse problem of the FETI-2 method is of the order  $6N_s = 6/H^2$  for plate bending problems, and  $15N_s = 15/H^2$  for shell problems. The coarse problem of the FETI-DP method is smaller: it is equal to  $3N_c \approx 3/H^2$  for plate bending problems, and  $6N_c \approx 6/H^2$  for shell problems. The corner mode based coarse problem of the FETI-2 method is sparse but indefinite; its solution deserves special attention. On the other hand, the coarse problem of the FETI-DP method is sparse and positive, and therefore can be handled by an off-the-shelf sparse solver. Furthermore, at each PCG iteration of the FETI-2 method, both rigid body mode and corner mode based corner problems must be solved. On the other hand, at each PCG iteration of the FETI-DP method, only one coarse problem (15) needs to be solved.

Noting the role of the quantity  $1/H$  in the above remarks, we formulate the following expectations

- E1: For second-order problems, the computational overhead associated with the solution at each PCG iteration of one or several coarse problems is smaller for the FETI-DP method than for the one-level FETI method. Assuming similar convergence rates, the FETI-DP method can be expected to significantly improve the performance of the one-level FETI method, for example by a factor two, when the number of subdomains is very large—say, of the order of a thousand.
- E2: For fourth-order plate and shell problems, the computational overhead associated with the solution at each PCG iteration of one or several coarse problems is significantly smaller for the FETI-DP method than for the FETI-2 method. Assuming similar convergence rates, the FETI-DP method can be expected to significantly improve the performance of the FETI-2 method, for example by a factor two, when the number of subdomains is reasonably large—say, of the order of a few hundreds.

Hence, the FETI-DP method offers important computational advantages per iteration over the previously established FETI methods. It remains however to investigate whether the FETI-DP method is numerically scalable and delivers iteration counts that are similar to those of the FETI and FETI-2 methods.

We have already noted in Section 2.2 that unlike in the FETI and FETI-2 methods, the null spaces of the subdomain problem matrices do not play any role in the present formulation of the FETI-DP method. This is because the interface problem (14) does not require any self-equilibrium condition to be satisfied by the Lagrange multipliers  $\lambda$ , which bypasses the necessity for computing the subdomain rigid body modes. Because extracting these null spaces by a ‘bullet-proof’ procedure is a challenging task [35], particularly for geometrically non-linear problems where some tangent stiffness matrices can lose one or several rotational rigid body modes, the FETI-DP solver offers a robust alternative to the one- and two-level FETI solvers.

### 3. NUMERICAL SCALABILITY

Next, we investigate the numerical scalability of the FETI-DP method with respect to the mesh size  $h$ , the subdomain size  $H$ , and the ratio  $H/h$ . These parameters characterize the problem size, the number of subdomains, and the number of elements per subdomain, respectively. For this purpose, we focus on a two-dimensional squared domain whose side is denoted by  $a$ , and which is discretized by  $1/h \times 1/h$  elements and partitioned into  $1/H \times 1/H$  subdomains. We consider both second-order elasticity and fourth-order plate problems. In all cases, we denote by  $N_{\text{dof}}$  and  $N_{\text{coarse}}$  the sizes of the global and coarse problems, respectively, adopt Definition D2 (see Section 2.1) for the corner points, and monitor the convergence of the iterative solvers using the following criterion:

$$\|Ku - f\|_2 \leq \varepsilon \times \|f\|_2 \quad (18)$$

where  $K$ ,  $u$ , and  $f$  denote the global stiffness matrix and displacement and force vectors, and  $\varepsilon = 10^{-6}$  except where otherwise specified.

Table I. Plane stress problem— $N_s = 64$  subdomains ( $8 \times 8$  mesh partition). Comparison of the iteration counts of the one-level FETI and FETI-DP methods equipped with their respective Dirichlet preconditioners.

$h$	$N_{\text{dof}}$	FETI # iterations	FETI-DP # iterations
1/40	3362	18	23
1/80	13 122	14	17
1/160	51 842	17	20
1/320	206 082	22	23
1/640	821 762	25	26

Table II. Plane stress problem— $N_{\text{dof}} = 821\,762$  ( $h = 1/640$ ). Comparison of the iteration counts of the one-level FETI and FETI-DP methods equipped with their respective Dirichlet preconditioners.

$H$	$N_s$	FETI # iterations	FETI-DP # iterations
1/10	100	26	27
1/16	256	28	26
1/20	400	27	25
1/40	1600	26	22
1/64	4096	23	19
1/128	16 384	18	16

### 3.1. Second-order elasticity problems

Here, we consider a plane stress problem. We set  $a=1$ , clamp one side of the square, and apply a distributed axial load on the opposite side. We set Young's modulus to  $E = 1.0e+07$ , and Poisson's ratio to  $\nu = 0.3$ . We consider several values of  $h$  and  $H$ , and in each case, solve the resulting systems of equations using both the one-level FETI and FETI-DP methods equipped with their respective Dirichlet preconditioners.

First, we fix  $H = 8$  ( $N_s = 64$  subdomains), and vary  $h$ . We report in Table I the performance results obtained for the FETI and FETI-DP methods. These results suggest that the FETI-DP method is as numerically scalable with respect to the mesh size  $h$  as the one-level FETI method.

Next, we fix  $h = 1/640$  ( $N_{\text{dof}} = 821\,762$ ) and vary the number of subdomains  $N_s$  by varying  $H$ . The performance results we report in Table II suggest that the FETI-DP method is not only numerically scalable with respect to the number of subdomains, but also outperforms the one-level FETI method iteration-wise.

Finally, we vary both  $h$  and  $H$  while keeping the ratio  $H/h = 10$ —that is, keeping the number of elements per subdomain constant and equal to 100. We report in Table III the performance results obtained for both the FETI and FETI-DP methods. These results suggest that the FETI-DP method is as numerically scalable with respect to the subdomain problem size as the one-level FETI method, and even delivers a better iteration count for large-scale problems. The reader can also observe that as predicted, the size  $N_{\text{coarse}}$  of the coarse problem of the FETI-DP method is in general 1.5 times smaller than that of the one-level FETI method for two-dimensional second-order elasticity problems.

Table III. Plane stress problem— $H/h = 10$  (100 elements per subdomain). Comparison of the iteration counts of the one-level FETI and FETI-DP methods equipped with their respective Dirichlet preconditioners.

$h$	$N_s$	$N_{\text{dof}}$	FETI $N_{\text{coarse}}$	FETI-DP $N_{\text{coarse}}$	FETI # iterations	FETI-DP # iterations
1/20	4	882	6	8	8	8
1/40	16	3362	36	36	12	14
1/80	64	13 122	168	140	14	17
1/160	256	51 842	720	540	18	18
1/320	1024	206 082	2976	2108	23	18
1/640	4096	821 762	12 096	8316	23	19

Table IV. Plate bending problem— $N_s = 64$  subdomains ( $8 \times 8$  mesh partition). Comparison of the iteration counts of the FETI-2 and FETI-DP methods equipped with their respective Dirichlet preconditioners.

$h$	$N_{\text{dof}}$	FETI-2 (a) # iterations	FETI-2 (b) # iterations	FETI-DP # iterations
1/40	5166	23	12	17
1/80	19 926	30	16	22
1/160	78 246	36	20	28
1/320	310 086	44	23	34
1/640	1 234 566	51	29	41

### 3.2. Fourth-order plate bending problems

Next, we consider a plate bending problem with the same geometry ( $a = 1$ ) and material properties as in the previous section. We set the thickness of the plate to  $t = 10^{-3}$ , clamp one of its sides, and apply a distributed bending load on the opposite side. We discretize this problem by triangular plate bending elements with 3 d.o.f. per node. Again, we consider several values of  $h$  and  $H$ , and in each case, solve the resulting systems of equations using both the FETI-2 and FETI-DP methods equipped with their respective Dirichlet preconditioners. For the FETI-2 method, we consider two options where (a) the continuity of the transverse displacement d.o.f. at the corners is ensured at each iteration, and (b) the continuity of all 3 d.o.f. at the corners is ensured at each iteration. Option (a) suffices to ensure the numerical scalability of the FETI-2 method [30]. Option (b) serves the purpose of further comparisons with the FETI-DP method whose coarse problem, in this case, is based on 3 d.o.f. per corner node.

First, we fix  $H = 8$  ( $N_s = 64$  subdomains), and vary  $h$ . We report in Table IV the performance results obtained for the FETI-2 and FETI-DP methods. These results suggest that the FETI-DP method is as numerically scalable with respect to the mesh size  $h$  as the FETI-2 method, and delivers an iteration count that is halfway between that of the FETI-2 method option (a), and that of the FETI-2 method option (b).

Next, we fix  $h = 1/640$  ( $N_{\text{dof}} = 1,234,566$ ) and vary the number of subdomains  $N_s$  between 64 and 4096 by varying  $H$  between  $1/8$  and  $1/64$ . The performance results we report in Table V suggest that the FETI-DP method is numerically scalable with respect to the number of subdomains.

Table V. Plate bending problem— $N_{\text{dof}} = 1\,234\,566$  ( $h = 1/640$ ). Comparison of the iteration counts of the FETI-2 and FETI-DP methods equipped with their respective Dirichlet preconditioners.

$H$	$N_s$	FETI-2 (a) # iterations	FETI-2 (b) # iterations	FETI-DP # iterations
1/8	64	51	29	41
1/10	100	47	28	40
1/16	256	47	27	37
1/20	400	47	25	35
1/40	1600	40	21	30
1/64	4096	36	20	28

Table VI. Plate bending problem— $H/h = 10$  (100 elements per subdomain). Comparison of the iteration counts of the FETI-2 and FETI-DP methods equipped with their respective Dirichlet preconditioners.

$h$	$N_s$	$N_{\text{dof}}$	FETI-2 (a) # iterations $N_{\text{coarse}}$	FETI-2 (b) # iterations $N_{\text{coarse}}$	FETI-DP # iterations $N_{\text{coarse}}$
1/20	4	1386	(12) 12	(24) 10	(12) 12
1/40	16	5166	(72) 24	(144) 14	(54) 19
1/80	64	19926	(336) 30	(672) 16	(210) 22
1/160	256	78246	(1440) 32	(2880) 16	(810) 24
1/320	1024	310086	(5952) 34	(11904) 18	(3162) 25
1/640	4096	1234566	(24192) 36	(48384) 20	(12474) 28

It performs a number of iterations that is halfway between those of the two options of the FETI-2 method. Note that for all considered FETI solvers, the number of iterations decreases when  $N_s$  increases.

Finally, we vary both  $h$  and  $H$  while keeping the ratio  $H/h = 10$ —that is, keeping the number of elements per subdomain constant and equal to 100. We report in Table VI the performance results obtained for both the FETI-2 and FETI-DP methods. We also report between parentheses the different sizes of the different coarse problems associated with the different FETI methods. Again, these performance results suggest that the FETI-DP method is as numerically scalable with respect to the subdomain problem size as the FETI-2 method. As in both previous numerical scalability tests (see Table IV and V), it delivers an iteration count that is halfway between that of the FETI-2 method option (a), and that of the FETI-2 method option (b). However, the reader can also observe in Table VI that the coarse problem of the FETI-DP method grows only as  $O(3N_s)$  and is smaller than that of the FETI-2 method option (a) which grows as  $O(6N_s)$ . It is also significantly smaller than that of the FETI-2 method option (b) which grows as  $O(12N_s)$ . For this reason, and because the FETI-2 methods must also solve twice at each PCG iteration the additional rigid-body-mode-based coarse problem (see Section 2.4), one can reasonably expect the FETI-DP solver to perform better CPU wise than both options of the FETI-2 method. This is also confirmed by the CPU performance results reported in Section 4.1.

Table VII. Plane stress problem— $N_{\text{dof}} = 821\,762$  ( $h = 1/640$ ). Performance comparisons on an 8-processor Origin 2000 of the one-level FETI and FETI-DP methods equipped with their respective Dirichlet preconditioners.

$H$	$N_s$	FETI # iterations	FETI-DP # iterations	FETI CPU (s)	FETI-DP CPU (s)
1/10	100	26	27	157	142
1/16	256	28	26	87	58
1/20	400	27	25	50	41
1/40	1600	26	22	47	25
1/64	4096	23	19	66	24
1/128	16384	18	16	262	57

#### 4. APPLICATIONS AND PERFORMANCE RESULTS

Finally, we report on the CPU performance of the FETI-DP method for a series of realistic problems. More specifically, we benchmark this new FETI method against the original FETI and FETI-2 methods, and the PSLDLT sparse direct solver. In all cases, we monitor the convergence of the FETI solvers using the criterion (18) with  $\varepsilon = 10^{-6}$ .

We perform all computations in 64-bit arithmetic on an Origin 2000 Silicon Graphics system with 24 processors and 12 gigabytes of memory. The PSLDLT sparse direct solver is part of the scientific library of the Silicon Graphics systems. It requires storing the target sparse matrix in the Harwell-Boeing format (also known as the Compressed Column Storage format). It is parallelized by Silicon Graphics on the O2000 multiprocessor, and is most likely written in assembly language. All FETI solvers are programmed in C++ and are not optimized for any specific computer architecture. All FETI computational kernels are parallelized except the forward and backward substitutions associated with the coarse problems of the one-level FETI and FETI-DP methods. These sparse triangular solves are performed in serial mode because they do not parallelize well. On the other hand, the forward and backward substitutions associated with the coarse problem of the FETI-2 method are parallelized; this is because the latter coarse problem gives rise to dense block algebra that parallelizes relatively well.

We also note that the sizes of the problems discussed here are such that memory swapping never occurs on our Origin 2000 system, and therefore the performance results we report are purely computational performance results.

##### 4.1. Back to the sample plane stress problem

First, we consider again the sample plane stress problem described in Section 3.1, and update Tables II and III with the corresponding CPU performance results.

For both the FETI and FETI-DP methods, increasing the number of subdomains for a fixed problem size reduces the computational complexity of the solution of the local problems, but increases the size of the interface problem. Hence, for both of these numerically scalable DD methods, increasing the number of subdomains up to a certain optimal number  $N_s^{\text{opt}}$  can decrease the total solution CPU time. Increasing  $N_s$  beyond that optimal number increases the total solution CPU time. The results reported in Table VII illustrate this behaviour and suggest that for this problem, the optimal number of subdomains for the FETI method is approximately  $N_s^{\text{opt}} = 1600$ ,

Table VIII. Plane stress problem— $H/h = 10$  (100 elements per subdomain). Performance comparisons on a single processor O2000 of the one-level FETI and FETI-DP methods equipped with their respective Dirichlet preconditioners.

$h$	$N_s$	$N_{\text{dof}}$	FETI # iterations	FETI-DP # iterations	FETI CPU (s)	FETI-DP CPU (s)
1/20	4	882	8	8	0.06	0.05
1/40	16	3362	12	14	0.24	0.24
1/80	64	13 122	14	17	1.3	1.5
1/160	256	51 842	18	18	13	13
1/320	1024	206 082	23	18	47	43
1/640	4096	821 762	23	19	252	192

while the optimal number of subdomains for the FETI-DP method is approximately  $N_s^{\text{opt}} = 4096$ . This suggests that the FETI-DP method is more suitable for fine mesh partitions than the FETI method, and therefore more suitable for massively parallel systems with thousands of processors such as the ASCI Option Red supercomputer.

The performance results reported in Table VII are for an 8-processor Origin 2000 configuration. They show that for a fixed problem size, the FETI-DP method equipped with its optimal number of subdomains is twice as fast as the FETI method equipped with its optimal number of subdomains. On the other hand, the performance results reported in Table VIII are for a single Origin 2000 processor. They suggest that for a fixed subdomain problem size and an increasing number of subdomains, the FETI-DP method is 30 per cent faster than the one-level FETI method.

As stated earlier, all FETI coarse problems are currently solved by a direct method where the factorization phase is performed once and in parallel, but all subsequent sparse forward and backward substitutions are performed in serial mode—because they do not parallelize well. Consequently, the parallel speed-up of a FETI solver is larger when the percentage of the total CPU time spent in the coarse problem solver is smaller—that is, when the coarse problem is smaller. This is illustrated by the performance results reported in Table VII for  $N_s = 4096$  and 8 processors, and Table VIII for  $h = 1/640$  and a single processor. These results show that for this plane stress problem, when the number of subdomains is as large as  $N_s = 4096$ , the FETI-DP method delivers a speed-up equal to 8 on an 8-processor O2000, but the one-level FETI method whose coarse problem is 1.5 times larger and must be solved twice at each PCG iteration delivers a speed-up equal to 3.8 only.

#### 4.2. A composite stiffened wing panel

Next, we consider the stress analysis of a stiffened composite wing panel from the V22 tiltrotor aircraft [36]. This panel contains interesting features such as ply drop-offs, ply interleaves, axial stiffeners, transverse ribs, clips, brackets, a large elliptical access hole, and 46 different material section properties. We design three different finite element models using two-noded beam, and three-noded triangular plate and shell elements: (a) model M1 with 56 916 d.o.f. (Figure 3), (b) model M2 with 223 620 d.o.f., and (c) model M3 with 885 924 d.o.f. We clamp the panel at one end, and constrain to zero the out-of-plane displacements and rotations about the longitudinal axis at the midpoint of the top of the fixture brackets. We specify a uniform end-shortening displacement field at the other end of the panel to simulate the compressive motion of a test machine.

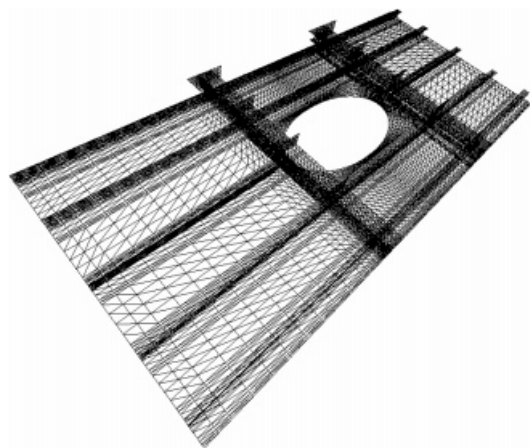


Figure 3. Finite element discretization of a V22 stiffened wing panel (model M1).

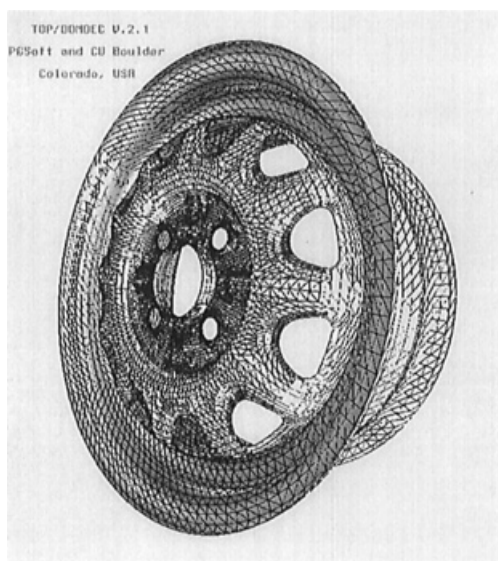


Figure 4. Finite element discretization of a car wheel.

Table IX. Stiffened composite wing panel from the V22 tiltrotor aircraft Performance comparisons on a 10-processor O2000 of the FETI-2 and FETI-DP methods equipped with their respective Dirichlet preconditioners.

FE	$N_{\text{dof}}$	$N_s$	FETI-2 $N_{\text{coarse}}$	FETI-DP $N_{\text{coarse}}$	FETI-2 # iter.	FETI-DP # iter.	FETI-2 CPU (s)	FETI-DP CPU (s)
M1	56 916	80	1656	1680	93	104	17	13
M2	223 620	140	2949	2796	129	130	77	55
M3	885 924	250	5348	4686	200	198	361	269

We perform several stress analyses of this panel on a 10-processor Origin 2000 system using both the FETI-2 and FETI-DP methods equipped with their respective Dirichlet preconditioners, and Definition D2 of the corner nodes. We generate all mesh partitions using the TOP/DOMDEC software package and its option for optimizing the subdomain aspect ratio [37]. For each problem size, we select a number of subdomains that is as optimal as possible for both the FETI-2 and FETI-DP solvers.

The performance results we report in Table IX show that

- (1) The FETI-2 and FETI-DP solvers deliver comparable iteration counts for all three finite element models. However, the FETI-DP solver is systematically 1.35 times faster than the FETI-2 solver.
- (2) Both the FETI-2 and FETI-DP solvers exhibit a CPU performance that scales reasonably well with the problem size. Indeed, when the size of this complex problem is increased by factors equal to 4 and 16, the CPU timings of both the FETI-2 and FETI-DP solvers increase by factors equal to 4 and 21, respectively.



Table X. Car wheel problem with 936 102 d.o.f. Parallel performance results on an O2000 system of the PSLDLT solver, and the FETI-2 and FETI-DP solvers equipped with their respective Dirichlet preconditioners.

$N_p$	FETI-2 (s)	PSLDLT (s)	FETI-DP (s)
1	2995	1631	1594
4	789	502	370
8	371	301	196
16	214	218	116
20	179	200	99
24	157	200	86

For the finite element model M3 of this specific shell problem, the FETI-2 solver stagnates at  $\varepsilon = 7.6e - 7$  (see criterion (18)). On the other hand, the FETI-DP solver stagnates much later at  $\varepsilon = 6.6e - 11$ . This suggests that the FETI-DP solver is not only faster than the FETI-2 solver, but can also reach significantly lower residuals.

Remark.. We note that for this problem and the mesh partitions generated for all three finite element models, the coarse problem of the FETI-2 method turns out to be not much larger than that of the FETI-DP method. This is because for all generated mesh partitions, it turns out that about 50 per cent of the corner points belong to two and only to two subdomains (see Definition D2 in Section 2.1). Each of these corner points contributes 3 rather than 9 unknowns to the FETI-2 coarse problem, while any corner point contributes 6 unknowns to the FETI-DP coarse problem. Consequently for this problem and the considered mesh partitions, the size of the FETI-DP coarse problem is only slightly smaller than that of the FETI-2 coarse problem, and the FETI-DP solver is only 1.35 times faster than the FETI-2 solver.

#### 4.3. A car wheel

Finally, we consider the stress analysis of an alloy wheel clamped at a few centre points and loaded by a set of concentrated forces at its top rim. The finite element model of this wheel contains 313 856 three-noded triangular shell elements and 936 102 d.o.f. (Figure 4). This problem is representative of a class of two-and-a-half-dimensional shell problems for which sparse solvers are in principle more computationally efficient than most iterative solvers. A preliminary investigation indicated that the optimal number of subdomains for the solution of this problem by the FETI-2 method equipped with the Dirichlet preconditioner and the corner modes is  $N_s^{\text{opt}} = 175$ , and that of the FETI-DP method equipped with its own Dirichlet preconditioner is  $N_s^{\text{opt}} = 500$ . For a change, we adopt for this problem Definition D1 for identifying the corner nodes (see Section 2.1).

For this shell problem with almost 1 million d.o.f., the parallel PSLDLT sparse direct solver requires 2601 megabytes of memory, and consumes 1631 s CPU on a single Origin 2000 processor. The FETI-2 solver requires 2638 megabytes of memory and 3226 s CPU on a single Origin 2000 processor. On the other hand, the FETI-DP method requires 2386 megabytes of memory and 1671 s CPU on the same O2000 processor. Hence, all three solvers require comparable amounts of memory. However, the serial FETI-DP solver is twice as fast as the serial FETI-2 solver, and almost as fast as the serial PSLDLT sparse direct solver, which is an important result.

We report in Table X the performance results obtained for all three solvers using an increasing number of processors  $N_p$  on the Origin 2000 system. The reader can observe that at  $N_p = 20$ , the

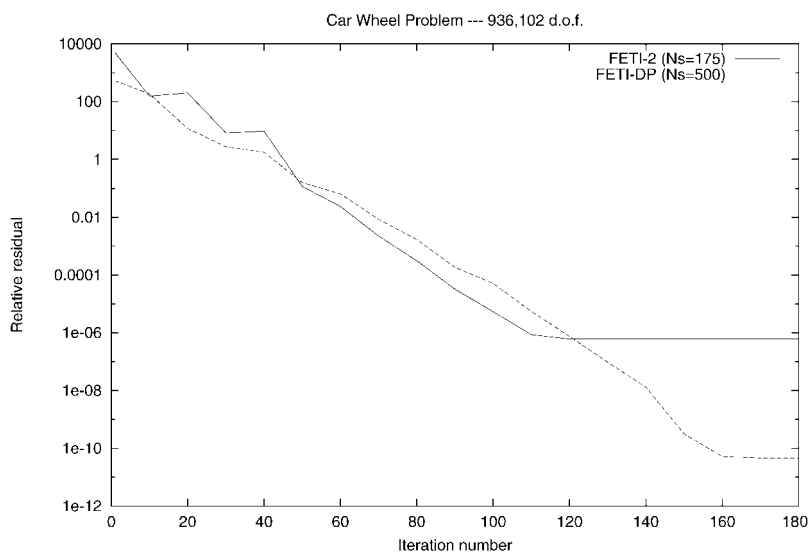


Figure 5. Convergence history.

PSLDT solver delivers a speed-up equal to 8.2, and that increasing the number of processors beyond  $N_p = 20$  does not reduce any further the solution time of this sparse solver. On the other hand, at  $N_p = 20$ , the FETI-2 and FETI-DP solvers deliver speed-ups equal to 16.7 and 16.1, respectively, and at  $N_p = 24$  these speed-ups are equal to 19 and 18.5, respectively. This shows that domain decomposition methods such as FETI, even when they require the solution of one or several coarse problems, are more amenable to parallel processing than sparse direct solvers. The FETI-2 solver delivers slightly better speed-ups than its FETI-DP counterpart because as stated earlier, the solution by forward and backward substitutions of the FETI-2 coarse problems involves dense block algebra and therefore parallelizes to some extent, whereas the solution of the FETI-DP coarse problem by forward and backward substitutions involves only sparse computations and therefore is currently serialized. Nevertheless, using 24 O2000 processors, the FETI-DP solver is still 1.8 times faster than the FETI-2 solver, and 2.3 times faster than the PSLDT sparse solver.

We also report in Figure 5 the convergence histories of both FETI-2 and FETI-DP methods for this car wheel problem. These histories show well that both FETI methods have similar convergence rates. Therefore the fact that the FETI-DP solver is faster than the FETI-2 solver is due to the differences in the number of coarse problems each method embeds, and the different sizes of these auxiliary problems. Most importantly, Figure 5 also shows that the FETI-DP method can reach significantly smaller residuals than the FETI-2 method.

## 5. OUTSTANDING ISSUE

For three-dimensional problems discretized by solid elements (brick, tetrahedra, ...), Definition D1 produces a large number of corner nodes. For computational efficiency purposes, only a subset of

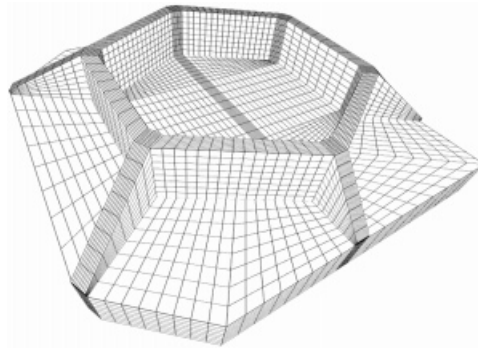


Figure 6. Finite element discretization of a diffraction grating system.

these nodes can be selected. Hence, we adopt for three-dimensional problems a modified version MD1 of Definition D1 where a node is identified as a corner if it belongs to at least four subdomains. This particular choice of four subdomains is due to the fact that for a three-dimensional uniform mesh partitioned into  $1/H \times 1/H \times 1/H$  subdomains, a ‘vertex’ of the mesh partition is connected to at least four subdomains. However, for three-dimensional irregular meshes partitioned into an arbitrary number of subdomains, Definition MD1 can cause some subdomain problem matrices to be singular. For this reason, we complement Definition MD1 by a fast postprocessing phase that guarantees that at least three non-colinear nodes are selected as corner nodes. This ensures that all subdomain problem matrices are then non-singular.

In order to illustrate the performance of the FETI-DP method for three-dimensional second-order problems, we consider the structural analysis of the diffraction grating system shown in Figure 6. This system is part of a satellite borne telescope spectrograph. Its finite element model contains 35 328 eight-noded brick elements and 120 987 d.o.f. The grating material of this structure is fused silica. When mounted, it must have face surface deflections below the micron level in 1G acceleration for accurate pre-launch alignment with the rest of the spectrograph. It must also be able to withstand accelerations up to 15G laterally and axially during launch. Several designs for this system have been conceived at the University of Colorado [38], and analysed by the FETI method. Here, we compare the performance results of the one-level FETI and FETI-DP methods when applied to the structural analysis of one design configuration. For this purpose, we equip both FETI methods with their respective lumped preconditioners because these are known to be more computationally efficient than the Dirichlet preconditioners for second-order problems. We report in Table XI the performance results obtained on a single processor Origin 2000 for two different numbers of subdomains. These results suggest that

- (1) Using Definition MD1, the size of the coarse problem of the FETI-DP method has a tendency to grow faster with the number of subdomains than that of the one-level FETI method.
- (2) The convergence rate of the FETI-DP method is slower than that of the one-level FETI method, and subsequently the FETI-DP solver is slower than the FETI solver.

The observations noted above are confirmed by several numerical tests that we have conducted for other three-dimensional second-order problems, using both the lumped and Dirichlet preconditioners. These observations suggest that for three-dimensional second-order problems, the FETI-DP

Table XI. Diffraction grating problem with 120 987 d.o.f. Performance results on a single processor O2000 of the one-level FETI and the FETI-DP solvers equipped with their lumped preconditioners.

$N_s$	FETI $N_{\text{coarse}}$	FETI-DP $N_{\text{coarse}}$	FETI # iterations	FETI-DP # iterations	FETI CPU (s)	FETI-DP CPU (s)
56	319	312	81	190	281	534
128	749	1116	51	125	115	273

method needs either a different preconditioner, or a different coarse problem. This outstanding issue is successfully dealt with in two sequel publications: in the companion paper [39], and Part II of this work. Indeed in Reference [39] and Part II, it is shown that the numerical scalability of the FETI-DP method and its computational efficiency can be restored for three-dimensional second-order problems by an appropriate enrichment of the coarse problem (15), and a careful selection of the corner nodes. This enrichment is based on the framework proposed in Reference [33] for accelerating the convergence of a DD method with Lagrange multipliers, and the methodology described in Reference [19] for enforcing a multipoint constraint in a FETI method.

## 6. CONCLUSIONS

Like the original FETI method, the FETI-DP method is a domain decomposition method with Lagrange multipliers. It is based on the same concept as the two-level FETI-2 method that was originally developed for the scalable iterative solution of fourth-order plate and shell problems. In both the FETI-2 and FETI-DP methods, the exact continuity of the displacement field at the subdomain corners is enforced at each preconditioned conjugate gradient (PCG) iteration, while the continuity of the displacement field at the other subdomain interface points is reached only at convergence. While the FETI-2 method employs corrective Lagrange multipliers to achieve this objective, the FETI-DP method relies for this purpose on a dual-primal formulation where the displacement and rotational degrees of freedom at the subdomain corners are considered as part of the fundamental unknowns. Unlike in the FETI and FETI-2 methods, the problem matrices associated with floating subdomains are never singular in the FETI-DP method. Consequently, the FETI-DP solver does not have to compute any subdomain rigid body mode, and is equally applicable to the solution of both static and dynamic problems. Therefore, the FETI-DP solver is more robust than the previously developed FETI solvers, and requires less code maintenance. Extensive numerical experiments suggest that the FETI-DP method is numerically scalable with respect to the size of the global problem, the size of the subdomain problem, and the number of subdomains. These numerical experiments also reveal that the FETI-DP solver can reach substantially lower residuals than the FETI and FETI-2 solvers. While most FETI methods require solving at least two coarse problems at each PCG iteration, the FETI-DP method requires solving only one coarse problem at each PCG iteration. Furthermore, for two-dimensional second-order elasticity problems and for fourth-order plate and shell problems, the coarse problem of the FETI-DP method is in general smaller than that of the FETI and FETI-2 methods. For these two reasons, even though the FETI-DP method exhibits the same convergence rate as the basic FETI method when applied to second-order problems, and the same convergence rate as the FETI-2 method when applied to

fourth-order problems, it is faster than both of these methods, particularly when the number of subdomains is very large. It is also faster than sparse direct solvers, even for some shell problems, particularly on parallel processors. For three-dimensional second-order problems, the FETI-DP method as described in this paper does not perform as well as the basic FETI method. However, in the companion paper [39] and Part II of this work, the FETI-DP method is further improved and shown to achieve numerical scalability and computational efficiency also for three-dimensional second-order problems.

#### ACKNOWLEDGEMENTS

Charbel Farhat acknowledges the support by the Sandia National Laboratories under Contract No. BD-2435. Michel Lesoinne and Kendall Pierson acknowledge the support by the Department of Energy under Award No. B347880/W-740-ENG-48. Patrick LeTallec acknowledges the support by INRIA under the NSF/INRIA collaboration program on Domain Decomposition and Parallelization Techniques in Scientific Computing.

#### REFERENCES

1. Farhat C. A Lagrange multiplier based divide and conquer finite element algorithm. *Journal of Computer and Systems Engineering* 1991; **2**:149–156.
2. Farhat C, Roux FX. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering* 1991; **32**:1205–1227.
3. Farhat C, Roux FX. An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**(1):379–396.
4. Farhat C. A saddle-point principle domain decomposition method for the solution of solid mechanics problems. In *Domain Decomposition Methods for Partial Differential Equations*, Keyes DE, Chan TF, Meurant GA, Scroggs JS, Voigt RG (eds). SIAM: Philadelphia, PA, 1992; 271–292.
5. Farhat C, Mandel J, Roux FX. Optimal convergence properties of the FETI domain decomposition method. *Computer Methods in Applied Mechanics and Engineering* 1994; **115**:367–388.
6. Mandel J, Tezaur R. Convergence of a substructuring method with Lagrange multipliers. *Numerische Mathematik* 1996; **73**:473–487.
7. Tezaur R. Analysis of Lagrange multiplier based domain decomposition. *Ph.D. Thesis*, University of Colorado, Denver, 1998.
8. Klawonn A, Widlund O. FETI and Neumann-Neumann iterative substructuring methods: connections and new results, private communication.
9. Farhat C, Roux FX. Implicit parallel processing in structural mechanics. *Computational Mechanics Advances* 1994; **2**:1–24.
10. Bhardwaj M, Day D, Farhat C, Lesoinne M, Pierson K, Rixen D. Application of the FETI method to ASCII problems: scalability results on one-thousand processors and discussion of highly heterogeneous problems. *International Journal for Numerical Methods in Engineering* 2000; **47**:513–536.
11. Farhat C, Chen PS. Tailoring domain decomposition methods for efficient parallel coarse grid solution and for systems with many right hand sides. *Contemporary Mathematics* 1994; **180**:401–406.
12. Farhat C, Crivelli L, Roux FX. Extending substructure based iterative solvers to multiple load and repeated analyses. *Computer Methods in Applied Mechanics and Engineering* 1994; **117**:195–200.
13. Farhat C, Gérardin M. On a component mode synthesis method and its application to incompatible substructures. *Computers and Structures* 1994; **51**:459–473.
14. Farhat C, Chen PS, Mandel J. A scalable Lagrange multiplier based domain decomposition method for implicit time-dependent problems. *International Journal for Numerical Methods in Engineering* 1995; **38**:3831–3858.
15. Rey C. Développement d'algorithmes parallèles de résolution en calcul non-linéaire de structures hétérogènes: application au cas d'une butée acier-élastomère. *Ph.D. Thesis*, Laboratoire de Modélisation et Mécanique des Structures, University of Paris VI, December 1994.
16. Roux FX. Parallel implementation of a domain decomposition method for nonlinear elasticity. In *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, Keyes D, Saad Y, Truhlar DG (eds), SIAM: Philadelphia, PA, 1995; 161–175.
17. Farhat C, Pierson K, Lesoinne M. The second generation of FETI methods and their application to the parallel solution of large-scale linear and geometrically nonlinear structural analysis problems. *Computer Methods in Applied Mechanics and Engineering* 2000; **184**:333–374.

18. Gérardin M, Coulon D, Delsemme J-P. Parallelization of the SAMCEF finite element software through domain decomposition and FETI algorithm. *International Journal of Supercomputer and Applications* 1997; **11**:286–298.
19. Farhat C, Lacour C, Rixen D. Incorporation of linear multipoint constraints in substructure based iterative solvers—Part I: a numerically scalable algorithm. *International Journal for Numerical Methods in Engineering* 1998; **43**:997–1016.
20. Rixen D, Farhat C. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *International Journal for Numerical Methods in Engineering* 1999; **44**:489–516.
21. de La Bourdonnaye A, Farhat C, Macedo A, Magoulès F, Roux FX. A non-overlapping domain decomposition method for the exterior Helmholtz problem. *Contemporary Mathematics* 1998; **218**:42–66.
22. Farhat C, Macedo A, Lesoinne M. A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. *Numerische Mathematik* 2000; **85**:283–308.
23. Papadrakakis M, Tsompanakis Y. Domain decomposition methods for parallel solution of sensitivity analysis problems. *Report 96-1*, Institute of Structural Analysis and Seismic Research, NTUA, Athens, Greece, 1996.
24. Rey C, Lene F. Generalized Krylov correction of the conjugate complement preconditioning to problems in computational fluid dynamics and computational electromagnetics. *Proceedings of the 9th International Conference on Domain Decomposition Methods*, Bergen, Norway, June 3–8, 1996.
25. Soulat D, Devries F. Mechanical criteria for the subdomains decomposition: applications to heterogeneous structures and composite materials. *Proceedings of the 9th International Conference on Domain Decomposition Methods*, Bergen, Norway, June 3–8, 1996.
26. Dureissex D, Ladevèze P. A two-level and mixed domain decomposition approach for structural analysis. *Contemporary Mathematics* 1998; **218**:238–245.
27. Dostal Z. Domain decomposition for semicoercive contact problems. *Contemporary Mathematics* 1998; **218**:82–93.
28. Lacour C. Non conforming domain decomposition method for plate problems. *Contemporary Mathematics* 1998; **218**:304–310.
29. Farhat C, Mandel J. Scalable substructuring by Lagrange multipliers in theory and in practice. *Proceedings of the 9th International Conference on Domain Decomposition Methods*, Bergen, Norway, June 3–8, 1996. Also published in *Domain Decomposition Methods for Partial Differential Equations*, Bjorstad P, Espedal M, Keyes D (eds), Domain Decomposition Press: Bergen, 1998; 20–30.
30. Farhat C, Mandel J. The two-level FETI method for static and dynamic plate problems—Part I: an optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering* 1998; **155**:129–152.
31. Farhat C, Chen PS, Mandel J, Roux FX. The two-level FETI method—Part II: extension to shell problems, parallel implementation and performance results. *Computer Methods in Applied Mechanics and Engineering* 1998; **155**: 153–180.
32. Mandel J, Tezaur R, Farhat C. A scalable substructuring method by Lagrange multipliers for plate bending problems. *SIAM Journal of Numerical Analysis* 1999; **36**:1370–1391.
33. Farhat C, Chen PS, Risler F, Roux FX. A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers. *International Journal for Numerical Methods in Engineering* 1998; **42**:257–288.
34. Rixen D, Farhat C, Tezaur R, Mandel J. Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver. *International Journal for Numerical Methods in Engineering* 1999; **46**:501–534.
35. Farhat C, Gérardin M. On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures. *International Journal for Numerical Methods in Engineering* 1998; **41**:675–696.
36. Davis DD, Krishnamurthy T, Stroud WJ, McCleary SL. An accurate nonlinear finite element analysis and test correlation of a stiffened composite wing panel. American Helicopter Society National Technical Specialists' Meeting on Rotorcraft Structures, Williamsburg, Virginia, October 29–31, 1991.
37. Farhat C, Maman N, Brown G. Mesh partitioning for implicit computations via iterative domain decomposition: impact and optimization of the subdomain aspect ratio. *International Journal for Numerical Methods in Engineering* 1995; **38**:989–1000.
38. Shipley A, Green JC, Andrews JP. The design and mounting of the gratings for the Far Ultraviolet Spectroscopic Explorer (FUSE). *SPIE* 1995; **2452**:185–196.
39. Farhat C, Lesoinne M, Pierson K. A scalable dual-primal domain decomposition method. *Numerical Linear Algebra and Applications* 2000; **7**(7–8):687–714.