Mechanism Free Domain Decomposition

D. M. Day [a,1], M. K. Bhardwaj [b,1], G. M. Reese [b,1], and J. S. Peery [b,1]

[a] *Organization 9214, Mail Stop 1110, Computational Mathematics and Algorithms, Sandia National Laboratories, Albuquerque, NM 87185-1110*

[b] *Organization 9142, Mail Stop 0847, Computational Solid Mechanics and Structural Dynamics Sandia National Laboratories, Albuquerque, NM 87185-0847*

Abstract

The simulation of three dimensional structural dynamics on massively parallel platforms places stringent requirements on the existing software infrastructure. A constrained and nonlinear graph partitioning problem that arises in scalable iterative substructuring methods, such as FETI methods, is identified. New sufficient criteria on a partition are presented that ensure the applicability of FETI methods, and improve the associated preconditioner. One dimensional finite elements in three dimensional structures are treated by an encapsulation method. The techniques are demonstrated on complex finite element model problems.

Post Script: This article appeared in *Computer Methods in Applied Mechanics and Engineering*, volume 192, number 7-8, pages 763-776, 2003.

1 Introduction

In 1996, the United States Department of Energy announced the Accelerated Strategic Computing Initiative aimed at enabling predictive simulation and virtual prototyping capabilities, and accelerating the development of high-performance computing through a focused initiative. Part of the ASCI initiative is the development at Sandia National Laboratories (Sandia) of Salinas [24], a scalable implicit structural dynamics package that enables structural dynamics simulations of unprecedented fidelity and complexity. Such large-scale finite element analyses, involving substantial computational effort, provide important information including vibrational and shock loads for components within larger systems, design optimization, frequency response informa-

[1] Supported by the Department of Energy through grant DE AC04 94AL85000

tion for guidance and space systems, modal data necessary for active vibration control, and characterization data for structural health monitoring.

Salinas is an implicit structural dynamics C++ code designed for optimal performance on massively parallel platforms. The Salinas package enables analysts to use structural dynamics finite element models with millions of degrees of freedom. Though Salinas is still under development, its many capabilities make it a powerful tool for structural dynamics. Salinas uses the Message Passing Interface standard ([10]) and is supported on parallel platforms including each ASCI platform and most shared memory machines. The purposes of this article are to demonstrate the reliability of Salinas, to present techniques developed to model structures of interest at Sandia, and to identify a research problem.

A design cycle consists of the following steps. The generation of the unstructured conforming finite element meshes is the first step and the most time consuming step by far. This paper concerns the impact of the second step, partitioning the mesh, on certain linear solvers. We use Chaco [12], and our results apply equally to other graph partitioners such as METIS [16] or Top-DomDec [6]. For eigenvalue problems, the shifted and inverted generalized symmetric semi-definite eigenvalue problem is solved using PARPACK [19]. Finite Element Tearing and Interconnecting (FETI) methods [5] are used to solve the resulting sequence of linear systems. Lastly, the analyst evaluates the modal data.

Graph partitioning tools for meshes and networks efficiently partition a graph into a user specified number of subgraphs so as to balance an additive measure of the subgraphs and to minimize interface sizes [16]. A graph with weights on the nodes and edges is partitioned to approximately minimize both the maximum subgraph weight (that depends on the node weights) and the interface size (that depends on the edge weights).

The performance of iterative substructuring methods depends on a different nonlinear cost function. Scalable domain decomposition based solvers require that each subgraph is a valid finite element discretization of a patch of the structure. The observation has motivated using different related additive measures such as the subdomain geometric aspect ratio [6], [26] or element type [21] in the graph partitioning problem and in the iterative repartitioning problem [3].

Partitions of a finite element model sometimes introduce spurious zero energy modes or **mechanisms**. If the partitioner introduces mechanisms, then iterative methods do not apply. The probability that mechanisms will be introduced by a black box partitioner increases with model complexity and the number of subgraphs. Though partitioning a simple model into a few subgraphs rarely introduces mechanisms, partitioning a complex model into thousands of subgraphs almost surely introduces at least one mechanism. For a finite element model using only continuum elements, mechanisms are related to the connectivity [15] of the subgraphs.

We present an algorithm to amend arbitrary partitions to satisfy a minimal set of 'validity' criteria required by scalable linear solvers for structural dynamics. The algorithm is implemented as a pre- and a post- process of an abstract graph partitioner that further decomposes each subdomain into elasticly connected components.

Summary: The article is organized as follows. In section two, the variety of finite element discretizations required for a general purpose structural dynamics code are specified. In section 2.1 sufficient criteria are given that a partition of a finite element model determines elasticly connected subdomains. Furthermore, a wrapper around an abstract graph partitioner is defined that produces elasticly connected subdomains by increasing the user specified number of processors. The preconditioner is improved and certain numerically unstable calculations are avoided (e.g. using finite precision arithmetic to determine if three points are collinear) by using sufficient criteria for elastic connections. Section three reviews iterative substructuring methods, and FETI-DP methods are considered in detail. Section four contains our numerical results. Section five discusses load balance issues, from the point of view of optimizing a partition from within a graph re-partitioning framework. Our results are summarized in section six.

2 Finite Element Mechanisms

Partitioning of a valid (properly connected) three dimensional (3D) finite element model determines a set of subgraphs. The finite element models corresponding to the subgraphs sometimes contain problematic mechanisms. We require a solution to the mechanism problem for 3D structures that contain some 1D and 2D finite elements. Sufficient validity criteria are presented that preclude mechanisms in 3D finite element models that may contain 1D or 2D finite elements. An algorithm is presented that ensures that each subdomain is mechanism free. The algorithm is similar to the mechanism buster [7] and the rigid parts detection algorithm [23]. The user specified number of processors is increased to correspond to the number of connected subdomains. The sufficient criteria lead to improved preconditioners. A second contribution of the new algorithm is that it works for models with 1D finite elements. The section is organized as follows. First the conforming finite elements used in Salinas for structural dynamics models are described. Second the associated primal (mesh nodes) and dual (elements) graphs are defined. Next mechanisms and elastic connections are defined. The combinatorical problems associated with trusses in realistic designs is discussed.

Assorted Low Order Finite Elements. 3D structural dynamics models include one, two, and three dimensional (finite) elements. Finite elements may share nodes, edges, or a face. 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). Figure 1 displays a three noded triangle with linear shape functions and a six noded triangle with quadratic shape functions. In this topological discussion shell and plate elements are not distinguished. Shell elements are quadrilateral or triangular. The solid elements are hexagonal, prismatic (wedge) or tetrahedral, and all have three unknowns per node. Beam and shell elements satisfy the discrete Kirchoff hypothesis [2] and have six unknowns per node.

Connectivity and decomposition issues are significantly more difficult for 1D elements than for solids and shells. A general purpose structural dynamics element library will contain a whole family of 1D finite elements. Beams activate translations and rotations for a total of six unknowns per node. Trusses typically provide stiffness in the axial direction only, but may also have rotational stiffness. Springs connect arbitrary degrees of freedom. Fortunately, for most large applications, these elements make up only a small part of the model.

Graphs. Graph partitioners operate on weighted graphs. We use elementary graph theory to address the mechanism problem. A graph is a set of vertices together with a set of edges between the vertices. A connected graph consists of one 'piece' or *connected component.* A disconnected graph has more than one connected component. Both a primal and a dual graph correspond to a finite element discretization. In the **primal graph** the vertices are the nodes (or mesh points), and the edges are the edges of each finite element. Edges in the primal graph have Euclidean lengths. In the **dual graph** the vertices are the finite elements, and the edges are the pairs of elements sharing a node. The element connectivity table in a finite element package is equivalent to the dual graph.

Rigid body motions are displacements of the nodes that leave all edge lengths invariant. A mechanism is a displacement of the nodes that leaves the edge lengths invariant, other than the rigid body motions. Both configurations in figure 2 contain mechanisms. Discrete Kirchoff elements eliminate some mechanisms. Neither configuration in figure 3 contains a mechanism. If K is the corresponding stiffness matrix, then the null space of K is the product of the rigid body motions and the mechanisms.

A set of finite elements is elasticly connected if their dual graph is connected and the ensemble has no mechanisms. In an elasticly connected structure, the

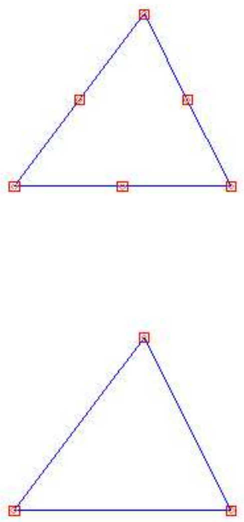

Fig. 1. Three and six noded triangles have linear and quadratic shape function, respectively.

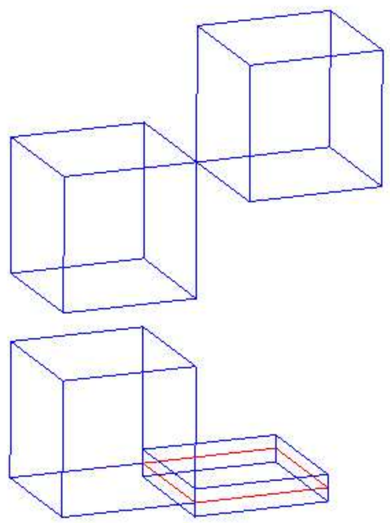

Fig. 2. Two examples of element connections that introduce mechanisms are displayed.

space of rigid body motions is spanned by translations along and rotations about a set of coordinate axes. Two elements sharing either three non-collinear nodes, or one discrete Kirchoff node [23], [7] are elasticly connected. The elements in figure 4 are elasticly connected. On the other hand a 1D element that is not a beam is never elasticly connected to one other element. A principle of design is that no natural substructure contains a mechanism. For this reason, we limit the detail of our characterization of mechanisms. The pairwise elasticly connected elements define a subgraph of the dual graph that we call the

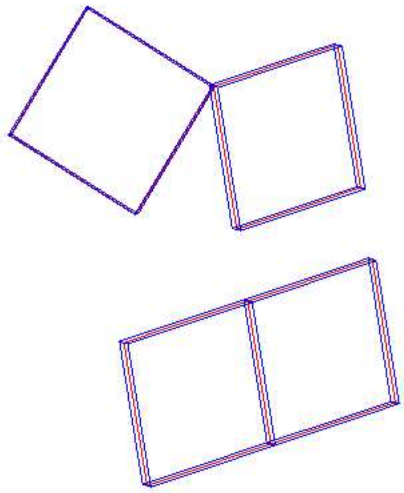

Fig. 3. Elastic connections between discrete Kirchoff quadrilateral elements may (lower edge connection) or may not (upper node connection) satisfy the strong criteria in Table 1.

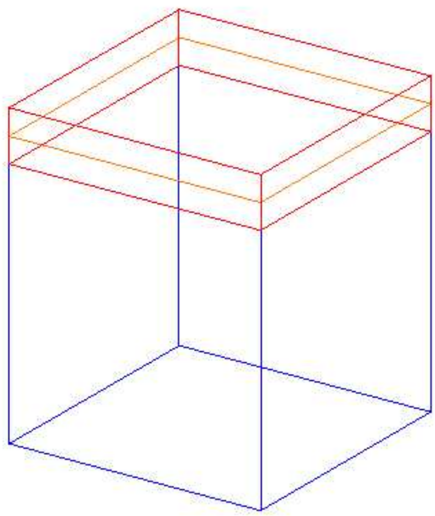

Fig. 4. An elastic connection between a shell and a solid element is illustrated.

elastic connectivity graph.

2.1 Sufficient Partition Validity Criteria

To simplify the exposition, we first present sufficient criteria for elastic connectivity in finite models containing no 1D elements. The connected and elasticly connected components of the subgraphs are determined using a breadth-first search of the dual graph in linear time (c.f. [7]). If the connected and elasticly connected components of the element connectity graph coincide, then the subdomain is mechanism free. Elastic connectivity constraint is satisfied by using the subgraph of elasticly connected elements and a graph partitioner that maintains connectivity [15]. Instead, in a post-process, the number of processors is increased to correspond to the number of elasticly connected subdomains.

In our experience using *necessary* and sufficient criteria to ensure elastic connectivity leads to poor preconditioners. Preconditioners locally approximate structural elasticity by the elasticity of a subdomain. Strengthening subdomain elasticity improves partition dependent preconditioners. A pair of solid elements sharing a node are elasticly connected if and only if they share at least three non-collinear nodes. It would be a design error to base the decision of whether or not two elements are elasticly connected on whether or not three shared nodes are collinear. For example a pair of isoparametric ten node tetrahedrons joined by a curvilinear edge share three noncollinear nodes.

Our criteria for a sufficiently elastic connection are given in Table 1. The definition of elastic connectivity excludes some valid connections. For example sharing an edge is not a sufficient connection between a six noded triangle and a three noded triangle. Also in figure 3 the vertex connection (top) is rejected. Our scheme forces such interfaces into the artificial boundaries determined by the partitioner. In decompositions of extraordinarily complex structures elastic connectivity has been ensured with an increase in the number of processors of about ten percent.

Elements	Order	Nodes
Solid - Solid	1	3
Solid - Solid	2	4
Solid - Shell	1	3
Solid - Shell	2	4
Shell - Shell	1	2
Shell - Shell	2	3

Table 1

Elastic element connections for solid and shell finite elements are summarized. Discrete Kirchoff elements are used for 2D geometries. First order elements have unknowns at vertices and second order elements have additional unknowns at edge midpoints. A sufficient number of common nodes to ensure a elastic connection appears in the last column.

If the assumptions **A1-3** below hold, then a partition of a model (possibly with 1D elements) determines elasticly connected subdomains.

A1. *The model is elasticly connected.*

A2. *The 1D finite elements are in the subdomain interiors.*

A3. *The 2D and 3D finite elements in each subdomain are pairwise elasticly connected* (see Table 1).

To justify this assertion, note that assumptions **A2-3** imply that a connected set of 1D elements is connected to one elasticly connected subdomain. The only possible mechanisms are local motions of the 1D elements. Assumption **A1** states that these elements have no mechanisms in the global model, constraining all local motions.

A mechanism free subdomain sometimes does not satisfy the assumptions. Assumption **A2** is satisfied by using large edge weights for 1D elements. If **A1-2** hold but not **A3**, then **A3** is ensured by increasing the user specified number of processors to the number of elasticly connected subgraphs. The weakness of our approach is that adding processors may violate **A2**. For FETI-DP methods we replace assumption **A3** by assumption **A3.1** and **A3.2** in section 3.1. Alternative techniques to adding processors for ensuring connectivity are discussed in section 5.

3 Iterative Substructuring Methods

In this section scalable and reliable methods for structures are reviewed, emphasizing some issues that arise in models of real structures. Multigrid methods are only mentioned. Iterative substructuring domain decomposition methods apply to general structural dynamics problems. The methods are characterized by exact subdomain solves and a null space assumption. Mechanism free domain decomposition is necessary for satisfying the null space assumption. Salinas uses the FETI-DP iterative substructuring method, which is described in detail in a subsection.

Direct substructuring methods are based on nested dissection sparse matrix orderings and sparse direct multifrontal solvers. The methods are robust but in 3D the computational complexity increases too quickly with problem size, thus creating the demand for scalable methods. By scalable, we mean that an n-times larger problem is solved on n-times as many processors in essentially the same CPU time. Solver development is difficult because on an unstructured mesh the relationship between the differential operator of linear elasticity and its matrix discretization is weak.

Iterative substructuring methods [25] are reliable and scalable. For certain model problems with coarse mesh with mesh size H and fine mesh size h and for certain methods, bounds on the condition number of the preconditioned linear system such as $O(1+\log(H/h))^3$ have been established. Iterative substructuring methods apply to problems with unstructured meshes. The convergence rate is independent of jumps in the partial differential equation coefficients aligned with the subdomain interfaces.

An outline of iterative substructuring methods follows. Consider a subdomain stiffness matrix A. If we reorder the unknowns so that the interior unknowns are first and the unknowns shared with other subdomains (boundary unknowns) are second, then

$$A = \begin{bmatrix} A_{ii} & A_{ib} \\ A_{bi} & A_{bb} \end{bmatrix}. \tag{1}$$

If A_{ii} is nonsingular, then the Schur complement of the boundary unknowns with respect to the interior unknowns is

$$S = A_{bb} - A_{bi} A_{ii}^{-1} A_{ib}. \tag{2}$$

If both A_{ii} and S are nonsingular, then

$$A^{-1} = \begin{bmatrix} I & A_{ii}^{-1} A_{ib} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{ii}^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ A_{bi} A_{ii}^{-1} & 0 \end{bmatrix}. \tag{3}$$

Primal iterative substructuring methods use preconditioners of the form

$$B = \begin{bmatrix} I & -B_{ii} A_{ib} \\ 0 & I \end{bmatrix} \begin{bmatrix} B_{ii} & 0 \\ 0 & B_s \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{bi} B_{ii} & 0 \end{bmatrix}.$$

A domain decomposition method with inexact solvers [25] results if the A_{ii}^{-1} that appears in the first and third terms in (3) is replaced with an approximation. Inexact substructuring methods iterate on both displacements and tractions (or Lagrange multipliers) [17], and have the drawback that ensuring the spectral equivalence of the preconditioned subdomain operators and approximate Schur complement operators is nontrivial. In contrast, exact methods iterate on the tractions only, making the reuse of the Krylov subspaces *more* efficient.

Iterative substructuring methods include the Balanced Neumann-Neumann method and FETI methods. In the Balanced Neumann-Neumann the reduced

Schur complement problem $Su = g$ is solved iteratively. The coarse grid is determined so that the preconditioner, an approximate inverse, is exact on a space that spans the null spaces of the subdomain Schur complements. FETI methods are based on partitions of the finite elements instead of the mesh nodes. The subdomain stiffness matrices are assembled, and Lagrange multipliers enforce an equilibrium.

Iterative substructuring methods have a null space assumption. In the abstract additive Schwarz framework, the global Schur complement problem is a direct sum of singular subdomain Schur complements. The null space assumption is that the preconditioner be a direct sum of subdomain preconditioners each with the same null space as the corresponding subdomain Schur complement.

A significant problem with FETI and Balancing Neumann-Neumann methods is the numerical instability of *sparse* direct solvers applied to singular linear systems. A common technique in the LDL^T factorization of a sparse symmetric positive semidefinite matrix is to neglect pivots below a threshold ([23] section 2). In equation (1) the sensitivity of the Schur complement, defined in equation (2), to perturbations of A_{ii} is governed by the matrix $A_{ii}^{-i}A_{ib}$ (see [13] section 10.3 pp. 210-217). After a small pivot appears, the factorization becomes unstable. The subtle point is that, even with *complete* pivoting, the bound on $A_{ii}^{-1}A_{ib}$ is hopelessly large. Ensuring that the factorization of A_{ii} is stable (as in [4], section 4.5) does not resolve the problem of a large $\|A_{ii}^{-i}A_{ib}\|_2$. Rank revealing LU factorizations using local maximum volume pivoting [22] are stable and *may* preserve sparsity.

Approximate Methods. Iterative methods that do not to exactly solve the subdomain problem are sometimes optimally efficient, and research continues on the application of these methods to general structures (with discrete Kirchoff elements, constraints, contact). A notable genus of scalable methods for linear elasticity is the smoothed aggregation algebraic multigrid (AMG) method for finite element models with solid and shell elements [28], [20], [27], [1]. The methods avoid large coarse grid problem by adding multiple levels, and use grid transfer operators (smoothers and prolongations) that preserve the subdomain rigid body modes. Mechanism free domain decomposition may facilitate parallel aggregation algorithms. The load balance problem for the grid transfer operators is similar to the load balance problem in iterative substructuring methods for the subdomain factors.

3.1 FETI-DP

FETI-DP is an iterative substructuring method. A source of the reliability of FETI-DP methods is that only nonsingular matrices are factored. The coarse problem is sparse and many fewer iterations are required for convergence than with the original FETI methods. Like all iterative substructuring methods, if a subdomain is not elasticly connected, FETI-DP may break down. Nonetheless, among the scalable methods for structures in the opinion of the authors FETI-DP [5] is the *least* sensitive to mechanisms.

Domain decomposition methods are based on a partition of the domain Ω into a disjoint union of subdomains $\overline{\Omega} = \cup_i \overline{\Omega}_i$. The degree of a node is the number of subdomains sharing the node. The subdomain interfaces are the nodes with degree greater than one. In unstructured 3D meshes, nodes of degree are as large as twenty are common, and it is nontrivial to distinguish edge nodes and vertex nodes.

FETI-DP methods involve corner nodes, a subset of the nodes with degree greater than two. One corner selection criterion is to ensure that on each subdomain the submatrix of the subdomain stiffness matrix corresponding to the interior nodes is nonsingular. In the FETI-DP initialization phase, first the remaining (non-corner) subdomain unknowns are eliminated. The corner unknowns are eliminated second, and the iterative solve then approximates the Lagrange multipliers. The interface nodes shared by $\overline{\Omega_i}$ and $\overline{\Omega_j}$ will be denoted Γ_{ij}. To ensure the validity of the second elimination, it suffices (this is not obvious) to select the corners constrain to constrain each nonempty Γ_{ij}. The coarse grid for FETI-DP tends to be about three times larger than the coarse grid for FETI methods.

The convergence rate of vertex based iterative substructuring methods, including FETI-DP, is competitive only if the coarse grid is augmented (cf the analysis of Algorithm A in [18]). The augmentation significantly increases the size of the coarse grid. Eigenvalue and linear transient problems solve a linear system with successive right-hand sides. Instead of augmentation, the initial Krylov subspaces are accumulated to accelerate the following iterations. The best performance is obtained by using more than a minimal (maximal independent) set of corners.

Singularities arise in the subdomain stiffness and coarse grid matrices due to mechanisms. It is possible to use the shift-invert Lanczos algorithm to detect eigenvalues of K (or (K, M)) less than a threshold [9], and eliminate the modes in FETI-DP methods by adding vertices to the coarse grid. The approach assumes the development of a reliable technique for augmenting the coarse grid, and depends on the shift and threshold parameters. It is nontrivial to make the technique robust without adding assumptions on mesh quality, material properties, and geometry. For example in section 4, the numeric singularities persist in a *shifted* eigenvalue problem $(K - M\sigma)\Phi = M\Phi(\Lambda - I\sigma)$ for $\sigma = -10^4$.

Validity with 1D Finite Elements: The assumptions on a partition sufficient to ensure the validity of the formulation of FETI-DP are less restrictive. The set of 2D and 3D elements in Ω_i consists of l_i elasticly connected connected components, $(\Omega_i^k)_{k=1}^{l_i}$ and the nodes on the boundary of Ω_i^k are Γ_i^k. The partition validity criteria for FETI-DP applied to models containing 1D,2D, and 3D elements follow.

A1. *The model is elasticly connected.*

A2. *The 1D finite elements are in the subdomain interiors.*

A3.1. *If $j \neq k$ then Γ_i^j and Γ_i^k are disjoint.*

A3.2. *If Γ_i^k and Γ_i^l both nontrivially intersect some Γ_{ij}, then $k = l$.*

A prerequisite for the next section which is not documented elsewhere in the literature is the load balance of the solution of the FETI-DP coarse grid problem. The n_c by n_c coarse grid problem is factored redundantly on each of the p processors. Each processor determines some of the rows of the inverse, so that an explicit inverse is distributed across the processors. During a solve, global communication is used to assemble the entire right-hand side on each processor. The standard approach is for each processor to determine store the rows corresponding to the corner equations on that subdomain, so that only one global communication is required[11]. Instead we partition the rows perfectly between the subdomains, and use a second global communication to assemble the entire solution on each subdomain.

4 Numerical Experiments

Examples are presented that demonstrate the reliability of FETI-DP methods on realistic model problems, provided that the subdomains are elasticly connected. Unrelated problems with rank revealing LDL^T factorizations and a projection technique that accelerates Krylov subspaces methods are observed. The increase in the number of processors required to maintain elastic connectivity is documented. Additional information is provided that shows how the the computational bottlenecks dominate the load balance problem for FETI methods. Improvements in load balance capabilities will make it possible to solve problems faster and also to solver larger problems.

Each failure of a software component during these experiments is carefully documented. Primarily, without maintaining elastic connectivity, FETI fails on every partition of the aerospace model problem due to mechanisms. We discuss the causes of the other failures. Next we compare the successful results (maintaining elastic connectivity) emphasizing issues related to load balance. The user specified number of processors increases to maintain the validity of each substructure, and the additional number of new processors is documented.

A limitation in our experiments is that once an algorithm fails, there is no further data. To illustrate the load balance problem for FETI methods, elasticly connected partitions are determined with and without element weights depending on the element type. Furthermore, for the aerospace model, some different corner selection algorithms for FETI-DP are compared.

The computational task for each model problem is to compute the ten lowest modes of the structure. In each case thirty-one linear systems are solved. The computational task is considered from the view point of those engineers that are interested in computing extraordinarily large numbers of modes.

A few words about the configuration of the linear solver are necessary. A high relative residual error threshold is used for the linear solver, 10^{-3}. If a domain decomposition method with inexact subdomain solves were used instead, a much smaller residual error threshold would be required to accurately approximate the mode shapes. There is no corner augmentation [5], and a Dirichlet preconditioner is used. For each linear solve after the first, a projection technique (see [7] section 3.1) is used to reduce the components of the residual that the preconditioner missed. A preconditioned conjugate gradient method is used that explicitly reorthogonalizes each direction vector against the first 2000 accumulated during the first few linear solves [8].

The user specified number of processors is always a power of 2 (64, 128 and 256). The number of processors reported always includes the number added to maintain elastic connectivity. For example if 277 processors are used, then 256 were requested and 21 were added.

These models are representative of the results observed by the authors with one exception. For meshes with a nonconforming interface tied together using beam elements, adding processors to maintain connectivity introduces an excessive number of subdomains. We recommend not using beams to tie meshes, and instead recommend using multipoint constraints.

The two model problems are an engine component and an aerospace component. The engine component is an exhaust intake manifold. The finite element model contains 203894 nodes and 193960 elements. There are 165992 eight nodes hexagons, 27024 six node prisms and 944 three node triangles. The computations on the engine model were performed on the ASCI Red platform (see http://www.sandia.gov/ASCI/Red/). The aerospace model is the electronics package of a structure of interest to Sandia. The finite element model contains 248226 nodes and 167928 elements. There are 152687 ten node tetrahedrons and 15241 six node triangles. The component is contained in a can, and the can is welded only at a few points to the rest of the component. For this reason, a subdomain may consist entirely of 2D elements. Modal anaysis is used here to approximate the stress induced on small features, such as a circuit board, by the vibration of the structure. Computations with the aerospace model were performed on the CPLANT platform (see http://www.cs.sandia.gov/cplant/), currently the worlds fastest Linux cluster. The CPLANT platform is composed of 1536 Compaq DS10L 1U servers connected via Myrinet networking hardware.

During the runs on the engine component, one break-down occurred. On 64 processors and without weights based on the elements, the per-processor memory ($256MB$) was insufficient. Certain nearly constant weights happen to work better than constant weights for the engine component. The weights are 3, 4, and 3 for the for the hexagonal, prismatic, and triangular elements respectively. Detailed timing information is displayed in Table 2. The difference between the Total time and the sums of the individual times is of course the time spent on other tasks. The condition number of the linear system is at least 10^6. The time to back-solve the coarse grid and subdomain stiffness linear systems is the dominant cost as usual.

Weights	**Proc**	**Its**	Proj	Prec	Orth	**Op**	Total
Yes	72	582	68	215	22	309	616
No	128	937	32	138	43	234	449
Yes	139	680	27	98	23	150	301
No	256	1091	14	69	45	131	261
Yes	268	778	14	41	22	97	177

Table 2
Computational results for the **engine** component. The columns have the following meanings. Weights (Yes/No) refers to whether or not node weights based on element types were used. P: the number of processors. Iterations: total number of FETI iterations for 31 linear solves. Project: Time spent (seconds) in projecting the right-hand sides Precond: Time spent (seconds) in applying the Dirichlet preconditioner. Orth: Time spent (seconds) in reorthogonalizing the CG direction vectors. Op: The time spent (seconds) back-solving the coarse grid and subdomain linear systems.

The aerospace model is a more difficult problem. If the subdomains were not elasticly connected, FETI always failed. There were also problems unrelated to subdomain connectivity. The aerospace model is a floating structure. The FETI-DP corner selection algorithm produces positive subdomain matrices, but the coarse grid linear system is singular. The pivot threshold method to determine a rank revealing LDL^T factorization failed for both $p = 2^7$ and $p = 2^8$. In both cases only four pivots were below the default threshold. In our experience, it is usually possible to find a successful threshold for a given problem.

Instead we shifted the eigenvalue problem. The shift is chosen such that the product of (the absolute value of) the shift and the largest entry of the mass matrix is approximately one. For the runs on 2^7 processors, we shifted by -10^5, and for the runs on 2^8 processors, we shifted by -10^4. The condition numbers of the shifted linear systems are at least 3×10^{11} and 3×10^{12} respectively.

A run on 2^8 processors stagnated for reasons that are not perfectly clear. For this partition, if no projection is done, then there is no stagnation, but the convergence rate is much worse than for the successful runs tabulated below. Salinas is designed so that the projection automatically turns off once stagnation is detected.

For the aerospace model different weights were used (resulting in different numbers of processors), and for a given partition, different corner selection algorithms were used. The computational results are tabulated in Table 3. The time to back-solve the coarse grid and subdomain stiffness linear systems is the dominant cost as usual.

Weights	**Proc**	**Its**	Nnz	Proj	Prec	Ortho	**Op**	Total
No	134	1118	2.5e+6	12	72	32	122	246
No	134	1397	2.4e+6	12	91	51	134	298
Yes	134	1116	4.6e+6	17	83	33	170	311
Yes	134	1166	2.2e+6	15	81	57	165	336
Yes	134	1484	2.2e+6	11	72	63	141	298
No	265	1173	9.8e+5	13	60	53	173	320
Yes	277	1149	1.5e+6	9	30	28	120	195

Table 3
Computational results for the **aerospace** component. The columns have the same meanings as in Table 2. The additional column Nnz reports the maximum number of nonzeros in the factor of a subdomain stiffness matrix.

For a fixed problem as the number of processors increases, the load balance deteriorates significantly (see figure 5). Also the number of iterations for partitions into essentially the same number of processors varies significantly. Elements weights do not resolve the load balance problems, but can help. For example the engine component run on 72 processors used too much memory with hex and wedge weights of three and two, and only fit in memory with weights three and four respectively. Load balance issues are discussed in the following section.

Solution time is proportional to the product of the number of iterations and the maximum number of nonzeros in a subdomain stiffness matrix. A better understanding of the relationship between the number of iterations and the partition is needed. The 3D geometric aspect ratio of a 2D or 1D subdomain is not defined. A well defined quantity related to aspect ratio is the maximum over the subdomains of the ratio of the subdomain diameter to the minimum element diameter. Alignment of the subdomain boundaries with material interfaces and the construction of convex subdomains are both beneficial.

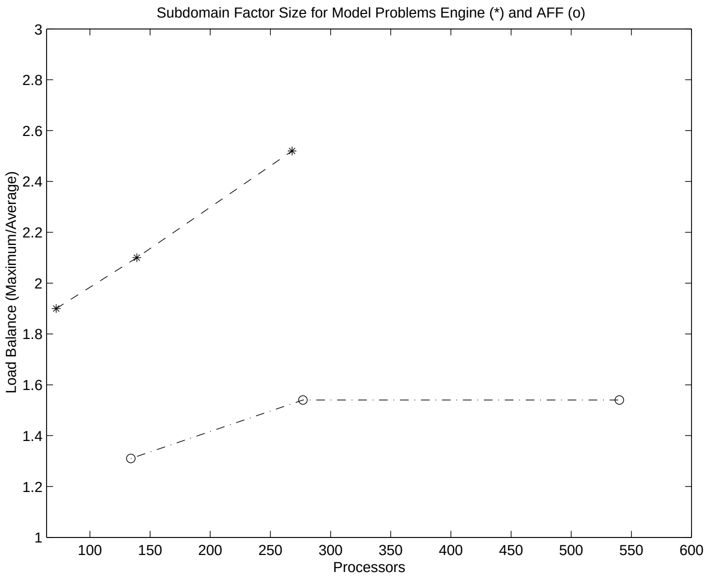

Fig. 5. Load balance for the number of nonzeros in the factors of the subdomain stiffness matrices.

5 Load Balance Issues

After the validity of the formulation, the next most important issue is performance. The load balance problem for domain decomposition methods with exact subdomain solves, such as FETI methods, is an important unsolved problem. We will discuss potential work on migration to balance the load for domain decomposition methods, in an attempt to rule out approaches that will not work. The problem is addressed here in a form that is amenable to iterative graph repartitioning methods. On the other hand, there can be no optimal solution of the problem until the dependence of the number of iterations remains on the partition is better understood.

In the future graph partitioners may address the constrained nonlinear cost functions associated with domain decomposition methods. For example load balance for FETI methods may be addressed from within the iterative *re*-partitioning frameworks under development for use with automatic mesh refinement (AMR) methods. These dynamic methods move data only between adjacent subdomains [15], [14], [3]. For FETI methods, the connected components of subdomains will migrated to ensure connectivity.

The difficulties with the approach stem from the nature of the cost function and the hidden constraints. Problems with aspect ratio are discussed at the end of §4. The density of a graph is the ratio of the number of edges to the number of vertices. A hidden constraint is that density of graph of the subdomain matrix must be nearly maximal. Before evaluating the cost function, one could migrate extra elasticly connected components to maximize graph density. Creating supernodes of 1D elements encapsulated by 2D and 3D elements would also be helpful. The number of nonzeros in the subdomain stiffness matrices may be determined by reordering the primal graph (defined in §2). This cost function is profoundly more expensive to evaluate than the cost functions for AMR and explicit dynamics. Another difficulty is that the cost function is not monotonic; it is possible to remove elements from a subdomain and increase the number of nonzeros in the factors. To approximate the relationship between the number of elements, n, and the number of nonzeros in the factors on the jth subdomain, one may use an *ansatz* such as $C_j n^{\alpha}$. Iteration halts once either the assumed relationship between the number of elements and the number of nonzeros is no longer predictive, or the load is balanced. One last point is that algorithms which maximize the minimum of the cost function avoid a network flow problem.

6 Conclusions

Salinas, a distributed memory software package, is a reliable tool for structural dynamics applications. The impact of the decomposition on the performance has been demonstrated. Graph partitioners introduce mechanisms in the partitions that cause the formulation of domain decomposition methods with a null space property, including FETI methods, to break down A post-process is presented that guarantees the elasticity of the subdomains containing 1,2, or 3D elements. New sufficient criteria for a elastic connection are used that improve upon existing criteria by determining better preconditioned linear systems. Furthermore, a new technique to treat bar and beam elements by encapsulation is presented.

Numerical evidence is presented to demonstrate the effectiveness of the techniques to ensure that the partitions are mechanism free. Lastly a graph partitioning problem for iterative substructuring problems is identified in a form suitable for iterative repartitioning methods.

Acknowledgement. The authors thank their colleagues Erik Boman, Robert Carr, Clark Dohrmann, Bruce Hendickson, Kendall Pierson and Mathew Saint-

John for their assistance and helpful comments over the course of this project. The authors also thank the referees for helpful comments.

References

[1] M. Adams. Evaluation of three unstructured multigrid methods on 3d finite element problems in solid mechanics. Technical Report SAND01-1636J, Sandia National Laboratories, 2001.

[2] D. J. Allman. A compatible triangular element including vertex rotations for plane elasticity problems. *Computers and Structures*, 19(1-2):1–8, 1996.

[3] R. Diekmann, R. Pries, F. Schlimbach, and C. Walshaw. Shape-optimized mesh partitioning and load balance for parallel adaptive FEM. *Parallel Computing*, 26(4):1555–1581, 2000.

[4] C. Farhat and M. Géradin. On the general solution by a direct method of a large-scale singular system of linear equations: Application to the analysis of floating structures. *Internat. J. Numer. Meths. Engrg.*, 44(4):675–696, February 1998.

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

[6] C. Farhat, N. Maman, and G. Brown. Mesh partitioning for implicit computations via iterative domain decomposition: Impact and optimization of the subdomain aspect ratio. *Internat. J. Numer. Meths. Engrg.*, 38:989–1000, 1995.

[7] C. Farhat, K. Pierson, and M. Lesoinne. The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. *Comput. Meth. Appl. Mech. Engrg.*, 184:333–374, 2000.

[8] C. Farhat and F. X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Internat. J. Numer. Meths. Engrg.*, 32:1205–1227, 1991.

[9] J. Fish, Y. Qu, and A. Suvorov. Towards robust two-level methods for indefinite systems. *Internat. J. Numer. Meths. Engrg.*, 45(10):1433–1456, August 1999.

[10] Message Passing Interface Forum. MPI: A message-passing interface standard. Technical report, Argonne National Laboratories, 1994.

[11] W.D. Gropp and D.E.Keyes. Domian decomposition methods in computational fluid mechanics. *Internat. J. Numer. Meths. Fluids.*, 14:147–165, 1992.

[12] Bruce Hendrickson and Robert Leland. The chaco user's guide: Version 2.0. Technical Report SAND94-2692, Sandia National Laboratories, 1994.

[13] N.J. Higham. *Accuracy and stability of numerical algorithms.* SIAM, Philadelphia, PA, USA, 1996.

[14] Y.F. Hu and R.J. Blake. An improved diffusion algorithm for dynamic load balancing. *Parallel Computing*, 25(4):417–444, April 1999.

[15] P. Ciarlet Jr. and F. Lamour. On the validity of a front-oriented approach to partitioning large sparse graphs with a connnectivity constraint. *Numerical Algorithms*, 12:193–214, 1996.

[16] G. Karypis and V. Kumar. Parallel multilevel k-way partitioning scheme for irregular graphs. *SIAM Rev.*, 41(2):278–300, June 1999.

[17] A. Klawonn and O. Widlund. A domain decomposition method with lagrange multipliers and inexact solvers for linear elasticity. *SIAM J. Sci. Comput.*, 22(4):1199–1219, 2000.

[18] A. Klawonn, O. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM J. Numer. Anal.*, 40(1):159–179, April 2002.

[19] R. B. Lehoucq, D.C. Sorensen, and C. Yang. *ARPACK Users' Guide*. SIAM, Philadelphia, PA, USA, 1998.

[20] J. Mandel, M. Brezina, and P. Vanek. Energy optimization of algebraic multigrid bases. *Computing*, 62(3):205–228, 1999.

[21] M.K.Bhardwaj and D.M.Day. Modifications to graph partitioning tools for use with FETI methods. In *Proceedings of the Fourteenth International Conference on Domain Decomposition Methods*, 2002. to appear.

[22] C.-T. Pan. On the existence and computation of rank-revealing LU factorizations. *Lin. Alg. Appl.*, 316:199–222, 2000.

[23] M. Papadrakakis and Y. Fragakis. An integrated geometric-algebraic method for solving semi-definite problems in structural mechanics. *Comput. Meth. Appl. Mech. Engrg.*, 190:6513–6532, 2001.

[24] G. Reese, M. Bhardwaj, D. Segalman, K. Alvin, and B. Driessen. Salinas - User's Notes. Technical Report SAND99-2801, Sandia National Laboratories, 1999.

[25] B. Smith, P. Bjørstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 1996.

[26] D. Vanderstraeten, C. Farhat, P. S. Chen, R. Keunings, and O. Zone. A retrofit and contraction based methodology for the fast generation and optimization of mesh partitions:beyond the minimum interface size criterion. *Comput. Meth. Appl. Mech. Engrg.*, 133:25–45, 1996.

[27] P. Vanek, M. Brezina, and R. Tezaur. Two-grid method for linear elasticity on unstructured meshes. *SIAM J. Sci. Comput.*, 21(3):900–923, December 1996.

[28] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3):179–196, 1996.