INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng. **46**, 501–533 (1999)

THEORETICAL COMPARISON OF THE FETI AND ALGEBRAICALLY PARTITIONED FETI METHODS, AND PERFORMANCE COMPARISONS WITH A DIRECT SPARSE SOLVER

DANIEL J. RIXEN[1], CHARBEL FARHAT[1,*], RADEK TEZAUR[2] AND JAN MANDEL[2]

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

[2] *Center for Computational Mathematics, University of Colorado at Denver, Denver, CO 80217-3364, U.S.A.*

SUMMARY

In this paper, we prove that the *Algebraic A*-FETI method corresponds to one particular instance of the original one-level FETI method. We also report on performance comparisons on an Origin 2000 between the one- and two-level FETI methods and an optimized sparse solver, for two industrial applications: the stress analysis of a thin shell structure, and that of a three-dimensional structure modelled by solid elements. These comparisons suggest that for topologically two-dimensional problems, sparse solvers are effective when the number of processors is relatively small. They also suggest that for three-dimensional applications, scalable domain decomposition methods such as FETI deliver a superior performance on both sequential and parallel hardware configurations. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: domain decomposition; FETI method; iterative solver; sparse solver; scalability

1. INTRODUCTION

In the last decade, the Finite Element Tearing and Interconnecting (FETI) method [1–4] has emerged as one of the most powerful iterative solvers for elliptic problems, and most popular domain decomposition (or substructuring) method in the computational mechanics community. This is reflected in the important number of papers from various scientific communities that continue to focus on the mathematical analysis of this method and its extension to increasingly challenging applications, and the large number of algorithms and variant methods that it has inspired during the last five years [5–14].

The FETI method is a Lagrange-multiplier-based iterative substructuring algorithm. It can be summarized as a two-step Preconditioned Conjugate Gradient (PCG) procedure where substructure problems with Dirichlet boundary conditions are solved in the preconditioning step, and related substructure problems with Neumann boundary conditions are solved in a second step. When equipped with the so-called Dirichlet preconditioner [3] and with the optional corner Lagrange

[*] Correspondence to: Charbel Farhat, Department of Aerospace Engineering Sciences, University of Colorado at Boulder, Boulder, CO 80309-0429, U.S.A. E-mail: charbel@boulder.colorado.edu

CCC 0029-5981/99/280501–33$17.50

Received 10 September 1998
Revised 3 February 1999

multipliers for plate and shell problems [2, 15–17], the condition number of its interface problem grows asymptotically only as

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

where h denotes the mesh size (and therefore is an indirect measure of the problem size), and H denotes the substructure size (and therefore is an indirect measure of the number of substructures). This condition number estimate establishes the numerical scalability of the FETI method with respect to both the problem size and the number of substructures—that is, its ability to solve larger problems using a larger number of substructures in almost a constant number of iterations. The parallel scalability of the FETI method—that is, its suitability for massively parallel computing —has also been demonstrated on a large number of massively parallel processors for several realistic structural and structural dynamics problems [15, 18]. Because of these numerical and parallel scalability properties, the FETI method has attracted the attention of both the academic community and finite element practitioners. In particular, it has been recognized by many finite element production code developers as a robust and powerful iterative solver on sequential as well as massively parallel computers.

The complexity of the FETI method depends on the order of the partial differential equation underlying the discrete problem to be solved, and can be characterized by the complexity of the associated 'coarse problem'. The latter problem is a relatively small size auxiliary problem that is generated by the FETI method in order to propagate the error globally during the PCG iterations, which accelerates convergence. For second-order elastostatic and elastodynamic problems, the coarse problem is constructed using the substructure rigid-body modes only, and the corresponding computational overhead is relatively insignificant. For fourth-order plate and shell problems, this coarse problem must be enriched by the so-called substructure corner modes [15, 16] in order to maintain the numerical scalability of the FETI method. This enriched coarse problem increases the complexity of the FETI method and transforms it into a genuine two-level algorithm known as the two-level FETI method. Hence, the two-level FETI algorithm is more computationally expensive per iteration than the original one-level FETI method. However, for plate and shell problems, the two-level FETI solver has a significantly better convergence rate than its one-level counterpart, and therefore can be significantly faster.

Recently, a series of papers [12, 19, 20] have appeared in the literature to propose an 'algebraic' variant of the FETI method denoted by A-FETI. In [12], the authors have labelled the original FETI method as a *differential* partitioned procedure, and have motivated their effort to develop an algebraic version of FETI by stating that 'differential and algebraic partitioning possess their distinct algorithmic advantages as well as disadvantages in the context of sequential staggered solution procedures'. To the best of our knowledge, domain decomposition methods go back to the century old Schwarz alternating procedure [21], and are usually classified into overlapping/non-overlapping and/or additive/multiplicative Schwarz methods. Being a variant, the A-FETI algorithm introduced in [12] strongly resembles the original FETI method. However, some details in the derivation of A-FETI and a number of conjectures made in [12, 19, 20] could suggest that A-FETI differs significantly from FETI. The most important of these conjectures, which was based on a performance comparison that turned out to be erroneous and therefore invalid, was that for plate problems, A-FETI should be capable of delivering a numerical performance that is close to that of the two-level FETI method without the need for enriching the coarse problem of the basic FETI method with the additional substructure corner modes [20]—in other words, at the same computational price as the basic FETI method. This intriguing conclusion and the numerous invitations made in [12, 20] for comparing the FETI and A-FETI methods have motivated us to write the present paper whose objectives are

1. To prove mathematically that the A-FETI method not only relies on the same concepts as the FETI method, but is identical to one particular instance of the FETI method.
2. To correct the invalid performance comparison offered in [20], and demonstrate that the A-FETI algorithm delivers exactly the same performance as the one-level FETI method, and is *not* numerically scalable for plate problems.
3. To demonstrate that the two-level FETI method is not only numerically scalable for plate and shell problems, but is also computationally (CPU-wise) more efficient than the A-FETI (or FETI) method at solving these fourth-order problems.
4. To compare the performance of the FETI methodology with that of the most advanced sparse direct solvers on both sequential and parallel computers for several realistic structural problems.

Objective 1 has been previously addressed albeit in a different form in [17]. Objectives 1–3 are discussed in Section 2 which also overviews the FETI and A-FETI methods in order to keep this paper as self-contained as possible. Objective 4 is addressed in Section 3, and Section 4 concludes this paper.

2. DOMAIN DECOMPOSITION WITH LAGRANGE MULTIPLIERS

2.1. The one-level FETI methodology

2.1.1. From a global primal problem to a dual interface problem. The general form of a linear or linearized algebraic problem arising from the discretization of a structural mechanics problem defined on a domain Ω can be written as

$$\mathbf{K}_{\mathrm{g}}\mathbf{u}_{\mathrm{g}} = \mathbf{f}_{\mathrm{g}} \tag{2}$$

where for a large class of applications, $\mathbf{K}_{\mathrm{g}}$ is a symmetric positive-definite or semi-definite matrix constructed from the assembly of element level stiffness matrices by introducing a global numbering of the degrees of freedom (d.o.f.) indicated here by the subscript g, $\mathbf{u}_{\mathrm{g}}$ is the vector of generalized displacements, and $\mathbf{f}_{\mathrm{g}}$ the vector of prescribed forces. A *non-overlapping* domain decomposition or substructuring method with matching interfaces for solving efficiently the above problem on a sequential or parallel processor can be devised by partitioning Ω into N_s non-overlapping substructures $\Omega^{(s)}$, expressing the local equilibrium of each substructure, and introducing displacement compatibility constraints at the substructure interfaces. If Lagrange multipliers are employed to enforce these constraints, the original mechanical problem can be reformulated as a saddle point problem whose Euler equations are given by [1]

$$\delta_{\mathbf{v}^{(s)},\boldsymbol{\mu}}\mathscr{L}(\mathbf{v}^{(s)},\boldsymbol{\mu}) = 0 \tag{3}$$

where $\delta_{\mathbf{v}^{(s)},\boldsymbol{\mu}}$ designates the variation with respect to the arguments $\mathbf{v}^{(s)}$ and $\boldsymbol{\mu}$, and $\mathscr{L}$ is the Lagrangian of the problem and can be written as

$$\mathscr{L}(\mathbf{v}^{(s)},\boldsymbol{\mu}) = \sum_{s=1}^{N_s}\left(\frac{1}{2}\mathbf{v}^{(s)^{\mathrm{T}}}\mathbf{K}^{(s)}\mathbf{v}^{(s)} - \mathbf{v}^{(s)^{\mathrm{T}}}\mathbf{f}^{(s)}\right) + \boldsymbol{\mu}^{\mathrm{T}}\sum_{s=1}^{N_s}\mathbf{B}^{(s)}\mathbf{v}^{(s)} \tag{4}$$

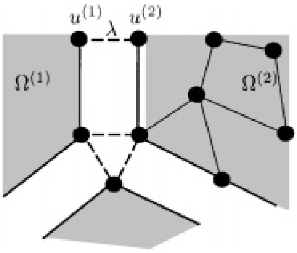


Figure 1. Interface assembly in the FETI formulation

Here, the superscript s designates a substructure quantity, the superscript T denotes the transpose of a quantity, $\mathbf{v}^{(s)}$ and $\boldsymbol{\mu}$ are some admissible substructure displacement and Lagrange multiplier fields, respectively, $\mathbf{K}^{(s)}$ is the substructure stiffness matrix, $\mathbf{f}^{(s)}$ the substructure vector of prescribed forces, and $\mathbf{B}^{(s)}$ the signed Boolean matrices that extracts from a substructure vector $\mathbf{v}^{(s)}$ its signed $(\pm)$ restriction to the substructure interface boundary. The Euler equations (3) lead to the following constraint problem which is equivalent to problem (2):

$$\mathbf{K}^{(s)}\mathbf{u}^{(s)} = \mathbf{f}^{(s)} - \mathbf{B}^{(s)^{\mathrm{T}}}\boldsymbol{\lambda}, \quad s = 1, \dots, N_s \tag{5}$$

$$\sum_{s=1}^{N_s} \mathbf{B}^{(s)}\mathbf{u}^{(s)} = \mathbf{0} \tag{6}$$

where $\mathbf{u}^{(s)}$ and $\boldsymbol{\lambda}$ are respectively, the substructure displacements and Lagrange multipliers that are the stationary points of $\mathscr{L}$. The mechanical interpretation of equations (5) and (6) is graphically depicted in Figure 1: each of equations (5) expresses the substructure equilibrium under the action of the prescribed forces $\mathbf{f}^{(s)}$ and the substructure interface forces $\mathbf{B}^{(s)^{\mathrm{T}}}\boldsymbol{\lambda}$, and equation (6) expresses the compatibility of the substructure displacement fields at the substructure interfaces. There is one compatibility equation for each pair of d.o.f. that connect at a substructure interface.

Let $\mathbf{K}$, $\mathbf{B}$, $\mathbf{u}$ and $\mathbf{f}$ be defined as follows:

$$\begin{gathered}\mathbf{K} = \begin{bmatrix} \mathbf{K}^{(1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}^{(N_s)} \end{bmatrix}, \quad \mathbf{B} = [\,\mathbf{B}^{(1)} \ \cdots \ \mathbf{B}^{(N_s)}\,] \\ \mathbf{u} = \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(N_s)} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(N_s)} \end{bmatrix}\end{gathered} \tag{7}$$

Using the above notation, the substructure equations of equilibrium (5) and compatibility (6) can be re-written in compact block diagonal form as follows

$$\begin{bmatrix} \mathbf{K} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \tag{8}$$

Once $\boldsymbol{\lambda}$ is determined, the substructure displacement fields can be recovered by solving concurrently the equilibrium equations to obtain

$$\mathbf{u}^{(s)} = \mathbf{K}^{(s)^+} \left(\mathbf{f}^{(s)} - \mathbf{B}^{(s)^{\mathrm{T}}} \boldsymbol{\lambda}\right) + \mathbf{R}^{(s)} \boldsymbol{\alpha}^{(s)} \tag{9}$$

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

$$\mathbf{R}^{(s)^{\mathrm{T}}} \left(\mathbf{f}^{(s)} - \mathbf{B}^{(s)^{\mathrm{T}}} \boldsymbol{\lambda}\right) = \mathbf{0} \tag{10}$$

Substituting equation (9) into the compatibility equation (6) and exploiting the solvability condition (10) transforms problem (8) into the interface problem

$$\begin{bmatrix} \mathbf{F}_{\mathrm{I}} & -\mathbf{G}_{\mathrm{I}} \\ -\mathbf{G}_{\mathrm{I}}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ -\mathbf{e} \end{bmatrix} \tag{11}$$

where

$$\begin{aligned}
\mathbf{F}_{\mathrm{I}} &= \sum_{s=1}^{N_s} \mathbf{B}^{(s)} \mathbf{K}^{(s)^+} \mathbf{B}^{(s)^{\mathrm{T}}} = \mathbf{B}\mathbf{K}^+\mathbf{B}^{\mathrm{T}} \\
\mathbf{d} &= \sum_{s=1}^{N_s} \mathbf{B}^{(s)} \mathbf{K}^{(s)^+} \mathbf{f}^{(s)} = \mathbf{B}\mathbf{K}^+\mathbf{f} \\
\mathbf{G}_{\mathrm{I}} &= [\, \mathbf{B}^{(1)}\mathbf{R}^{(1)} \ \cdots \ \mathbf{B}^{(N_s)}\mathbf{R}^{(N_s)} \,] = \mathbf{B}\mathbf{R} \\
\boldsymbol{\alpha} &= [\, \boldsymbol{\alpha}^{(1)^{\mathrm{T}}} \ \cdots \ \boldsymbol{\alpha}^{(N_s)^{\mathrm{T}}} \,]^{\mathrm{T}} \\
\mathbf{e} &= [\, \mathbf{f}^{(1)^{\mathrm{T}}}\mathbf{R}^{(1)} \ \cdots \ \mathbf{f}^{(N_s)^{\mathrm{T}}}\mathbf{R}^{(N_s)} \,]^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}}\mathbf{f}
\end{aligned} \tag{12}$$

Here $\mathbf{K}^+$ is the block diagonal matrix storing the $\mathbf{K}^{(s)^+}$ substructure matrices, and the columns of $\mathbf{R}$ store the rigid-body modes of the floating substructures

$$\mathbf{K}^+ = \begin{bmatrix} \mathbf{K}^{(1)^+} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{K}^{(N_s)^+} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}^{(1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}^{(N_s)} \end{bmatrix} \tag{13}$$

Because $\boldsymbol{\lambda}$ is a dual variable to the primal variables $\mathbf{u}^{(s)}$, the interface problem (11) is called a dual interface problem. It partially defines the one-level FETI that was originally introduced in [1, 3, 4, 18]. This interface problem is best solved by an iterative algorithm, the choice of which completes the description of the one-level FETI method.

2.1.2. Iterative solution of the dual interface problem. In the FETI method, the interface problem (11) is solved by a preconditioned conjugate projected gradient (PCPG) algorithm. More specifically, the indefinite interface problem (11) is transformed into a semi-definite system of equations by eliminating the self-equilibrium condition $\mathbf{G}_\mathrm{I}^\mathrm{T}\boldsymbol{\lambda}=\mathbf{e}$ using the splitting

$$\boldsymbol{\lambda}=\boldsymbol{\lambda}^0+\mathbf{P}(\mathbf{Q})\bar{\boldsymbol{\lambda}} \tag{14}$$

where $\boldsymbol{\lambda}^0$ is a particular solution of $\mathbf{G}_\mathrm{I}^\mathrm{T}\boldsymbol{\lambda}=\mathbf{e}$ of the form

$$\boldsymbol{\lambda}^0=\mathbf{Q}\mathbf{G}_\mathrm{I}(\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{Q}\mathbf{G}_\mathrm{I})^{-1}\mathbf{e} \tag{15}$$

and $\mathbf{P}(\mathbf{Q})$ is a projection matrix defined for a given matrix $\mathbf{Q}$ by

$$\mathbf{P}(\mathbf{Q})=\mathbf{I}-\mathbf{Q}\mathbf{G}_\mathrm{I}(\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{Q}\mathbf{G}_\mathrm{I})^{-1}\mathbf{G}_\mathrm{I}^\mathrm{T} \tag{16}$$

Note that for any matrix $\mathbf{Q}$, $\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{P}(\mathbf{Q})=\mathbf{0}$.

Applying the splitting (14) to the interface problem (11) leads to the alternative symmetric positive-semi-definite interface problem

$$\mathbf{P}(\mathbf{Q})^\mathrm{T}\mathbf{F}_\mathrm{I}\mathbf{P}(\mathbf{Q})\bar{\boldsymbol{\lambda}}=\mathbf{P}(\mathbf{Q})^\mathrm{T}(\mathbf{d}-\mathbf{F}_\mathrm{I}\boldsymbol{\lambda}^0) \tag{17}$$

This generalized formulation of the FETI interface problem in which $\mathbf{Q}$ is not necessarily the identity matrix was first proposed in [18]. For any given matrix $\mathbf{Q}$, the alternative interface problem (17) can be solved by a standard PCG algorithm.

So far, two different preconditioners have been proposed: the mathematically optimal Dirichlet preconditioner $\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{D}-1}$ introduced in [3], and the computationally economical lumped preconditioner $\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{L}-1}$ proposed in earlier works [1, 4]. If each substructure stiffness matrix is partitioned as

$$\mathbf{K}^{(s)}=\begin{bmatrix}\mathbf{K}_\mathrm{ii}^{(s)} & \mathbf{K}_\mathrm{ib}^{(s)}\\ \mathbf{K}_\mathrm{ib}^{(s)^\mathrm{T}} & \mathbf{K}_\mathrm{bb}^{(s)}\end{bmatrix} \tag{18}$$

where the subscripts i and b designate the substructure interior and interface boundary d.o.f., respectively, then the Dirichlet preconditioner can be written as

$$\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{D}-1}=\mathbf{W}\left(\sum_{s=1}^{N_s}\mathbf{B}^{(s)}\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{S}_\mathrm{bb}^{(s)}\end{bmatrix}\mathbf{B}^{(s)^\mathrm{T}}\right)\mathbf{W} \tag{19}$$

and the lumped preconditioner is given by

$$\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{L}-1}=\mathbf{W}\left(\sum_{s=1}^{N_s}\mathbf{B}^{(s)}\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{K}_\mathrm{bb}^{(s)}\end{bmatrix}\mathbf{B}^{(s)^\mathrm{T}}\right)\mathbf{W} \tag{20}$$

In the above expressions of $\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{D}-1}$ and $\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{L}-1}$, $\mathbf{W}$ is a diagonal matrix storing in each of its entries the inverse of the multiplicity of an interface d.o.f. [2, 18, 22]—that is, the inverse of the number of substructures to which an interface d.o.f. belongs—and $\mathbf{S}_\mathrm{bb}^{(s)}$ is the following substructure primal Schur complement

$$\mathbf{S}_\mathrm{bb}^{(s)}=\mathbf{K}_\mathrm{bb}^{(s)}-\mathbf{K}_\mathrm{ib}^{(s)^\mathrm{T}}\mathbf{K}_\mathrm{ii}^{(s)^{-1}}\mathbf{K}_\mathrm{ib}^{(s)} \tag{21}$$

Note that $\mathbf{K}_\mathrm{ii}^{(s)}$ is non-singular since it corresponds to the system matrix with all interface boundaries fixed and thus $\mathbf{K}_\mathrm{ii}^{(s)^{-1}}$ exists. Both Dirichlet and lumped preconditioners have been recently extended

```
1. Initialize
     $\boldsymbol{\lambda}^0 = \mathbf{Q}\mathbf{G}_\mathrm{I}(\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{Q}\mathbf{G}_\mathrm{I})^{-1}\mathbf{e}$
     $\mathbf{w}^0 = \mathbf{P}(\mathbf{Q})^\mathrm{T}(\mathbf{d} - \mathbf{F}_\mathrm{I}\boldsymbol{\lambda}^0)$
2. Iterate $k = 0, 1, \ldots$ until convergence
     $\mathbf{y}^k = \mathbf{P}(\mathbf{Q})\bar{\mathbf{F}}_\mathrm{I}^{-1}\mathbf{w}^k$
     $\mathbf{p}^k = \mathbf{y}^k - \sum_{i=0}^{k-1} \frac{\mathbf{y}^{k^\mathrm{T}}\mathbf{F}_\mathrm{I}\mathbf{p}^i}{\mathbf{p}^{i^\mathrm{T}}\mathbf{F}_\mathrm{I}\mathbf{p}^i}\mathbf{p}^i$
     $\eta^k = \frac{\mathbf{p}^{k^\mathrm{T}}\mathbf{w}^k}{\mathbf{p}^{k^\mathrm{T}}\mathbf{F}_\mathrm{I}\mathbf{p}^k}$
     $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \eta^k\mathbf{p}^k$
     $\mathbf{w}^{k+1} = \mathbf{w}^k - \eta^k\mathbf{P}(\mathbf{Q})^\mathrm{T}\mathbf{F}_\mathrm{I}\mathbf{p}^k$
```

Box 1. The FETI PCPG method

in [23] for addressing more efficiently heterogeneous problems. In particular, it was shown in [23] that such extensions can be simply designed by redefining appropriately the scaling matrix $\mathbf{W}$.

The iterative one-level FETI solver is summarized in Box 1. A more detailed description of this algorithm can be found in [18].

Remark. (1) The straightforward formulation of the substructure displacement compatibility equations generates redundant constraints at the crosspoints of a mesh partition, and therefore $\mathbf{B}^\mathrm{T}$ does not have in general a full column rank [24, 25]. It follows that $\mathbf{F}_\mathrm{I}$ and $\bar{\mathbf{F}}_\mathrm{I}^{-1}$ are in general semi-definite, and that the solution $\boldsymbol{\lambda}$ of the interface problem (17) is not unique. However, the corresponding substructure displacement solutions $\mathbf{u}^{(s)}$ are unique. While programming tricks can be invoked to avoid redundant compatibility constraints at the crosspoints, it is preferable to keep the full redundancy in the system because it accelerates the convergence of the FETI method. A rational explanation of this issue as well as a mechanical interpretation of this result can be found in [23].

(2) The matrix $\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{Q}\mathbf{G}_\mathrm{I}$ defines an auxiliary coarse problem that couples all the substructure computations, propagates the error globally, and accelerates convergence. It is because of this coarse problem that when the Dirichlet preconditioner is used, the condition number of the FETI interface problem (17) can be bounded by a polylogarithmic function of H/h (see 1), where H and h denote the substructure and mesh sizes, respectively. We remind the reader that such a bound proves the numerical scalability of the FETI method with respect to both the problem size and number of substructures [10, 16].

(3) If the given global structure is fully restrained—that is, if $\mathbf{K}_\mathrm{g}$ is not singular—$\mathbf{G}_\mathrm{I}$ has full column rank and $\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{G}_\mathrm{I}$ is non-singular [18]. In that case, it makes sense to restrict the choice of the matrix $\mathbf{Q}$ by the condition that $\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{Q}\mathbf{G}_\mathrm{I}$ be also non-singular. Furthermore, if $\mathbf{Q}$ is chosen among symmetric matrices, $\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{Q}\mathbf{G}_\mathrm{I}$ becomes symmetric and thus more economical to compute, store, and factor. In [18] it was suggested that a good choice for $\mathbf{Q}$ is $\mathbf{Q} = \bar{\mathbf{F}}_\mathrm{I}^{-1}$, where $\bar{\mathbf{F}}_\mathrm{I}^{-1}$ is either the Dirichlet or the lumped preconditioner used in the FETI iterations.

(4) If the given global structure is not fully restrained—that is, if $\mathbf{K}_\mathrm{g}$ is singular—the matrix $\mathbf{G}_\mathrm{I}^\mathrm{T}\mathbf{Q}\mathbf{G}_\mathrm{I}$ becomes also singular [18]. In that case, the FETI method is still applicable and the null space of $\mathbf{K}_\mathrm{g}$ can be retrieved using the strategy described in [26].

(5) For most problems, the simplest choice $\mathbf{Q}=\mathbf{I}$ is computationally the most effective one. Choosing $\mathbf{Q}=\mathbf{Q}^\mathrm{D}=\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{D}-1}$ or $\mathbf{Q}=\mathbf{Q}^\mathrm{L}=\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{L}-1}$ was originally recommended in [18] for heterogeneous problems, and was further investigated in [27].

(6) In practice, the FETI algorithm is always used with a reorthogonalization procedure to accelerate convergence, as indicated in Box 1. In [18], it was shown that such a strategy is cost-effective for substructure problems because reorthogonalization is applied only to the interface Lagrange multiplier unknowns.

The FETI method has also been extended to problems with multiple right-hand sides [28, 29], transient dynamic analysis [30–32], fourth-order plate and shell problems [2, 15, 16], non-linear structural analysis [33–35], and problems with multipoint constraints [36]. The specific extension of FETI to plate and shell problems has initiated the development of what is known today as the two-level FETI method (see [2, 15, 16] for fourth-order structural problems, and [37] for a general presentation of the two-level FETI method).

2.1.3. FETI operators and local primal Schur complements. An alternative expression of the FETI interface problem (11) can be obtained by eliminating the substructure internal d.o.f. from the decomposed problem (8). While such a transformation does not offer any particular computational advantage, it can simplify the mathematical analysis of the FETI method in some circumstances. For example, in [3, 18], such a transformation was performed to justify the Dirichlet preconditioner $\bar{\mathbf{F}}_\mathrm{I}^{\mathrm{D}-1}$. Here, the substructure internal d.o.f. are eliminated from the formulation of the FETI interface problem in order to derive a few expressions that are needed for establishing the connection between the FETI and *A*-FETI methods.

Using the same partitioning as in (18), $\mathbf{u}^{(s)}$ can be written as

$$\mathbf{u}^{(s)}=\begin{bmatrix}\mathbf{u}_\mathrm{i}^{(s)}\\ \mathbf{u}_\mathrm{b}^{(s)}\end{bmatrix} \tag{22}$$

and $\mathbf{u}_\mathrm{i}^{(s)}$ can be expressed in terms of $\mathbf{u}_\mathrm{b}^{(s)}$ by solving equation (5) for the internal d.o.f.:

$$\mathbf{u}_\mathrm{i}^{(s)}=\mathbf{K}_\mathrm{ii}^{(s)^{-1}}(\mathbf{f}_\mathrm{i}^{(s)}-\mathbf{K}_\mathrm{ib}^{(s)}\mathbf{u}_\mathrm{b}^{(s)}) \tag{23}$$

Let $\mathbf{B}_\mathrm{b}^{(s)}$ be the submatrix of $\mathbf{B}^{(s)}$ that operates on $\mathbf{u}_\mathrm{b}^{(s)}$, so that

$$\mathbf{B}^{(s)}=[\mathbf{0}\quad \mathbf{B}_\mathrm{b}^{(s)}] \tag{24}$$

and define

$$\mathbf{B}_\mathrm{b}=[\mathbf{B}_\mathrm{b}^{(1)}\quad \dots\quad \mathbf{B}_\mathrm{b}^{(N_s)}] \tag{25}$$

Substituting equations (23) and (24) in the expression of the Lagrangian $\mathscr{L}$ defined in (4) gives

$$\mathscr{L}(\mathbf{v}_\mathrm{b}^{(s)},\boldsymbol{\mu})=\sum_{s=1}^{N_s}\left(\frac{1}{2}\mathbf{v}_\mathrm{b}^{(s)^\mathrm{T}}\mathbf{S}_\mathrm{bb}^{(s)}\mathbf{v}_\mathrm{b}^{(s)}-\mathbf{v}_\mathrm{b}^{(s)^\mathrm{T}}\mathbf{f}_\mathrm{b}^{*(s)}\right)+\boldsymbol{\mu}^\mathrm{T}\sum_{s=1}^{N_s}\mathbf{B}_\mathrm{b}^{(s)}\mathbf{v}_\mathrm{b}^{(s)} \tag{26}$$

where $\mathbf{S}_{\mathrm{bb}}^{(s)}$ are the substructure primal Schur complements defined in (21), and $\mathbf{f}_{\mathrm{b}}^{*(s)}$ are given by

$$\mathbf{f}_{\mathrm{b}}^{*(s)} = \mathbf{f}_{\mathrm{b}}^{(s)} - \mathbf{K}_{\mathrm{ib}}^{(s)^{\mathrm{T}}} \mathbf{K}_{\mathrm{ii}}^{(s)^{-1}} \mathbf{f}_{\mathrm{i}}^{(s)} \tag{27}$$

Seeking the saddle point of $\mathscr{L}(\mathbf{v}_{\mathrm{b}}^{(s)}, \boldsymbol{\mu})$ leads to the following algebraic system of equations:

$$\begin{bmatrix} \mathbf{S}_{\mathrm{bb}} & \mathbf{B}_{\mathrm{b}}^{\mathrm{T}} \\ \mathbf{B}_{\mathrm{b}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathrm{b}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathrm{b}}^{*} \\ \mathbf{0} \end{bmatrix} \tag{28}$$

where

$$\mathbf{S}_{\mathrm{bb}} = \begin{bmatrix} \mathbf{S}_{\mathrm{bb}}^{(1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{S}_{\mathrm{bb}}^{(N_s)} \end{bmatrix}, \quad \mathbf{f}_{\mathrm{b}}^{*} = \begin{bmatrix} \mathbf{f}_{\mathrm{b}}^{*(1)} \\ \vdots \\ \mathbf{f}_{\mathrm{b}}^{*(N_s)} \end{bmatrix} \quad \text{and} \quad \mathbf{u}_{\mathrm{b}} = \begin{bmatrix} \mathbf{u}_{\mathrm{b}}^{(1)} \\ \vdots \\ \mathbf{u}_{\mathrm{b}}^{(N_s)} \end{bmatrix} \tag{29}$$

The reader can observe that problem (28) is identical to problem (8) with the internal d.o.f. condensed out.

The interface problem associated with equations (28) is obtained by the same procedure described in Section 2.1.1. First, $\mathbf{u}_{\mathrm{b}}$ is extracted from the first of equations (28) as follows:

$$\mathbf{u}_{\mathrm{b}} = \mathbf{S}_{\mathrm{bb}}^{+}(\mathbf{f}_{\mathrm{b}}^{*} - \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}) + \mathbf{R}_{\mathrm{b}}\boldsymbol{\alpha} \tag{30}$$

where $\mathbf{R}_{\mathrm{b}}$ stores the null spaces $\mathbf{R}_{\mathrm{b}}^{(s)}$ of $\mathbf{S}_{\mathrm{bb}}^{(s)}$. Note that

1. $$\mathbf{K}^{(s)}\mathbf{R}^{(s)} = \mathbf{0} \Leftrightarrow \begin{cases} \mathbf{R}_{\mathrm{i}}^{(s)} = -\mathbf{K}_{\mathrm{ii}}^{(s)^{-1}} \mathbf{K}_{\mathrm{ib}}^{(s)} \mathbf{R}_{\mathrm{b}}^{(s)} \\ \mathbf{S}_{\mathrm{bb}}^{(s)} \mathbf{R}_{\mathrm{b}}^{(s)} = \mathbf{0} \end{cases} \tag{31}$$

   which shows that $\mathbf{R}_{\mathrm{b}}^{(s)} = \operatorname{Ker} \mathbf{S}_{\mathrm{bb}}^{(s)}$ is the restriction of $\mathbf{R}^{(s)} = \operatorname{Ker} \mathbf{K}^{(s)}$ to the substructure interface boundary d.o.f.
2. The solvability (substructure self-equilibrium) condition associated with equation (30) is

   $$\mathbf{R}_{\mathrm{b}}^{\mathrm{T}}(\mathbf{f}_{\mathrm{b}}^{*} - \mathbf{B}_{\mathrm{b}}^{T}\boldsymbol{\lambda}) = \mathbf{0} \tag{32}$$

   which in view of equations (25), (27) and (31) can also be written as

   $$\mathbf{G}_{\mathrm{I}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{e} \tag{33}$$

   where

   $$\mathbf{G}_{\mathrm{I}} = \mathbf{B}\mathbf{R} = \mathbf{B}_{\mathrm{b}}\mathbf{R}_{\mathrm{b}} \tag{34}$$

   $$\mathbf{e} = \mathbf{R}^{\mathrm{T}}\mathbf{f} = \mathbf{R}_{\mathrm{b}}^{\mathrm{T}}\mathbf{f}_{\mathrm{b}}^{*} \tag{35}$$

Then, substituting equation (30) in the second of equations (28) (the compatibility equation) and using equation (33) leads to the following expression of the FETI interface problem:

$$\begin{bmatrix} \mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}} & -\mathbf{G}_{\mathrm{I}} \\ -\mathbf{G}_{\mathrm{I}}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{f}_{\mathrm{b}}^{*} \\ -\mathbf{e} \end{bmatrix} \tag{36}$$

In [18, Section 5.1.5], it was shown that

$$\mathbf{F}_{\mathrm{I}} = \mathbf{B}\,\mathbf{K}^{+}\mathbf{B}^{\mathrm{T}} = \mathbf{B}_{\mathrm{b}}\,\mathbf{S}_{\mathrm{bb}}^{+}\,\mathbf{B}_{\mathrm{b}}^{\mathrm{T}} \tag{37}$$

Hence, from the comparison of both expressions (11) and (36) of the same dual interface problem, it follows that

$$\mathbf{d} = \mathbf{B}\mathbf{K}^{+}\mathbf{f} = \mathbf{B}_{\mathrm{b}}\,\mathbf{S}_{\mathrm{bb}}^{+}\,\mathbf{f}_{\mathrm{b}}^{*} \tag{38}$$

which shows that the entire FETI method can be formulated in terms of the substructure primal Schur complements.

Finally, we point out that the Dirichlet (19) and lumped (20) preconditioners can be written in block form as follows:

$$\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}} = \mathbf{W}\mathbf{B}_{\mathrm{b}}\,\mathbf{S}_{\mathrm{bb}}\,\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\,\mathbf{W}, \quad \bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{L}^{-1}} = \mathbf{W}\mathbf{B}_{\mathrm{b}}\,\mathbf{K}_{\mathrm{bb}}\,\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\,\mathbf{W} \tag{39}$$

2.2. The algebraic FETI method

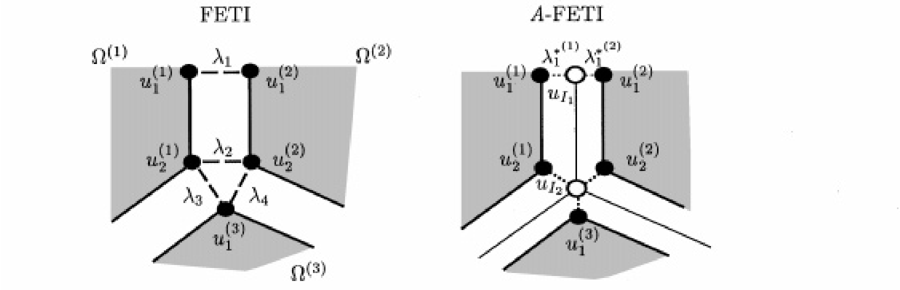
Recently, a version of the FETI method called the algebraic FETI method has been introduced in [12, 19, 20], and named by its developers as the *A*-FETI method. Here, we summarize this method using the same notation and concepts as those previously used for describing the one-level FETI method.

2.2.1. A three-field formalism with discrete Lagrange multipliers. Consider again the solution by a substructuring method of a structural problem represented by equation (2). In the FETI methodology, at each interface $\Gamma_{\mathrm{I}}^{(s,q)}$ between two substructures $\Omega^{(s)}$ and $\Omega^{(q)}$, a single field of Lagrange multipliers is introduced to enforce the compatibility of the substructure displacement fields $\mathbf{u}^{(s)}$ and $\mathbf{u}^{(q)}$ on $\Gamma_{\mathrm{I}}^{(s,q)}$. Using the partitioning introduced in (22) and (24), this compatibility, which has already been expressed in equation (6), can be re-written as

$$\sum_{s=1}^{N_s} \mathbf{B}_{\mathrm{b}}^{(s)}\mathbf{u}_{\mathrm{b}}^{(s)} = \mathbf{B}_{\mathrm{b}}\mathbf{u}_{\mathrm{b}} = \mathbf{0} \tag{40}$$

where as before, the subscript b designates the interface d.o.f.

Alternatively, one can define a unique displacement field at the substructure interfaces, $\mathbf{u}_{\mathrm{I}}$, and introduce at each local interface $\Gamma_{\mathrm{I}}^{(s,q)}$ a first Lagrange multiplier field to enforce the compatibility between $\mathbf{u}_{\mathrm{b}}^{(s)}$ and the corresponding component of $\mathbf{u}_{\mathrm{I}}$, and a second Lagrange multiplier field to enforce the compatibility between $\mathbf{u}_{\mathrm{b}}^{(q)}$ and the corresponding component of $\mathbf{u}_{\mathrm{I}}$. Such a strategy is often referred to in the domain decomposition community as a three-field method (two Lagrange multiplier fields and $\mathbf{u}_{\mathrm{I}}$). It is a popular discretization method when the given substructures have *non-matching* discrete interfaces (for example, see [38–40]). In that case, *non-discrete* Lagrange multipliers are employed for enforcing a weak form of the compatibility of the displacement fields at the substructure interfaces. However, when the substructures have matching interfaces and discrete Lagrange multipliers are employed for enforcing interface compatibility—in other words, for algebraic applications—it is not clear that a three-field approach can offer any advantage over a conventional hybrid method, because in that case both approaches are identical from a mathematical viewpoint.



$$\underbrace{\left[\begin{array}{ccccc} \multicolumn{2}{c}{\overbrace{\qquad\quad}^{B_b^{(1)}}} & \multicolumn{2}{c}{\overbrace{\qquad\quad}^{B_b^{(2)}}} & \overbrace{\quad}^{B_b^{(3)}} \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{array}\right]}_{B_b} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ u_1^{(2)} \\ u_2^{(2)} \\ u_1^{(3)} \end{bmatrix} = \mathbf{0} \qquad \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ u_1^{(2)} \\ u_2^{(2)} \\ u_1^{(3)} \end{bmatrix} = \begin{array}{c} L_b^{(1)} \left\{ \vphantom{\begin{array}{c} 1 \\ 0 \end{array}} \right. \\ L_b^{(2)} \left\{ \vphantom{\begin{array}{c} 1 \\ 0 \end{array}} \right. \\ L_b^{(3)} \left\{ \vphantom{1} \right. \end{array} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}}_{L_b} \begin{bmatrix} u_{I_1} \\ u_{I_2} \end{bmatrix}$$

Figure 2. Interface assembly for the FETI and A-FETI methods

The A-FETI method borrows the formalism of a three-field substructuring method, and yet targets algebraic applications arising from substructures with matching interfaces. Consequently, it replaces the interface compatibility equation (6) by an equivalent form as described below:

$$\sum_{s=1}^{N_s} \mathbf{B}_{\mathrm{b}}^{(s)} \mathbf{u}_{\mathrm{b}}^{(s)} = \mathbf{B}_{\mathrm{b}} \begin{bmatrix} \mathbf{u}_{\mathrm{b}}^{(1)} \\ \mathbf{u}_{\mathrm{b}}^{(2)} \\ \vdots \\ \mathbf{u}_{\mathrm{b}}^{(N_s)} \end{bmatrix} = \mathbf{0}$$

$$\Leftrightarrow$$

$$\begin{bmatrix} \mathbf{u}_{\mathrm{b}}^{(1)} \\ \mathbf{u}_{\mathrm{b}}^{(2)} \\ \vdots \\ \mathbf{u}_{\mathrm{b}}^{(N_s)} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{\mathrm{b}}^{(1)} \\ \mathbf{L}_{\mathrm{b}}^{(2)} \\ \vdots \\ \mathbf{L}_{\mathrm{b}}^{(N_s)} \end{bmatrix} \mathbf{u}_{\mathrm{I}} = \mathbf{L}_{\mathrm{b}} \mathbf{u}_{\mathrm{I}} \tag{41}$$

where $\mathbf{L}_{\mathrm{b}}^{(s)}$ is the assembly matrix that identifies each local interface vector $\mathbf{u}_{\mathrm{b}}^{(s)}$ with the appropriate component of the global interface vector $\mathbf{u}_{\mathrm{I}}$ (see Figure 2). Like any three-field method, A-FETI employs two separate Lagrange multiplier (discrete) fields $\boldsymbol{\lambda}^{*^{(s)}}$ and $\boldsymbol{\lambda}^{*^{(q)}}$ to enforce on each local interface $\Gamma_{\mathrm{I}}^{(s,q)}$ the compatibility between the substructure displacement fields $\mathbf{u}^{(s)}$ and $\mathbf{u}^{(q)}$ (a star superscript is used here to distinguish the Lagrange multipliers employed in the A-FETI method from those employed in the original FETI method). However, because the A-FETI solver assumes that the substructures have matching interfaces, the Lagrange multipliers it generates verify $\boldsymbol{\lambda}^{*^{(s)}} = \boldsymbol{\lambda}^{*^{(q)}}$ as will be explained later. Note also that the equivalence expressed in (41) implies that

$$\operatorname{Ker}\mathbf{B}_{\mathrm{b}} = \operatorname{Im}\mathbf{L}_{\mathrm{b}} \tag{42}$$

which can be verified in the simple example of Figure 2.

In the A-FETI formulation, the restriction of each substructure displacement field to the substructure interface boundary is decomposed into one component $\tilde{\mathbf{u}}_{\mathrm{b}}^{(s)}$ that induces deformations, and another component that represents the potential rigid-body motion of $\Omega^{(s)}$:

$$\mathbf{u}_{\mathrm{b}}^{(s)} = \tilde{\mathbf{u}}_{\mathrm{b}}^{(s)} + \mathbf{R}_{\mathrm{b}}^{(s)}\boldsymbol{\alpha}^{(s)} \tag{43}$$

where $\mathbf{R}_{\mathrm{b}}^{(s)}$ is the restriction of the substructure rigid-body modes $\mathbf{R}^{(s)}$ to the interface boundary d.o.f. and, as shown in (31), it is also the nullspace of the primal Schur complement $\mathbf{S}_{\mathrm{bb}}^{(s)}$ introduced in (21). Note that equation (43) is identical to the FETI equation (30).

From equations (41) and (43), it follows that A-FETI's counterpart of FETI's Lagrangian $\mathscr{L}$ (26) is given by [41]

$$\begin{aligned}\mathscr{F}(\tilde{\mathbf{v}}_{\mathrm{b}}^{(s)}, \boldsymbol{\mu}^{*(s)}, \boldsymbol{\beta}^{(s)}, \mathbf{v}_{\mathrm{I}}) = \sum_{s=1}^{N_{\mathrm{s}}} \Bigg\{ & \frac{1}{2}\tilde{\mathbf{v}}_{\mathrm{b}}^{(s)^{\mathrm{T}}}\mathbf{S}_{\mathrm{bb}}^{(s)}\tilde{\mathbf{v}}_{\mathrm{b}}^{(s)} - (\tilde{\mathbf{v}}_{\mathrm{b}}^{(s)} + \mathbf{R}_{\mathrm{b}}^{(s)}\boldsymbol{\beta}^{(s)})^{\mathrm{T}}\mathbf{f}_{\mathrm{b}}^{*(s)} \\ & + \boldsymbol{\mu}^{*(s)^{\mathrm{T}}}(\tilde{\mathbf{v}}_{\mathrm{b}}^{(s)} + \mathbf{R}_{\mathrm{b}}^{(s)}\boldsymbol{\beta}^{(s)} - \mathbf{L}_{\mathrm{b}}^{(s)}\mathbf{v}_{\mathrm{I}})\Bigg\}\end{aligned} \tag{44}$$

where $\mathbf{f}^{*(s)}$ has been defined in (27). The stationary point of the above functional $\mathscr{F}$ is obtained by setting to zero each of the variations of $\mathscr{F}$ with respect to $\tilde{\mathbf{v}}_{\mathrm{b}}^{(s)}$, $\boldsymbol{\mu}^{*(s)}$, $\boldsymbol{\beta}^{(s)}$, and $\mathbf{u}_{\mathrm{I}}$, which leads to the following set of equations:

$$\begin{bmatrix} \mathbf{S}_{\mathrm{bb}} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{R}_{\mathrm{b}} & -\mathbf{L}_{\mathrm{b}} \\ \mathbf{0} & \mathbf{R}_{\mathrm{b}}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{L}_{\mathrm{b}}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_{\mathrm{b}} \\ \boldsymbol{\lambda}^{*} \\ \boldsymbol{\alpha} \\ \mathbf{u}_{\mathrm{I}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathrm{b}}^{*} \\ \mathbf{0} \\ \mathbf{R}_{\mathrm{b}}^{\mathrm{T}}\mathbf{f}_{\mathrm{b}}^{*} \\ \mathbf{0} \end{bmatrix} \tag{45}$$

where the block diagonal forms $\mathbf{S}_{\mathrm{bb}}$ and $\mathbf{f}_{\mathrm{b}}^{*}$ have been defined in (29), and

$$\begin{gathered}\mathbf{R}_{\mathrm{b}} = \begin{bmatrix} \mathbf{R}_{\mathrm{b}}^{(1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}_{\mathrm{b}}^{(N_{\mathrm{s}})} \end{bmatrix} \\ \tilde{\mathbf{u}}_{\mathrm{b}} = \begin{bmatrix} \tilde{\mathbf{u}}_{\mathrm{b}}^{(1)} \\ \vdots \\ \tilde{\mathbf{u}}_{\mathrm{b}}^{(N_{\mathrm{s}})} \end{bmatrix}, \quad \boldsymbol{\lambda}^{*} = \begin{bmatrix} \boldsymbol{\lambda}^{*(1)} \\ \vdots \\ \boldsymbol{\lambda}^{*(N_{\mathrm{s}})} \end{bmatrix}, \quad \boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}^{(1)} \\ \vdots \\ \boldsymbol{\alpha}^{(N_{\mathrm{s}})} \end{bmatrix}\end{gathered} \tag{46}$$

As in the FETI method, the first of equations (45) expresses the equilibrium of the substructures $\Omega^{(s)}$, and the second of these equations expresses the compatibility of their displacement fields at the substructure interfaces. The third of equations (45) is identical to the self-equilibrium condition (32) characterizing the FETI method. The fourth of equations (45) results from the three-field formalism of the A-FETI method: it states that the Lagrange multipliers acting on both sides of an interface $\Gamma_{\mathrm{I}}^{(s,q)}$ must be in equilibrium. This "action–reaction" principle is naturally satisfied in the FETI method because for FETI, a single Lagrange multiplier is defined for each pair of connecting d.o.f.

Equations (45) summarize the formulation of the *A*-FETI method. Before discussing the associated iterative solution algorithm, we make two important observations

(a) The fourth of equations (45), $\mathbf{L}_b^T\boldsymbol{\lambda}^* = 0$, is equivalent to stating

$$\boldsymbol{\lambda}^* \in \operatorname{Ker} \mathbf{L}_b^T$$

which in view of (42) is also equivalent to writing

$$\boldsymbol{\lambda}^* \in \operatorname{Im} \mathbf{B}_b^T$$

It follows that $\mathbf{L}_b^T\boldsymbol{\lambda}^* = 0$ can be eliminated from the system of equations (45) by writing

$$\boldsymbol{\lambda}^* = \mathbf{B}_b^T\boldsymbol{\gamma} \tag{47}$$

From (42) it also follows that

$$\mathbf{B}_b\mathbf{L}_b = 0$$

and therefore the global interface displacement field $\mathbf{u}_I$ can be eliminated from equations (45) by premultiplying the second of these equations by $\mathbf{B}_b$. Hence, taking account of relations (34) and (35), the governing equations of the *A*-FETI method can be re-written as

$$\begin{bmatrix} \mathbf{S}_{bb} & \mathbf{B}_b^T & \mathbf{0} \\ \mathbf{B}_b & \mathbf{0} & \mathbf{G}_I \\ \mathbf{0} & \mathbf{G}_I^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_b \\ \boldsymbol{\gamma} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b^* \\ \mathbf{0} \\ \mathbf{e} \end{bmatrix} \tag{48}$$

Furthermore, $\tilde{\mathbf{u}}_b$ can be eliminated from equations (48) by substituting in that system

$$\tilde{\mathbf{u}}_b = \mathbf{S}_{bb}^+(\mathbf{f}_b^* - \mathbf{B}_b^T\boldsymbol{\gamma}) \tag{49}$$

in which case one recovers exactly the dual FETI in terms of the local Schur complements as in (36), with $\boldsymbol{\gamma}$ playing the role of the FETI lagrange multipliers $\boldsymbol{\lambda}$. In summary, from this first observation, and more specifically from equation (47), we conclude that *A*-FETI's Lagrange multipliers $\boldsymbol{\lambda}^*$ are related to FETI's Lagrange multipliers $\boldsymbol{\lambda}$ by

$$\boldsymbol{\lambda}^* = \mathbf{B}_b^T\boldsymbol{\lambda} \tag{50}$$

(b) From the previous observation, it follows that the substructuring formulation (45) with four set of variables can be transformed into a classical hybrid substructuring decomposition formulation by eliminating the global interface variable $\mathbf{u}_I$, and enforcing naturally the interface equilibrium of the Lagrange multipliers $\boldsymbol{\lambda}^*$. Indeed, the *A*-FETI iterative solver performs exactly such a transformation. However, instead of following the procedure outlined above, it utilizes for this purpose a suitable projector, as explained next.

2.2.2. Yet another expression of the dual interface problem. In [20], the authors propose to eliminate the unknown $\mathbf{u}_I$ from equations (45) by introducing the following symmetric projector

$$\mathbf{P}_\ell = \mathbf{I} - \mathbf{L}_b(\mathbf{L}_b^T\mathbf{L}_b)^{-1}\mathbf{L}_b^T \tag{51}$$

which verifies $\mathbf{L}_b^T\mathbf{P}_\ell=\mathbf{0}$ and $\mathbf{P}_\ell=\mathbf{P}_\ell\mathbf{P}_\ell^T=\mathbf{P}_\ell^T$. Given that $\boldsymbol{\lambda}^*$ satisfies $\mathbf{L}_b^T\boldsymbol{\lambda}^*=0$, $\boldsymbol{\lambda}^*$ can be written as

$$\boldsymbol{\lambda}^*=\mathbf{P}_\ell\boldsymbol{\lambda}_\ell \tag{52}$$

Since $(\mathbf{L}_b^T\mathbf{L}_b)$ is the diagonal matrix of the interface multiplicities, it follows that the projector $\mathbf{P}_\ell$ enforces the usage of Lagrange multipliers $\boldsymbol{\lambda}^*$ that not only satisfy interface equilibrium, but are also constructed by averaging some given Lagrange multipliers $\boldsymbol{\lambda}_\ell$. Such a strategy is similar to the multiplicity scaling employed by the preconditioners of FETI (see equations (19) and (20)).

Substituting equation (52) into equations (45) and premultiplying the second set of equations in (45) by $\mathbf{P}_\ell^T$ leads to

$$\begin{bmatrix}\mathbf{S}_{bb} & \mathbf{P}_\ell & \mathbf{0}\\ \mathbf{P}_\ell^T & \mathbf{0} & \mathbf{P}_\ell^T\mathbf{R}_b\\ \mathbf{0} & \mathbf{R}_b^T\mathbf{P}_\ell & \mathbf{0}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{u}}_b\\ \boldsymbol{\lambda}_\ell\\ \boldsymbol{\alpha}\end{bmatrix}=\begin{bmatrix}\mathbf{f}_b^*\\ \mathbf{0}\\ \mathbf{e}\end{bmatrix} \tag{53}$$

The above system of equations is similar to system (48), except that the multipliers $\boldsymbol{\lambda}_\ell$ are defined on each side of an interface $\Gamma_I^{(s,q)}$, whereas in equations (48), $\boldsymbol{\lambda}$ is uniquely defined between any pair of connecting d.o.f. It follows that in general, the *A*-FETI method requires about twice as many Lagrange multipliers as the one-level FETI method.

The last of equations (53) is enforced by introducing a projector $\mathbf{P}_R$ that is similar to FETI's projector $\mathbf{P}$ and given by

$$\mathbf{P}_R=\mathbf{I}-\mathbf{P}_\ell^T\mathbf{R}_b(\mathbf{R}_b^T\mathbf{P}_\ell\mathbf{R}_b)^{-1}\mathbf{R}_b^T\mathbf{P}_\ell \tag{54}$$

such that $\mathbf{R}_b^T\mathbf{P}_\ell\mathbf{P}_R=\mathbf{0}$. Performing the following splitting

$$\boldsymbol{\lambda}_\ell=\boldsymbol{\lambda}_\ell^0+\mathbf{P}_R\boldsymbol{\lambda}_R$$

where

$$\boldsymbol{\lambda}_\ell^0=\mathbf{P}_\ell^T\mathbf{R}_b(\mathbf{R}_b^T\mathbf{P}_\ell\mathbf{R}_b)^{-1}\mathbf{e} \tag{55}$$

and premultiplying the second set of equations in (53) by $\mathbf{P}_R^T$ transforms (53) into

$$\begin{bmatrix}\mathbf{S}_{bb} & \mathbf{P}_\ell\mathbf{P}_R\\ \mathbf{P}_R^T\mathbf{P}_\ell^T & \mathbf{0}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{u}}_b\\ \boldsymbol{\lambda}_R\end{bmatrix}=\begin{bmatrix}\mathbf{f}_b^*-\mathbf{P}_\ell\boldsymbol{\lambda}_\ell^0\\ \mathbf{0}\end{bmatrix} \tag{56}$$

Comparing the above system of equations to the FETI equations (28), one can observe that the *A*-FETI method can already be viewed as a FETI method where each Lagrange multiplier is duplicated on the other side of each $\Gamma_I^{(s,q)}$ with the appropriate sign, and where the constraint matrix $\mathbf{B}_b$ is a non-Boolean operator constructed as $\mathbf{B}_b=\mathbf{P}_R^T\mathbf{P}_\ell^T=\mathbf{P}_R\mathbf{P}_\ell=\mathbf{P}_\ell\mathbf{P}_R=\mathbf{B}_b^T$ [17].

Finally, eliminating $\tilde{\mathbf{u}}_b$ from equations (56) via

$$\tilde{\mathbf{u}}_b=\mathbf{S}_{bb}^+(\mathbf{f}_b^*-\mathbf{P}_\ell\boldsymbol{\lambda}_\ell^0-\mathbf{P}_\ell\mathbf{P}_R\boldsymbol{\lambda}_R) \tag{57}$$

leads to yet another form of the dual interface problem that is adopted by the *A*-FETI method and is given by

$$(\mathbf{P}_R^T\mathbf{P}_\ell^T\mathbf{S}_{bb}^+\mathbf{P}_\ell\mathbf{P}_R)\boldsymbol{\lambda}_R=\mathbf{P}_R^T\mathbf{P}_\ell^T\mathbf{S}_{bb}^+(\mathbf{f}_b^*-\mathbf{P}_\ell\boldsymbol{\lambda}_\ell^0) \tag{58}$$

```
1. Initialize
    $\boldsymbol{\lambda}^{*^0} = \mathbf{P}_\ell^{\mathrm{T}}\mathbf{R}_{\mathrm{b}}(\mathbf{R}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}_\ell\mathbf{R}_{\mathrm{b}})^{-1}\mathbf{e}$
    $\mathbf{w}^{*^0} = \mathbf{P}_R^{\mathrm{T}}\mathbf{P}_\ell^{\mathrm{T}}\mathbf{S}_{\mathrm{bb}}^{+}(\mathbf{f}_{\mathrm{b}}^{*} - \boldsymbol{\lambda}_\ell^{*^0})$
2. Iterate $k = 0, 1, \ldots$ until convergence
    $\mathbf{y}^{*^k} = \mathbf{P}_\ell\mathbf{P}_R\mathbf{S}_{\mathrm{bb}}\mathbf{w}^{*^k}$
    $\mathbf{p}^{*^k} = \mathbf{y}^{*^k} - \sum_{i=0}^{k-1} \frac{\mathbf{y}^{*^{k^{\mathrm{T}}}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{p}^{*^i}}{\mathbf{p}^{*^{i^{\mathrm{T}}}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{p}^{*^i}}\mathbf{p}^{*^i}$
    $\eta^{*^k} = \frac{\mathbf{p}^{*^{k^{\mathrm{T}}}}\mathbf{w}^{*^k}}{\mathbf{p}^{*^{k^{\mathrm{T}}}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{p}^{*^k}}$
    $\boldsymbol{\lambda}^{*^{k+1}} = \boldsymbol{\lambda}^{*^k} + \eta^{*^k}\mathbf{p}^{*^k}$
    $\mathbf{w}^{*^{k+1}} = \mathbf{w}^{*^k} - \eta^{*^k}\mathbf{P}_R^{\mathrm{T}}\mathbf{P}_\ell^{\mathrm{T}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{p}^{*^k}$
```

Box 2. The *A*-FETI PCPG method

2.2.3. The A-FETI iterative solution algorithm. The *A*-FETI iterative algorithm can then be summarized as the PCG method applied to the solution of the interface problem (58). As stated in [20], this scheme borrows from the original FETI method the Dirichlet preconditioner that is in this case given by the block diagonal matrix of the substructure Schur complements $\mathbf{S}_{\mathrm{bb}}$.

The *A*-FETI algorithm is outlined in Box 2. For further details, we refer the readers to [12, 19, 20].

2.3. A-FETI is one instance of the FETI method

So far, we have shown that the *A*-FETI method employs the same mechanical concepts, a similar splitting of the Lagrange multipliers, a similar projector based on the rigid body modes, a similar scaling procedure, and the same preconditioner as the FETI method. However, because it employs 'right' and 'left' Lagrange multipliers at each substructure interface $\Gamma_{\mathrm{I}}^{(s,q)}$, whereas the FETI method employs a single Lagrange multiplier to connect two substructure interface d.o.f., the *A*-FETI method has often been described as an alternative to the FETI method with a non-singular dual interface problem. In this section, we prove mathematically that the *A*-FETI iterative solver is not an alternative but one particular instance of the FETI method that is obtained for a specific choice of the matrix $\mathbf{Q}$ (see Section 2.1.2). Furthermore, we show that this specific choice is a degenerate form of the two choices $\mathbf{Q}^{\mathrm{D}} = \bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}}$ and $\mathbf{Q}^{\mathrm{L}} = \bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{L}^{-1}}$ that were first proposed in [18] and recalled in Section 2.1.2 of this paper. Using the block matrix notation introduced in equations (46), $\mathbf{Q}^{\mathrm{D}}$ and $\mathbf{Q}^{\mathrm{L}}$ can be re-written as

$$\mathbf{Q}^{D} = \mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W} \quad \text{and} \quad \mathbf{Q}^{\mathrm{L}} = \mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{K}_{\mathrm{bb}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W} \tag{59}$$

2.3.1. Mathematical preliminaries. The matrix $\mathbf{B}_{\mathrm{b}}$ of the FETI method has entries equal to either -1, or 0, or $+1$. Redundant constraints are generated at the crosspoints. Without loss of generality, it can be assumed that $\mathbf{u}_{\mathrm{b}}$ is reordered so that the local degrees of freedom corresponding to one global degree of freedom are numbered consecutively. In that case, $\mathbf{B}_{\mathrm{b}}$ can be written in block form as

$$\mathbf{B}_{\mathrm{b}} = [\mathbf{B}_{\mathrm{b},j}] = [\mathbf{B}_{\mathrm{b},1} \ \mathbf{B}_{\mathrm{b},2} \ \dots]$$

where each block $\mathbf{B}_{\mathrm{b},j}$ consists of the columns of $\mathbf{B}_{\mathrm{b}}$ that are indexed by the local degrees of freedom corresponding to the global degree of freedom j. For example, for the problem graphically depicted in Figure 2, the compatibility equations can be written as

$$\left[\begin{matrix} \overbrace{\begin{matrix}1 & -1\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{matrix}}^{\mathbf{B}_{\mathrm{b},1}} & \overbrace{\begin{matrix}0 & 0 & 0\\ -1 & 1 & 0\\ -1 & 0 & 1\\ 0 & -1 & 1\end{matrix}}^{\mathbf{B}_{\mathrm{b},2}}\end{matrix}\right] \begin{bmatrix} u_1^{(1)} \\ u_1^{(2)} \\ u_2^{(1)} \\ u_2^{(2)} \\ u_1^{(3)} \end{bmatrix} = \mathbf{0} \tag{60}$$

Each block $\mathbf{B}_{\mathrm{b},j}$ thus has a number of columns equal to the multiplicity m_j of the corresponding interface.

Lemma 1. The matrix $\mathbf{B}_{\mathrm{b}}$ *satisfies*

$$\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{B}_{\mathrm{b}} = \mathrm{diag}\,(\mathbf{B}_{\mathrm{b},j}^{\mathrm{T}}\mathbf{B}_{\mathrm{b},j}) = \mathrm{diag}\ (m_j\mathbf{I}_{m_j} - \mathbf{E}_{m_j}) \tag{61}$$

where $\mathbf{I}_{m_j}$ *is the identity matrix of dimension* m_j, *and* $\mathbf{E}_{m_j}$ *is a square matrix of dimension* m_j *with all entries equal to* 1.

Proof. Since a substructure degree of freedom corresponds to only one global degree of freedom, the sets of indices of non-zero rows of $\mathbf{B}_{\mathrm{b},j_1}$ and $\mathbf{B}_{\mathrm{b},j_2}$ for two global degrees of freedom $j_1 \neq j_2$ do not overlap. Hence, $\mathbf{B}_{\mathrm{b},j_1}^{\mathrm{T}}\mathbf{B}_{\mathrm{b},j_2} = \mathbf{0}$ for $j_1 \neq j_2$, which proves the first equality in (61).

Let j be a fixed global degree of freedom. We now show that $\mathbf{B}_{\mathrm{b},j}^{\mathrm{T}}\mathbf{B}_{\mathrm{b},j} = m_j\mathbf{I}_{m_j} - \mathbf{E}_{m_j}$. Define an oriented graph $G = (N, E)$. The set N of nodes of G is the set of the indices of the columns of $\mathbf{B}_{\mathrm{b},j}$. The set E of the edges of the graph G consists of all pairs (n_1, n_2) from N such that $(\mathbf{B}_{\mathrm{b}}\mathbf{u}_{\mathrm{b}})_{(n_1,n_2)} = \mathbf{u}_{n_1} - \mathbf{u}_{n_2}$. (This is a complete graph where each edge has an orientation; such a graph is known as a *tournament*, and $\mathbf{B}_{\mathrm{b},j}$ is a *tournament matrix*.) Let $\boldsymbol{\lambda} = \mathbf{B}_{\mathrm{b}}\mathbf{u}_{\mathrm{b}}$ and let n_1 be a fixed node of the graph G. Then,

$$\begin{aligned}(\mathbf{B}_{\mathrm{b},j}^{\mathrm{T}}\boldsymbol{\lambda})_{n_1} &= \sum_{n_2:e=(n_1,n_2)\in E} (\boldsymbol{\lambda})_{\mathrm{e}} - \sum_{n_2:e=(n_2,n_1)\in E} (\boldsymbol{\lambda})_{\mathrm{e}} \\ &= \sum_{n_2:e=(n_1,n_2)\in E} (\mathbf{u}_{n_1} - \mathbf{u}_{n_2}) - \sum_{n_2:e=(n_2,n_1)\in E} (\mathbf{u}_{n_2} - \mathbf{u}_{n_1}) \\ &= (m_j - 1)\mathbf{u}_{n_1} - \sum_{n_2 \in N:n_2\neq n_1} \mathbf{u}_{n_2} \\ &= m_j\mathbf{u}_{n_1} - \sum_{n_2\in N} \mathbf{u}_{n_2}. \qquad \square\end{aligned}$$

Lemma 2. The following relations hold:

$$(\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}})^2 = \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}} \tag{62}$$

$$\mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}} = \mathbf{W}\mathbf{B}_{\mathrm{b}} \tag{63}$$

Proof. From Lemma 1 and the definition of $\mathbf{W}$, it follows that $(\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}})=\mathrm{diag}\,(\mathbf{B}_{\mathrm{b},j}^{\mathrm{T}}(1/m_j)\mathbf{B}_{\mathrm{b},j})$. Thus (62) can be verified for each diagonal block separately

$$\begin{aligned}\left(\mathbf{B}_{\mathrm{b},j}^{\mathrm{T}}\frac{1}{m_j}\mathbf{B}_{\mathrm{b},j}\right)\left(\mathbf{B}_{\mathrm{b},j}^{\mathrm{T}}\frac{1}{m_j}\mathbf{B}_{\mathrm{b},j}\right) &= \frac{1}{m_j^2}(m_j\mathbf{I}-\mathbf{E}_{m_j})^2\\ &= \frac{1}{m_j^2}(m_j^2\mathbf{I}-2m_j\mathbf{E}_{m_j}+\mathbf{E}_{m_j}^2)\\ &= \frac{1}{m_j^2}m_j(m_j\mathbf{I}-\mathbf{E}_{m_j}) = \mathbf{B}_{\mathrm{b},j}^{\mathrm{T}}\frac{1}{m_j}\mathbf{B}_{\mathrm{b},j}\end{aligned}$$

Next, we prove (63).

$$\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}} = \mathbf{B}_{\mathrm{b}}\mathrm{diag}\left(\mathbf{I}_{m_j}-\frac{1}{m_j}\mathbf{E}_{m_j}\right) = \mathbf{B}_{\mathrm{b}} - \mathbf{B}_{\mathrm{b}}\mathrm{diag}\left(\frac{1}{m_j}\mathbf{E}_{m_j}\right)$$

The last term on the right-hand side is equal to the zero matrix since the only possibly non-zero contributions are scalar multiples of $\mathbf{B}_{\mathrm{b},j}\mathbf{E}_{m_j}$. But since each non-zero row of $\mathbf{B}_{\mathrm{b},j}$ contains exactly one -1 and one 1, $\mathbf{B}_{\mathrm{b},j}\mathbf{E}_{m_j}=0$. □

Theorem 1. $\mathbf{L}_{\mathrm{b}}$ *has full column rank and* $\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}}$ *verifies*

$$\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}} = \mathbf{I}-\mathbf{L}_{\mathrm{b}}(\mathbf{L}_{\mathrm{b}}^{\mathrm{T}}\mathbf{L}_{\mathrm{b}})^{-1}\mathbf{L}_{\mathrm{b}}^{\mathrm{T}} = \mathbf{P}_{\ell} \tag{64}$$

Proof. Matrix $\mathbf{L}_{\mathrm{b}}$ defines the global-to-local mapping. Its columns correspond to global degrees of freedom and its rows to the union of all the substructure degrees of freedom. $\mathbf{L}_{\mathrm{b},ij}=1$ whenever the global degree of freedom j coincides with the substructure degree of freedom i, and $\mathbf{L}_{\mathrm{b},ij}=0$ otherwise. Hence, $\mathbf{L}_{\mathrm{b}}$ has full column rank.

Define $\mathbf{P}_W$ as

$$\mathbf{P}_W = \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}}$$

From Lemma 2, it follows that $\mathbf{P}_W$ is a projector. Since $\mathbf{W}$ is a diagonal matrix, $\mathbf{P}_W$ is symmetric and therefore is an orthogonal projector. Note that $(\mathbf{L}_{\mathrm{b}}^{\mathrm{T}}\mathbf{L}_{\mathrm{b}})^{-1}$ exists because $\mathbf{L}_{\mathrm{b}}$ has a full column rank, and that $\mathbf{P}_{\ell}=\mathbf{I}-\mathbf{L}_{\mathrm{b}}(\mathbf{L}_{\mathrm{b}}^{\mathrm{T}}\mathbf{L}_{\mathrm{b}})^{-1}\mathbf{L}_{\mathrm{b}}^{\mathrm{T}}$ is an orthogonal projector. Given that an orthogonal projector is completely determined by its nullspace, it suffices to show that $\mathrm{Ker}\,\mathbf{P}_W=\mathrm{Ker}\,\mathbf{P}_{\ell}$ in order to prove that $\mathbf{P}_W=\mathbf{P}_{\ell}$.

Let $\mathbf{u}_{\mathrm{b}}\in\mathrm{Ker}\,\mathbf{P}_{\ell}$. Then $\mathbf{u}_{\mathrm{b}}=\mathbf{L}_{\mathrm{b}}(\mathbf{L}_{\mathrm{b}}^{\mathrm{T}}\mathbf{L}_{\mathrm{b}})^{-1}\mathbf{L}_{\mathrm{b}}^{\mathrm{T}}\mathbf{u}_{\mathrm{b}}$, and therefore $\mathbf{u}_{\mathrm{b}}\in\mathrm{Im}\,\mathbf{L}_{\mathrm{b}}$. From (42), it follows that $\mathbf{B}_{\mathrm{b}}\mathbf{u}_{\mathrm{b}}=\mathbf{0}$, and therefore $\mathbf{u}_{\mathrm{b}}\in\mathrm{Ker}\,\mathbf{P}_W$. Hence, $\mathrm{Ker}\,\mathbf{P}_{\ell}\subset\mathrm{Ker}\,\mathbf{P}_W$. On the other hand, let $\mathbf{u}_{\mathrm{b}}\in\mathrm{Ker}\,\mathbf{P}_W$. Then, $(\mathbf{B}_{\mathrm{b}}\mathbf{u})^{\mathrm{T}}\mathbf{W}(\mathbf{B}_{\mathrm{b}}\mathbf{u})=\mathbf{0}$. Since $\mathbf{W}$ is positive definite, it follows that $\mathbf{u}_{\mathrm{b}}\in\mathrm{Ker}\,\mathbf{B}_{\mathrm{b}}$, which in view of (42) implies that $\mathbf{u}_{\mathrm{b}}\in\mathrm{Im}\,\mathbf{L}_{\mathrm{b}}$, and therefore $\mathbf{P}_{\ell}\mathbf{u}_{\mathrm{b}}=\mathbf{0}$. It follows that $\mathrm{Ker}\,\mathbf{P}_W\subset\mathrm{Ker}\,\mathbf{P}_{\ell}$, and hence $\mathrm{Ker}\,\mathbf{P}_W=\mathrm{Ker}\,\mathbf{P}_{\ell}$. □

2.3.2. Relationship between A-FETI and the original FETI method. Here, we prove mathematically that the *A*-FETI algorithm is identical to the one-level FETI method equipped with the Dirichlet preconditioner and with $\mathbf{Q}=\mathbf{W}$.

Lemma 3. All search directions $\mathbf{p}^k$ *and residuals* $\mathbf{w}^k$ *generated by the one-level FETI algorithm equipped with the Dirichlet preconditioner and* $\mathbf{Q}=\mathbf{W}$ *satisfy*

$$\mathbf{p}^{k^{\mathrm{T}}}\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^k=\mathbf{p}^{k^{\mathrm{T}}}\mathbf{w}^k \tag{65}$$

Proof. From Box 1, it follows that the successive search directions $\mathbf{p}^k$ can be expressed as

$$\begin{aligned}\mathbf{p}^k&=\mathbf{y}^k+\sum_{i=0}^{k-1}\zeta_i\mathbf{p}^i=\sum_{i=0}^{k}\zeta_i'\mathbf{y}^i\\&=\sum_{i=0}^{k}\zeta_i'\mathbf{P}\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}}\mathbf{w}^i\end{aligned} \tag{66}$$

where ζ_i and ζ_i' are some scalars. Also from Box 1 and equations (37) and (38), it follows that

$$\begin{aligned}\mathbf{w}^k&=\mathbf{P}^{\mathrm{T}}(\mathbf{d}-\mathbf{F}_{\mathrm{I}}\boldsymbol{\lambda}^k)\\&=\mathbf{P}^{\mathrm{T}}\mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}^{+}(\mathbf{f}_{\mathrm{b}}^{*}-\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^k)\end{aligned} \tag{67}$$

Using (66), (67) and (39), one finds

$$\begin{aligned}\mathbf{p}^{k^{\mathrm{T}}}\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^k&=\left(\sum_{i=0}^{k}\zeta_i'\mathbf{w}^{i^{\mathrm{T}}}\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}}\mathbf{P}^{\mathrm{T}}\right)\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}^{\mathrm{T}}\mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}^{+}(\mathbf{f}_{\mathrm{b}}^{*}-\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^k)\\&=\left(\sum_{i=0}^{k}\zeta_i'\mathbf{w}^{i^{\mathrm{T}}}\right)\mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}\left(\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}^{\mathrm{T}}\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}^{\mathrm{T}}\mathbf{B}_{\mathrm{b}}\right)\mathbf{S}_{\mathrm{bb}}^{+}\left(\mathbf{f}_{\mathrm{b}}^{*}-\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^k\right)\end{aligned} \tag{68}$$

Because $(\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}})^2=(\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}})$ (see (62) in Lemma 2) and $\mathbf{G}_{\mathrm{I}}=\mathbf{B}_{\mathrm{b}}\mathbf{R}_{\mathrm{b}}$, the following holds:

$$\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}(\mathbf{W})^{\mathrm{T}}\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}(\mathbf{W})^{\mathrm{T}}\mathbf{B}_{\mathrm{b}}=\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}(\mathbf{W})^{\mathrm{T}}\mathbf{P}(\mathbf{W})^{\mathrm{T}}\mathbf{B}_{\mathrm{b}} \tag{69}$$

where $\mathbf{P}(\mathbf{W})$ is $\mathbf{P}(\mathbf{Q}=\mathbf{W})$ as defined in (16). Hence, for $\mathbf{P}=\mathbf{P}(\mathbf{W})$ expression (68) simplifies to

$$\begin{aligned}\mathbf{p}^{k^{\mathrm{T}}}\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^k&=\left(\sum_{i=0}^{k}\zeta_i'\mathbf{w}^{i^{\mathrm{T}}}\right)\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}}\mathbf{P}(\mathbf{W})^{\mathrm{T}}\mathbf{P}(\mathbf{W})^{\mathrm{T}}\mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}^{+}\left(\mathbf{f}_{\mathrm{b}}^{*}-\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^k\right)\\&=\left(\sum_{i=0}^{k}\zeta_i'\mathbf{w}^{i^{\mathrm{T}}}\right)\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}}\mathbf{P}(\mathbf{W})^{\mathrm{T}}\mathbf{w}^k\\&=\mathbf{p}^{k^{\mathrm{T}}}\mathbf{w}^k\end{aligned} \tag{70}$$

Note that Lemma 3 holds even in the absence of floating subdomains. In that case, $\mathbf{P}=\mathbf{I}$ and the result (70) follows from equation (68). □

Theorem 2. At each iteration k, the A-FETI algorithm generates the same approximate solution of problem (2) *as the one-level FETI method equipped with the Dirichlet preconditioner and* $\mathbf{Q}=\mathbf{W}$.

Proof. First, we note that using equation (64), A-FETI's projector $\mathbf{P}_{\mathrm{R}}$ can be re-written as

$$\begin{aligned}\mathbf{P}_{\mathrm{R}} &= \mathbf{I} - \mathbf{P}_{\ell}^{\mathrm{T}}\mathbf{R}_{\mathrm{b}}(\mathbf{R}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}_{\ell}\mathbf{R}_{\mathrm{b}})^{-1}\mathbf{R}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}_{\ell}\\ &= \mathbf{I} - \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{G}_{\mathrm{I}}(\mathbf{G}_{\mathrm{I}}^{\mathrm{T}}\mathbf{W}\mathbf{G}_{\mathrm{I}})^{-1}\mathbf{G}_{\mathrm{I}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}} \qquad (71)\end{aligned}$$

so that

$$\begin{aligned}\mathbf{P}_{\ell}\mathbf{P}_{\mathrm{R}} &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}} - \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{G}_{\mathrm{I}}(\mathbf{G}_{\mathrm{I}}^{\mathrm{T}}\mathbf{W}\mathbf{G}_{\mathrm{I}})^{-1}\mathbf{G}_{\mathrm{I}}^{\mathrm{T}}\mathbf{W}\mathbf{B}_{\mathrm{b}}\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}(\mathbf{I} - \mathbf{W}\mathbf{G}_{\mathrm{I}}(\mathbf{G}_{\mathrm{I}}^{\mathrm{T}}\mathbf{W}\mathbf{G}_{\mathrm{I}})^{-1}\mathbf{G}_{\mathrm{I}}^{\mathrm{T}})\mathbf{W}\mathbf{B}_{\mathrm{b}}\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}(\mathbf{W})\mathbf{W}\mathbf{B}_{\mathrm{b}} \qquad (72)\end{aligned}$$

Next, we prove by induction that when the Dirichlet preconditioner is selected and the matrix $\mathbf{Q}$ is set to $\mathbf{Q}=\mathbf{W}$, the successive iterates computed by the A-FETI solver summarized in Box 2 are identical to those generated by the one-level FETI method summarized in Box 1. We remind the reader that because A-FETI employs a three-field formalism, a vector of Lagrange multiplier iterates $\boldsymbol{\lambda}^{*^k}$ generated by this algorithm is identical to a vector of Lagrange multipliers $\boldsymbol{\lambda}^k$ generated by the one-level FETI method if $\boldsymbol{\lambda}^{*^k}=\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^k$ (see equation (50)).

As a first step of the proof by induction, we show that the initial solutions computed by both FETI algorithms are identical.

Using equation (64), one finds that

$$\begin{aligned}\boldsymbol{\lambda}^{*^0} &= \mathbf{P}_{\ell}^{\mathrm{T}}\mathbf{R}_{\mathrm{b}}(\mathbf{R}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}_{\ell}\mathbf{R}_{\mathrm{b}})^{-1}\mathbf{e}\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{G}_{\mathrm{I}}(\mathbf{G}_{\mathrm{I}}^{\mathrm{T}}\mathbf{W}\mathbf{G}_{\mathrm{I}})^{-1}\mathbf{e}\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^0 \qquad (73)\end{aligned}$$

where $\boldsymbol{\lambda}^0$ is the initialization of $\boldsymbol{\lambda}$ specified in (15) with $\mathbf{Q}=\mathbf{W}$. From equations (72), (73), (37) and (38), it follows that the first residual computed by A-FETI satisfies

$$\begin{aligned}\mathbf{w}^{*^0} &= \mathbf{P}_{\mathrm{R}}^{\mathrm{T}}\mathbf{P}_{\ell}^{\mathrm{T}}\mathbf{S}_{\mathrm{bb}}^{+}(\mathbf{f}_{\mathrm{b}}^{*} - \boldsymbol{\lambda}^{*^0})\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}(\mathbf{W})^{\mathrm{T}}\mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}^{+}(\mathbf{f}_{\mathrm{b}}^{*} - \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^0)\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}(\mathbf{W})^{\mathrm{T}}(\mathbf{d} - \mathbf{F}_{\mathrm{I}}\boldsymbol{\lambda}^0)\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^0 \qquad (74)\end{aligned}$$

where $\mathbf{w}^0$ is the initial residual of the one-level FETI method summarized in Box 1. Using now equations (74), (72) and (39), the first search direction computed by the A-FETI solver can be written as

$$\begin{aligned}\mathbf{p}^{*^0} = \mathbf{y}^{*^0} &= \mathbf{P}_{\ell}\mathbf{P}_{\mathrm{R}}\mathbf{S}_{\mathrm{bb}}\mathbf{w}^{*^0}\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}(\mathbf{W})\mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^0\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}(\mathbf{W})\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}}\mathbf{w}^0\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}(\mathbf{W})\mathbf{y}^0\\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{p}^0 \qquad (75)\end{aligned}$$

and from equations (74), (75) and (37) and Lemma 3, it follows that the coefficient η^{*^0} satisfies

$$\begin{aligned}\eta^{*^0} &= \frac{\mathbf{p}^{*^{0^{\mathrm{T}}}}\mathbf{w}^{*^0}}{\mathbf{p}^{*^{0^{\mathrm{T}}}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{p}^{*^0}} \\ &= \frac{\mathbf{p}^{0^{\mathrm{T}}}\mathbf{B}_{\mathrm{b}}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^0}{\mathbf{p}^{0^{\mathrm{T}}}\mathbf{F}_{\mathrm{I}}\mathbf{p}^0} \\ &= \frac{\mathbf{p}^{0^{\mathrm{T}}}\mathbf{w}^0}{\mathbf{p}^{0^{\mathrm{T}}}\mathbf{F}_{\mathrm{I}}\mathbf{p}^0} \\ &= \eta^0 \end{aligned} \tag{76}$$

Next, we assume that for $i=0,\ldots,k-1$

$$\begin{aligned}\boldsymbol{\lambda}^{*^i} &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^i, \quad & \mathbf{w}^{*^i} &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^i \\ \mathbf{p}^{*^i} &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{p}^i, \quad & \eta^{*^i} &= \eta^i \end{aligned} \tag{77}$$

and conclude our proof by induction by showing that the above results also hold for $i=k$.

From assumptions (77) it follows that

$$\begin{aligned}\boldsymbol{\lambda}^{*^k} &= \boldsymbol{\lambda}^{*^{k-1}} + \eta^{*^{k-1}}\mathbf{p}^{*^{k-1}} \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^{k-1} + \eta^{k-1}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{p}^{k-1} \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}(\boldsymbol{\lambda}^{k-1} + \eta^{k-1}\mathbf{p}^{k-1}) \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\boldsymbol{\lambda}^k \end{aligned} \tag{78}$$

Similarly, from assumptions (77) and equations (37) and (72) it follows that

$$\begin{aligned}\mathbf{w}^{*^k} &= \mathbf{w}^{*^{k-1}} - \eta^{*^{k-1}}\mathbf{P}_{\mathrm{R}}^{\mathrm{T}}\mathbf{P}_{\ell}^{\mathrm{T}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{p}^{*^{k-1}} \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^{k-1} - \eta^{k-1}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{P}^{\mathrm{T}}(\mathbf{W})\mathbf{B}_{\mathrm{b}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{p}^{k-1} \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}(\mathbf{w}^{k-1} - \eta^{k-1}\mathbf{P}^{\mathrm{T}}(\mathbf{W})\mathbf{F}_{\mathrm{I}}\mathbf{p}^{k-1}) \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{W}\mathbf{w}^k \end{aligned} \tag{79}$$

From the above result and equations (39) and (37) we deduce

$$\begin{aligned}\mathbf{y}^{*^k} &= \mathbf{P}_{\ell}\mathbf{P}_{\mathrm{R}}\mathbf{S}_{\mathrm{bb}}\mathbf{w}^{*^k} \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{P}(\mathbf{W})\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}}\mathbf{w}^k \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{y}^k \end{aligned} \tag{80}$$

$$\begin{aligned}\mathbf{p}^{*^k} &= \mathbf{y}^{*^k} - \sum_{i=0}^{k-1}\frac{\mathbf{y}^{*^{k^{\mathrm{T}}}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{p}^{*^i}}{\mathbf{p}^{*^{i^{\mathrm{T}}}}\mathbf{S}_{\mathrm{bb}}^{+}\mathbf{p}^{*^i}}\mathbf{p}^{*^i} \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\left(\mathbf{y}^k - \sum_{i=0}^{k-1}\frac{\mathbf{y}^{k^{\mathrm{T}}}\mathbf{F}_{\mathrm{I}}\mathbf{p}^i}{\mathbf{p}^{i^{\mathrm{T}}}\mathbf{F}_{\mathrm{I}}\mathbf{p}^i}\mathbf{p}^i\right) \\ &= \mathbf{B}_{\mathrm{b}}^{\mathrm{T}}\mathbf{p}^k \end{aligned} \tag{81}$$

and from equations (81) and (37) and Lemma 3 it follows that

$$\begin{aligned}\eta^{*^k} &= \frac{\mathbf{p}^{*^{k^{\mathrm{T}}}}\mathbf{w}^{*^k}}{\mathbf{p}^{*^{k^{\mathrm{T}}}}\mathbf{S}^{+}_{\mathrm{bb}}\mathbf{p}^{*^k}} \\ &= \frac{\mathbf{p}^{k^{\mathrm{T}}}\mathbf{B}_{\mathrm{b}}\mathbf{B}^{\mathrm{T}}_{\mathrm{b}}\mathbf{W}\mathbf{w}^k}{\mathbf{p}^{k^{\mathrm{T}}}\mathbf{F}_{\mathrm{I}}\mathbf{p}^k} \\ &= \frac{\mathbf{p}^{k^{\mathrm{T}}}\mathbf{w}^k}{\mathbf{p}^{k^{\mathrm{T}}}\mathbf{F}_{\mathrm{I}}\mathbf{p}^k} \\ &= \eta^k \end{aligned} \tag{82}$$

which completes the proof by induction that, when the Dirichlet preconditioner is selected and the matrix $\mathbf{Q}$ is set to $\mathbf{Q}=\mathbf{W}$, the successive iterates computed by the A-FETI solver are identical to those generated by the one-level FETI method.

In the event where all subdomains are non-floating, $\mathbf{R}=\mathbf{0}$, $\mathbf{P}=\mathbf{I}$ and $\mathbf{P}_{\mathrm{R}}=\mathbf{I}$. In that case, equation (73) becomes $\boldsymbol{\lambda}^{*^0}=\mathbf{B}^{\mathrm{T}}_{\mathrm{b}}\boldsymbol{\lambda}^0=\mathbf{0}$, and the reader can check that all the results of the above proof remain valid. □

Hence, we conclude that

The A-FETI method is a particular instance of the FETI method equipped with the Dirichlet preconditioner that corresponds to the case $\mathbf{Q}=\mathbf{W}$.

Next, we note that from the definitions (15) and (16) of $\boldsymbol{\lambda}^0(\mathbf{Q})$ and $\mathbf{P}(\mathbf{Q})$, $\mathbf{Q}$ always operates on $\mathbf{G}_{\mathrm{I}}$. Since $\mathbf{G}_{\mathrm{I}}=\mathbf{B}\mathbf{R}$, then $\mathbf{Q}$ always operates on an image of $\mathbf{B}$, and from equation (63) of Lemma 2 it follows that

$$\mathbf{P}(\mathbf{W})=\mathbf{P}(\mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{B}^{\mathrm{T}}_{\mathrm{b}}\mathbf{W}) \tag{83}$$

$$\boldsymbol{\lambda}^0(\mathbf{W})=\boldsymbol{\lambda}^0(\mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{B}^{\mathrm{T}}_{\mathrm{b}}\mathbf{W}) \tag{84}$$

Therefore, we can also state that

The A-FETI method is a particular instance of the FETI method equipped with the Dirichlet preconditioner that corresponds to the case

$$\mathbf{Q}=\mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{B}^{\mathrm{T}}_{\mathrm{b}}\mathbf{W} \tag{85}$$

From equations (59) and (85), it follows that the specific choice of the matrix $\mathbf{Q}$ adopted by the A-FETI algorithm is a degenerate form of the two choices $\mathbf{Q}^{\mathrm{D}}=\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{D}^{-1}}$ and $\mathbf{Q}^{\mathrm{L}}=\bar{\mathbf{F}}_{\mathrm{I}}^{\mathrm{L}^{-1}}$ that were previously proposed in [18]. This degenerate form does not account for any stiffness properties of the given substructures, and therefore cannot be suitable for heterogeneous problems.

It remains to investigate whether setting $\mathbf{Q}=\mathbf{W}$ (or its equivalent $\mathbf{W}\mathbf{B}_{\mathrm{b}}\mathbf{B}^{\mathrm{T}}_{\mathrm{b}}\mathbf{W}$) improves the performance of the original FETI method equipped with $\mathbf{Q}=\mathbf{I}$.

2.4. Corrected performance comparisons of the one-level FETI, A-FETI, and two-level FETI algorithms

In [19, 20], the authors have applied the A-FETI algorithm to the iterative solution of systems of equations associated with the structural analysis of a square cantilever plate subjected to a

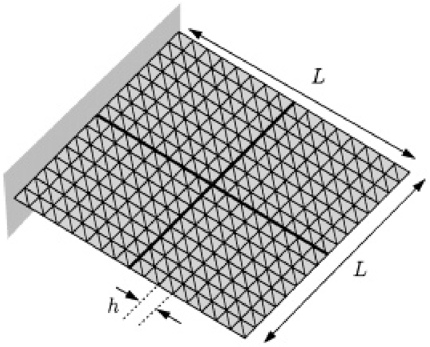


Figure 3. Cantilever plate problem discretized by 3-node ANDES elements—uniform mesh partition

uniform pressure load (Figure 3), and discretized by three-noded ANDES plate bending elements [42]. This model problem was previously solved in [2] by the original one- and the two-level FETI methods. However, whereas in [2] the square plate was decomposed into *irregularly shaped* substructures that are representative of more realistic problems, *uniform* mesh partitions were considered in [19, 20]. Yet, the authors of the A-FETI algorithm have compared in [19, 20] their performance results obtained for uniform mesh partitions with the performance results published in [2] for the FETI and two-level FETI methods using arbitrary mesh partitions. Given that the convergence of substructure based iterative methods depends on the shape of the substructures, and more specifically on their aspect ratios [43, 44], it follows that the comparisons reported in [19, 20] and the conclusions formulated in these references are *invalid*. Here, we clear up this confusion and correct the conclusions. We consider both cases corresponding to uniform and irregular mesh partitions. In each case, we apply all of the original one-level FETI method equipped with $\mathbf{Q}=\mathbf{I}$ and $\mathbf{Q}=\mathbf{W}$, the A-FETI algorithm, and the two-level FETI method to the solution of the benchmark plate problem graphically depicted in Figure 3. For all of these methods, we employ the same convergence criterion given below

$$\frac{||\mathbf{K}_{\mathrm{g}}\mathbf{u}_{\mathrm{g}}^{k}-\mathbf{f}_{\mathrm{g}}||}{||\mathbf{f}_{\mathrm{g}}||}<\varepsilon \tag{86}$$

where $||\cdot||$ denotes the Eucledian norm of a vector. All computations are performed on a single O2000 processor in double precision arithmetic.

Note that

(i) The stopping criterion (86) is based on the *exact* primal residual, and not on the estimated primal residual developed in [18] and adopted for the benchmark problems reported in [2]. This stopping criterion is applied in the same manner to both FETI and A-FETI solvers.

(ii) All auxiliary coarse problems arising in the FETI and A-FETI methods are solved by the same direct method.

(iii) Because the arbitrary mesh partitions that were employed in [2] for the cantilever plate problem discussed here were no longer available at the time of writing this paper, a new set of arbitrary mesh partitions have been generated for the purposes of this work.

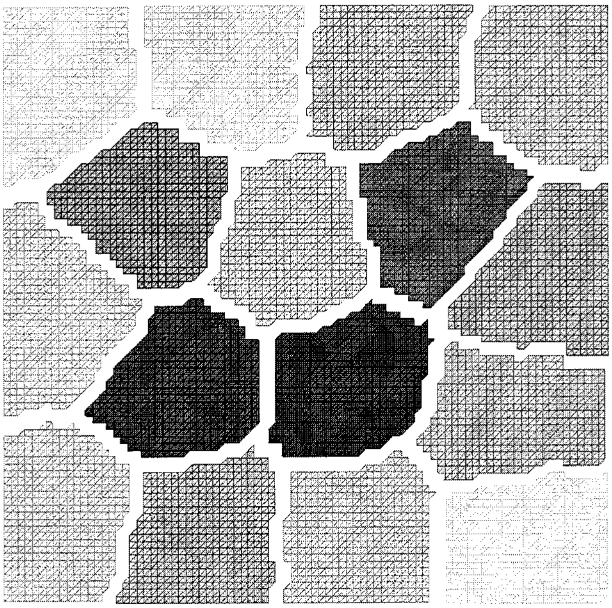


Figure 4. Irregular decomposition into 16 substructures

(iv) For all the reasons outlined above, the iteration counts reported in this paper for the case of irregular mesh partitions may slightly differ from those published in [2]. However, for the same reasons outlined above, the comparisons offered in this paper between the performances of the FETI and A-FETI solvers are technically sound.

First, we consider five different finite element discretizations characterized by a uniform mesh size h, and for each mesh a uniform $\sqrt{N_s} \times \sqrt{N_s}$ partition. *Even though we have mathematically proved in this paper that A-FETI is the one-level FETI method equipped with* $\mathbf{Q}=\mathbf{W}$, *we have programmed both methods and verified that for this and every other problem we have investigated, both of them generate at each iteration k the same numerical results, and converge in the same number of iterations*. We report in Table I the number of iterations for convergence and the total solution CPU time in seconds (between parentheses) obtained for each FETI method equipped with the Dirichlet preconditioner, and for $\varepsilon = 10^{-6}$.

Next, we consider all but the first of the uniform meshes investigated above, and for each one of them, we generate an irregular decomposition such as that depicted in Figure 4. We report in Table II the performance results obtained for $\varepsilon = 10^{-6}$.

From the performance results displayed in Table I and Table II, we formulate the following conclusions:

1. The A-FETI algorithm is the same as the one-level FETI method equipped with $\mathbf{Q}=\mathbf{W}$.
2. The one-level FETI method equipped with $\mathbf{Q}=\mathbf{W}$—and therefore the A-FETI algorithm—performs essentially the same, iteration-wise and CPU-wise, as the one-level FETI method equipped with $\mathbf{Q}=\mathbf{I}$.

Table I. Cantilever plate problem—uniform mesh partitions. Dirichlet pre-conditioner – $\varepsilon = 10^{-6}$

h/L	N_s	FETI ($\mathbf{Q} = \mathbf{I}$)	FETI ($\mathbf{Q} = \mathbf{W}$) and A-FETI	Two-level FETI
1/16	2×2	14 itr. (0·07 s)	14 itr. (0·07 s)	12 itr. (0·08 s)
	4×4	30 itr. (0·22 s)	30 itr. (0·22 s)	18 itr. (0·21 s)
	8×8	36 itr. (0·71 s)	35 itr. (0·69 s)	15 itr. (0·19 s)
1/40	4×4	42 itr. (1·05 s)	42 itr. (1·05 s)	26 itr. (0·96 s)
1/80	4×4	49 itr. (6·29 s)	49 itr. (6·29 s)	31 itr. (5·06 s)
1/90	6×6	81 itr. (12·07 s)	81 itr. (12·07 s)	33 itr. (6·54 s)
1/120	8×8	124 itr. (33·30 s)	123 itr. (33·21 s)	35 itr. (12·67 s)

Table II. Cantilever plate problem—arbitrary mesh partitions. Dirichlet pre-conditioner – $\varepsilon = 10^{-6}$

h/L	N_s	FETI ($\mathbf{Q} = \mathbf{I}$)	FETI ($\mathbf{Q} = \mathbf{W}$) and A-FETI	Two-level FETI
1/40	16	53 itr. (1·35 s)	53 itr. (1·35 s)	29 itr. (0·98 s)
1/80	16	63 itr. (8·24 s)	64 itr. (8·39 s)	40 itr. (6·95 s)
1/90	36	108 itr. (16·55 s)	109 itr. (16·71 s)	44 itr. (8·29 s)
1/120	64	153 itr. (43·10 s)	153 itr. (43·10 s)	50 itr. (17·20 s)

3. For a given uniform discretization characterized by a mesh size h and a specified number of substructures N_s, each FETI method performs better iteration-wise when the mesh partition is uniform rather than irregular.
4. Contrary to what was speculated in [19, 20], neither the one-level FETI method equipped with $\mathbf{Q} = \mathbf{I}$, nor the A-FETI method—that is, the one-level FETI method equipped with $\mathbf{Q} = \mathbf{W}$—are numerically scalable fourth-order plate problems, neither with respect to the mesh size h nor the number of substructures N_s.
5. The two-level FETI method is numerically scalable for fourth-order plate problems. This result, which has been mathematically established in several previous papers, is by now a well-known result [2, 16].
6. Contrary to the conjectures published in [19, 20], the two-level FETI method not only outperforms the original one-level FETI and A-FETI methods in the number of iterations required for convergence, but also in the total solution CPU time. More specifically, the two-level FETI method can be more than twice as fast as each of the one-level FETI method equipped with $\mathbf{Q} = \mathbf{I}$ and the A-FETI methods at solving the plate problem investigated herein.

Finally, in order to complete the comparison of the original one-level FETI method ($\mathbf{Q} = \mathbf{I}$) and the A-FETI algorithm, we consider the solution of the plane stress problem illustrated in Figure 5. This second-order elasticity problem is discretized here by 4-noded plane stress finite elements. The performance results reported in Table III highlight again the fact that the A-FETI algorithm has essentially the same convergence rate as the basic one-level FETI method. This conclusion is also supported by the convergence curves given in Figure 6 for the case $h/L = 1/40$ and a 4×4 mesh partition.

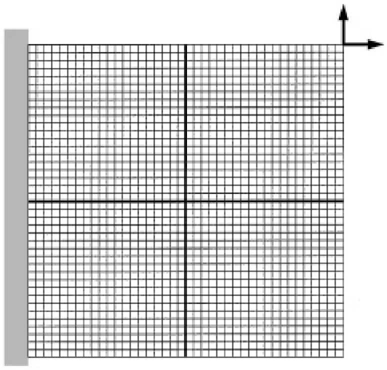


Figure 5. Plane stress problem—uniform mesh partition

Table III. Plane stress problem—Regular mesh partitions. Dirichlet preconditioner $-\varepsilon = 10^{-8}$

h/L	N_s	FETI ($\mathbf{Q} = \mathbf{I}$)	FETI ($\mathbf{Q} = \mathbf{W}$) A-FETI
1/8	2×2	9 itr.	9 itr.
	4×4	12 itr.	11 itr.
1/16	2×2	11 itr.	11 itr.
	4×4	13 itr.	13 itr.
	8×8	14 itr.	14 itr.
1/40	2×2	12 itr.	12 itr.
	4×4	17 itr.	17 itr.
	8×8	18 itr.	18 itr.

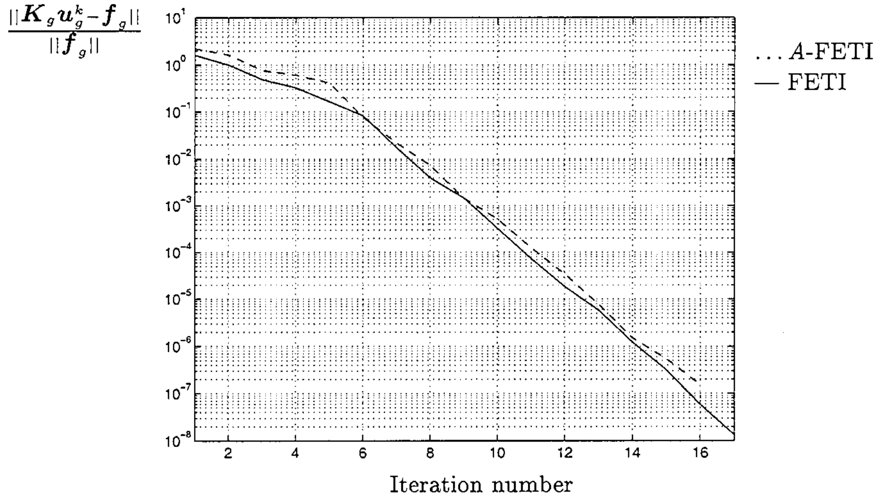


Figure 6. Plane stress problem—$h/L = 1/40$ and 4×4 mesh partition

3. PERFORMANCE COMPARISONS WITH AN OPTIMIZED SPARSE SOLVER

Here, we turn our attention to two realistic structural problems that are representative of challenging industrial applications, and to performance comparisons between the one-level FETI method, the two-level FETI method, and an optimized sparse solver. Indeed, most if not all production and/or commercial finite element packages rely today on a sparse solver when the systems of equations they generate are to be solved by a direct method. Previous comparisons of a FETI method and a direct solver have focused mostly if not exclusively on the skyline solver, because the latter is widely available in the academic community if not easily programmable. Today, many computer hardware manufacturers provide an optimized sequential and/or parallel sparse solver in their scientific software library, which eases the process of benchmarking iterative solvers against the most powerful and popular direct methods. In this work, we use for this purpose the PSLDLT library that is available on any Silicon Graphics system, which requires that the given sparse matrix be stored in the Harwell-Boeing format (also known as the Compressed Column Storage format), and which is parallelized by Silicon Graphics on the O2000 multiprocessor. We also use the optimal equation renumbering scheme provided by this library.

The first problem we consider is a large-scale *thin* shell problem. It allows us to demonstrate that for fourth-order problems, the two-level FETI method is not only mathematically superior to one-level FETI algorithms, but is also more computationally efficient. Previously [15], we have reported that depending on the computational platform used for carrying out the finite element analysis, and on the chosen number of substructures, the two-level FETI method may or may not significantly outperform CPU-wise the one-level FETI method when applied to the solution of shell problems. However, the computer implementation of the two-level FETI method has recently undergone a significant number of improvements, among which we note the solution of the second-level coarse problem by an efficient parallel direct method rather than an iterative one, and the ability to choose a number of substructures independently from the number of processors available on the given parallel computer [45]. Here, we show that as a result of these improvements, the two-level FETI method is the FETI method of choice for shell problems, no matter how large is the desired number of substructures.

The second problem we discuss in this section is a mid-scale solid mechanics problem whose stiffness matrix requires over 1 Gbyte of memory when stored in a sparse compact format.

In both cases, we perform all computations in 64-bit arithmetic on a 24-processor Origin 2000 system with 8 Gbytes of memory. We point our that while the PSLDLT library is optimized for the Origin 2000 system, and is most likely written in assembly language, our FETI algorithms are programmed in C++ and are not optimized for any specific computer architecture. 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.

3.1. A car wheel problem

The mesh shown in Figure 7 corresponds to the finite element discretization of an alloy wheel by 78 464 three-noded shell elements. It contains 39 269 nodes and 235 614 d.o.f. The ratio of the thickness of this wheel to its diameter is equal to 150. Hence, this is a thin shell problem [46]. The wheel is clamped at a few center points, and a set of concentrated forces are applied at its top rim. Several sequential and parallel linear static stress analyses are performed

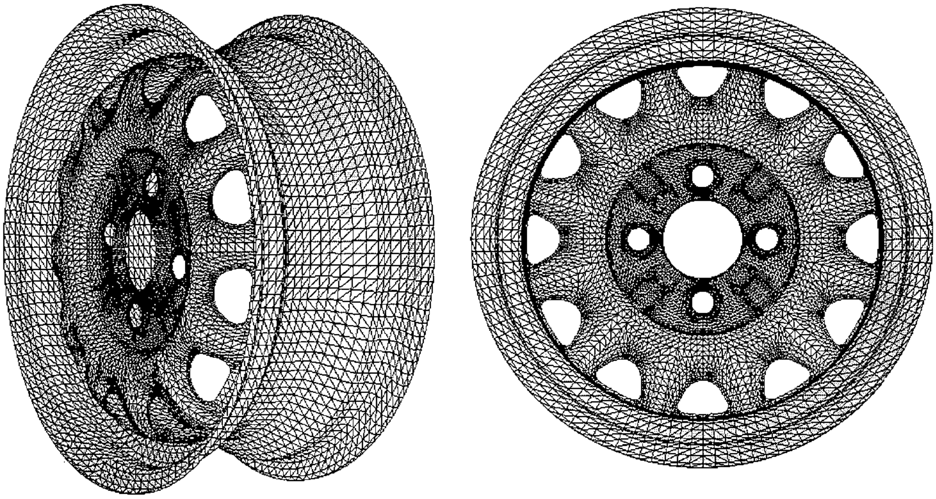


Figure 7. Finite element discretization of a car wheel

Table IV. Car wheel problem—235 614 d.o.f.—8 Origin 2000 processors. Dirichlet preconditioner $-\varepsilon = 10^{-5}$

N_s	Two-level FETI	One-level FETI
40	67 itr. (77·8 s and 581 Mb)	203 itr. (161·4 s and 625 Mb)
80	86 itr. (57·8 s and 484 Mb)	270 itr. (150·6 s and 545 Mb)
120	86 itr. (54·1 s and 453 Mb)	272 itr. (130·6 s and 518 Mb)
140	93 itr. (59·3 s and 440 Mb)	291 itr. (136·0 s and 507 Mb)

using the sparse direct solver, and both the one- and two-level FETI substructure based iterative solvers.

The parallel solution using the PSLDLT sparse solver on 8 Origin 2000 processors of the system of equations associated with the mesh shown in Figure 7 requires 502 Mbytes of memory, and consumes 42·9 s CPU. The parallel performance results obtained for the solution of the same system of equations using the Dirichlet preconditioned one- and two-level FETI methods on the same 8 Origin 2000 processors are summarized in Table IV for various mesh partitions. These performance results show that

(1) The two-level FETI method is numerically scalable with respect to the number of substructures N_s, but the one-level FETI method is not.

(2) For a fixed problem size, increasing the number of substructures reduces the computational complexity of the solution of the local problems, but increases the size of the interface problem. Hence, for a numerically scalable domain decomposition method, increasing the number of substructures up to a certain critical number can decrease the total solution CPU time. Increasing N_s beyond that critical number increases the total solution CPU time. The results

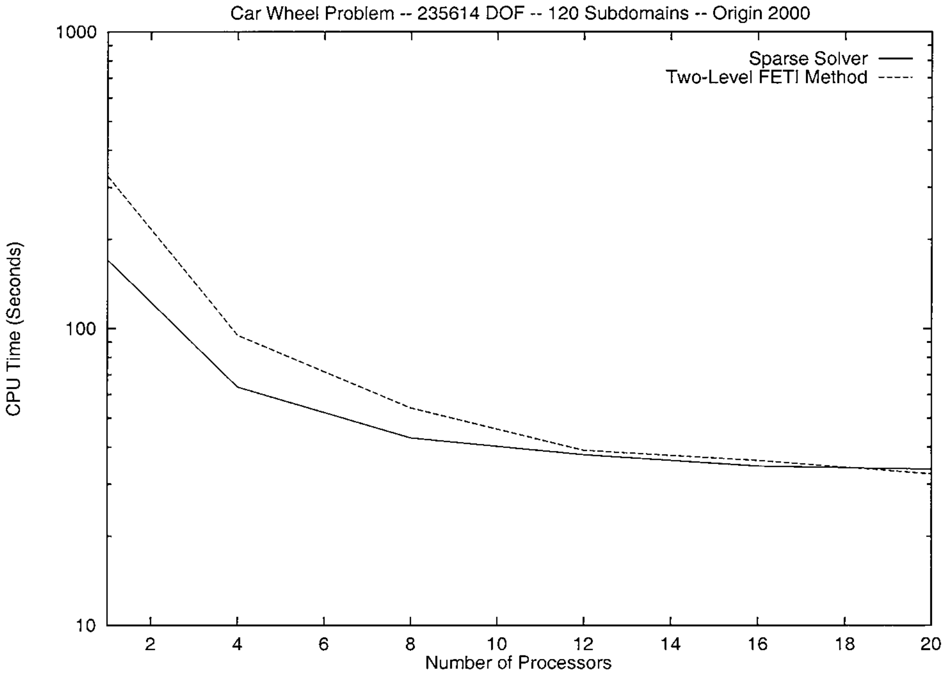


Figure 8. Car wheel coarse mesh—parallel performance results

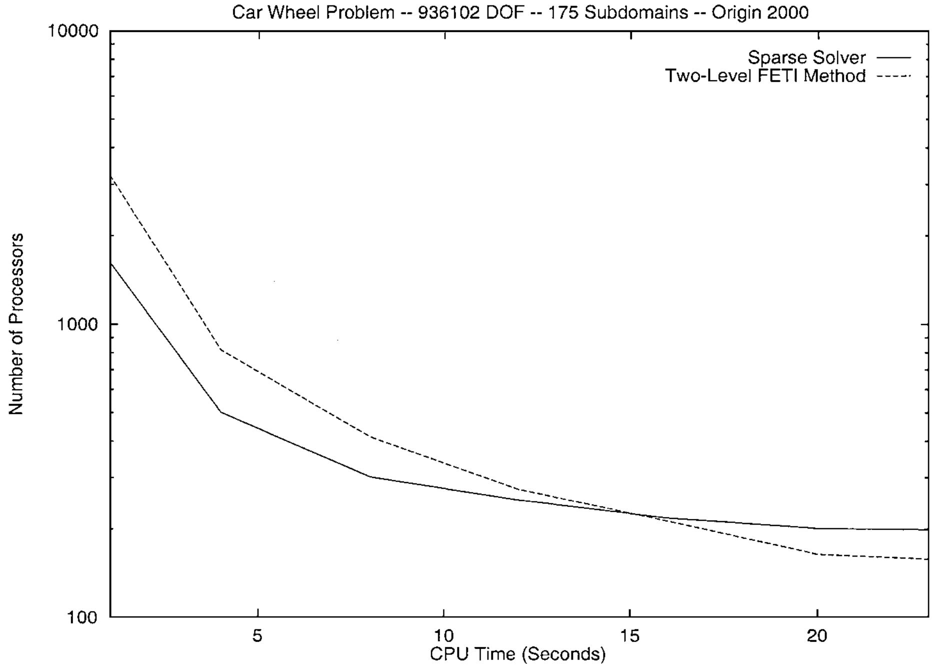


Figure 9. Car wheel fine mesh—parallel performance results

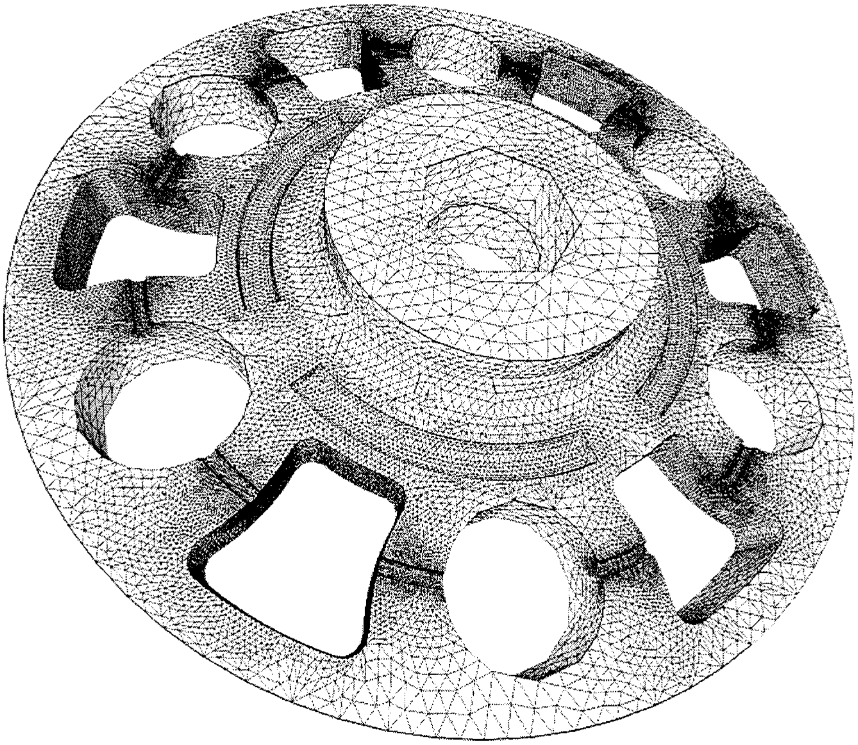


Figure 10. Finite element discretization of a wheel carrier

reported in Table IV suggest that for the FETI methodology and this car wheel problem, the critical number of substructures is $N_s^* = 120$.

(3) For each considered mesh partition, the two-level FETI method converges in about 3 times fewer iterations than the one-level FETI algorithm, and outperforms it CPU-wise by a factor larger than 2.

(4) For this problem, the memory requirement of the one-level FETI algorithm is higher than that of the two-level FETI method because both algorithms are equipped with full reorthogonalization [18]. Since the one-level FETI algorithm performs on average 3 times more iterations than the two-level FETI method, it consumes more memory for the storage of all search directions.

(5) Also for this problem, the memory requirement of the two-level FETI method is comparable to that of the sparse solver. However, the sparse solver is 1·3 times faster.

The performance results of the two-level FETI method and the highly optimized PSLDLT sparse solver for the same car wheel problem but a number of Origin 2000 processors ranging between 1 and 20 are reported in Figure 8. These results indicate that the two-level FETI method is more amenable to parallel processing than a sparse solver. Indeed, for this problem, the PSLDLT sparse solver is about twice as fast as the two-level FETI method on a single processor configuration of the O2000 machine. However, as the number of processors is increased, the two-level FETI method

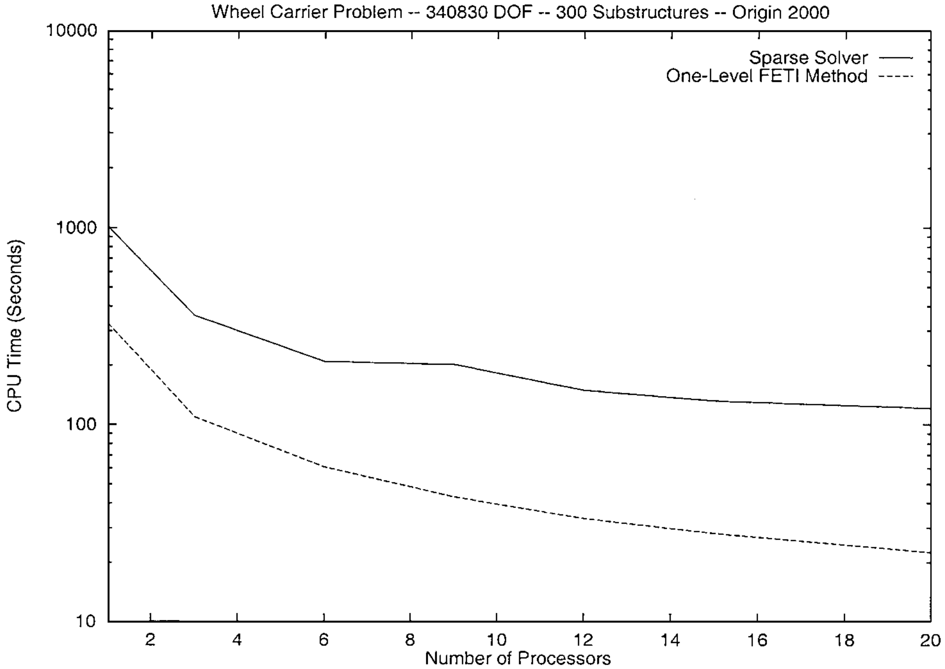


Figure 11. Wheel carrier problem—parallel performance results

exhibits a better parallel scalability than the sparse solver, and outperforms it on a 20-processor configuration. The same trend can be observed in Figure 9 for a finer mesh of the car wheel problem containing 313 856 three-noded shell elements, 156 017 nodes, and 936 102 d.o.f., and partitioned into 175 substructures. For the latter problem, the two-level FETI method outperforms the PSLDLT sparse solver for a number of processors greater or equal to 16.

3.2. A wheel carrier problem

Next, we consider the static stress analysis of the wheel carrier shown in Figure 10. The finite element mesh contains 542 144 four-noded tetrahedra elements, 113 610 nodes, and 340 830 d.o.f. The one-level FETI method is the appropriate FETI method for this second-order elasticity problem.

The solution of this problem by the PSLDLT sparse solver requires 1·3 Gbytes of memory, and consumes 1011·4 s CPU on a single Origin 2000 processor. On the other hand, its solution by the one-level FETI method equipped with the lumped preconditioner, $\varepsilon = 10^{-5}$, and using 300 substructures—this turned out to be the optimal number of substructures—requires only 658 megabytes and consumes only 327·9 seconds CPU on the same single Origin 2000 processor. Hence, for this problem, the one-level FETI method requires half the storage space needed by the sparse solver, and is 3 times faster.

The parallel performances of the sparse and one-level FETI method applied to the solution of this wheel carrier problem are summarized in Figure 11. Topologically, three-dimensional problems such as this one produce more fill-ins for a sparse solver than topologically two-dimensional problems such as the shell discretization discussed in the previous section. Consequently, for a fixed size problem, the sparse solver performs more computations in the former case than in the latter case, and therefore parallelizes better. These two conclusions are illustrated in Figure 11 which shows that, for the wheel carrier problem, the PSLDLT sparse solver is outperformed by the one-level FETI method on average by a factor equal to 5, even though it exhibits a better parallel speedup than when applied to the first shell problem of Section 3.1.

4. CONCLUSIONS

The recently developed algebraic variant of the one-level FETI method called A-FETI is a three-field formulation of the one-level FETI method. In this paper, we have proved mathematically that A-FETI not only relies on the same mechanical and numerical concepts as FETI, but is exactly identical to one particular instance of the one-level FETI methodology. We have also shown numerically that A-FETI delivers the same performance results as the one-level FETI method, and is not numerically scalable for plate and shell problems, neither with respect to the problem size nor the number of substructures. For plate and shell problems, among the family of FETI algorithms, only the two-level FETI method equipped with the so-called Dirichlet preconditioner and the substructure corner modes is numerically scalable with respect to both the problem and mesh partition sizes. Furthermore, using both academic and industrial examples, we have shown that the two-level FETI method is not only numerically scalable for plate and shell problems, but also outperforms one-level FETI algorithms CPU-wise, typically by a factor larger than two. Shell problems are often topologically two-dimensional. For such problems, the two-level FETI method delivers a performance that is comparable to that of a highly optimized direct sparse solver. However, for topologically three-dimensional problems discretized by solid elements, the one-level FETI method outperforms a highly optimized direct sparse solver by several factors and is more memory efficient.

REFERENCES

1. Farhat C. A lagrange multiplier based divide and conquer finite element algorithm. *Journal of Computational Systems Engineering* 1991; **2**(2/3):149–156.
2. 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.
3. Farhat C, Mandel J, Roux FX. Optimal convergence properties of the FETI domain decomposition method. In *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, Keyes D, Saad Y, Truhlar DG (eds). SIAM: Philadelphia, PA, 1994.
4. Farhat C, Roux FX. A method of finite tearing and interconnecting and its parallel solution algorithm. *Computer Methods in Applied Mechanics and Engineering* 1991; **32**:1205–1227.
5. Dinh QV, Fanion T. Applications of dual schur complement preconditioning to problems in computational fluid dynamics and computational electromagnetics. In *Proceedings of the Ninth Conference on Domain Decomposition*, Bergen, Norway, June 1996.
6. Dostal Z, Friedlander A, Santos S. Solution of coercive and semicoercive contact problems by FETI domain decomposition. *Contemporary Mathematics* 1998; **218**:82–93.
7. Klawonn A, Widlund OB. A domain decomposition method with Lagrange multipliers for second order elasticity. In *Proceedings of the 11th International Conference on Domain Decomposition Methods*, Greenwich, U.K., July 1998.
8. Lacour C. Non conforming domain decomposition method for plate problems. *Contemporary Mathematics* 1998; **218**:304–310.
9. Mandel J. Balancing domain decomposition. *Communications in Applied Numerical Methods and Engineering* 1993; **9**:233–241.
10. Mandel J, Tezaur R. Convergence of a substructuring method with Lagrange multipliers. *Numerische Mathematik* 1996; **73**:473–487.

11. Papadrakakis M, Tsompanakis Y. Domain decomposition methods for parallel solution of sensitivity analysis problems. *Technical Report 96-1*, Institute of Structural Analysis and Seismic Research, NTUA, Athens, Greece, 1996.
12. Park KC, Justino MR, Felippa CA. An algebraically partitioned FETI method for parallel structural analysis: algorithm description. *International Journal for Numerical Methods in Engineering* 1997; **40**:2717–2737.
13. Risler F, Rey C. On the reuse of Ritz vectors for the solution of nonlinear elasticity problems by domain decomposition methods. *Contemporary Mathematics* 1998; **218**:334–340.
14. Soulat D, Devries F. Mechanical criteria for the subdomains decomposition: applications to heterogeneous structures and composite materials. In *Proceedings of the Ninth Conference on Domain Decomposition*, Bergen, Norway, June 1996.
15. Farhat C, Chen PS, 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.
16. Mandel J, Tezaur R, Farhat C. A scalable substructuring method by Lagrange multipliers for plate bending problems. *UCD/CCM Report* 84, Center for Computational Mathematics, University of Colorado at Denver 1996. *SIAM Journal on Numerical Analysis*, (in press).
17. Tezaur R. *Analysis of Lagrange multiplier based domain decomposition. Ph.D. Thesis*, University of Colorado at Denver, 1998.
18. Farhat C, Roux FX. Implicit parallel processing in structural mechanics. *Computational Mechanics Advances* 1994; **2**(1):1–124.
19. Felippa CA, Park KC. A direct flexibility method. *Computational Methods in Applied Mechanics and Engineering* 1997; **149**:319–337.
20. Justino MR, Park KC, Felippa CA. An algebraically partitioned FETI method for parallel structural analysis: performance evaluation. *International Journal for Numerical Methods in Engineering* 1997; **40**:2739–2758.
21. Schwarz HA. *Gesammelte Mathematische Abhandlungen*, Vol. 2, Springer: Berlin, 1890; 133–143. First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, 1870; **15**:272–286.
22. Glowinski R, Golub GH, Meurant GA, Périaux J, (eds). *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*. SIAM: Philadelphia, PA, U.S.A., 1988.
23. 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.
24. 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 D, Chan TF, Meurant GA, Scroggs JS, Voigt RG (eds). SIAM, 1992; 271–292.
25. 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.
26. Rixen D. Dual schur complement method for semi-definite problems. *Contemporary Mathematics* 1998; **18**:341–348.
27. Rixen D. Substructuring and dual methods in structural analysis. *Ph.D. Thesis*, Université de Liège, Belgium, Collection des Publications de la Faculté des Sciences appliquées, no. 175, 1997.
28. Farhat C, Chen PS, Stern P. Towards the ultimate iterative substructuring method: combined numerical and parallel scalability, and multiple load cases. *International Journal for Numerical Methods in Engineering* 1994; **37**(11): 1945–1975.
29. Farhat C, Crivelli L, Roux FX. Extending substructures based iteratives solvers to multiple load and repeated analyses. *Computer Methods in Applied Mechanics and Engineering* 1994; **117**:195–209.
30. Farhat C, Chen PS, Mandel J. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *International Journal for Numerical Methods in Engineering* 1995; **38**(22):3831–3853.
31. Farhat C, Crivelli L, Roux FX. A transient FETI methodology for large-scale parallel implicit computations in structural mechanics. *International Journal for Numerical Methods in Engineering* 1994; **37**(11):1945–1975.
32. Farhat C, Hemez F. Improving the convergence rate of a transient substructuring iterative method using the rigid body modes of its static equivalent. In *Structures, Structural Dynamics and Material Conference*, New Orleans, USA, April 10–12, 1995. 36th AIAA/ASME/ASCE/AHS/ASC. AIAA-95-1271.
33. Géradin M, Coulon D, Delsemme JP. Parallelization of the SAMCEF finite element software through domain decomposition and FETI algorithm. *International Journal of Supercomputer Applications* 1997; **11**:286–298.
34. Rey Ch, Devries F, Lene F. Parallelism in non linear computation of heterogeneous structures. *Calculateurs Parallèles* 1995; **7**(3).
35. Roux FX. Parallel implementation of a domain decomposition method for non-linear elasticity problems. In *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, Keyes D, Saad Y, Truhlar D (eds). SIAM, 1994; 161–175.
36. Farhat C, Lacour C, Rixen D. Incorporation of linear multipoint constraints in substructure based iterative solvers, Part I: a numerically scalable algorithm. *Computer Methods in Applied Mechanics and Engineering* 1998; **43**:997–1016.
37. Farhat C, Chen PS, 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.

38. Aminpour MA, Ransom JB, McCleary SL. Coupled analysis of independently modeled finite element subdomains. In *Structural Dynamics and Material Conference*, Dallas, USA, 13–15 April 1992. 33rd AIAA/ASME/AHS/ASC Structures.
39. Brezzi F, Fortin M. *Mixed and Hybrid Finite Elements Methods*, Computational Mathematics. Springer: Berlin, 1991.
40. Brezzi F, Marini LD. A three-field domain decomposition method. In *Sixth Conference on Domain Decomposition Methods for Partial Differential Equations*, Kuznetsov YA, Quarteroni A, Periaux J, Widlund OB (eds). Como, Italy, 15–19 June 1992. AMS: Providence, RI, USA, 1993; 27–34.
41. Park KC, Felippa CA. A variational framework for solution method developments in structural mechanics. *Transactions of the ASME* 1998; **65**:242–249.
42. Militello C, Felippa C. The first ANDES elements: 9-dof plate bending triangles. *Computational Methods in Applied and Mechanical Engineering* 1991; **93**:217–246.
43. 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.
44. Le Tallec P, Vidrascu M. Solving large scale structural problems on parallel computers using domain decomposition techniques. In *Parallel Solution Methods in Computational Mechanics*. Wiley: New York, 1997; 49–82.
45. Lesoinne M, Pierson K. An efficient FETI implementation on distributed shared memory machines with independent numbers of subdomains and processors. *Contemporary Mathematics* 1998; **18**:318–324.
46. Zienkiewicz OC, Taylor RL. *The Finite Element Method*, 4th ed., Vol. 2. McGraw-Hill: New York, 1991.