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

# 16. Modifications to Graph Partitioning Tools for use with FETI methods

M.K. Bhardwaj[1], D.M. Day[2]

**1. Introduction and Summary.** Engineering solutions are presented for certain massively parallel implementation issues associated with FETI domain decomposition methods [2]. A wrapper around a graph partitioner is defined so that a computational domain is decomposed into subdomains that may be used with FETI methods. The techniques described here may find use with other domain decomposition methods for structural dyamics in which the subdomain matrices are factored. Our solution methodology is imperfect, but it is the most robust way known to the authors to use an off the shelf graph partitioner with FETI methods.

A unique aspect of finite element methods in structual dynamics is the variety of elements combined in a model. FETI methods employ partitions of the dual or element connectivity graph. This article contributes a set of weights depending on the element type that improve load balance with FETI methods.

A serious problem with FETI methods is that incompletely connected subdomains result in subdomain mechanisms that are difficult to characterize geometrically. A general definition of element connectivity is given, and used in a post process of the partition that further decomposes each subdomain into its connected components.

The discussion is organized as follows. The remainder of this section reviews certain relevant aspects of structural dynamics and describes the model problems. Section two concerns element weights. The resulting load imbalance is presented for the more problematic of the two models. The third section addresses subdomain mechanisms and connectivity. Numerical results and conclusions are presented in section four. Numerical examples are also integrated into the exposition.

The United States Department of Energy (DOE) has supported work at Sandia National Laboratory on full systems analysis. The goal is to simulate designs of applications of interest to DOE using massively parallel platforms. The development of hardware, software and algorithms for these tasks is challenging.

A design cycle of component models culminates in an evaluation based on an analysis of a few hundred of the smallest eigenvalues and eigenvectors. The first step in our design cycle is to generate a conforming mesh for the full system using a commercial mesh generation package such as Patran or Sandia's Cubit framework. Second the mesh is partitioned using Chaco. Parallel graph partitioning packages such as including METIS [4] and Zoltan are available. Graph partitioners have also been developed specifically for use with FETI methods (e.g. TopDomDec). Our comments apply to all of these tools. A finite element code is used to build matrices, such as Salinas. The inverted generalized symmetric semi-definite eigenvalue problem is solved using PARPACK. FETI methods are used to solve the resulting sequence of linear systems [1].

Graph partitioning software packages routinely determine partitions in which processor loads vary by less than one tenth of one percent. Domain decomposition algorithms have more specific requirements on a partition than are addressed by graph partitioners. For FETI methods the weight of a subdomain depends primarily on the number of nonzeros in the Cholesky factor of the stiffness matrix, a nonlinear objective function. For FETI-DP methods, another critical variable is the size of the resulting coarse grid linear system. The techniques described here significantly reduce the size of the FETI-DP coarse grid linear

[1]Sandia National Labs, mkbhard@sandia.gov
[2]Sandia National Labs dmday@sandia.gov

system. The interface size and roughness are secondary contributions.

Another set of problems stem from the linear elasticity equation. In the dual formulation, FETI-1 ([3]), the subdomain stiffness matrices are singular. A six dimensional null space may be determined from the geometry (coordinates) for each face connected component (defined in section 3) of the subdomain. In a subdomain that is not face connected, it is possible for the Cholesky factorization routine to incorrectly reveal the null space. The problem persists in dual-primal methods, and is addressed through sophisticated corner node selection methods. As the number of processors increases, the probability that an off the shelf graph partitioning tool will introduce mechanisms also increases.

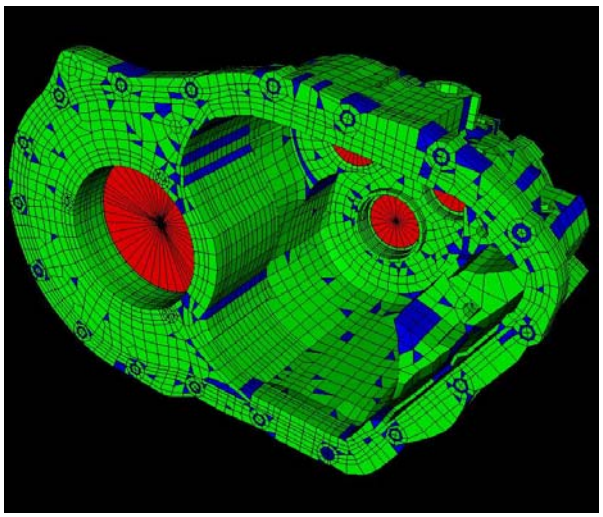

Figure 1.1: Engine model

Model problem one is an engine manifold (see Figure 1.1) and model problem two is the electronics package from a structure of interest at Sandia (see Figure 1.2. For both problems PARPACK needs to solve 31 linear systems in order to approximate the ten smallest modes. The engine model has 203894 nodes (three unknowns per node) and 193960 elements. Most of the elements are eight node hexagons, and the other elements are six node wedge elements and six node triangles. The computations on the engine model were performed on the ASCI Red platform (see http://www.sandia.gov/ASCI/Red/). The component model has 248226 nodes (three unknowns per node) and 167928 elements. The elements are six node triangles and ten node tetrahedrons. Computations with the component model were performed on the CPLANT platform (see http://www.cs.sandia.gov/cplant/), the worlds fastest Linux cluster. The CPLANT platform is composed of 1536 Compag DS10L 1U servers connected via Myrinet networking hardware.

**2. Elements and Weights.** Finite element models of aerospace structures routinely contain many different element types. The ratio of unknowns to elements is asymptotically constant on homogeneous submeshes with simple topologies. Unfortunately the load balance problem for FETI methods is not linear, depending on the number of nonzeros in the Cholesky factor of each subdomain matrix.

The nonlinear load balance problem is addressed by the selection of element weights. Initially a nearby linear problem is solved. The asymptotic ratio of unknowns to elements for a regular mesh is used as an initial guess for the element weight. The weights were then

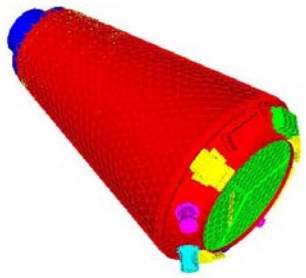

Figure 1.2: Component model

calibrated on a few model problems.

Structural dynamics models make use of a one, two and three dimensional elements. One dimensional elements, including truss and bar elements, are all converted to beam elements in our finite element code. The shell elements are quadrilateral and triangular. Beam and shell elements have six unknowns per node. The additional drilling degrees of freedom at each node are essential for maintaining subdomain connectivity. The solid elements are hexagonal, prismatic or tetrahedral, and all have three unknowns per node. An element may have nodes only at the vertices (linear shape functions) or nodes at both the vertices and the midpoints of the edges (quadratic shape functions).

Element weights balancing the number of subdomain unknowns for models with regular meshes are known. The element weight is the ratio of the number of unknowns to the number of elements. In two dimensions the number of knowns is asymptotically equal to the sum of the unknowns per node and three times the number of nodes per edge (see §1.9 of [5]). There are similar formulas for solid elements. These element weights vary by an order of magnitude.

Balancing the unknowns per subdomain does not solve the load balance problem for FETI methods. Subdomains consisting of irregular solid elements may have Cholesky factors with relatively large numbers of nonzeros. An example of such a subdomain is presented in Figure 2.1. Furthermore a subdomain that consists entirely of shell elements usually comes from a two dimensional subcomponent (e.g. an aero-shell); such a subdomain has a relatively sparse Cholesky factorization and a one dimensional boundaries.

The weights of the solid elements have been experimentally increased to decrease the load imbalance. One set of sub-optimal weights are used for all models. Reports of load imbalance problems ceased once the graph partitioner was modified to use the weights listed in 2.1.

The load balance for the component model partitioned into 540 subdomains using the weights is depicted in Figure 2.2. The data for partitions into 137 and 277 is similar. In each case the ratio of the maximum to the average for *both* unknowns and nonzeros is 3/2;

The subdomain stiffness matrices with the most nonzeros still correspond to irregularly meshed solid elements. The large spread in the number of nonzeros in the Cholesky factorization represented how inexactly the nonlinear load balance problem is solved. The processors

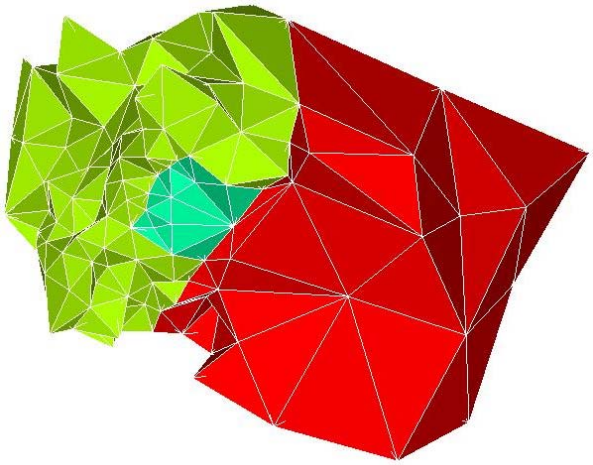

Figure 2.1: The subdomain with largest Cholesky factor in the 137 subdomain partition of the component is shown

with large numbers of unknowns are subdomains of the aero-shell. The source of the extremely small subdomains will be explained in the next section.

**3. Mechanisms and Connectivity.** A feature of FETI-1 methods is that singular matrices must be accurately factored. This is possible if the subdomain is face connected and the nodes are ordered so that either the last three nodes are 3 unknowns per node nodes and the nodes are not collinear, or the node is a 6 unknown per node node. One of the three main properties of FETI-DP methods is that only nonsingular matrices are factored. The other two nice properties of FETI-DP are that the coarse problem is sparse and that fewer iterations are required for convergence. For an arbitrary partition the null space of the subdomain stiffness matrix could be anything.

FETI-DP is more reliable than FETI-1, but is still sensitive to the partition. If a subdomain is not face connected, the corner nodes may not eliminate the entire null space. A feature of FETI-DP is that the coarse grid problem is approximately three times larger. Evidence will be presented in the next section that the load balance techniques developed here usually result in smaller coarse grid problems.

Figure 3.1 depicts the subgraph assigned to one processor. If we define two elements that share a node to be connected (nodal connectivity), then the subgraph has four connected components. Note that there is a triangular element that shares a node but not an edge with it's neighbors.

Here a more restrictive definition of connectivity is used, face connectivity. Two solid elements (or a solid and a shell) are face connected if they share a face. A shell element and a solid element that share three or more nodes are face connected. Two shell elements are face connected if they share an edge. A beam element is face connected if it shares a node with another beam element or a shell element. The subgraph in Figure 3.1 has five face connected components. In the remainder of this work connectivity always refers to face connectivity.

Connectivity is ensured by assigning each extra connected component of each subdomain to an additional processor. For the models considered, this results in a modest increase in the number of processors (see Figure 3.2).

| *Element* | *Number of Nodes* | *Weight* |
|---|---|---|
| *Wedge* | 6 | 2 |
| *Wedge* | 15 | 12 |
| *Tet* | 4 | 1 |
| *Tet* | 10 | 3 |
| *Hex* | 8 | 3 |
| *Hex* | 20 | 12 |
| *Tri* | 3 | 3 |
| *Tri* | 6 | 12 |
| *Quad* | 4 | 6 |
| *Quad* | 8 | 12 |
| *Beam* | 2 | 1 |

Table 2.1: graph weights for elements with linear (e.g. Tet4) or quadratic (c.f. Tet10) shape functions.

The number of face connected components of a partition may be acceptably large. Examples of such meshes have come to the attention of the authors in which beam elements have been painstakingly used to sow together nonconformal solid meshes.

**4. Results and Conclusions.** Results are presented for the engine and component models.

The engine model problem is solved efficiently on $2^7$ or $2^8$ processors. Though the engine model consists mostly of hexagonal elements, the modified graph partitioner is noticably more efficient. On approximately $2^8$ processors the improved partitions reduce the time required to solve 31 linear systems from 261 seconds to 172 seconds.

The $2^9$ processor runs illustrate different failure modes with two different corner selection strategies. For one corner selection strategy, with the improved partition the factorization of the coarse grid nonetheless erroneously detects zero pivots, and slow convergence results. It is noteworthy that our framework is not yet robust. For the other corner selection strategy, the improved partition is much more efficient due to the reduction in the size of the coarse grid problem.

The component model contains many triangular elements, and better illustrates the improvements in the partitions. Only results for the component model with the improved partition are presented. For the standard partition, initially FETI-DP broke down due to singular subdomain matrices across the processor range. Singular subdomain matrices were avoided by solving the shifted problem ( $K + M10^5$ ). Unfortunately memory is insufficient to factor the shifted stiffness matrices on 128, 256 or 512 processors using a serial or a parallel linear solver for the coarse grid.

For the improved partitions and using the serial coarse grid solver FETI-DP is successful on 137 or 277 subdomain partitions, but on the 540 subdomain partition, memory was exhausted. For the 540 subdomain partition, FETI-DP succeeded using the DSCPACK parallel coarse grid linear solver.

In summary a technique for improving the partitions determined by an off-the-shelf graph partitioner have been presented. A carefully calibrated set of element weights is used to maintain load balance. Furthermore extra subdomains are added to ensure the face connectivity of the subdomains. The technique also results in smaller coarse grid problems for FETI-DP.

REFERENCES

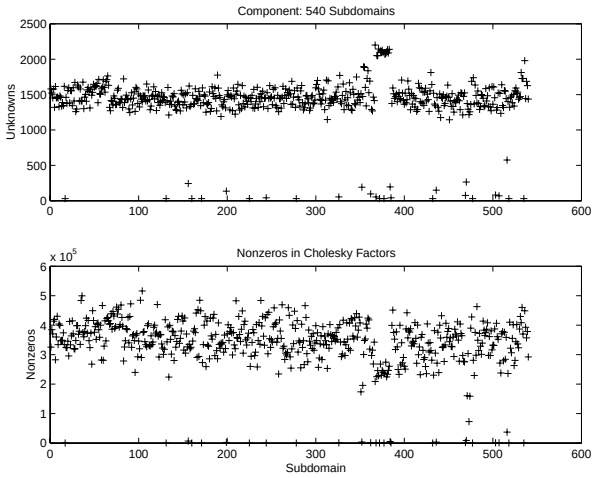


Figure 2.2: The processor loads for the component model partitioned into 540 subdomains is displayed. For both the number of unknowns and the number of nonzeros, the ratio to the maximum to the average is 3/2. Similar results are observed for partitions into 137 and 277 subdomains.

[1] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen. Application of the FETI method to ASCI problems - scalability results on one thousand processors and discussion of highly heterogeneous problems. *Int. J. Numer. Meth. Engrg.*, 47:513–535, 2000.

[2] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, and D. Rixen. FETI-DP: A dual-primal unified FETI method – part I: A faster alternative to the two-level FETI method. *Int. J. Numer. Meth. Engrg.*, 50:1523–1544, 2001.

[3] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115:367–388, 1994.

[4] G. Karypis and V. Kumar. Metis, unstructured graph partitioning and sparse matrix ordering system. version 2.0. Technical report, University of Minnesota, Department of Computer Science, Minneapolis, MN 55455, August 1995.

[5] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, N.J., 1973.

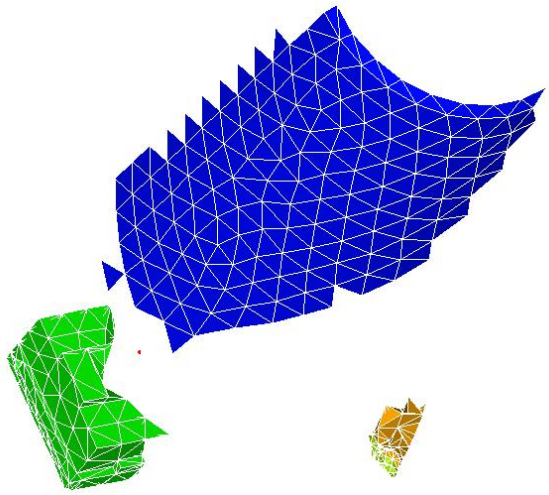

Figure 3.1: A disconnected subdomain computed by a graph partitioner for a 128 subdomain partition of the component model is depicted. The subdomain has four node-connected components, and five face-connected components.

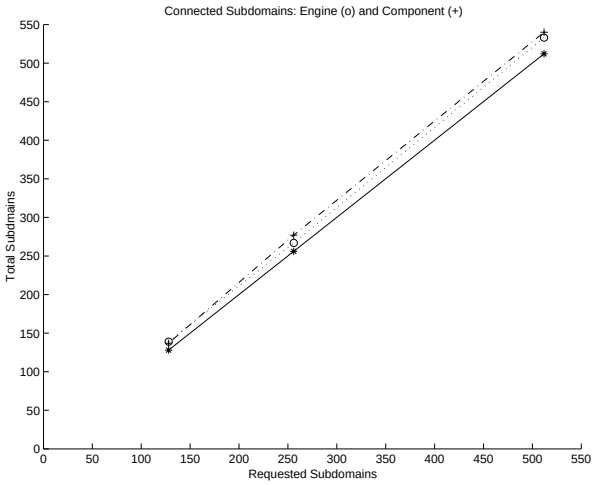


Figure 3.2: The figure displays the number of subdomains determined if 128, 256 and 512 subdomain partitions are requested for both the engine and the component model. The solid * line depicts no extra subdomains. The dash dot + line corresponds to the component model, and the dotted o line corresponds to the engine model. In the latter two cases each extra connected component of each subdomain is assigned to an additional subdomain.

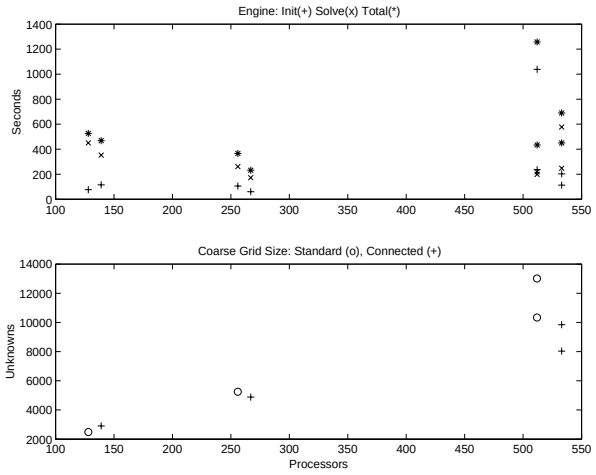


Figure 4.1: The figure displays the results for the engine model. The upper figure shows the initialization time (+), solve time (gap between (x) and (+)) and total time to compute the ten lowest modes on different numbers of processors. The lower figure shows the corresponding number of coarse grid unknowns for FETI-DP with the standard partition (o) and with the weighted partition maintaining connectivity (+). The $2^9$ processor runs were run twice with different corner selection strategies.

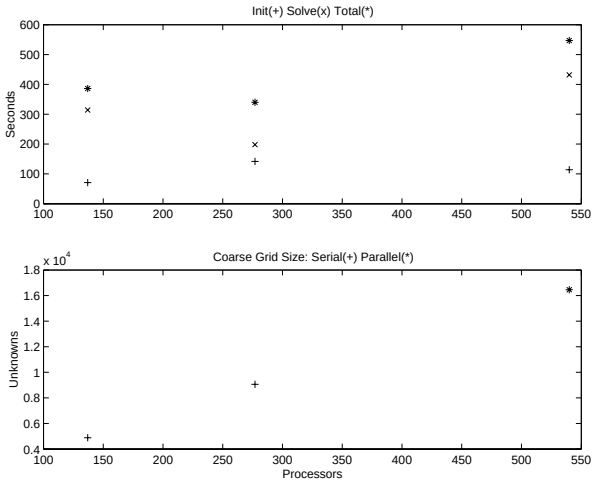


Figure 4.2: The results for the component with the weighted partition maintaining connectivity. No data is shown with the standard partitions due to break downs.