Contemporary Mathematics
Volume **218**, 1998
B 0-8218-0988-1-03008-9

Parallel Implementation of Direct Solution Strategies for the Coarse Grid Solvers in 2-level FETI Method

François-Xavier Roux and Charbel Farhat

1. Introduction

The FETI method is based on introducing Lagrange multipliers along interfaces between subdomain to enforce continuity of local solutions [**4**]. It has been demonstrated to be numerically scalable in the case of second-order problems, thanks to a built-in "coarse grid" projection [**3**].

For high-order problems, especially with plate or shell finite element models in structural analysis, a two-level preconditioning technique for the FETI method has been introduced [**2**]. Computing the preconditioned gradient requires the solution of a coarse grid problem that is of the same kind as the original FETI problem, but is associated with a small subset of Lagrange multipliers enforcing continuity at cross-points. This preconditioner gives optimal convergence property for plate or shell finite element models [**5**].

This approach has been recently generalized to various local or partial continuity requirements in order to derive a general methodology for building second-level preconditioners [**1**].

For a sake of simplicity, the first method advocated for solving the coarse grid problems in distributed memory environment has been the same projected gradient as for the first-level FETI method [**6**] [**7**]. But with the increased complexity of the generalized 2-level FETI method, this approach leads to poor performance on machines with high performance compute nodes.

In the present paper this new preconditioning technique is reinterpreted in a simple algebraic form, in order to derive algorithms based on direct solution techniques to solve efficiently the coarse grid problems in distributed memory environment. Performance results for real-life applications are given.

2. Notations

In each subdomain, Ω_i, the local displacement field is solution of the linear elasticity equations with imposed forces on the interfaces with other subdomains:

$$K_i u_i = B_i^t \lambda + b_i \qquad (1)$$

1991 *Mathematics Subject Classification.* Primary 65N55; Secondary 65Y05.
Key words and phrases. Parallel Algorithms, Multi-Level Preconditioners.

where K_i is the stiffness matrix, u_i the displacement field, B_i a signed boolean matrix associated with the discrete trace operator, and λ the Lagrange multiplier, equal to the interaction forces between subdomains.

The continuity requirement along the interfaces is written as follows:

$$\sum_i B_i u_i = 0 \tag{2}$$

where the signed discrete trace matrices B_i are such that if subdomains Ω_i and Ω_j are connected by the interface Γ_{ij}, then restriction of equation (2) on Γ_{ij} is: $u_i - u_j = 0$.

If the boundary of subdomain Ω_i does not contain a part of the external boundary with prescribed displacements, the local problem has only Neumann boundary conditions and then the matrix K_i is positive semi-definite.

If K_i^+ is a pseudo-inverse of matrix K_i, and if columns of matrix R_i form a basis of the kernel of K_i (rigid body motions), equation (1) is equivalent to:

$$\left\{ \begin{array}{r} u_i = K_i^+(b_i + B_i^t\lambda) + R_i\alpha_i \\ R_i^t(b_i + B_i^t\lambda) = 0 \end{array} \right. \tag{3}$$

Introducing u_i given by equation (3) in the continuity condition (2) gives:

$$\sum_i B_i K_i^+ B_i^t \lambda + \sum_i B_i R_i \alpha_i = -\sum_i B_i K_i^+ b_i \tag{4}$$

With the constraint on λ set by the second equation of (3), the global interface problem can be written:

$$\begin{bmatrix} F & G \\ G^t & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ c \end{bmatrix} \tag{5}$$

With:

- $F = \sum B_i K_i^+ B_i^t$, dual Schur complement matrix,
- $G\alpha = \sum B_i R_i \alpha_i$, jump of rigid body motions defined by α_i in Ω_i,
- $(G^t\lambda)_i = R_i^t B_i^t \lambda$,
- $d = -\sum B_i K_i^+ b_i$,
- $c_i = -R_i^t b_i$.

In the following sections, we shall use the term "rigid body modes" for Lagrange multipliers in the image space of G.

3. Parallelization of the rigid body projection

3.1. Rigid body projection. Thanks to the fact that the number of admissibility constraints related to the second set of equations:

$$(G^t\lambda)_i = R_i^t B_i^t \lambda \tag{6}$$

the hybrid condensed system (5) can be solved in practice by a projected gradient algorithm. The projection associated with the rigid body modes can be explicitly computed:

$$P = I - G(G^tG)^{-1}G^t \tag{7}$$

The computation of the product by projection P requires products by G and G^t and the solution of systems with form:

$$(G^tG)\alpha = G^t g \tag{8}$$

The product by G^t can be performed independently in each subdomain, the product by G requires exchanging data through interfaces between neighboring subdomains. The product by $(G^tG)^{-1}$ requires the solution of a coarse grid problem associated with rigid body motions coefficients in each subdomain. This problem has the same kind of algebraic structure as a finite element problem whose elements are the subdomains and whose degrees of freedom in each element are the subdomain rigid body motions coefficients.

3.2. Forming and factorization of (G^tG). The (G^tG) matrix has a sparse block structure. If subscripts i and j are associated with subdomains Ω_i and Ω_j, the block $(G^tG)_{ij}$ representing entries in (G^tG) associated with influence between rigid body modes in Ω_i and Ω_j is equal to:

$$(G^tG)_{ij} = R_i^t B_i^t B_j R_j = (B_i R_i)^t (B_j R_j) \tag{9}$$

The columns of B_iR_i and B_jR_j are respectively the traces on interfaces of rigid body motions of subdomains Ω_i and Ω_j. The entries of these columns are simultaneously non zero only on interface Γ_{ij}. So, the computation of block $(G^tG)_{ij}$ just requires the values of the traces on Γ_{ij} of rigid body motions of subdomains Ω_i and Ω_j.

In order to minimize computation costs and memory requirements, the following algorithm can be implemented to compute and factorize the (G^tG) matrix.

1. Store in each subdomain Ω_i, the traces of rigid body motions on each interface Γ_{ij} with neighboring subdomain Ω_j. This requires the storage for each interface Γ_{ij} of a matrix $(B_iR_i)_j$ with number of rows equal to the number of degrees of freedom on interface Γ_{ij} and number of columns equal to the number of rigid body motions in Ω_i.
2. Exchange $(B_iR_i)_j$ matrices with neighboring subdomains. This means that subdomain Ω_i sends matrix $(B_iR_i)_j$ to subdomain Ω_j and receives from it matrix $(B_jR_j)_i$ whose number of rows is equal to the number of degrees of freedom on interface Γ_{ij} and number of columns equal to the number of rigid body motions in Ω_j.
3. In subdomain Ω_i, compute the following matrix-matrix products for each interfaceΓ_{ij}:

$$\begin{array}{lcl} (G^tG)_{ij} & = & (B_iR_i)_j^t (B_jR_j)_i \\ (G^tG)_{ii} & = & (G^tG)_{ii} + (B_iR_i)_j^t (B_iR_i)_j \end{array} \tag{10}$$

   Now, subdomain Ω_i has a part of the sparse block structure of matrix (G^tG).
4. Assemble the complete (G^tG) matrix via global data transfer operations (“GATHER”).
5. Factorize the complete (G^tG) matrix via Choleski factorization using optimal renumbering strategy. Note that numerical pivoting strategy may also be implemented to detect global rigid body motions in the case of a not clamped global problem. In this case, the global rigid boy motions form the kernel of the (G^tG) matrix.
6. Compute the rows of the $(G^tG)^{-1}$ matrix associated to the rigid body modes in the neighborhood of subdomain, including the subdomain itself and its neighbors (see next section).

3.3. Computation of rigid body modes projection. The projected gradient is given by:

$$Pg = g - G(G^tG)^{-1}G^tg \tag{11}$$

The computation of G^tg is purely local. In subdomain Ω_i, $(G^tg)_i = R_i^tB_i^tg$.

The main step of the projection is the product by $(G^tG)^{-1}$. As the global (G^tG) matrix has been factorized, the product by $(G^tG)^{-1}$ requires a forward-backward substitution. To perform it, the global G^tg vector must be assembled.

Once $\alpha = (G^tG)^{-1}G^tg$ has been computed a product by G must be performed to compute the projected gradient. As $G\alpha$ is defined as:

$$G\alpha = \sum B_iR_i\alpha_i \tag{12}$$

its restriction on interface Γ_{ij} is given by:

$$(G\alpha)_{ij} = (B_iR_i)_j\alpha_i + (B_jR_j)_i\alpha_j \tag{13}$$

This means that subdomain Ω_i can compute the projected gradient on its interfaces without any data transfer, provided that it has the values of α in all its neighboring subdomains. Furthermore, it does not need at all the values of α in the other subdomains.

Hence, the computation of the projection can be parallelized with minimal data transfer, provided that each subdomain computes the solution of the coarse problem:

$$(G^tG)\alpha = G^tg \tag{14}$$

for its neighborhood including the subdomain itself and its neighbors. Only the rows of the $(G^tG)^{-1}$ matrix associated to the neighborhood are required in the subdomain to do so. These rows form a matrix with number of rows equal to the number of rigid body motions in the neighborhood and number of columns equal to the total number of rigid body modes. Thanks to the symmetry of (G^tG), the rows of this matrix are equal to the columns of $(G^tG)^{-1}$ associated to the neighborhood. So, computing this matrix noted $(G^tG)^{-1}_{zone\_i}$ requires a forward-backward substitution with complete matrix (G^tG) for each row. The computation of this matrix represents the last step of the forming and factorization of (G^tG), as indicated in the previous section.

So, computing the projected gradient in parallel requires the following steps:

1. Compute $(G^tg)_i = R_i^tB_i^tg$ in each subdomain Ω_i.
2. Gather complete $\beta = G^tg$ in each subdomain via a global data transfer operation.
3. Compute components of $\alpha = (G^tG)^{-1}\,\beta$ in neighborhood of each subdomain Ω_i. This means compute the matrix-vector product:

$$(G^tG)^{-1}_{zone\_i}\,\beta \tag{15}$$

4. Compute $G\alpha$ in each subdomain Ω_i as:

$$(G\alpha)_{ij} = (B_iR_i)_j\alpha_i + (B_jR_j)_i\alpha_j \tag{16}$$

   for each interface Γ_{ij}.
5. Compute $Pg = g - G\alpha$ in each subdomain.

The only transfer this algorithm requires is the gathering of $\beta = G^tg$ in step 2.

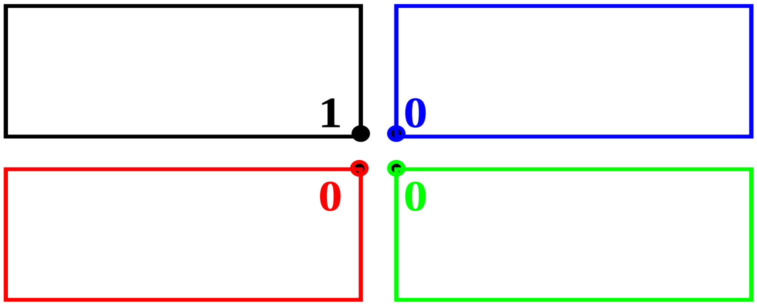

FIGURE 1. A "corner motion" for a scalar problem

4. The second level FETI preconditioner

4.1. Definition of corner modes. In this section, the second level FETI preconditioner is presented in the case of a coarse grid defined as the so-called "corner modes". The objective consists in constraining the Lagrange multiplier to generate local displacement fields that are continuous at interface cross-points. To get a practical formulation of this constraint, it can be observed that requiring the continuity of displacement fields at interface cross-points is equivalent to imposing their jump to be orthogonal to the jump of "corner motions" defined as displacement fields with unit value in one space direction at a node connected to a crosspoint as in Figure (1).

Note C_i the set of corner motions in subdomain Ω_i, then the Lagrange multiplier λ satisfies the continuity requirement of associated displacement fields at interface cross-points if the projected gradient satisfies:

$$(B_iC_i)^tPg = 0 \ \forall i \ \Leftrightarrow \ (\sum_i B_iC_i\gamma_i)^tPg = 0 \ \forall \gamma \tag{17}$$

4.2. Coarse grid space. Let us define "corner modes" from corner motions and the associated global operator C in the same way as rigid body modes and operator G are defined from rigid body motions:

$$C\gamma = \sum B_iC_i\gamma_i \tag{18}$$

The second-level FETI preconditioner consists in building the search direction vector w from the projection of the gradient Pg in order to satisfy the constraint of generating local displacement fields that are continuous at interface cross-points. As the jump of displacement fields created by w is equal to PFw, this constraint can be written in the following way:

$$C^tPFw = C^tP^tFw = 0 \tag{19}$$

In order to satisfy this constraint, the search direction vector w must be constructed from Pg corrected by a vector in the image space of C:

$$w = Pg + C\gamma \tag{20}$$

The search direction vector w must also satisfy the constraint of orthogonality to the traces of rigid body modes. This constraint can be written $Pw = w$. So w must have the following form:

$$w = Pg + C\gamma + G\beta \text{ with } Pw = w \tag{21}$$

So, the coarse grid space associated with the second-level FETI preconditioner must contain both image spaces of C and G. This means that it is the image space of the matrix noted $[CG]$ whose first columns are the columns of C and last columns the columns of G.

4.3. Second-level FETI problem . By definition of the rigid body modes projection P, PFw satisfies:

$$G^tPFw = G^tP^tFw = 0 \tag{22}$$

From the definition of the coarse grid space (21), the search direction vector must have the following form:

$$w = Pw = Pg + P(C\gamma + G\beta) \tag{23}$$

The formulation of constraints in equations (19) and (22), entails that γ and β satisfy the following problem:

$$\begin{array}{rcl} C^tP^tFP(C\gamma + G\beta) & = & -C^tP^tFPg \\ G^tP^tFP(C\gamma + G\beta) & = & -G^tP^tFPg \end{array} \tag{24}$$

These equations can be rewritten as:

$$[CG]^t \; P^tFP \; [CG] \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -[CG]^t \; P^tFPg \tag{25}$$

This system is precisely a coarse FETI problem, posed in the subspace of Lagrange multipliers defined as the image space of $[CG]$. With this coarse grid preconditioner, the solution algorithm appears clearly as a two-level FETI method: at each iteration of projected conjugate gradient at the fine level, an additional preconditioning problem of the same type has to be solved at the coarse grid level.

4.4. Rigid body projection for coarse grid vectors. The rigid body projection takes a simple form for vectors belonging to the coarse grid space. In general, for any Lagrange multiplier μ, the rigid body projection is defined by:

$$P\mu = \mu + G\delta \text{ with } G^t(\mu + G\delta) = 0 \tag{26}$$

In the same way, for a vector in the coarse grid space, the projection can be written:

$$P(C\gamma + G\beta) = C\gamma + G\beta + G\delta = C\gamma + G(\beta + \delta) \tag{27}$$

correction δ being defined by the constraint:

$$G^t(C\gamma + G(\beta + \delta)) = 0 \tag{28}$$

So, $(\beta + \delta)$ satisfies:

$$(G^tG)(\beta + \delta)) = -G^tC\gamma \tag{29}$$

Hence, the projection of a vector in the coarse grid space is given by the following equation:

$$P(C\gamma + G\beta) = C\gamma - G(G^tG)^{-1}G^tC\gamma \tag{30}$$

This equation illustrates the fact that the effective degrees of freedom of the second-level FETI preconditioner are the corner modes coefficients. The rigid body modes have been added to the coarse grid space just for allowing coarse grid vectors to satisfy the rigid body constraint. For any set of corner modes coefficients γ, the only coarse grid vector satisfying the rigid body constraint is given by equation (30).

Let us note R_{GC} the matrix defined by:

$$R_{\mathrm{GC}} = -(G^tG)^{-1}G^tC \tag{31}$$

$R_{\mathrm{GC}}\gamma$ represents the coefficients of the rigid body modes correction to apply on the corner mode generated by coefficients γ for satisfying the rigid body constraint.

Note P_{coarse} the following matrix:

$$P_{coarse} \ = \ \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} [\ I \ 0\ ] \tag{32}$$

Equation (30) can be rewritten:

$$P\ [CG] \ \begin{bmatrix} \gamma \\ \beta \end{bmatrix} \ = \ [CG]\ P_{coarse} \begin{bmatrix} \gamma \\ \beta \end{bmatrix} \tag{33}$$

So, actually P_{coarse} is the rigid body projection for coarse grid vectors in the natural coarse grid basis defined by columns of $[CG]$.

4.5. Projected coarse grid problem. The second-level FETI problem (25) can be rewritten as:

$$(\ P\ [CG]\ )^t\ F\ (\ P\ [CG]\ ) \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -(\ F\ P\ [CG]\ )^t Pg \tag{34}$$

Thanks to the definition of the rigid body projection for coarse grid vectors given by equation (33), a new formulation of the second-level FETI problem can be derived from equation (34):

$$P^t_{coarse}\ (\ [CG]^t\ F\ [CG]\ )\ P_{coarse} \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -P^t_{coarse}\ (\ F\ [CG]\ )^t\ Pg \tag{35}$$

Note F_{coarse} the projection of the FETI operator in the coarse grid space:

$$F_{coarse} \ = \ [CG]^t\ F\ [CG] \tag{36}$$

From the definition of P_{coarse} in equation (32), the second-level FETI problem can be rewritten:

$$\begin{bmatrix} I \\ 0 \end{bmatrix} \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t F_{coarse} \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} [\ I\ 0\ ] \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = - \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t (\ F\ [CG]\ )^t\ Pg \tag{37}$$

Thanks to the definition of the rigid body projection for coarse grid vectors,the only degrees of freedom of equation (37) are coefficients of γ. Eliminating null blocks in this equation finally leads to the projected coarse grid problem that defines γ:

$$\begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t F_{coarse} \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} \gamma = - \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t (\ F\ [CG]\ )^t\ Pg \tag{38}$$

From equation (23) and (30), the search direction vector satisfying both corner modes constraint and rigid body modes constraint is given by:

$$w = Pw = Pg + P(C\gamma + G\beta) = Pg + \ [CG] \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} \gamma \tag{39}$$

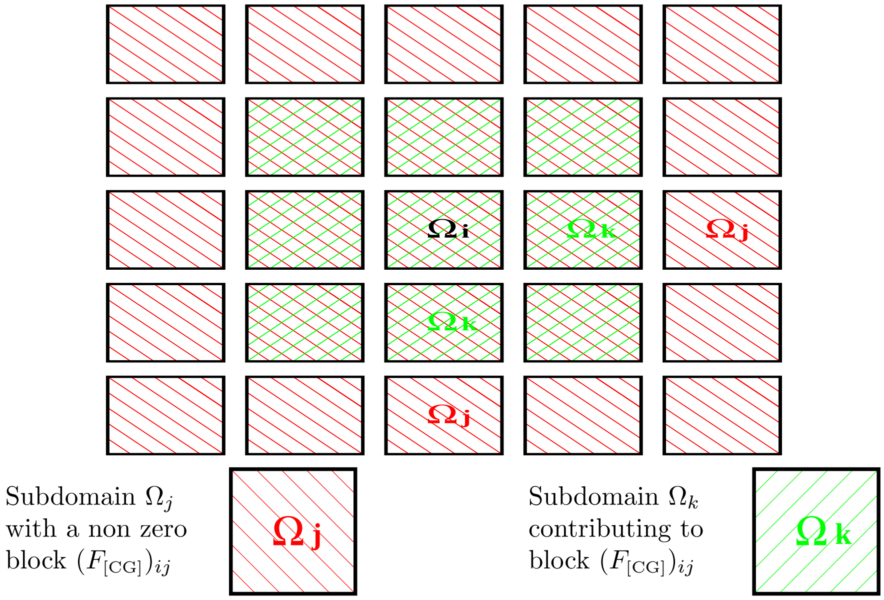
Ω_i
Ω_k
Ω_j
Ω_k
Ω_j
Subdomain Ω_j with a non zero block $(F_{[\mathrm{CG}]})_{ij}$
Ω_j
Subdomain Ω_k contributing to block $(F_{[\mathrm{CG}]})_{ij}$
Ω_k

FIGURE 2. Contribution of subdomains to blocks of F_{coarse}

5. Parallel computation of the projected coarse grid FETI matrix

5.1. Forming of F_{coarse}. The projection of FETI operator on the coarse grid space is defined in equation (36). The coarse grid matrix $[CG]$ and the first-level FETI matrix F can be written:

$$\begin{array}{lcl} [CG] & = & \sum_j B_j[C_jR_j] \\ F & = & \sum_k B_kK_k^+B_k^t \end{array} \tag{40}$$

where $[C_jR_j]$ is the matrix whose columns are the corner and rigid body motions of subdomain Ω_j.

From this equation, it can be derived that F_{coarse} has a sparse block structure, such that the block of interaction between coarse grid degrees of freedom in subdomains Ω_i and Ω_j is defined as:

$$(F_{coarse})_{ij} \;=\; (\; B_i[C_iR_i]\;)^t\;(\;\sum_k B_kK_k^+B_k^t\;)\;(\;\sum_j B_j[C_jR_j]\;) \tag{41}$$

As a consequence of the fact that the trace operator B_n is non zero only on interfaces of subdomain Ω_n, a block of form:

$$(B_i[C_iR_i])^t\;(\;B_kK_k^+B_k^t\;)\;B_j[C_jR_j] \tag{42}$$

is non zero only if subdomain Ω_k is neighbor of both subdomains Ω_i and Ω_j. Such a block is the contribution of subdomain Ω_k to the block matrix $(F_{coarse})_{ij}$. This means that the contribution of subdomain Ω_k to the coarse-grid FETI matrix F_{coarse} is a dense square matrix with dimension equal to the sum of numbers of corner and rigid body motions in its neighborhood, including the subdomain itself and all its neighbors. Figure (2) show the dependency between subdomains for the computation of $(F_{coarse})_{ij}$.

5.2. Local contribution to F_{coarse}. From equation (42), it appears that subdomain Ω_k can compute its contribution to the F_{coarse} matrix, if it has all the traces of corner and rigid body motions in its neighborhood. So, forming the contribution of a subdomain to the F_{coarse} matrix can be organized according to the algorithm described below.

1. Store in each subdomain Ω_k, the traces of corner motions on each interface Γ_{kj} with neighboring subdomain Ω_j. This requires the storage for each interface Γ_{kj} of a matrix $(B_kC_k)_j$ with number of rows equal to the number of degrees of freedom on interface Γ_{kj} and number of columns equal to the number of corner motions in Ω_k.
2. Exchange $(B_kC_k)_j$ matrices with neighboring subdomains. This means that subdomain Ω_k sends matrix $(B_kC_k)_j$ to subdomain Ω_j and receives from it matrix $(B_jC_j)_k$ whose number of rows is equal to the number of degrees of freedom on interface Γ_{kj} and number of columns equal to the number of corner motions in Ω_j.
3. In subdomain Ω_k, perform a forward-backward substitution for each local corner and rigid body motion, and for each corner and rigid body motion of neighboring subdomains, in order to compute the following matrices:

$$K_k^+ B_k^t B_j C_j \text{ and } K_k^+ B_k^t B_j R_j \tag{43}$$

   for $j = k$ or j such that Ω_j is a neighbor of Ω_k.

4. Store the traces of resulting vectors interface by interface. On each interface Γ_{ki}, these traces form two sets of matrices $(B_kK_k^+B_k^tB_jC_j)_i$ and $(B_kK_k^+B_k^tB_jR_j)_i$ whose number of rows is equal to the number of degrees of freedom on interface Γ_{ki} and number of columns respectively equal to the number of corner or rigid body motions in Ω_j
5. In subdomain Ω_k, for each interfaceΓ_{ki}, compute the following matrix-matrix products:

$$\begin{array}{ll} (\ B_iC_i\ )_k^t\ (\ B_kK_k^+B_k^t\ \ B_jC_j\ )_i & (\ B_iC_i\ )_k^t\ (\ B_kK_k^+B_k^t\ \ B_jR_j\ )_i \\ (\ B_iR_i\ )_k^t\ (\ B_kK_k^+B_k^t\ \ B_jC_j\ )_i & (\ B_iR_i\ )_k^t\ (\ B_kK_k^+B_k^t\ \ B_jR_j\ )_i \end{array} \tag{44}$$

   In the same way, the following blocks must be computed and added to the contribution of other interfaces:

$$\begin{array}{ll} (\ B_kC_k\ )_i^t\ (\ B_kK_k^+B_k^t\ \ B_jC_j\ )_i & (\ B_iC_i\ )_k^t\ (\ B_kK_k^+B_k^t\ \ B_jR_j\ )_i \\ (\ B_kR_k\ )_i^t\ (\ B_kK_k^+B_k^t\ \ B_jC_j\ )_i & (\ B_iR_i\ )_k^t\ (\ B_kK_k^+B_k^t\ \ B_jR_j\ )_i \end{array} \tag{45}$$

   Now, subdomain Ω_k has its contribution to the $(F_{coarse})_{ij}$ block, for any pair of its neighboring subdomains Ω_i and Ω_j.

5.3. Contribution of subdomain to the projected coarse grid FETI problem. If the restriction of coarse grid on neighborhood of subdomain Ω_k, including Ω_k itself, is written $I^{(k)}$, then the projected coarse grid problem of equation (38) is constructed from the local contributions to the coarse grid FETI matrix as follows:

$$\begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t F_{coarse} \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} = \sum_k \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t (I^{(k)})^t\ F_{coarse}^{(k)}\ (I^{(k)}) \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} \tag{46}$$

where $F_{coarse}^{(k)}$ is the contribution of Ω_k to the coarse grid FETI matrix constructed in the way presented in section 5.2.

So, once $F^{(k)}_{coarse}$ has been formed, the computation of the contribution of subdomain Ω_k to the projected coarse grid FETI matrix only requires the rows of R_{GC} associated with rigid body modes in the neighborhood of Ω_k.

In the same way, once the projected coarse grid problem of equation (38) is solved, the computation of the vector w as described in equation (39) can be performed without any data transfer in each subdomain, provided that the complete solution γ and the rows of R_{GC} associated with rigid body modes in the neighborhood are known.

5.4. Forming of the rigid body projection for coarse grid vectors. From the previous section, it appears that each subdomain must construct the rows of the rigid body projection associated with the rigid body modes of its neighborhood. In subdomain Ω_k, these rows form the following matrix:

$$I^{(k)} \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} = I^{(k)} \begin{bmatrix} I \\ -(G^tG)^{-1}G^tC \end{bmatrix} \tag{47}$$

The G^tC matrix has exactly the same kind of sparse block structure as G^tG and can be formed in parallel using the same methodology. The (G^tG) matrix has already been formed and factorized in each subdomain. The only problem to compute the rows of R_{GC} associated with the rigid body modes in the neighborhood of subdomain Ω_k lies in the fact that all the entries corresponding to corner motions in ALL subdomains must be computed.

The following algorithm can be implemented.

1. In subdomain Ω_k, compute the following matrix-matrix products for each interfaceΓ_{kj}:

$$\begin{array}{rcl} (G^tC)_{jk} & = & (B_jR_j)^t_k(B_kC_k)_j \\ (G^tC)_{kk} & = & (G^tC)_{kk} - (B_jR_j)^t_k(B_kC_k)_j \end{array} \tag{48}$$

   The result is scattered in a matrix whose number of columns is equal to the number of corner motions in subdomain Ω_k and number of rows equal to the total number of rigid body modes. This matrix forms the subset of columns of G^tC associated with corner modes of subdomain Ω_k.

2. Compute $-(G^tG)^{-1}G^tC$ for all columns of G^tC associated with corner modes of subdomain Ω_k. This requires a number of forward-backward substitutions, using the COMPLETE factorization of (G^tG), equal to the number of corner motions in subdomain Ω_k. Now, subdomain Ω_k has all the columns of

$$R_{\mathrm{GC}} = -(G^tG)^{-1}G^tC \tag{49}$$

   associated with its own corner modes.

3. Exchange the columns of R_{GC} associated with corner modes of subdomain Ω_k with neighboring subdomains. Now, subdomain Ω_k has all the columns of R_{GC} associated with all corner modes in its neighborhood, itself included.
4. Subdomain Ω_k has all the COLUMNS of R_{GC} associated with its own corner modes. To get the ROWS of R_{GC} associated with its rigid body modes, a global matrix transposition through data exchange between all processors must be performed.
5. A last data transfer operation with neighboring subdomains must be performed in order to get rows of R_{GC} associated with rigid body modes in the whole neighborhood of each subdomain.

6. Build the matrix:

$$I^{(k)} \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} \tag{50}$$

The rows of R_{GC} associated with rigid body modes in the whole neighborhood of each subdomain form the lower part of this matrix, the upper part is just the boolean restriction to the subset of corner modes in the whole neighborhood of each subdomain.

5.5. Forming and factorization of the projected coarse grid FETI matrix . The forming and factorization of the complete projected coarse grid FETI matrix can be performed in 4 steps.

1. In each subdomain Ω_k, compute the matrix-matrix product:

$$\begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t (I^{(k)})^t \, F^{(k)}_{coarse} \, (I^{(k)}) \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} \tag{51}$$

   The result is a square matrix of dimension equal to the total number of corner modes. It is the contribution of subdomain Ω_k to the projected coarse grid FETI matrix.
2. Assemble the projected coarse grid FETI matrix through a global data transfer. This is a reduction with add operation.
3. Build the pseudo-inverse of the projected coarse grid FETI matrix via Choleski factorization with numerical pivoting. This matrix is not full rank when there is some redundancy of the corner modes.
4. Compute the rows of the pseudo-inverse of the projected coarse grid FETI associated with the corner modes in the neighborhood of subdomain, including the subdomain itself and its neighbors.

6. Parallel solution of the second-level FETI problem

6.1. Right-hand side of the projected coarse grid problem. From equation (38) it can be derived that the first step of the computation of the right-hand side of the projected coarse grid problem requires the computation of:

$$(\ F\ [CG]\ )^t\ Pg\ =\ (\ (\ \sum_k B_k K_k^+ B_k^t\ )\ (\ \sum_j B_j [C_j R_j]\ )\ )^t\ Pg \tag{52}$$

All the data to compute the contribution of subdomain Ω_k to this product are already present, thanks to the computation of the local contribution to F_{coarse}, as explained in section 5.2. Also, the columns of the rigid body projection for the coarse grid problem have been computed locally.

Hence, the parallelization of the computation of the right-hand side of the projected coarse grid problem can be performed as follows:

1. In subdomain Ω_k, for each interfaceΓ_{ki}, compute the following matrix-vector products:

$$\begin{array}{c} (\ B_k K_k^+ B_k^t\ B_j C_j\ )_i^t\ (Pg)_{ki} \\ (\ B_k K_k^+ B_k^t\ B_j R_j\ )_i^t\ (Pg)_{ki} \end{array} \tag{53}$$

   for all subdomains Ω_j in the neighborhood of Ω_k, itself included.
   Scatter the resulting vectors in a vector of dimension equal to the sum of corner and rigid body motions in all subdomains in the locations associated with the corner and rigid body modes of the neighborhood of Ω_k. If $j = k$,

the contributions of all interfaces must be added. This vector represents the contribution of subdomain Ω_k to $(F [CG])^t Pg$.

2. Assemble $(F [CG])^t Pg$ via global data exchange. This is a reduction with add operation.
3. Compute in each subdomain Ω_k the part of the product:

$$-\begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t (F [CG])^t Pg \tag{54}$$

associated with corner modes of Ω_k. Only the columns of the matrix R_{GC} associated with corner modes of Ω_k and that have been computed during the forming of the rigid body projection for the coarse grid, are needed.

This procedure gives the restriction to each subdomain of the right-hand side of the projected coarse grid problem.

4. Gather complete right-hand side of the projected coarse grid problem in each subdomain via a global data transfer operation.

6.2. Solution of the projected coarse grid FETI problem. Once the complete right-hand side has been gathered in each subdomain, the computation of the solution γ of the projected coarse grid problem (38) in subdomain Ω_i just requires a product by the matrix:

$$\left(\begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}^t F_{coarse} \begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix} \right)^{-1}_{zone\_i} \tag{55}$$

using the rows of the pseudo-inverse of the projected coarse grid FETI associated with the corner modes in the neighborhood of subdomain computed as explained in section 5.5.

6.3. Computation of the search direction vector. From equation (39) it can be observed that the search direction vector w can be computed locally in each subdomain, provided that entries of the solution of the projected coarse grid FETI problem γ and of vector $\beta = R_{\mathrm{GC}}\gamma$ associated to the subdomain and its neighbors are known.

The procedure described in the previous section has given the entries of the solution of the projected coarse grid FETI problem γ associated to the subdomain and its neighbors. The computation of the search direction vector w can be completed as follows.

1. In each subdomain, compute the contribution to β by computing the matrix-vector product:

$$\beta = R_{\mathrm{GC}}\gamma \tag{56}$$

for columns of R_{GC} associated with corner modes of subdomain.

2. Assemble β through a global data transfer. This is a reduction with add operation. Extract the entries of β associated to the subdomain and its neighbors.
3. In subdomain Ω_i, for each interface Γ_{ij}, compute the following matrix-vector products:

$$w_{ij} = Pg_{ij} + (B_iC_i)_j\gamma_i + (B_jC_j)_i\gamma_j + (B_iR_i)_j\beta_i + (B_jR_j)_i\beta_j \tag{57}$$

This step computes without any data transfer the restriction to subdomain of:

$$w = Pg + C\gamma + G\beta = Pg + [CG]\begin{bmatrix} I \\ R_{\mathrm{GC}} \end{bmatrix}\gamma \tag{58}$$

6.4. Computation of the starting λ. For a given λ^0 satisfying the rigid body constraint:

$$(G^t\lambda^0)_i = -R_i^t b_i \tag{59}$$

in each subdomain Ω_i, note g^0 the gradient $F\lambda^0 - d$. The corner mode correction of λ^0, w, must be of form:

$$w = P(C\gamma + G\beta) \tag{60}$$

The corrected initial λ, is $\lambda^0 + w$, and the associated corrected gradient is $g^0 + Fw$. The correction w must be such that the projected corrected gradient satisfy:

$$\begin{array}{rcl} C^tP(g^0 + Fw) & = & 0 \\ G^tP(g^0 + Fw) & = & 0 \end{array} \tag{61}$$

From these equations, the same development as in section 4.3 leads to the second level FETI problem:

$$[CG]^tP^tFP[CG]\begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -[CG]^tPg^0 \tag{62}$$

This problem is similar to the one of equation (25), except for the right-hand-side. As, by definition of the rigid body projection P, G^tPg^0 is equal to 0, the right-hand side can be even simplified:

$$[CG]^tP^tFP[CG]\begin{bmatrix} \gamma \\ \beta \end{bmatrix} = -C^tPg^0 \tag{63}$$

This problem can be solved in the same way as for the search direction vector, except for the right-hand side that is simpler and can be computed exactly in the same way as the right-hand side of a rigid body projection, using corner motions C_i instead of rigid body motions G_i.

7. Application

7.1. "Interface averaging" modes. The second-level preconditioner has been presented in the previous sections for the case of a coarse grid associated with "corner modes". This preconditioner has been demonstrated to be very efficient for dealing with the singularity at cross-points for high order problems like plate and shell finite element problems. In this case, the coarse grid is built for imposing a local continuity requirement. But the same approach can be used to enforce more global continuity requirements, like for instance continuity of mean value of each component of the displacement field on each interface between two subdomains.

The corresponding modes, called interface averaging modes are simply the jumps of local constant motion on a single interface as in Figure 3. Once again, building the coarse grid space as the set of jumps of local special motions makes it simpler to define and gives automatically admissible Lagrange multipliers. This approach also allows to define efficient coarse grid preconditioner in the case where local Neumann problems are well posed, especially for time-dependent problems.

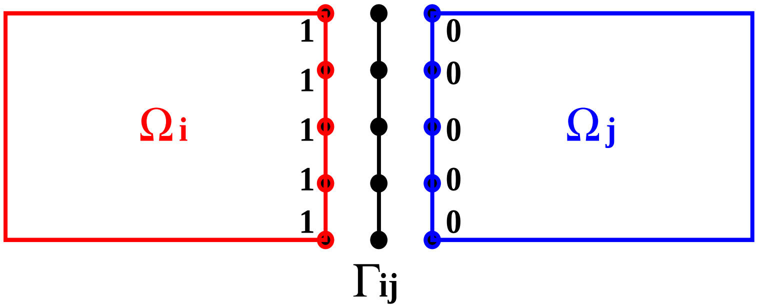

FIGURE 3. Generation of a single interface averaging mode for a scalar problem

TABLE 1. Solution time on IBM-SP2 with corner and interface averaging modes

Numb. of	1-level FETI		2-level FETI corner modes		2-level FETI corner + averaging	
Domains	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
16	52	21	26	11	19	8.4
32	91	14	28	5	20	3.5

In such a case, there is no rigid body projection to play the role of a first-level coarse grid preconditioner, and the standard FETI method is not numerically scalable.

7.2. Parallel performance. To illustrate the efficiency of the second-level FETI preconditioner parallelized by the solution technique presented in this paper, a small shell problem, with 25600 nodes and 6 degrees of freedom per node, has been solved on an IBM-SP2 system with either 16 or 32 processors and one domain per processor.

The reason why a small problem has been chosen is the following: the objective is to demonstrate that with the parallelization technique developed in this paper, the 2-level FETI method is not only numerically scalable, but its implementation on distributed memory systems actually gives scalable performances. The main drawback with the coarse grid preconditioners is the fact that their implementation on distributed memory machines can be very inefficient because the number of data transfers they require is high and their granularity is very low. Of course, for a given target architecture, if the local size of the problem is large enough, the cost for the forward-backward substitution may remain dominant. But to be really scalable, a method must be efficient even in the case where the size of the local problems is not very large.

The stopping criterion is the same in all cases presented in this paper and is related to the global residual:

$$||Ku - b||/||b|| < 10^{-6} \qquad (64)$$

Also, in all cases presented in this paper, the local optimal Dirichlet preconditioner is used. Table 1 gives a comparison of the elapsed parallel times for the iterations of 1-level and 2-level FETI methods with various coarse grid preconditioners. It

Table 2. Solution time on IBM-SP2 with corner and interface averaging modes

1-level FETI		2-level FETI corner modes		2-level FETI corner + averaging	
Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
163	106	25	11	16	7.8

Table 3. Number of modes and times for building the second-level preconditioner

1-level FETI		2-level FETI corner modes		2-level FETI corner + averaging	
Numb. of Rigid Body Modes	Dirichlet + Neumann(s)	Numb. of Modes	Set-up Time(s)	Numb. of Modes	Set-up Time(s)
304	17	588	20	924	40

demonstrates clearly that the cost for the parallel solution of the coarse grid problems is small enough to ensure that the solution time decreases in the same proportion as the number of iterations. Enforcing the corner continuity is enough to make the method scalable for this kind of shell problem. Nevertheless, the averaging modes give a significant decrease of the number of iterations. Speed-ups with the 2-level FETI method are even super-linear, thanks to the decrease of the bandwidth of local matrices with larger number of subdomains.

7.3. Constructing cost for the coarse grid preconditioner. In the previous section, only the timings for the iterations were given. But the cost for assembling, factorizing and inverting the second-level FETI operator can be far from negligible in comparison with the time for the initial factorization of the local Dirichlet and Neumann problems matrices.

In order to illustrate this point, a larger shell model problem with 100000 nodes and 600000 degrees of freedom has been decompose in 64 subdomains. With such a large number of subdomains, the global number of coarse grid modes with both corner and averaging modes can be nearly one thousand. It would clearly not make sense to use such a large number of coarse grid modes for solving a small problem with only a few thousands degrees of freedom. Nevertheless, with 64 subdomains, the number of nodes per subdomain is less than 2000, hence the time for factorizing the local Dirichlet and Neumann matrices is rather small.

Table 2 features the number of iterations and the parallel times for the iterations of 1-level and 2-level FETI methods with various coarse grid preconditioners. Table 3 shows the total number of rigid body modes and of corner and corner + averaging modes, and it gives a comparison of times spent on each processor for factorizing the local Dirichlet and Neumann matrices and for assembling and factorizing the second-level FETI preconditioner.

In the case of the largest coarse grid space, this table shows that the time for forming the second-level preconditioner can be more than two times larger than the factorization time of local Dirichlet and Neumann matrices. Hence, this time may be considered too high, especially in comparison with the time for the iterations shown in Table 2.

Nevertheless these results are quite good. First, as discussed above, the global size of this problem is not that large for such a number of subdomains. The time for local factorizations and forward-backward substitutions increases faster with the number of nodes per subdomain than the time for building the second-level FETI preconditioner. So, for larger problems, the comparison will be more in favor of the 2-level method.

Secondly, this time to build the second-level FETI preconditioner is payed only once in the case of multiple right-hand-side, and the overall efficiency will be even better in such a case.

Thirdly, for the timings presented here, the factorization of the second-level FETI matrix has been computed in sequential. It would be possible to use a parallel skyline method to perform this factorization that represents nearly half the time spent in the construction of the second-level FETI preconditioner. Even a very low efficiency would be enough to make the factorization time itself negligible.

8. Conclusion

Thanks to the algebraic interpretation of the 2-level FETI method presented here a parallel implementation methodology has been designed. It allows using global direct solvers for the second-level FETI preconditioner but keeps working with a simple description of the interfaces at subdomain level.

The actual efficiency obtained with this approach makes feasible the enrichment of the coarse grid spaces used in the preconditioner, making the overall method faster and more robust.

References

1. F. Risler C. Farhat, P. C. Chen and F.-X. Roux, *A simple and unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers*, Comput. Meths. Appl. Mech. Engrg. (in press).
2. C. Farhat and J. Mandel, *The two-level FETI method for static and dynamic plate problems-part1: an optimal iterative solver for biharmonic systems*, Comput. Meths. Appl. Mech. Engrg. (in press).
3. C. Farhat, J. Mandel, and F.-X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Meths. Appl. Mech. Engrg. **115** (1994), 367–388.
4. C. Farhat and F.-X. Roux, Implicit parallel processing in structural mechanics (J. Tinsley Oden, ed.), Computational Mechanics Advances, vol. 2, Nort-Holland, 1994, pp. 1–124.
5. J. Mandel, R. Tezaur, and C. Farhat, *An optimal Lagrange multiplier based domain decomposition method for plate bending problems*, SIAM J. Sc. Stat. Comput. (in press).
6. F.-X. Roux, *Parallel implementation of a domain decomposition method for non-linear elasticity problems*, Domain-Based Parallelism and Problem decomposition Methods in Computational Science and Engineering (Philadelphia) (Youcef Saad David E. Keyes and Donald G. Truhlar, eds.), SIAM, 1995, pp. 161–176.
7. F.-X. Roux and C. Farhat, *Parallel implementation of the two-level FETI method*, Domain Decomposition Methods for Partial Differential Equations, 10th International Conference, Bergen, Norway, 1996 (M. Espedal P. Bjorstad and D. Keyes, eds.), Wiley and sons.

High Performance Computing Dpt, ONERA, BP72, F92322 Chatillon CEDEX, France
E-mail address: roux@onera.fr

Center for Space Structures, University of Colorado at Boulder, Boulder CO 80309-0526
E-mail address: charbel@alexandra.colorado.edu