Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

19. A FETI-DP Corner Selection Algorithm for three-dimensional problems

M. Lesoinne[1]

1. Introduction. The FETI-DP algorithm is a numerically scalable iterative domain decomposition method for static and dynamic problems. It was first derived as an alternative to the two-level FETI method for fourth order problems [1] and later extended to three dimensional second order problems [5, 2]. Later, several authors have showed that FETI-DP is scalable for scalar and mechanical problems [6] even in the presence of heterogeneities [4].

As it is derived from the two-level FETI method for fourth order problems, the choice of corner in such problems has to follow the same rules [3], however, for second order, three dimensional problems, the FETI-DP implementations remain flexible on the choice of corners. However a few constraints have to be placed on their choices, so that the resulting subdomain matrices and the resulting coarse problem is non-singular.

This article describes a robust algorithm for the selection of corners for three-dimensional problems that guarantees that none of the matrices involved in the FETI operator will be singular.

2. The Dual-Primal FETI Method. Let Ω be partitioned into a set of N_s, non-overlapping subdomains (or substructures) Ω^s. Select a set of points called corner points on which the degrees of freedom will remain primal variable. The mechanical interpretation of this particular method of mesh splitting can be viewed as making incisions into the mesh but leaving the corner points attached. This is analogous to the "tearing" stage of FETI. The "interconnecting" stage occurs only on the subdomain interfaces which now excludes the corner points (see Figure 2.1). By splitting, u^s into two sub-vectors such that:

$$u = \begin{bmatrix} u_r \\ u_c \end{bmatrix} = \begin{bmatrix} u_r^1 \\ \vdots \\ u_r^{N_s} \\ u_c \end{bmatrix} \tag{2.1}$$

where u_r^s is the remaining subdomain solution vector and u_c is a global/primal solution vector over all defined corner degrees of freedom. The solution at the corner points is continuous by definition when the solution vector is constructed in this manner. Using this notation, we can split the subdomain stiffness matrix into:

$$K^s = \begin{bmatrix} K_{rr}^s & K_{rc}^s \\ K_{rc}^{s^T} & K_{cc}^s \end{bmatrix} \tag{2.2}$$

[1]Assistant Professor, Department of Aerospace Engineering and Sciences and Center for Aerospace Structures University of Colorado at Boulder Boulder, CO 80309-0429, U.S.A. Email: Michel@Colorado.EDU

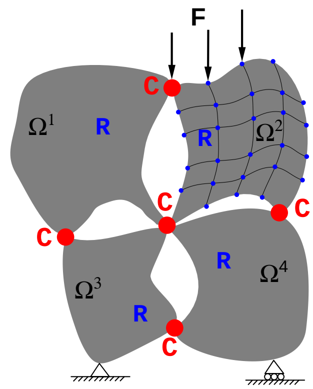

Figure 2.1: Dual-Primal Mesh Partitions

Then the FETI-DP equilibrium equations can be written using the following matrix partitioning where the subscripts c and r denote the corner and the remainder degrees of freedom.

$$\begin{bmatrix} K_{rr}^1 & \dots & 0 & K_{rc}^1 B_c^1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & K_{rr}^{N_s} & K_{rc}^{N_s} B_c^{N_s} \\ {B_c^1}^T {K_{rc}^1}^T & \dots & {B_c^{N_s}}^T {K_{rc}^{N_s}}^T & \sum_{s=1}^{N_s} {B_c^s}^T K_{cc}^s B_c^s \end{bmatrix} \begin{bmatrix} u_r^1 \\ \vdots \\ u_r^{N_s} \\ u_c \end{bmatrix} = \begin{bmatrix} f_r^1 - {B_r^1}^T \lambda \\ \vdots \\ f_r^{N_s} - {B_r^{N_s}}^T \lambda \\ \sum_{s=1}^{N_s} {B_c^s}^T f_c^s \end{bmatrix} \tag{2.3}$$

While the compatibility equations of interface displacements take the form:

$$\sum_{s=1}^{N_s} B_r^s u_r^s = 0 \tag{2.4}$$

In the preceeding, the corner stiffness matrix, $K_{cc} = \sum_{s=1}^{N_s} {B_c^s}^T K_{cc}^s B_c^s$ is a global stiffness quantity, B_c^s maps the local corner equation numbering to global corner equation numbering, f_r^s is the external force applied on the r degrees of freedom, ${B_r^s}^T$ is a boolean matrix that extracts the interface of a subdomain, and λ are the Lagrange multipliers.

Let K_{rr} denote the block diagonal subdomain stiffness matrix restricted to the remaining, r, points, K_{rc} the block column vector of r-c coupling stiffness matrices, f_r the block column vector of subdomain force vectors, K_{cc} the global corner stiffness matrix and using the "rc" notation, we can rewrite the equilibrium compatibility equations in the more compact form:

$$\begin{bmatrix} K_{rr} & K_{rc} & B_r^T \\ K_{rc}^T & K_{cc} & 0 \\ B_r & 0 & 0 \end{bmatrix} \begin{bmatrix} u_r \\ u_c \\ \lambda \end{bmatrix} = \begin{bmatrix} f_r \\ f_c \\ 0 \end{bmatrix} \tag{2.5}$$

In this formulation, the FETI-DP operator is a schur-complement obtained by eliminating the u_r and u_c degrees of freedom. The elimination of the u_r degrees of freedom is a subdomain per subdomain operation, while the elimination of the u_c degrees of freedom is a global operation that provides the FETI-DP operator with a coarse problem, coupling all the subdomains together.

Though this approach is scalable for two-dimensional problems and for plates and shells, it was shown that for second order three-dimensional problems, an augmented coarse problem is necessary.

The augmented FETI-DP system is obtained by adding new coarse degrees of freedom in the form of new Lagrange multipliers μ that are used to guarantee that at each iteration, the residual is orthogonal to a subspace Q:

$$Q^T B u_r = 0 \tag{2.6}$$

Thus leading to the system of equations:

$$\left(\begin{array}{c|ccc|c} K_{rr} & K_{rc} & B^t Q & & B^t \\ \hline K_{cr} & K_{cc} & 0 & & 0 \\ Q^t B & 0 & 0 & & 0 \\ \hline B & 0 & 0 & & 0 \end{array}\right) \left(\begin{array}{c} u_r \\ \hline u_c \\ \mu \\ \hline \lambda \end{array}\right) = \left(\begin{array}{c} f_r \\ \hline f_c \\ 0 \\ \hline 0 \end{array}\right) \tag{2.7}$$

In this set of equations, the first line is the set of subdomain by subdomain equations while the second and third lines represent the coarse problem which is global. By doing the Schur complement of the u_r equations, we obtain the coarse matrix:

$$\widetilde{K}_{cc} = \begin{pmatrix} K_{cc} - K_{cr} K_{rr}^{-1} K_{rc} & -K_{cr} K_{rr}^{-1} B^t Q \\ -Q^t B K_{rr}^{-1} K_{rc} & -Q^t B K_{rr}^{-1} B^t Q \end{pmatrix} \tag{2.8}$$

3. Preliminary Observations. There are two essential conditions that the corner selection should satisfy:

1. Each subdomain stiffness matrix should be non singular.
2. The resulting coarse problem matrix should be non singular.

Additionally, as they do not contribute significantly to the convergence rate, keeping the number of corner nodes low reduces the overall cost of the computation and improves its scalability.

3.1. Non-Singular $K_{rr}^{(s)}$. The non singularity of each subdomain $K_{rr}^{(s)}$ can be guaranteed simply by making sure that every subdomain has either 3 non-colinear corner nodes in 3 dimensions or 2 non-coincidental corner nodes in 2 dimensions.

3.2. Non-Singular $\widetilde{K}_{cc}$ and Pivoting. As presented here, the FETI-DP method only requires that $\widetilde{K}_{cc}$ be non singular for the corner degrees of freedom. However this matrix is not positive and without pivoting, zero diagonal terms could appear during the factorization on one of the corner degree of freedom. It is to be noticed that a singularity on one of the augmentation degree of freedom can be dealt with simply by eliminating the augmentation degree of freedom. Such an occurrence only affects the convergence rate but does not otherwise adversely affect the method. However it is imperative that no singularity appears on the corner degrees of freedom.

We note that is is always possible to deal with the occurrence of a zero pivot in the factorization of K_{cc}^*, the corner node portion of $\widetilde{K}_{cc}$ by using pivoting if we assume the global coarse matrix to be non-singular. However pivoting solver are generally complex and usually have a slightly lower performance when compared with non-pivoting solver. Moreover, guaranteeing that the augmentation correctly addresses the singularity in K_{cc}^* is by no means a trivial task.

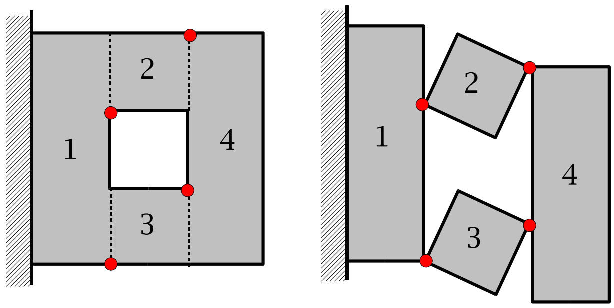

Figure 4.1: Coarse Problem Mechanism

Fortunately, it can be guaranteed that no zero pivot will appear on the corner degree of freedoms if K^*_{cc} is non singular. Because of this simple remark, we propose to build our corner selection algorithm to guarantee that K^*_{cc} is non singular. This choice will make pivoting unnecessary and consequently simplify the implementation and improve the performance of the code.

3.3. Subdomains as Super-Elements. In order to facilitate the discussion of the non-singularity of K^*_{cc}, we first notice that

$$K^*_{cc} = \sum_{s=1}^{N_{sub}} K^{(s)}_{cc} - K^{(s)}_{cr} K^{-1}_{rr(s)} K^{(s)}_{rc} \tag{3.1}$$

is an assembly of subdomain as Super-Elements where only the corner nodes are kept for attaching subdomains together. We will assume in what follows that every subdomain created by the decomposer is free of any internal mechanism. This is to say that in three dimensions, each subdomain, before the application of any boundary condition has exactly 6 rigid body modes, while in two dimensions, each subdomain has exactly 3 rigid body modes.

4. An Ad-Hoc Algorithm. In our early implementation of the FETI-DP algorithm, we extended the two-dimensional view of corners to three-dimensions by using the following algorithm:

1. Pick nodes with more than 4 neighbors as corner nodes
2. Post-guarantee the non singularity of K_{rr}

Unfortunately this algorithm generally leads to a large number of corners and more importantly, it does not offer any guarantee as to the non-singularity of the $\widetilde{K}_{cc}$ matrix. Figure 4.1 shows a two dimensional example. In this problem there are no points where three or more subdomains meet. Therefore, the corners have been chosen to guarantee the non singularity of all the subdomains – i.e. in this case, at least two corner nodes per subdomain. It can be seen that the resulting system has a spurious mechanism.

5. A Robust Algorithm. To keep the following discussion clear, let us introduce two important definitions:

Mechanism-Free entity: a set of elements such that when combined together, there is no mechanism between any part of the set

Subdomain to Subdomain Face: the set of nodes shared by two given subdomains

Under our assumption about the decomposer, every subdomain is a *Mechanism-Free entity*.

It is easy to check that, using only corner nodes to attach subdomains together, two *Mechanism-Free entities* can be combined into a single composite *Mechanism-Free entity* if they share at least 3 non colinear corner nodes in three dimensions or 2 non colocated nodes in two dimensions.

We also note that when a subdomain is merged with any *Mechanism-Free entity*, it will guarantee that its local $K_{rr}^{(s)}$ matrix is non singular.

Thus, by recursively combining pairs of *Mechanism-Free entities* until the whole set of subdomains has been merged into a single entity, we can attain our goals, thus leading to:

Corner Selection Algorithm

1. Mark Corner Candidates on Each Subdomain Face
2. Declare Each Subdomain a Mechanism-Free Entity
3. Iterate Until all Subdomain are Assembled into a Single Entity:
   (a) For each Entity, Choose 2 Preferred Neighboring Entities by:
      i. Favoring Already Picked Corners.
      ii. Maximizing the Area Formed by the Corner Nodes Joining the 2 Entities.
   (b) Check if Previous Choices of Corner Create a Tie between Entities
   (c) Merge Entities, Favoring Pre-Existing ties, then Paring

In the first step of the algorithm, we pick candidate corner nodes from which all corner nodes will be chosen. In most three-dimensional problems, the faces between neighboring subdomain are two-dimensional and therefore if we have at least three non-colinear corner nodes on such faces, we can guarantee that we can tie each subdomain to a neighbor as a Mechanism-Free Entity, guaranteeing by the same operation that the subdomain $K_{rr}^{(s)}$ will be non-singular.

We will note that there are some special cases to deal with. It is possible to have a structure in which subdomains are attached by faces that are all two dimensions lower than the dimension of the problem –i.e. single nodes in 2 dimensions. This means that even when use all the potential corner nodes, it is not possible to tie two subdomains into a single *Mechanism-Free entity* just by one face. In such a case, the algorithm may end up with a final set of *Mechanism-Free entities* that it cannot guarantee can be tied into a single one. In such a case, we will take all the remaining corner candidates shared by at least two entities as corners. Assuming the global problem was mechanism-free, the resulting choice will guarantee the non singularity of K_{cc}^{*}. If the resulting K_{cc}^{*} remains singular however, we will conclude that the global problem was singular and an error can be generated for the user.

6. Numerical Results.

We present two numerical examples of large three-dimensional structures. The first model is of a car engine component and has 985,340 degrees of freedom. It is made of four noded tetrahedra (see Figure 6). The second model, illustrated in Figures 6 has 2,437,104 degrees of freedom. Both models were run first with the Ad-Hoc algorithm and secondly with new algorithm. The results show the number of corners generated, the total number of degrees of freedom of the coarse problem, the memory used by the $\widetilde{K}_{cc}$ matrix, the number of iterations and the total elapsed time for the solution.

Figure 6.1: Engine Gas Collector Geometry

Algorithm	Number of Corners	Coarse Pb Size	Memory Usage	Iteration Count	Solution Time
Old	2,285	15,692	60MB	38	501s
New	1,571	13,679	44MB	41	490s

Table 6.1: Results for the Engine Gas Collector

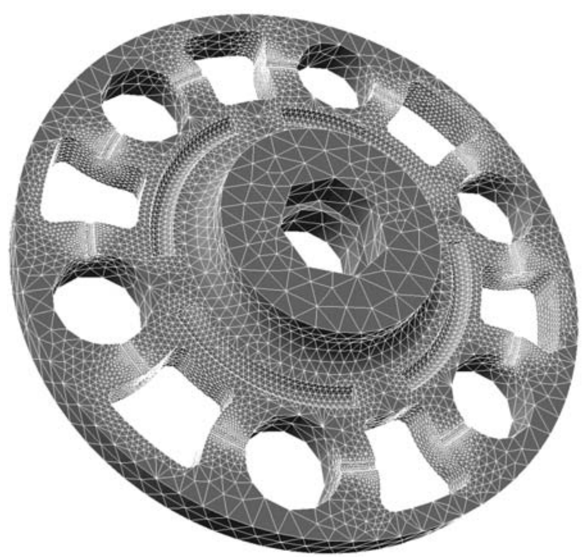

Figure 6.2: Wheel Carrier Geometry

Algorithm	Number of Corners	Coarse Pb Size	Memory Usage	Iteration Count	Solution Time
Old	3,163	28,572	154MB	104	1272s
New	2,210	25,095	118MB	104	1218s

Table 6.2: Results for the Wheel Carrier Problem

The engine component was run using 8 CPUs on an SGI Origin 2000 machine. The results show that the number of corner nodes was reduced by roughly 30% while the total number of degree of freedom in the coarse problem is reduced by 12.5% This reduction leads to a saving of memory of 27% We observe a slight but increase in the number of iterations to reach the solution, however the smaller coarse problem leads to a lower cost per iteration and a shorter factorization of the coarse matrix. As a result, the overall timing is about 2% faster.

The wheel carrier shows similar effects. The reduction in number of corner is again roughly one third while the reduction in number of coarse degrees of freedom is lower. In this case, the number of iterations remains unaffected by the number of corners and the overall execution time is faster with the smaller coarse problem.

7. Conclusions. We have presented an algorithm for the selection of corner nodes for three-dimensional problems for the FETI-DP algorithm. This algorithm offers the benefit over the previous Ad-Hoc algorithm of guaranteeing that no zero pivot will appear during the factorization of the coarse problem.

With two large examples of three-dimensional problems, it was shown that the improved algorithm leads to a reduction of number of corners by roughly one third and is accompanied by a very small decrease of convergence rate. However, the coarse problem matrix is smaller and the resulting reduction in factorization cost as well as a reduction in the cost for the solution of the coarse problem at each iteration results in a slight reduction of the overall solution time.

REFERENCES

[1] P. L. K. P. C. Farhat, M. Lesoinne and D. Rixen. Feti-dp: A dual-primal unified feti method - part i: A faster alternative to the two-level feti method. *International Journal for Numerical Methods in Engineering*, 2000. in press.

[2] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.

[3] C. Farhat and J. Mandel. The two-level feti method for static and dynamic plate problems - part i: an optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering*, 155:129–152, 1998.

[4] A. Klawonn, O. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. Technical Report 815, Courant Institute of Mathematical Sciences, Department of Computer Science, April 2001. to appear in SIAM J. Numer. Anal.

[5] M. Lesoinne and K. Pierson. Feti-dp: An efficient, scalable, and unified dual-primal feti method. In T. Chan, T. Kako, H. Kawarada, and O. Pironneau, editors, *Domain Decomposition Methods in Sciences and Engineering*, pages 421–428. DDM.org, 1999.

[6] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.