SAND REPORT

SAND2003-4391
Unlimited Release
Printed December 2003

A Study of Two Domain Decomposition Preconditioners

Clark R. Dohrmann

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Of£ce of Scienti£c and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Spring£eld, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-4391
Unlimited Release
Printed December 2003

A Study of Two Domain Decomposition Preconditioners

Clark R. Dohrmann
Structural Dynamics Research Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0847

Abstract

Large-scale finite element analysis often requires the iterative solution of equations with many unknowns. Preconditioners based on domain decomposition concepts have proven effective at accelerating the convergence of iterative methods like conjugate gradients for such problems. A study of two new domain decomposition preconditioners is presented here. The first is based on a substructuring approach and can viewed as a primal counterpart of the dual-primal variant of the finite element tearing and interconnecting method called FETI-DP. The second uses an algebraic approach to construct a coarse problem for a classic overlapping Schwarz method. The numerical properties of both preconditioners are shown to scale well with problem size. Although developed primarily for structural mechanics applications, the preconditioners are also useful for other problems types. Detailed descriptions of the two preconditioners along with numerical results are included.

Acknowledgment

The author acknowledges the support of the LDRD program (project number 26515) administered through the Engineering and Computer Sciences Research Foundations. The author wishes to express his gratitude to Kendall Pierson for many helpful discussions of FETI-DP and methods for corner and edge selection. Thanks are also extended to Garth Reese for providing the interface between Salinas and the preconditioners of this study, and to Manoj Bhardwaj for running some examples and for use of his domain decomposition tool. Finally, the author is grateful to Jan Mandel for taking an interest in the first of the two preconditioners of this study and for developing an accompanying mathematical theory.

Contents

Figures

Tables

1 Introduction

The purpose of this report is to document an investigation of two new preconditioners based on domain decomposition concepts. This investigation was part of a three-year LDRD project funded jointly by the Engineering and Computer Sciences Research Foundations. Although development of the preconditioners was motivated largely by structural mechanics applications, the preconditioners are applicable to other problem types as well.

Effective preconditioners are key to the success of iterative methods like conjugate gradients when applied to large-scale finite element analysis. For example, consider a simple elasticity problem in one dimension discretized with N elements. The condition number of the system of finite element equations varies with N^2. Consequently, it takes more and more conjugate gradient iterations to solve the equations to a specified accuracy as the number of elements increases. The goal of a preconditioner is to keep the required number of iterations nearly independent of N at little additional cost.

Preconditioners based on domain decomposition concepts have proven effective at accelerating the convergence of iterative methods when applied to finite element discretizations of partial differential equations. The advantages of these preconditioners are particularly apparent in parallel computing environments with large numbers of processors. Two new domain decomposition preconditioners are presented in the following sections. The first is based on a substructuring approach and can be viewed as a primal counterpart of the dual-primal variant of the finite element tearing and interconnecting method called FETI-DP. Section 2 is an updated version of an earlier Sandia document (SAND2002-2724J) and will appear in *SIAM Journal on Scientific Computing*. A mathematical theory for this preconditioner now exists and appears in the journal *Numerical Linear Algebra with Applications*. The second preconditioner is described in Section 3 and uses an algebraic-based partition of unity to construct a coarse problem for a classic two-level overlapping Schwarz method. The numerical properties of both preconditioners are shown to scale well with problem size.

2 A Preconditioner for Substructuring Based on Constrained Energy Minimization

Abstract

A preconditioner for substructuring based on constrained energy minimization concepts is presented. The preconditioner is applicable to both structured and unstructured meshes and offers a straightforward approach for the iterative solution of second and fourth-order structural mechanics problems. The approach involves constraints associated with disjoint sets of nodes on substructure boundaries. These constraints provide the means for preconditioning at both the substructure and global levels. Numerical examples are presented which demonstrate the good performance of the method in terms of iterations, compute time, and condition numbers of the preconditioned equations.

2.1 Introduction

This study is focused on a preconditioner for the iterative solution of substructuring problems. The basic idea of substructuring is to decompose the domain of a finite element mesh into non-overlapping substructures $\Omega_1, \ldots, \Omega_N$ such that each element is contained in exactly one substructure. In the direct method of substructuring [1], all degrees of freedom (dofs) not shared by two or more substructures are removed via static condensation. One then obtains a much smaller system of equations that only involves dofs on substructure boundaries. This smaller system is then solved by a direct method.

Such an approach is effective for small to moderate sized problems, but it may be impractical for larger ones because the smaller system might still be too large for a direct method. An effective iterative approach for larger substructuring problems is balancing domain decomposition (BDD) [2]. BDD performs well for second-order problems in both two and three dimensions, but requires modifications to effectively address fourth-order problems like plates [3]. A related FETI method [4] also requires modifications for fourth-order problems [5]. The method presented here has many similarities with a dual-primal version of FETI [6] called FETI-DP, but important differences include: 1) the primary variables for iterative solution are displacements rather than Lagrange multipliers, 2) the coefficient matrix for the coarse problem is never indefinite, and 3) multilevel extensions for very large problems appear to be more straightforward.

The method also has similarities with a Neumann-Neumann domain decomposition method for plate and shell problems [7]. If only corner constraints are used, then the substructure spaces, coarse space, and the substructure bilinear forms are identical. The difference is that the present method uses an additive rather than multiplicative coarse grid correction. Consequently, it is possible to use a different bilinear form on the coarse space resulting in a sparser matrix for the coarse problem. In addition, the present method is applicable to 3D problems and allows for more general types of constraints.

Other preconditioners for substructuring exist, but most are only applicable to problems where the finite element mesh is a refinement of a previously existing coarser mesh [8]. Like BDD and FETI, the present method does not require a preexisting coarser mesh and is applicable to unstructured meshes. From a practical point of view, the preconditioner can be implemented using existing software for factoring sparse symmetric definite matrices. The formulation of the preconditioner is presented in the following section. Examples are presented in Section 2.5 which demonstrate its good numerical performance. A mathematical theory for the method is presented elsewhere [9].

2.2 Preconditioner

The discrete internal energy E_i of substructure Ω_i can be expressed as

$$E_i = u_i^T K_i u_i / 2 \tag{1}$$

where K_i and u_i are the stiffness matrix and degree of freedom (dof) vector of Ω_i. The superscript T in (1) and elsewhere denotes transpose. The matrix K_i is assumed to be either symmetric positive definite or symmetric positive semidefinite in this study. The substructure dof vector u_i is related to the global dof vector u by the equation

$$u_i = R_i u \tag{2}$$

where each row of R_i contains exactly one nonzero entry of unity. The assembled finite element equations are expressed as

$$Ku = f \tag{3}$$

where f is the global force vector and the assembled stiffness matrix K is given by

$$K = \sum_{i=1}^{N} R_i^T K_i R_i \tag{4}$$

Preconditioning at both the substructure and global levels is closely linked to solutions of constrained energy minimization problems. Let ϕ_i^j denote the solution to the problem of minimizing E_i subject to the constraints

$$C_i u_i = e_j \tag{5}$$

where C_i is a constraint matrix and e_j is column j of the identity matrix. Each row of C_i is associated with a coarse dof common to two or more substructures. More details on the matrix C_i are presented later. Define

$$\Phi_i = \left[\begin{array}{ccc} \phi_i^1 & \cdots & \phi_i^{n_{ci}} \end{array} \right] \tag{6}$$

where n_{ci} is the number of rows in C_i. It follows from Lagrange's method for constrained minimization that

$$\left[\begin{array}{cc} K_i & C_i^T \\ C_i & 0 \end{array} \right] \left[\begin{array}{c} \Phi_i \\ \Lambda_i \end{array} \right] = \left[\begin{array}{c} 0 \\ I \end{array} \right] \tag{7}$$

where Λ_i is a matrix of Lagrange multipliers and I is the identity matrix. Let u_{ci} denote a vector of length n_{ci} of coarse dofs for Ω_i. The vector u_{ci} is related to the global vector of coarse dofs u_c by

$$u_{ci} = R_{ci} u_c \tag{8}$$

where each row of R_{ci} contains exactly one nonzero entry of unity.

Let u_{Ii} denote a vector which contains all dofs in Ω_i that are not shared with any other substructures. In other words, u_{Ii} contains all dofs internal to Ω_i. The vector u_{Ii} is related to u_i by the equation

$$u_{Ii} = R_{Ii} u_i \tag{9}$$

where each row of R_{Ii} contains exactly one nonzero entry of unity. Note that the sparse matrices R_i, R_{ci} and R_{Ii} are all used simply for bookkeeping purposes and never actually formed.

Returning now to the constraint matrices, each row of C_i is associated with a particular set of nodes on the boundary of Ω_i. These sets are classified as either corners or edges. Corners consist of single nodes and are chosen as follows. The first corner $c_1^{ij} \in \mathcal{N}_{ij}$ is chosen as a node shared by the largest number of substructures where $\mathcal{N}_{ij}$ is the set of nodes shared by Ω_i and Ω_j. The second corner $c_2^{ij} \in \mathcal{N}_{ij}$ is chosen as a node with greatest distance from c_1^{ij}. For problems in three dimensions, a third corner $c_3^{ij} \in \mathcal{N}_{ij}$ is chosen as a node for which the area of the triangle connecting

c_1^{ij}, c_2^{ij} and c_3^{ij} is maximized. If the angle between the line segments (c_1^{ij}, c_2^{ij}) and (c_1^{ij}, c_3^{ij}) is less than say 0.01 radians, then c_3^{ij} is no longer considered a corner. This approach is repeated for all i and j to obtain the set of corners. Such an approach is nearly identical to one described by a colleague (see Acknowledgments) and used in the parallel structural dynamics code Salinas [10]. Salinas was developed at Sandia National Laboratories and currently uses a FETI-DP implementation for its solver.

Let S_i denote the set of all nodes on the boundary of Ω_i excluding corners. The set S_i is partitioned into disjoint subsets called edges via the following equivalence relation. Two nodes are related to each other if the substructures containing the two nodes are identical. In other words, each node of S_i is contained in exactly one edge, and all nodes of a given edge are contained in exactly the same set of substructures. Using all such edges can lead to a large number of rows in C_i for certain problems. Thus, it may be useful to consider only a single edge associated with each $\mathcal{N}_{ij}$. The edge $\mathcal{E}_{ij} \subset \mathcal{N}_{ij}$ associated with $\mathcal{N}_{ij}$ in this study is one with the largest number of nodes.

The sets of corners and edges for all substructures are collectively grouped into $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_{N_n}\}$. Contributions from the corners to $\mathcal{M}$ are always included in this study, but those from the edges may or may not. Let u_{ik} denote the dof in row k of u_i. For the matrix C_i, the entry in column k of the row for component p of $\mathcal{M}_j$ is given by

$$c_{ikjp} = \begin{cases} s_{ik} & \text{if} \quad n(u_{ik}) \in \mathcal{M}_j \quad \text{and} \quad c(u_{ik}) = p \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $n(u_{ik})$ and $c(u_{ik})$ are the node and component numbers of dof u_{ik} and p is an integer typically between 1 and 6. The scalar s_{ik} is the sum of all diagonal entries of the global stiffness matrix K associated with $n(u_{ik})$. The rows of C_i are then scaled so that the sum of entries in each row equals unity.

In order to distribute residuals to the substructures, it is necessary to define weights for each substructure dof. There are two cases to consider. If u_{ik} is not involved in any constraints, i.e. column k of C_i is zero, then the weight for u_{ik} is given by

$$w_{ik} = s_{ik}^i / s_{ik} \tag{11}$$

where s_{ik}^i is the sum of all diagonal entries of the substructure stiffness matrix K_i associated with $n(u_{ik})$. Define the substructure coarse stiffness matrix K_{ci} as

$$K_{ci} = \Phi_i^T K_i \Phi_i \tag{12}$$

If there is a nonzero entry in column k of C_i, then define s_{ik}^{ci} as the sum of all diagonal entries of K_{ci} associated with $n(u_{ik})$. The global counterpart, s_{ik}^c, is defined similarly

where diagonal entries of the global coarse stiffness matrix K_c, see (17) below, are used instead. In this case, the weight for u_{ik} is given by

$$w_{ik} = s_{ik}^{ci}/s_{ik}^{c} \tag{13}$$

The weights w_{ik} are used to form the diagonal substructure weight matrix W_i defined as

$$W_i = \mathrm{diag}(w_{i1}, \ldots, w_{in_i}) \tag{14}$$

where n_i is the number of rows in u_i. The substructure weight matrices form a partition of unity in the sense that

$$\sum_{i=1}^{N} R_i^T W_i R_i = I \tag{15}$$

Given a residual vector r associated with the iterative solution of (3), the preconditioned residual $M^{-1}r$ is obtained using the following algorithm.

1. Calculate the coarse grid correction v_1:

$$v_1 = \sum_{i=1}^{N} R_i^T W_i \Phi_i R_{ci} K_c^{-1} r_c \tag{16}$$

where

$$K_c = \sum_{i=1}^{N} R_{ci}^T K_{ci} R_{ci} \tag{17}$$

$$r_c = \sum_{i=1}^{N} R_{ci}^T \Phi_i^T W_i R_i r \tag{18}$$

2. Calculate the substructure correction v_2:

$$v_2 = \sum_{i=1}^{N} R_i^T W_i z_i \tag{19}$$

where z_i is obtained from the solution of

$$\begin{bmatrix} K_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} z_i \\ \lambda_i \end{bmatrix} = \begin{bmatrix} W_i R_i r \\ 0 \end{bmatrix} \tag{20}$$

3. Calculate the static condensation correction v_3:

$$v_3 = \sum_{i=1}^{N} R_i^T R_{Ii}^T (R_{Ii} K_i R_{Ii}^T)^{-1} R_{Ii} R_i r_1 \tag{21}$$

where

$$r_1 = r - K(v_1 + v_2) \tag{22}$$

4. Calculate the preconditioned residual:

$$M^{-1} r = v_1 + v_2 + v_3 \tag{23}$$

Residuals associated with dofs in substructure interiors are removed prior to the first iteration via a static condensation correction. These residuals then remain zero for all subsequent iterations.

The preconditioner M looks much like other two-level additive Schwarz preconditioners, but there is an important distinction. Notice that v_1 is not a Galerkin or variational coarse grid correction. That is, v_1 does not equal $\Phi(\Phi^T K \Phi)^{-1} \Phi^T r$ for some interpolation matrix Φ. A Galerkin coarse grid correction could be used, but it leads to more coupling in K_c and increased complexity for code implementations. It is important to note from (17) that two coarse dofs are coupled in K_c only if both dofs appear together in at least one substructure. There is greater coupling between dofs in the coarse problem for BDD than the present approach. This is true because coarse dofs associated with two different substructures can be coupled in BDD even if they are not directly adjacent to one another. We also note that K_c is always at least positive semidefinite. This is not the case for the coarse problem coefficient matrix of FETI-DP if an augmented coarse problem is used [6].

2.3 Implementation Details

Notice from (7) and (20) the need to solve indefinite systems of equations. In order to make use of existing sparse solvers for definite systems, it is useful to consider the following partitions of the vectors z_i and $W_i R_i r$.

$$z_i = \begin{bmatrix} z_{ic} \\ z_{ir} \end{bmatrix}, \quad W_i R_i r = \begin{bmatrix} r_{ic} \\ r_{ir} \end{bmatrix} \tag{24}$$

where the subscript c denotes dofs associated with corners and the subscript r denotes the complement. Equation (20) is then rewritten as

$$\begin{bmatrix} K_{cc} & K_{cr} & I & 0 \\ K_{rc} & K_{rr} & 0 & C_r^T \\ I & 0 & 0 & 0 \\ 0 & C_r & 0 & 0 \end{bmatrix} \begin{bmatrix} z_c \\ z_r \\ \lambda_c \\ \lambda_r \end{bmatrix} = \begin{bmatrix} r_c \\ r_r \\ 0 \\ 0 \end{bmatrix} \tag{25}$$

where all subscript i's are dropped for notational convenience. Solving (25) for z_r and back substituting the result leads to

$$z_r = K_{rr}^{-1}(r_r - C_r^T \lambda_r) \tag{26}$$

and

$$(C_r K_{rr}^{-1} C_r^T)\lambda_r = C_r K_{rr}^{-1} r_r \tag{27}$$

The order of the reduced linear system (27) is only the number of edge constraints used in C_i. As such, it can be factored effectively using software from the LAPACK library [11]. Provided the corner dofs remove any singularities, matrix vector products of the form $K_{rr}^{-1}x$ can be obtained using a sparse matrix solver for definite systems of equations.

2.4 Multilevel Extension

The two-level preconditioner described in this section requires solutions of equations at both the local (substructure) and global (coarse problem) levels. Difficulties will arise for very large problems if either K_c or the coefficient matrix in (20) becomes too large for direct factorization. If K_c is too large, then one can apply the preconditioner recursively to obtain approximate solutions of (16). The primary reason for this option is made evident by comparing (4) and (17). Notice that the stiffness matrix K_c for the coarse problem is identical in form to the stiffness matrix for the original problem. That is, both are obtained by assembling stiffness matrices of "elements". In the case of K_c, the substructure coarse stiffness matrices K_{ci} are assembled. Similarly, the node sets described earlier play the role of coarse nodes. Thus, one can construct a preconditioner M_c for K_c just like M is constructed for K. Given a coarse problem residual r_c, $K_c^{-1}r_c$ in (16) can simply be replaced by $M_c^{-1}r_c$. Such an approach has shown promising initial results and will be investigated further in another study. It is unclear if a similar option is available for the current formulation of FETI-DP, but some work has been done with approximate solution techniques for a related domain decomposition method [12].

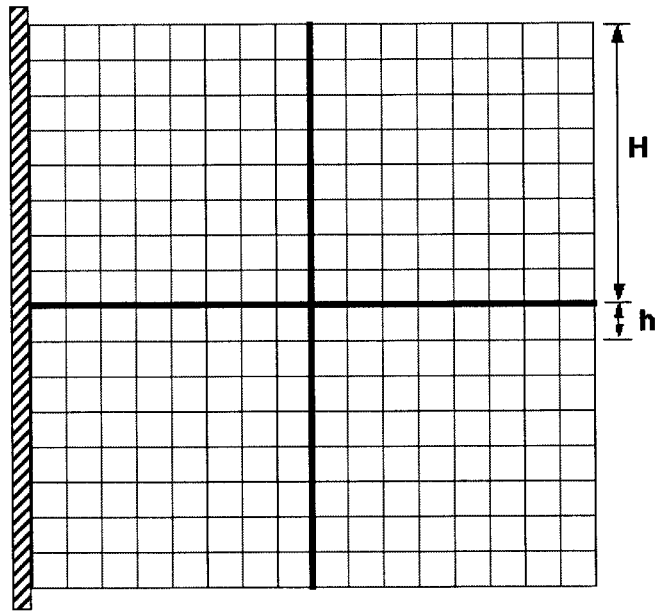

Figure 2.1. Mesh used for 2D scalability studies. In the figure $h = 1/16$ and $H = 1/2$.

2.5 Numerical Examples

The first set of examples demonstrates the numerical scalability of the preconditioner with respect to the number of substructures and the number of elements per substructure. Consider a plane stress problem for a square of unit length with all dofs on the left side constrained to zero. The domain is decomposed into $(1/H)^2$ square substructures each containing $(H/h)^2$ square quadrilateral elements (see Figure 2.1). Thus, the length of each substructure and element equals H and h, respectively. The elastic modulus is set to 30×10^6 and Poisson's ratio to 0.3. Unit forces are applied to all nodes on the right edge of the mesh in the horizontal direction. Results for including different node sets in $\mathcal{M}$ are designated by C for corners only and CE for corners and edges. The number of equations in the coarse problem is denoted by N_c.

The number of preconditioned conjugate gradient iterations needed to achieve a relative residual tolerance $||r||_2/||f||_2$ of 10^{-6} are shown in Table 2.1 for increasing numbers of substructures. Also shown in the table are condition number estimates obtained using the connection between conjugates gradients and the Lanczos

Table 2.1. Iterations (iter), condition number estimates (κ), and number of coarse problem equations (N_c) for 2D problems with increasing numbers of substructures (N) and $H/h = 8$.

	plane stress						plate bending					
	C			CE			C			CE		
N	iter	κ	N_c	iter	κ	N_c	iter	κ	N_c	iter	κ	N_c
16	14	5.3	36	8	2.4	84	21	6.3	54	12	2.7	126
64	17	5.9	140	10	2.7	364	25	6.6	210	12	2.7	546
144	18	6.0	308	10	2.8	836	26	6.7	462	12	2.7	1254
256	18	6.1	540	10	2.8	1500	26	6.7	810	12	2.7	2250
400	18	6.1	836	10	2.8	2356	27	6.8	1254	12	2.7	3534

method [13]. The condition number estimates are for the preconditioned matrix $M^{-1/2}KM^{-1/2}$ (see (3) and (23)). The estimates are lower bounds on the actual condition numbers and were obtained from the extremal eigenvalues of a tridiagonal matrix of dimension equal to the number of iterations. Notice that the number of iterations and condition number estimates grow very slowly as the number of substructures increases. Notice also that better performance is obtained if both corners and edges are included, but the size of the coarse problem is also larger.

Results for a fixed number of substructures ($N = 16$) and increasing values of H/h are shown in Table 2.2. Notice that the number of iterations and condition numbers grow slowly with the number of elements per substructure. Results for a plate bending problem with the same geometry, material properties, and a thickness of 0.01 are also shown in Tables 2.1 and 2.2. Results were obtained using discrete Kirchoff triangular elements with unit forces applied to all nodes on the right edge of the mesh in the out of plane direction. Notice in the two tables that the same trends are apparent for both the plane stress (second-order) and plate bending (fourth-order) problems.

To study the effects of material properties jumps, consider Figure 2.2 where the elastic modulus $E = 1$ and $\nu = 0.3$ throughout the square domain except in the center region $[1/4, 3/4] \times [1/4, 3/4]$ where $E = \sigma$ and $\nu = 0.3$. The boundary conditions are the same as the ones in the previous examples. Results for 16 substructures with $H/h = 6$ are shown in Table 2.3 for different values of σ. Results for an analogous Laplace equation problem with the same material property jumps are also reported. In Table 2.3 each substructure contains a single material. That is, the substructures are aligned with material interfaces. Notice that the number of iterations and

Table 2.2. Iterations and condition number estimates for 2D problems with 16 substructures ($H = 1/4$) and increasing numbers of elements per substructure.

	plane stress				plate bending				Laplace equation			
	C		CE		C		CE		C		CE	
H/h	iter	κ	iter	κ	iter	κ	iter	κ	iter	κ	iter	κ
4	12	3.7	6	1.6	17	4.2	9	1.8	9	2.2	4	1.1
8	14	5.3	8	2.4	21	6.3	12	2.7	10	3.0	5	1.3
16	16	7.2	10	3.4	25	8.8	15	3.6	12	3.8	6	1.5
32	19	9.5	11	4.7	30	12	18	4.8	13	4.8	7	1.7
64	22	12	13	6.1	34	15	21	6.1	14	5.9	8	2.0

Table 2.3. Iterations and condition number estimates for 2D problems with material property jumps. The number of substructures is 16 ($H = 1/4$) and $H/h = 6$. Material property jumps are aligned with substructure boundaries.

	plane stress				plate bending				Laplace equation			
	C		CE		C		CE		C		CE	
σ	iter	κ	iter	κ	iter	κ	iter	κ	iter	κ	iter	κ
10^{-3}	13	4.5	8	1.8	19	5.2	10	2.1	9	2.4	5	1.1
10^{-2}	13	4.5	8	1.7	19	5.2	10	2.1	9	2.4	5	1.1
1	13	4.6	7	2.0	20	5.4	10	2.3	10	2.6	5	1.2
10^2	14	4.3	8	2.2	18	6.0	11	2.3	10	2.6	5	1.2
10^3	14	4.3	8	2.2	19	6.1	12	2.3	10	2.6	5	1.2

condition numbers remain bounded independently of σ for all three problem types.

Results for the same mesh ($h = 1/24$) with 9 substructures ($H = 1/3$) are shown in Table 2.4. Here the substructure boundaries do not align with the material interfaces. Notice for both the plate bending and Laplace equation problems that the number of iterations and condition numbers remain bounded independently of σ whether or not edges are included in $\mathcal{M}$. In contrast, the condition numbers continue to increase with larger values of σ for the plane stress problems if only corners are used. Notice that the number of iterations increases by less than a factor of two when σ increases from 1 to 10^4 while there appears to be a near linear relationship between the condition number and σ for larger values of σ.

Similar results for three-dimensional problems are shown in Tables 2.5 through 2.8 where fully-integrated 8-node hexahedral elements were used. Results from the

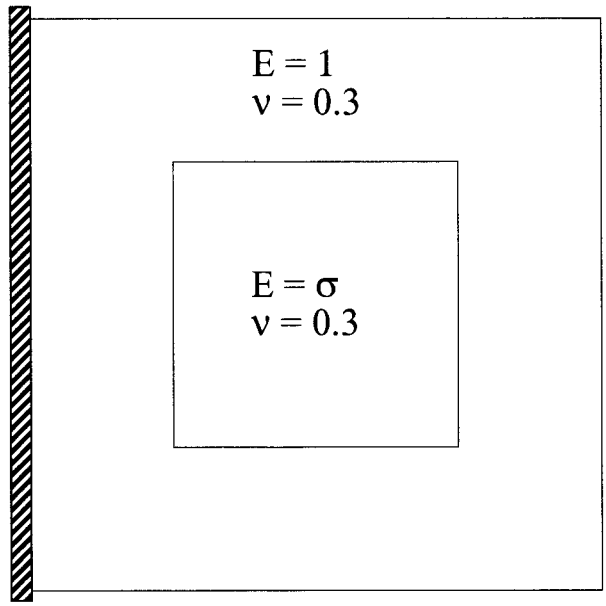

Figure 2.2. Geometry and boundary conditions for 2D problems with material property jumps.

Table 2.4. Iterations and condition number estimates for 2D problems with material property jumps. The number of substructures is 9 ($H = 1/3$) and $H/h = 8$. Material property jumps are not aligned with substructure boundaries.

	plane stress				plate bending				Laplace equation			
	C		CE		C		CE		C		CE	
σ	iter	κ	iter	κ	iter	κ	iter	κ	iter	κ	iter	κ
10^{-3}	14	6.3e2	8	2.8	24	27	16	5.2	7	3.5	4	1.2
10^{-2}	14	66	8	2.7	24	15	16	4.9	8	3.4	4	1.2
1	12	4.8	7	1.7	18	6.1	11	2.5	8	2.8	5	1.2
10^{2}	17	27	10	2.1	19	5.9	13	2.4	8	2.4	6	1.2
10^{3}	18	2.6e2	10	2.1	19	5.9	13	2.4	8	2.4	6	1.2
10^{4}	18	2.5e3	11	2.1	20	5.9	13	2.4	8	2.4	6	1.2

Table 2.5. Iterations, condition number estimates, and number of coarse problem equations for 3D elasticity problems with increasing numbers of substructures (N) and $H/h = 4$.

	C			CE		
N	iter	κ	N_c	iter	κ	N_c
64	27	18	288	9	2.2	1044
216	31	19	870	9	2.2	3840
512	32	19	1932	9	2.1	9492
1000	32	19	3618	9	2.1	19008

Table 2.6. Iterations and condition number estimates for 3D elasticity problems problems with 64 substructures ($H = 1/4$) and increasing numbers of elements per substructure.

	present approach				FETI-DP (Salinas)	
	C ($N_c = 288$)		CE ($N_c = 1044$)		C ($N_c = 288$)	ACP ($N_c = 1368$)
H/h	iter	κ	iter	κ	iter	iter
4	27	18	9	2.2	29	9
8	46	53	13	4.1	49	13
12	61	96	15	5.6	65	15
16	66	144	16	6.9	68	17

FETI-DP implementation used in Salinas are also shown in Tables 2.6 through 2.8 where the designation ACP means that an augmented coarse problem was used. The coefficient matrix for the augmented coarse problem of FETI-DP is indefinite. In contrast, the matrix K_c is positive definite for both cases C and CE.

Results in Table 2.5 suggest that the preconditioner scales well with respect to the number of substructures for 3D problems. Results in Table 2.6 suggest that much better scalability with respect to the number of elements per substructure is obtained if both corners and edges are included. Likewise, better performance is observed for FETI-DP by including an augmented coarse problem. Results for problems with material property jumps are shown in Tables 2.7 and 2.8. In this case the entire domain has $E = 1$ and $\nu = 0.3$ except for the center region $[1/4, 3/4] \times [1/4, 3/4] \times [1/4, 3/4]$ where $E = \sigma$ and $\nu = 0.3$. If material property and substructure boundaries are aligned, then good performance is obtained whether or not edges are included as shown in Table 2.7. Results in Table 2.8 suggest that much better performance can

Table 2.7. Iterations and condition number estimates for 3D elasticity problems with material property jumps. The number of substructures is 64 ($H = 1/4$) and $H/h = 6$. Material property jumps are aligned with substructure boundaries.

	present approach				FETI-DP (Salinas)	
	C ($N_c = 288$)		CE ($N_c = 1044$)		C ($N_c = 288$)	ACP ($N_c = 1368$)
σ	iter	κ	iter	κ	iter	iter
10^{-3}	36	33	12	3.3	38	14
10^{-2}	37	33	12	3.3	38	14
1	37	34	11	3.2	40	11
10^{2}	39	37	12	2.7	42	12
10^{3}	41	38	12	2.7	43	13

Table 2.8. Iterations and condition number estimates for 3D elasticity problems with material property jumps. The number of substructures is 27 ($H = 1/3$) and $H/h = 8$. Material property jumps are not aligned with substructure boundaries.

	present approach				FETI-DP (Salinas)	
	C ($N_c = 132$)		CE ($N_c = 402$)		C ($N_c = 132$)	ACP ($N_c = 510$)
σ	iter	κ	iter	κ	iter	iter
10^{-3}	37	156	11	5.3	40	31
10^{-2}	37	146	11	5.1	40	19
1	31	48	10	3.1	32	11
10^{2}	47	89	16	6.8	49	27
10^{3}	74	7.0e2	18	10	76	51
10^{4}	78	6.8e3	20	11	78	60

Table 2.9. Results for 3D elasticity problems with $H/h = 22$. Each substructure is assigned to one processor and the designation CLIP is for the present approach.

	Salinas CLIP		Salinas FETI-DP	
N	iter	time (sec)	iter	time(sec)
64	52	86	58	94
216	62	99	75	110
512	65	118	77	126
1000	68	123	79	133

be obtained by including both corners and edges when large material property jumps not aligned with substructure boundaries are present. Greater sensitivity to material property jumps is apparent in Table 2.8 for FETI-DP with an augmented coarse problem (ACP) compared to CE of the present approach. Although not shown, preliminary results obtained by suitable scaling of the augmented coarse problem constraint equations removed much of the sensitivity of FETI-DP results for this problem.

Results for much larger 3D elasticity problems with $H/h = 22$ and a relative residual tolerance of 10^{-3} are shown in Table 2.9. The problems were run on the Department of Energy's Accelerated Strategic Computing Initiative option Q supercomputer. The 1000 substructure problem has over 32 million dofs. Notice that the numbers of iterations and compute times for a Salinas implementation of the present approach, designated CLIP, are comparable to those for the FETI-DP implementation. The results in Tables 2.9-2.11 are for using corner constraints only. In this case, the primary computational and memory requirements are very similar for CLIP and FETI-DP. In particular, both require factorizations of the matrices K_c, $R_{Ii}K_iR_{Ii}^T$, and K_{rr} (see (17), (21), (26)). Thus, large performance differences are not expected.

The final two examples are for unstructured meshes. One is for the diffraction grating model shown in Figure 2.3 and described in [6]. This model consists of 35328 eight-node hexahedral elements, 40329 nodes, and a single material with $\nu = 0.17$. All dofs of six selected nodes are constrained and unit loads are applied to each node in the vertical direction. The second example is for the joint leg model shown in Figure 2.4. This model consists of 269852 ten-node tetrahedral elements, 374281 nodes, and a single material with $\nu = 0.3$. All dofs of nodes on the bottom surface are constrained and each node is subjected to a unit load in the vertical direction. For both examples the relative residual tolerance is 10^{-6}. Mesh decompositions into substructures were obtained using a code based on the graph partitioning software Chaco [14]. Results

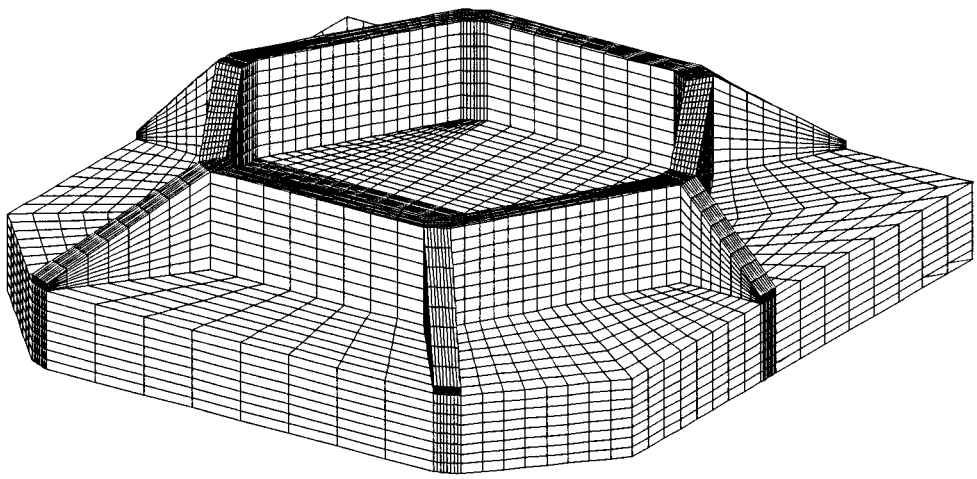

Figure 2.3. Diffraction grating model with 35328 eight-node hexahedral elements and 40329 nodes.

for these two problems are shown in Tables 2.10 and 2.11. Although the numbers of iterations are generally larger than those for the structured meshes, they remain quite reasonable. Comparable performance of the present approach and FETI-DP is evident for these two problems.

2.6 Conclusions

The results of this study are encouraging. The preconditioner has the attractive feature of exploiting the good numerical stability and efficiency of existing sparse solvers for symmetric definite matrices. The preconditioner was observed to have very good numerical scalability with respect to the number of substructures in structured mesh studies. Good scalability with respect to the number of elements per substructure was also observed, but three-dimensional second-order problems required that both corners and edges be included in the constraints. These observations are consistent with those for other approaches like FETI-DP [6] and Schur complement methods [8]. The performance of the preconditioner was observed to be fairly insensitive to large material property jumps provided the material interfaces were aligned with

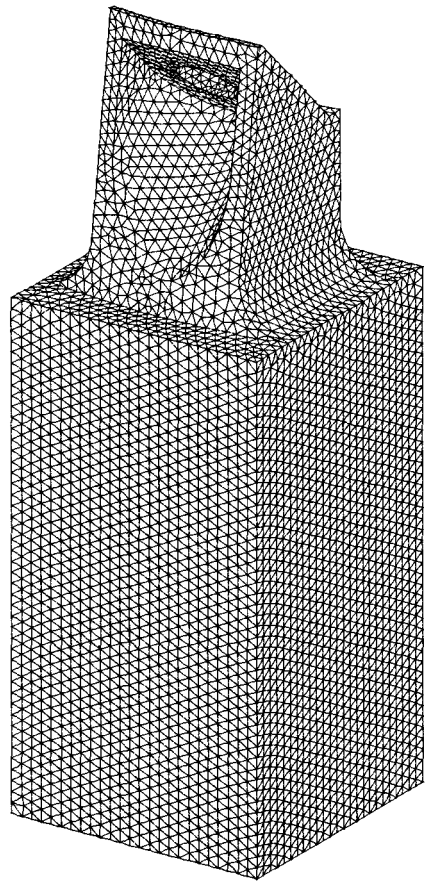

Figure 2.4. Joint leg model with 269852 ten-node tetrahedral elements and 374281 nodes.

Table 2.10. Results for diffraction grating model.

	Salinas CLIP			Salinas FETI-DP		
N	iter	time (sec)	N_c	iter	time (sec)	N_c
8	52	23	81	53	22	81
16	75	10	198	72	9	198
32	78	4	504	76	3	504
64	78	2	1008	75	2	1008

Table 2.11. Results for joint leg model.

	Salinas CLIP			Salinas FETI-DP		
N	iter	time (sec)	N_c	iter	time (sec)	N_c
120	70	21	2745	74	20	2760
140	62	16	3381	66	14	3411
201	62	15	4662	66	13	4710
250	57	13	5886	60	11	5934

substructure boundaries. Results for second-order problems suggest that better performance is obtained by including both corners and edges when such alignment does not occur. Results from more realistic unstructured meshes demonstrated that the preconditioner is competitive with an existing approach.

Some remaining issues need to be addressed for improvement. First, it would be useful to have an effective method for selecting additional corners and edges to improve performance for very poorly conditioned problems. Second, the performance of the multilevel extension should be investigated further. Recall that the multilevel extension is obtained by recursive application of the preconditioner to coarse problem stiffness matrices. Such an extension would be beneficial for problems with very large numbers of substructures. Third, it would be useful to extend the method to preconditioning of mixed formulations of elasticity. Such an extension would be useful for elasticity problems with nearly incompressible materials or for related Stokes problems.

2.7 References

[1] J. S. Przemieniecki. *Theory of Matrix Structural Analysis.* Dover Publications, Inc., New York, 1985. Reprint of McGraw Hill, 1968.

[2] J. Mandel. Balancing domain decomposition. *Comm. Numer. Meth. Engrg.*, 9:233–241, 1993.

[3] P. Le Tallec, J. Mandel, and M. Vidrascu. Balancing domain decomposition for plates. *Contemp. Math.*, 180:515–524, 1994.

[4] C. Farhat and F. X. Roux. A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm. *Internat. J. Numer. Methods Engrg.*, 32:1205–1227, 1991.

[5] 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. *Comput. Methods Appl. Mech. Engrg.*, 155:129–151, 1998.

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

[7] P. Le Tallec, J. Mandel, and M. Vidrascu. A neumann-neumann domain decomposition algorithm for solving plate and shell problems. *SIAM J. Numer. Anal.*, 35:836–867, 1998.

[8] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, New York, 1996.

[9] J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Lin. Alg. Appl.*, 10:639–659, 2003.

[10] G. Reese, D. Segalman, M. Bhardwaj, K. Alvin, B. Driessen, K. Pierson, and T. Walsh. *Salinas - User's Notes.* Sandia National Laboratories, 2003.

[11] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide.* Society for Industrial and Applied Mathematics, Philadelphia, Philadelphia, 1992.

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

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins Univ. Press, Baltimore, MD, 1989. Second Edition.

[14] B. Hendrickson and R. Leland. The chaco user's guide version 2.0. Technical report, SAND95-2344, Sandia National Laboratories, Albuquerque, NM, 1995.

3 An Algebraic Coarse Space for Domain Decomposition Methods

Abstract

An algebraic coarse space for domain decomposition methods is presented. The space is constructed using a partition of unity that is based on solutions of problems local to each subdomain. Unique to the method is the construction of the partition and the treatment of rotational degrees of freedom for fourth-order plate and shell problems. The coarse space is used in the formation of a two-level overlapping Schwarz preconditioner. Numerical examples demonstrate good performance of the preconditioner independent of the number of subdomains and the number of elements per subdomain provided the overlap is proportional to subdomain size.

3.1 Introduction

The primary motivation for this study is the need for simple and effective coarse problems in two-level domain decomposition preconditioners [1]. Without an effective coarse problem, the number of iterations needed to approximately solve a system of equations may grow as the number of subdomains increases. Sometimes an auxiliary coarse finite element mesh can be used to provide a coarse problem. Often, however, this is not possible and one must appeal to an alternative "algebraic" approach. Some possible options in this regard include smoothed aggregation [2, 3], balancing domain decomposition [4], and partition of unity [5] approaches.

Of the approaches mentioned above, the proposed one is most closely related to [5]. Like the cited approach, the coarse space is obtained using a partition of unity. The distinguishing features of the present work are the algebraic construction of the partition and the treatment of rotational degrees of freedom for fourth-order plate and shell problems. Recall for a one-level overlapping Schwarz preconditioner the requirement to solve, either exactly or approximately, a series of problems local to each subdomain. Solutions of these problems are then combined, either in an additive or multiplicative manner, to construct the preconditioner. The same types of calculations needed for solutions of the subdomain problems are used here in the construction of the partition of unity. Thus, one is able to take advantage of existing solution methods for the Schwarz preconditioner to construct the coarse space.

For smoothed aggregation approaches, one starts with a tentative prolongator

matrix $\hat{P}$ whose columns span the null space of the system operator. The final prolongator P is then obtained via a smoothing process that reduces the energy associated with each column of $\hat{P}$. This smoothing process typically leads to more nonzeros in P than $\hat{P}$ and requires matrix norm estimates. For the present approach, a similar reduction in energy is obtained directly by multiplying tentative prolongator like quantities by a partition of unity. A large part of the present approach can be viewed simply as an algebraic method for constructing a "smooth" partition of unity. Compared with the coarse space for balancing domain decomposition, the present one leads to fewer nonzeros in the coarse problem matrix.

The method is described in the following section. Corrections to the coarse space needed for fourth-order structural mechanics problems are also described. Numerical examples are provided in the third section where the coarse space is used in a two-level overlapping Schwarz preconditioner. The examples demonstrate the good numerical properties of the preconditioner.

3.2 Coarse Space

To begin, consider a non-overlapping decomposition of a finite element mesh into subdomains $\Omega_1, \ldots, \Omega_N$ such that each element is contained in exactly one subdomain. Overlapping subdomain Ω_i^p is obtained by including the p rows of elements directly adjacent to Ω_i. That is, Ω_i^1 contains all elements with at least one node in Ω_i. Likewise, Ω_i^2 contains all elements with at least one node in Ω_i^1, and so on. A degree of freedom is in the set $\mathcal{S}_i^p$ if at least one element in Ω_i^p contains this dof. The boundary set $\bar{\mathcal{S}}_i^p$ consists of all dofs in $\mathcal{S}_i^{p+1}$ that are not in $\mathcal{S}_i^p$. The subdomain dof vector u_i contains all dofs in $\mathcal{S}_i^p$ and can be expressed in terms of the global dof vector u by the equation

$$u_i = R_i u \tag{1}$$

where each row of R_i has exactly one nonzero entry of unity. Similarly, the stiffness matrix associated with Ω_i^p is expressed as

$$A_i = R_i A R_i^T \tag{2}$$

where A is the stiffness matrix for the entire problem and superscript T denotes matrix transpose. It is assumed in this study that A is symmetric and positive definite. Recall for a one-level, additive Schwarz preconditioner [1] that the preconditioned residual $s(r)$ can be expressed as

$$s(r) = \sum_{i=1}^{N} R_i^T M_i^{-1} R_i r \tag{3}$$

where r is the residual and $M_i^{-1}R_i r$ is either an exact or approximate solution of the equation $A_i x_i = R_i r$. For example, the action of M_i^{-1} on a vector could be obtained using a multigrid preconditioner, an incomplete factorization, or a sparse approximate inverse. Attention is restricted in this study to the case where M_i^{-1} equals A_i^{-1}.

The coarse space approximation of u is expressed in terms of a sum over all overlapping subdomains by the equation

$$u_c = \sum_{i=1}^{N} R_i^T \Phi_i c_i \tag{4}$$

where Φ_i is a subdomain interpolation matrix and c_i is a vector of coarse dofs for Ω_i^p.

An outline of the overall process to obtain Φ_i is now given. To begin, a matrix F_i is constructed based on the nodal coordinates of dofs in u_i. The goal here is for the columns of F_i to provide a force basis to generate a low-energy displacement basis represented by the matrix Ψ_i. An approximate inverse of A_i is obtained from Ψ_i and its diagonal entries are used to construct the partition of unity. The partition of unity is then used to construct Φ_i by scaling the null space of the differential operator. In the case of plate bending and shell problems, the quality of Φ_i is improved using a simple procedure. Finally, a correction based on energy minimization concepts is made to Φ_i.

Let

$$f_i = F_i c_i \tag{5}$$

where there is a one-to-one correspondence between the rows of f_i and u_i. Recall that the columns of F_i provide a force basis used to generate a low-energy displacement basis. The columns of F_i can be chosen in several different ways. One option is to use the eigenvectors associated with the smallest of eigenvalues of A_i. Another option is to associate each column of F_i with a simple loading condition such as a spatially varying body force. In order to avoid the potential cost of the first option and the bookkeeping requirements of the second, the following procedure is adopted. Let $x^{j,m}$ denote the spatial coordinate in direction m of the node associated with row j of u. For scalar problems like the Laplace equation, the entry in row j and column k of F is given by

$$F^{j,k} = x^{j,k-1} \tag{6}$$

for $k = 1, \ldots, d+1$ where d is the spatial dimension and $x^{j,0} \equiv 1$. In the case of elasticity and 3D shell problems,

$$F^{j,d(m-1)+k} = \delta^{j,m} x^{j,k-1} \tag{7}$$

for $k = 1, \ldots, d+1$ and $m = 1, \ldots, d$. The scalar $\delta^{j,m} = 1$ if the dof associated with row j of u is in direction m and $\delta^{j,m} = 0$ otherwise. For 2D plate problems with three dofs per node, m varies from 1 to 3. The matrix F_i is given by

$$F_i = R_i F \tag{8}$$

The total potential energy of Ω_i^p is defined as

$$W_i = u_i^T A_i u_i / 2 - u_i^T f_i \tag{9}$$

Substitution of (5) into (9) and minimization of W_i with respect to u_i leads to

$$A_i u_i = F_i c_i \tag{10}$$

An approximate solution of (10) consistent with the Schwarz preconditioner is given by

$$u_i = \Psi_i c_i \tag{11}$$

where

$$\Psi_i = M_i^{-1} F_i \tag{12}$$

Note if an exact factorization of A_i is available, then an efficient and more direct route to constructing Ψ_i is to simply use a subset of the eigenvectors associated with the smallest eigenvalues of the eigenproblem $A_i \Psi_i = G_i \Psi_i \Lambda$ where G_i is the mass matrix, if available, or the identity matrix. In this case, the matrix F_i need not be constructed at all.

If f_i is no longer constrained by (5) and W_i is minimized subject to the constraint (11), one then obtains

$$u_i = B_i f_i \tag{13}$$

where

$$B_i = \Psi_i (\Psi_i^T A_i \Psi_i)^{-1} \Psi_i^T \tag{14}$$

If for some reason $\Psi_i^T A_i \Psi_i$ is singular, e.g. columns of Ψ_i may be linearly dependent for very small subdomains, then its inverse can be replaced in practice by a pseudo-inverse since the matrix dimensions are small. The matrix B_i can be viewed as an approximate inverse of A_i since $B_i F_i = A_i^{-1} F_i$ if $M_i^{-1} = A_i^{-1}$.

Define

$$P_i = \mathrm{diag}(B_i)(R_i B^{-1} R_i^T) \tag{15}$$

where

$$B = \sum_{i=1}^{N} R_i^T \mathrm{diag}(B_i) R_i \tag{16}$$

One can confirm that $P_1, \ldots, P_N$ forms a partition of unity. That is,

$$\sum_{i=1}^{N} R_i^T P_i R_i = I \tag{17}$$

where I is the identity matrix.

Given a particular problem type such as linear elasticity, the columns of the null space matrix N_e span the associated null space of A in the absence of any Dirichlet boundary conditions. For the scalar Laplace equation, N_e is a vector of the same length as u with all its entries equal to unity. For 3D elasticity problems that may include shell elements, the entries in row j of N_e are given by

$$N_e^j = d^{jT} C^j \tag{18}$$

where the 6 by 1 vector d^j has a single nonzero entry of unity in the row corresponding to the local dof number associated with row j of u and

$$C^j = \begin{bmatrix} I & S^j \\ 0 & I \end{bmatrix}, \qquad S^j = \begin{bmatrix} 0 & x^{j,3} & -x^{j,2} \\ -x^{j,3} & 0 & x^{j,1} \\ x^{j,2} & -x^{j,1} & 0 \end{bmatrix} \tag{19}$$

For example, $d^{jT} = [0\,1\,0\,0\,0\,0]$ if the dof associated with row j of u is a translation in the 2-direction. Similarly, $d^{jT} = [0\,0\,0\,0\,0\,1]$ if the dof associated with row j of u is a rotation about the 3-direction. The matrix N_e for plate bending problems is constructed in a similar manner, but details are omitted here for brevity.

Let x_i^m denote the average coordinate in direction m associated with all dofs in u_i. The subdomain interpolation matrix Φ_i is obtained by scaling N_e by the partition of unity according to

$$\Phi_i = P_i R_i N_e C_i \tag{20}$$

where C_i equals C^j in (19) with $-x_i^m$ replacing $x^{j,m}$. The primary reason for including C_i in (20) is so that the average translation associated with columns 4 through 6 of Φ_i is zero. One can confirm that the standard zero energy modes for the unconstrained global system can be represented exactly by (4). In particular,

$$N_e = \sum_{i=1}^{N} R_i^T \Phi_i \bar{C}_i \tag{21}$$

where $\bar{C}_i$ equals C^j in (19) with x_i^m replacing $x^{j,m}$.

The quality of the matrix Φ_i can be improved using energy minimization concepts. To this end, let

$$\Phi_i = R_{ix}^T \Phi_{ix} + R_{if}^T \Phi_{if} \tag{22}$$

where each row of R_{ix} and R_{if} has exactly one nonzero entry of unity. The subscripts x and f in (22) refer to fixed and free dofs, respectively. Let d_i^k denote the dof in row k of u_i. Row k of R_{ix}^T is nonzero either if $d_i^k \in \mathcal{S}_j^p$ only for $j = i$ or $d_i^k \in \bar{\mathcal{S}}_j^p$ for at least one $j \neq i$. In words, this means that either Ω_i^p is the only subdomain containing d_i^k or that d_i^k is on the boundary of at least one other subdomain. Define the energy associated with the matrix Φ_i as

$$E_i = \text{trace}(\Phi_i^T A_i \Phi_i)/2 \tag{23}$$

where trace denotes the sum of diagonal entries. Substitution of (22) into (23) and minimization of E_i with respect to the entries of Φ_{if} leads to

$$\Phi_{if} = -(R_{if} A_i R_{if}^T)^{-1} R_{if} A_i R_{ix}^T \Phi_{ix} \tag{24}$$

In this study, Φ_{if} is calculated using (24) and Φ_i is then updated according to (22). A similar procedure is used in [5] where Φ_{if} is the harmonic extension of Φ_{ix}. The energy of the updated Φ_i, as measured by E_i, will always be less than or equal to that of the unmodified Φ_i. Note that correct use of (24) requires the exact solution of a sparse system of equations of dimension typically much less than that of A_i. If an approximate solution is used instead, then it may not be possible to faithfully represent any column of N_e by (4) without additional corrections.

For elasticity problems without rotational dofs or for scalar problems like the Laplace equation, the procedure just described to obtain Φ_i does not need to be modified. Unfortunately, for problems with rotational dofs, the energy associated with the columns of Φ_i is too large. To help understand why this is the case, note that all rotational dofs are constrained to zero in the first three columns of Φ_i. A state of deformation with nonzero values for the translational dofs and zero values for all rotational dofs can have very high energy. In order to address this issue, let

$$u_i = R_{it}^T u_{it} + R_{ir}^T u_{ir} \tag{25}$$

where u_{it} and u_{ir} contain translational and rotational dofs, respectively. Each row of R_{it} and R_{ir} has exactly one nonzero entry of unity. Minimization of the energy term $u_i^T A_i u_i$ subject to the constraint $u_{it} = R_{it} \Phi_i c_i$ leads to the modified interpolation matrix

$$\tilde{\Phi}_i = R_{it}^T R_{it} \Phi_i + R_{ir}^T \Phi_{ir} \tag{26}$$

where

$$\Phi_{ir} = -(R_{ir} A_i R_{ir}^T)^{-1} R_{ir} A_i R_{it}^T R_{it} \Phi_i \tag{27}$$

Note that the rows of Φ_i associated with translational dofs remain unchanged while those associated with rotational dofs are modified to minimize energy. Note also that

the exact solution implied by (27) can be replaced by an approximate one. Because Φ_i has been modified, it is now necessary to make corrections so that all rigid body modes of the unconstrained global system can be represented exactly by u_c. Details are provided below for problems with 3D shells. A similar procedure applies to 2D plate bending problems.

With reference to (21), define

$$\tilde{N}_e = \sum_{i=1}^{N} R_i^T \tilde{\Phi}_i \bar{C}_i \tag{28}$$

Ideally, $\tilde{N}_e = N_e$ but this will not be the case for rows of N_e corresponding to rotational dofs. In order to address this issue, columns 1 through 3 of $\tilde{\Phi}_i$ are modified first according to

$$\tilde{\Phi}_i \leftarrow \tilde{\Phi}_i + P_i(N_e - \tilde{N}_e) \tag{29}$$

The final three columns of $\tilde{N}_e$ are then calculated using (28) and the final three columns of $\tilde{\Phi}_i$ modified according to (29). This two-step procedure is motivated by the block triangular structure of $\bar{C}_i$ and ensures that $\tilde{N}_e = N_e$. The procedure to calculate the subdomain interpolation matrix Φ_i is summarized below.

1. Depending on the problem type, determine the entries in the matrix F using either (6) or (7).
2. Calculate F_i, Ψ_i and B_i using (8), (12), and (14).
3. Calculate the partition of unity matrix P_i according to (15) and determine the null space matrix N_e (see (18) and surrounding discussion).
4. Calculate Φ_i using (20). For plate bending and shell problems, modify Φ_i according to (29).
5. Calculate Φ_{if} using (24) and update Φ_i according to (22).

Step 5 can be omitted if direct solvers are to be avoided altogether.

The primary differences between the procedure just described and that of [5] are now discussed. First, the partition of unity is constructed in this study using either approximate or exact solutions of local subdomain problems. In contrast, the approach of [5] uses a simpler approach based on the layer of neighboring nodes in which a particular dof appears. Although simpler and easier to implement, such an approach does not account for the size or shape of the elements in the mesh. The

approach of [5] also involves a method to modify the partition to have a controlled decay to zero near the constrained boundary. A similar modification could improve the coarse space for the present approach, but was avoided to simplify implementation on a parallel computer. A final difference is that guidance is not provided in [5] on constructing the coarse space for plate and shell problems.

The final part of this section describes the two-level Schwarz preconditioner used in the numerical examples. With reference to (4), let

$$c_i = E_i c \tag{30}$$

where

$$c = \left[\begin{array}{ccc} c_1^T & \cdots & c_N^T \end{array} \right]^T \tag{31}$$

Substitution of (30) in (4) leads to

$$u_c = \Phi c \tag{32}$$

where

$$\Phi = \left[\begin{array}{ccc} R_1^T \Phi_1 E_1 & \cdots & R_N^T \Phi_N E_N \end{array} \right] \tag{33}$$

The coarse problem correction for a residual r is given by

$$c(r) = \Phi (\Phi^T A \Phi)^{-1} \Phi^T r \tag{34}$$

Given a residual r, the preconditioned residual $z(r)$ for a two-level Schwarz method with multiplicative coarse problem correction and additive subdomain correction is obtained as follows:

1. Calculate $c(r)$ from (34) and set $r_1 = r - Ac(r)$.

2. Calculate $s(r_1)$ from (3) and set $r_2 = r_1 - As(r_1)$.

3. Calculate $c(r_2)$ from (34) and set $z(r) = c(r) + s(r_1) + c(r_2)$.

Additive and approximate coarse problem corrections are also possible, but are not investigated here. Note after the first iteration of conjugate gradients that $c(r) = 0$ in Step 1 since all subsequent residuals are orthogonal to Φ^T.

Table 3.1. Iterations (iter) and condition number estimates (κ) for 2D problems with increasing numbers of subdomains (N) for $H/h = 8$. The overlap parameter p has a fixed value of 1. The number of equations in the coarse problems for the plane stress, plate bending, and Laplace equation problems equals $3N$, $3N$, and N, respectively.

	plane stress		plate bending		Laplace equation	
N	iter	κ	iter	κ	iter	κ
16	15	5.0	38	27	13	4.7
64	16	5.0	43	30	14	4.7
144	16	5.0	47	32	14	4.7
256	16	5.0	50	33	14	4.7
400	16	4.9	51	33	14	4.7

3.3 Numerical Examples

The first set of examples demonstrates the numerical scalability of the preconditioner with respect to the number of subdomains and the number of elements per subdomain. Consider a plane stress problem for a square of unit length with all dofs on the left side constrained to zero (see Figure 2.1). The domain is decomposed into $(1/H)^2$ square subdomains each containing $(H/h)^2$ square 4-node quadrilateral elements. Thus, the length of each subdomain and element equals H and h, respectively. The elastic modulus E is set to 30×10^6 and Poisson's ratio ν to 0.3. Unit forces are applied to all nodes on the right edge of the mesh in the horizontal direction. Results are also presented for analogous plate bending and Laplace equation problems. For the plate bending problems, discrete Kirchoff triangular elements of thickness 0.01 are used with unit forces applied to all nodes on the right edge of the mesh in the out of plane direction.

The number of preconditioned conjugate gradient iterations needed to achieve a relative residual tolerance of 10^{-6} are shown in Table 3.1 for increasing numbers of subdomains. Also shown in the table are condition number estimates obtained using the connection between conjugates gradients and the Lanczos method [6]. The estimates are lower bounds on the actual condition numbers and were obtained from the extremal eigenvalues of a tridiagonal matrix of dimension equal to the number of iterations. Notice that the number of iterations and condition number estimates remain nearly constant as the number of subdomains increases.

Results for a fixed number of subdomains ($N = 16$) and increasing values of H/h

Table 3.2. Iterations, condition number estimates, and minimum ($e_{\min}$) and maximum ($e_{\max}$) eigenvalues of the coarse problem stiffness matrix for 2D problems with 16 subdomains ($H = 1/4$) and increasing numbers of elements per subdomain. The overlap is given by $p = (H/h)/4 - 1$.

	plane stress				plate bending				Laplace equation			
H/h	iter	κ	$e_{\min}$	$e_{\max}$	iter	κ	$e_{\min}$	$e_{\max}$	iter	κ	$e_{\min}$	$e_{\max}$
8	18	7.1	4.6e5	2.2e8	52	2.2e2	2.6	4.9e4	14	7.2	1.2e7	2.9e8
16	18	7.2	4.6e5	2.3e8	57	2.3e2	2.8	5.6e4	15	7.4	1.2e7	3.0e8
32	18	7.2	4.6e5	2.3e8	59	2.3e2	2.9	5.8e4	15	7.5	1.2e7	3.0e8
64	19	7.2	4.6e5	2.3e8	60	2.3e2	2.9	5.8e4	15	7.6	1.2e7	3.0e8

are shown in Table 3.2. In this case, the coarse dofs of the four leftmost subdomains are all constrained to zero. Without these constraints the numbers of iterations and condition numbers are lower, but the insensitivity of the quantities in Table 3.2 to the ratio H/h is less clear. Also shown in the table are the minimum and maximum eigenvalues of the coarse problem stiffness matrix $\Phi^T A \Phi$. Notice for all three problem types that the number of iterations, condition numbers, and eigenvalues all remain nearly constant provided the overlap parameter p is proportional to H/h.

To study the effects of material properties jumps, consider a problem where the elastic modulus $E = 1$ and $\nu = 0.3$ throughout the entire square domain except in the center region $[1/4, 3/4] \times [1/4, 3/4]$ where $E = \sigma$ and $\nu = 0.3$. The boundary conditions are the same as in the previous examples. Results for 16 subdomains with $H/h = 6$ are shown in Table 3.3 for different values of σ. In Table 3.3 each subdomain contains a single material. That is, the subdomains are aligned with material interfaces. Notice that the number of iterations and condition numbers do not change significantly with σ for all three problem types. Results for the same mesh ($h = 1/24$) with 9 subdomains ($H = 1/3$) are shown in Table 3.4. Here the subdomain boundaries do not align with the material interfaces. Although the results are not as good as those in Table 3.3, the numbers of iterations are not significantly larger.

Similar results for 3D problems are shown in Tables 3.5-3.8 where fully-integrated 8-node hexahedral elements were used. Results in Table 3.5 suggest that the preconditioner scales well with respect to the number of subdomains for 3D problems. Similarly, good scalability with respect to the number of elements per subdomain is apparent in Table 3.6. For Tables 3.7 and 3.8 the entire cube domain has $E = 1$ and $\nu = 0.3$ except for a centered cube region of dimension $1/2$ where $E = \sigma$ and $\nu = 0.3$.

Table 3.3. Iterations and condition number estimates for 2D problems with material property jumps. The number of subdomains is 16 ($H = 1/4$) and $H/h = 6$. Material property jumps are aligned with subdomain boundaries and the overlap p is fixed at 1.

	plane stress		plate bending		Laplace equation	
σ	iter	κ	iter	κ	iter	κ
10^{-3}	13	4.0	27	9.6	12	3.5
10^{-2}	13	4.0	26	9.0	12	3.5
1	13	3.8	26	9.6	11	3.6
10^{2}	14	4.0	27	10.8	13	4.0
10^{3}	15	4.0	29	10.9	15	4.0

Table 3.4. Iterations and condition number estimates for 2D problems with material property jumps. The number of subdomains is 9 ($H = 1/3$) and $H/h = 8$. Material property jumps are not aligned with subdomain boundaries and the overlap p is fixed at 1.

	plane stress		plate bending		Laplace equation	
σ	iter	κ	iter	κ	iter	κ
10^{-3}	13	4.8	31	22	11	4.0
10^{-2}	14	4.8	32	22	12	4.0
1	15	4.7	34	25	13	4.7
10^{2}	20	12	49	1.9e2	17	14
10^{3}	22	22	60	8.5e2	17	15
10^{4}	23	27	65	2.0e3	19	25
10^{5}	24	27	76	2.4e3	19	25

Table 3.5. Iterations and condition number estimates for 3D problems with increasing numbers of subdomains for $H/h = 4$. The overlap parameter p has a fixed value of 1. The number of equations in the coarse problems for the elasticity and Laplace equation problems equals $6N$ and N, respectively.

	elasticity		Laplace equation	
N	iter	κ	iter	κ
64	21	8.6	17	7.1
216	21	8.6	17	7.0
512	22	8.6	17	7.0
1000	22	9.0	18	9.3

Table 3.6. Iterations, condition number estimates, and minimum and maximum eigenvalues of the coarse problem stiffness matrix for 3D problems with 64 subdomains and increasing numbers of elements per subdomain. The overlap is given by $p = (H/h)/4 - 1$.

	elasticity				Laplace equation			
H/h	iter	κ	$e_{\min}$	$e_{\max}$	iter	κ	$e_{\min}$	$e_{\max}$
4	25	23	8.7e4	6.8e7	21	30	3.0e5	7.9e6
8	26	23	8.8e4	6.9e7	22	29	3.0e5	7.9e6
12	26	23	8.8e4	6.9e7	23	29	3.0e5	8.1e6
16	27	23	8.8e4	6.9e7	23	29	3.0e5	8.2e6

As with the 2D examples, good performance is obtained for both 3D examples with material property jumps.

Results for much larger 3D elasticity problems with $H/h = 22$ and a relative residual tolerance of 10^{-3} are shown in Table 3.9. The problems were run on the Department of Energy's Accelerated Strategic Computing Initiative option Q supercomputer. The 1000 subdomain problem has over 32 million dofs and was analyzed using the parallel structural dynamics code Salinas [7]. Results were obtained using Salinas implementations of the present approach with $p = 0$, a primal substructuring approach (CLIP) [8, 9], and FETI-DP [10]. Notice that the numbers of iterations and compute times for a Salinas implementation of the present approach, designated CLASP, are competitive with those for the CLIP and FETI-DP implementations. The results in the table for CLIP and FETI-DP are for using corner constraints only.

Table 3.7. Iterations and condition number estimates for 3D problems with material property jumps. The number of subdomains is 64 and $H/h = 6$. Material property jumps are aligned with subdomain boundaries and the overlap p is fixed at 1.

	elasticity		Laplace equation	
σ	iter	κ	iter	κ
10^{-3}	20	7.5	17	6.6
10^{-2}	20	7.4	17	6.6
1	20	7.5	17	6.6
10^{2}	22	7.8	21	13
10^{3}	24	12	23	16
10^{4}	27	15	27	24
10^{5}	30	26	30	35
10^{6}	34	35	32	44

Table 3.8. Iterations and condition number estimates for 3D problems with material property jumps. The number of subdomains is 27 and $H/h = 8$. Material property jumps are not aligned with subdomain boundaries and the overlap p is fixed at 1.

	elasticity		Laplace equation	
σ	iter	κ	iter	κ
10^{-3}	21	9.5	18	8.8
10^{-2}	21	9.5	18	8.7
1	22	9.8	19	9.2
10^{2}	31	38	28	46
10^{3}	35	54	30	57
10^{4}	37	70	31	59
10^{5}	39	76	32	59
10^{6}	40	77	32	59

Table 3.9. Iterations and times in seconds for 3D elasticity problems with $H/h = 22$. Each subdomain is assigned to one processor and the designation CLASP is for the present approach.

	Salinas CLASP		Salinas CLIP		Salinas FETI-DP	
N	iter	time	iter	time	iter	time
64	29	40	52	86	58	94
216	30	43	62	99	75	110
512	31	55	65	118	77	126
1000	31	62	68	123	79	133

Table 3.10. Results for 2D unstructured mesh problems shown in Figure 3.1. The overlap p was set to the integer nearest to $\sqrt{N_{elem}}/4 - 1$ where N_{elem} is the number of elements in the mesh.

			plane stress		Laplace equation	
N_{node}	N_{elem}	p	iter	κ	iter	κ
290	239	0	19	7.1	17	6.4
978	877	1	20	7.8	19	7.2
3604	3403	3	22	9.3	20	8.5
12800	12396	6	29	16	25	11

A possible explanation for the smaller number of iterations required by CLASP is that a multiplicative coarse problem correction is used rather than an additive one. The primary reason for the significant reduction in time is that one fewer factorization of a large sparse matrix is required by the present approach.

The final example deals with the unstructured meshes shown in Figure 3.1 of a square domain with three circular cutouts. The mesh decompositions into 16 subdomains were obtained using a code based on the graph partitioning software Chaco [11]. The boundary conditions are the same as those for the previous 2D examples. Results for plane stress and Laplace equation problems solved to a relative residual tolerance of 10^{-6} are shown in Table 3.10. In order to simplify communications for parallel programming, if any dof of $S_i^p - S_i^0$ is contained in at least one S_j^p for which Ω_i and Ω_j are not originally adjacent, then this dof is removed from S_i^p. Although the results shown are not as impressive as the ones in Table 3.2, they do show that very good results can be obtained for unstructured meshes.

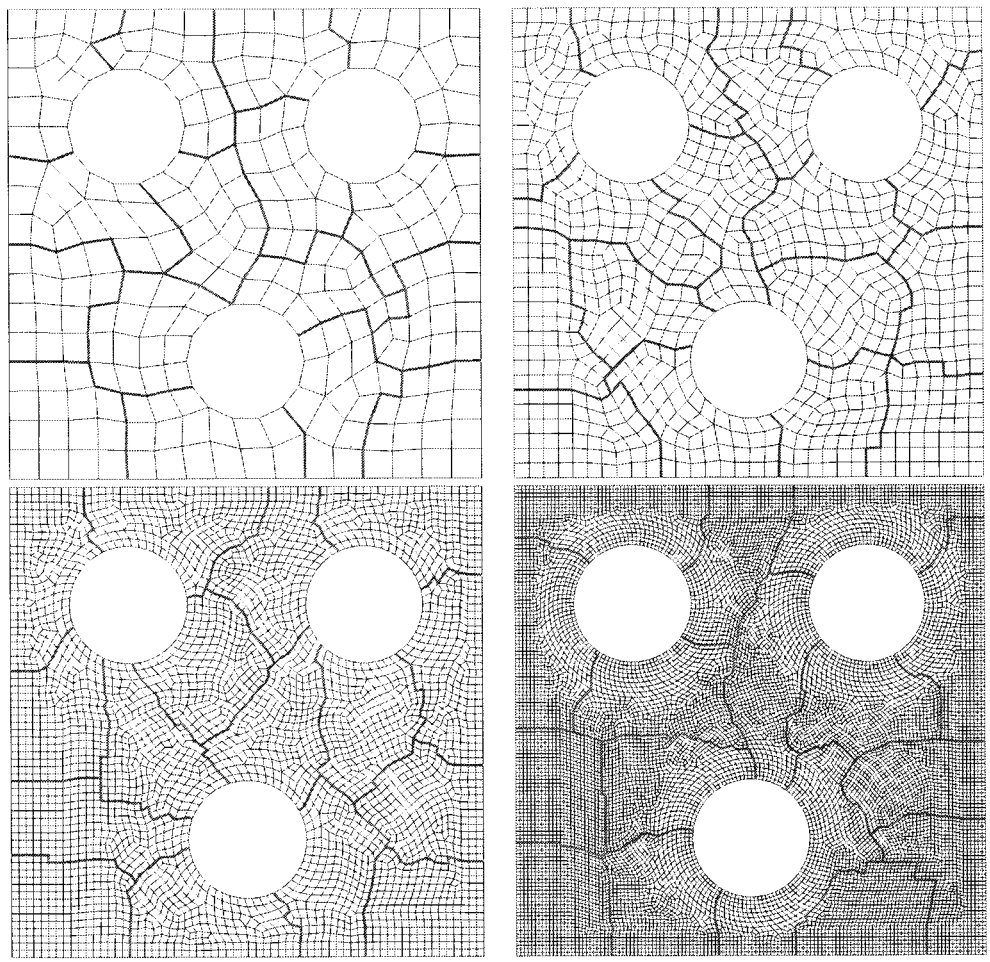

Figure 3.1. Finite element meshes and subdomains for final example.

3.4 Conclusions

A new algebraic coarse space for domain decomposition methods was presented. An attractive feature of the coarse space is that it can be constructed using solution techniques for one-level preconditioners. Thus, one has the ability to construct a two-level method from a one-level method at little additional cost. The coarse space was used in the formation of a two-level overlapping Schwarz preconditioner and applied to some example problems. Results from numerical studies suggest that the preconditioner scales well both with respect to the number of subdomains and the number of elements per subdomain. Results for a large-scale benchmark problem also show that the method is competitive with other solution methods. Good results were obtained for example problems with material property jumps whether or not the jumps were aligned with subdomain boundaries.

3.5 References

[1] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, New York, 1996.

[2] M. Brezina and Petr Vaněk. A black–box iterative solver based on a two–level Schwarz method. *Computing*, 63:233–263, 1999.

[3] P. Vaněk, M. Brezina, and R. Tezaur. Two-grid method for linear elasticity on unstructured meshes. *SIAM J. Sci. Comput.*, 21(3):900–923, 1999.

[4] J. Mandel. Balancing domain decomposition. *Comm. Numer. Meth. Engrg.*, 9:233–241, 1993.

[5] M. Sarkis. Partition of unity coarse spaces: Enhanced versions, discontinuous coefficients and applications to elasticity. *Fourteenth International Conference on Domain Decomposition Methods*, pages 149–158, January 2002.

[6] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins Univ. Press, Baltimore, MD, 1989. Second Edition.

[7] G. Reese, D. Segalman, M. Bhardwaj, K. Alvin, B. Driessen, K. Pierson, and T. Walsh. *Salinas - User's Notes.* Sandia National Laboratories, 2003.

[8] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, in press.

[9] J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Lin. Alg. Appl.*, 10:639–659, 2003.

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

[11] B. Hendrickson and R. Leland. The chaco user's guide version 2.0. Technical report, SAND95-2344, Sandia National Laboratories, Albuquerque, NM, 1995.

DISTRIBUTION:

1 Prof. Jan Mandel
Department of Mathematics
University of Colorado at Denver
Campus Box 170
Denver, CO 80217-3364

1 Prof. Charbel Farhat
Department of Aerospace Engineering Sciences
University of Colorado at Boulder
Campus Box 429
Boulder, CO 80309-0429

1 Prof. Olof Widlund
Department of Computer Science
Courant Institute
251 Mercer Street
New York, NY 10012

1 MS 0310
Leland, Robert W, 9220

1 MS 0323
Chavez, Donna L, 1011

1 MS 0835
Alvin, Kenneth F, 9142

1 MS 0835
McGlaun, J Michael, 9140

1 MS 0835
Pierson, Kendall H, 9142

1 MS 0835
Walsh, Timothy F, 9142

1 MS 0841
Bickel, Thomas C, 9100

1 MS 0847
Bhardwaj, Manoj K, 9142

1 MS 0847
Blanford, Mark L, 9142

10 MS 0847
Dohrmann, Clark R, 9124

1 MS 0847
Heinstein, Martin W, 9142

1 MS 0847
Key, Samuel W, 9142

1 MS 0847
Morgan, Harold S, 9120

1 MS 0847
Redmond, James M, 9124

1 MS 0847
Reese, Garth M, 9142

1 MS 1110
Bochev, Pavel B, 9214

1 MS 1110
Day, David, 9214

1 MS 1110
Heroux, Michael A, 9214

1 MS 1110
Hetmaniuk, Ulrich L, 9214

1 MS 1110
Lehoucq, Richard, 9214

1 MS 1110
Womble, David E, 9214

1 MS 1111
Boman, Erik G, 9215

1 MS 1111
Hendrickson, Bruce A, 9215

1 MS 9217
Adams, Mark F, 9214

1 MS 9217
Hu, Jonathan, 9214

1 MS 9217
Tuminaro, Raymond S, 9214

2 MS 0899
Technical Library, 9616

1 MS 9018
Central Technical Files, 8945-1