SIAM J. Sci. Comput.
Vol. 25, No. 1, pp. 246–258

A PRECONDITIONER FOR SUBSTRUCTURING BASED ON CONSTRAINED ENERGY MINIMIZATION[*]

CLARK R. DOHRMANN[†]

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 that demonstrate the good performance of the method in terms of iterations, compute time, and condition numbers of the preconditioned equations.

Key words. substructuring, domain decomposition, iterative methods, FETI, balancing domain decomposition

AMS subject classifications. 65F10, 65F50, 65N22, 65N55

DOI. 10.1137/S1064827502412887

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 nonoverlapping substructures $\Omega_1, \dots, \Omega_N$ 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 involves only dofs on substructure boundaries. This smaller system is then solved by a direct method.

Such an approach is effective for small to moderately sized problems, but it may be impractical for larger ones because the smaller system still might 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 such as plates [3]. A related finite element tearing and interconnecting (FETI) method [4] also requires modifications for fourth-order problems [5]. The method presented here shares many similarities with a dual-primal version of FETI [6] called FETI-DP, but important differences include the following: 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 shares 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

[*] Received by the editors August 14, 2002; accepted for publication (in revised form) April 28, 2003; published electronically October 2, 2003. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/sisc/25-1/41288.html

[†] Structural Dynamics Research Department, Sandia National Laboratories, Mail Stop 0847, Albuquerque, NM 87185-0847 (crdohrm@sandia.gov).

identical. The difference in the present method is that it uses an additive rather than a multiplicative coarse grid correction. Consequently, it is possible to use a different bilinear form on the coarse space, which results 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 in which 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 3 that demonstrate its good numerical performance. A mathematical theory for the method was developed shortly after the original submission of this paper and will be presented elsewhere [9].

2. Preconditioner.

The discrete internal energy of 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 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

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

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 containing 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

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

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 are never actually formed.

Returning now to the constraint matrices, we see that 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 the 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 $\mathcal{N}_{ij}$ is the 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 be included. 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

$$w_{ik} = s^i_{ik}/s_{ik}, \tag{11}$$

where s^i_{ik} 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^{ci}_{ik} as the sum of all diagonal entries of K_{ci} associated with $n(u_{ik})$. The global counterpart, s^c_{ik}, 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

$$w_{ik} = s^{ci}_{ik}/s^c_{ik}. \tag{13}$$

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

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

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 between them. 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 in 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].

2.1. 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 lead to

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

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

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.

2.2. 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].

3. 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 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 the inclusion of 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} is shown in Table 1 for increasing numbers of substructures. Also shown in the table are condition number estimates obtained using the connection between conjugate gradients and the Lanczos 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 grows 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. Notice that the number of iterations and condition numbers grows 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 1 and 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 property jumps, consider Figure 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

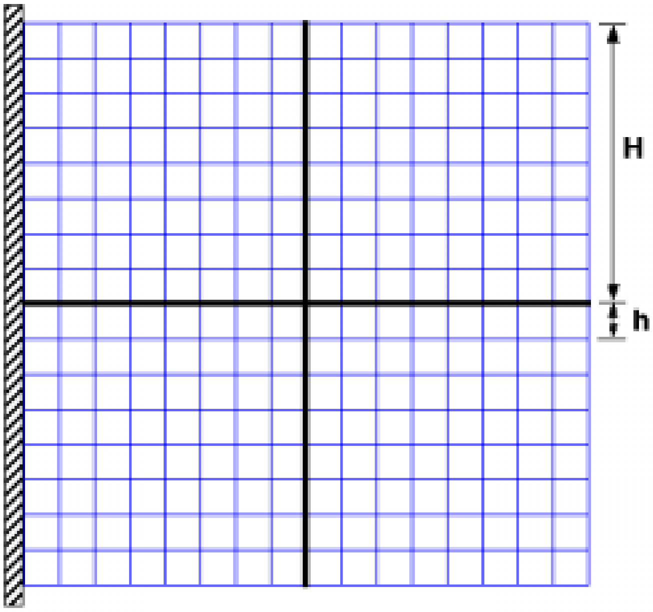

FIG. 1. *Mesh used for 2D scalability studies. In the figure $h = 1/16$ and $H = 1/2$.*

TABLE 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

are the same as the ones in the previous examples. Results for 16 substructures with $H/h = 6$ are shown in Table 3 for different values of σ. Results for an analogous Laplace equation problem with the same material property jumps are also reported. In Table 3 each substructure contains a single material. That is, the substructures are aligned with material interfaces. Notice that the number of iterations and condition numbers remains 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 4. Here the substructure boundaries do not align with the material interfaces. Notice that for both the plate bending and Laplace equation problems the number of iterations and condition numbers remains 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 2 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.

Table 2

Iterations and condition number estimates for 2*D 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

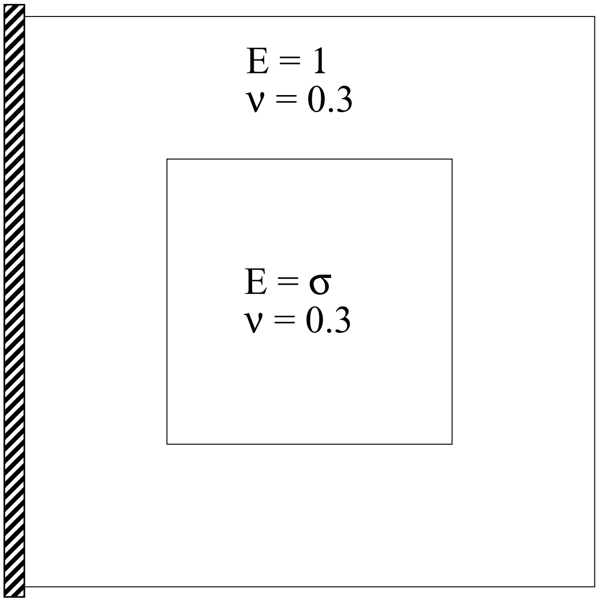

Fig. 2. *Geometry and boundary conditions for* 2*D problems with material property jumps.*

Similar results for 3D problems are shown in Tables 5–8, where fully integrated 8-node hexahedral elements were used. Results from the FETI-DP implementation used in Salinas are also shown in Tables 6–8, where the designation ACP means that an augmented coarse problem was used. The coefficient matrix for the ACP of FETI-DP is indefinite. In contrast, the matrix K_c is positive definite for both cases C and CE.

Results in Table 5 suggest that the preconditioner scales well with respect to the number of substructures for 3D problems. Results in Table 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 ACP. Results for problems with material property jumps are shown in Tables 7 and 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 7. Results in Table 8 suggest that much better performance can be obtained by including both corners and

Table 3

Iterations and condition number estimates for 2*D 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

Table 4

Iterations and condition number estimates for 2*D 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

edges when large material property jumps not aligned with substructure boundaries are present. Greater sensitivity to material property jumps is apparent in Table 8 for FETI-DP with an ACP compared to CE of the present approach. Although not shown, preliminary results obtained by suitable scaling of the ACP 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 9. The problems were run on the U.S. 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 9–11 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 3 and described in [6]. This model consists of 35328 8-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 4. This model consists of 269852 10-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

Table 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 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

substructures were obtained using a code based on the graph partitioning software Chaco [14]. Results for these two problems are shown in Tables 10 and 11. Although the numbers of iterations are generally larger than those for the structured meshes, they remain quite reasonable. Comparable performances of the present approach and FETI-DP are evident for these two problems.

4. 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 3D second-order problems required that both corners and edges be included in the constraints. These observations are consistent with those for other approaches such as 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 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.

Table 7

Iterations and condition number estimates for 3*D 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 8

Iterations and condition number estimates for 3*D 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

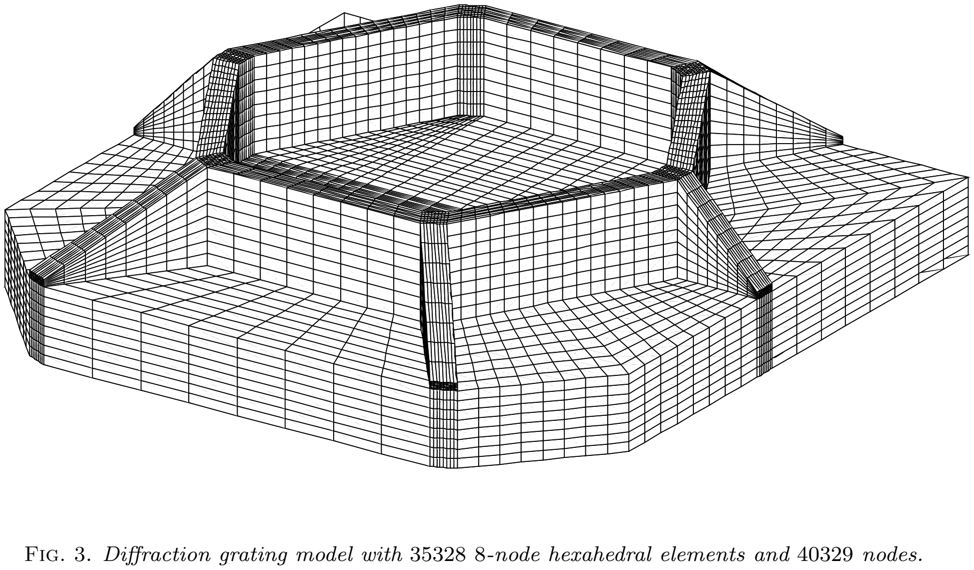

Fig. 3. *Diffraction grating model with* 35328 8*-node hexahedral elements and* 40329 *nodes.*

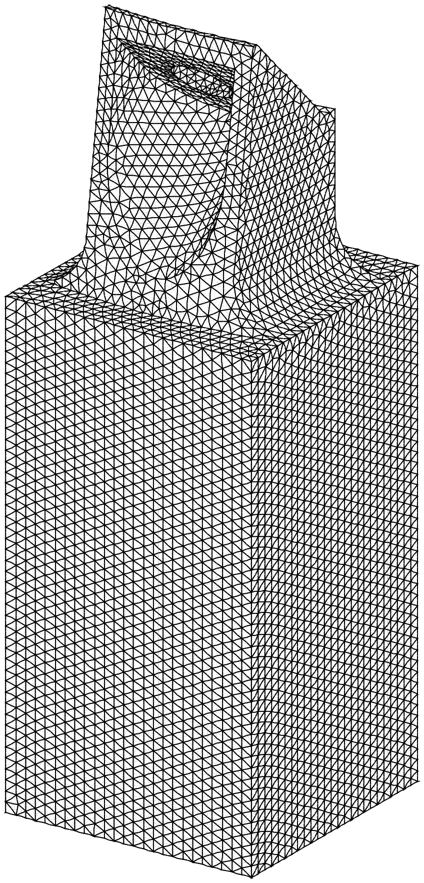

Fig. 4. *Joint leg model with* 269852 10-*node tetrahedral elements and* 374281 *nodes.*

Table 9

Results for 3D *elasticity problems with* $H/h = 22$. *Each substructure is assigned to one processor and CLIP designates 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

Table 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 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

Acknowledgments. The author wishes to express his gratitude to Kendall Pierson for providing the diffraction grating model and for many helpful discussions of FETI-DP and methods for corner and edge selection. Thanks are also extended to Manoj Bhardwaj for running some examples and for use of his domain decomposition tool; to Garth Reese for providing the interface between Salinas and the preconditioner of this study; and to Ronald Hopkins for the geometry model of the joint leg. Finally, the author is grateful to Jan Mandel for taking an interest in the preconditioner and for developing an accompanying mathematical theory.

REFERENCES

[1] J. S. Przemieniecki, *Theory of Matrix Structural Analysis*, McGraw-Hill, New York, 1968. Reprinted by Dover, New York, 1985.

[2] J. Mandel, *Balancing domain decomposition*, Comm. Numer. Methods Engrg., 9 (1993), pp. 233–241.

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

[4] C. Farhat and R. X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, Internat. J. Numer. Methods Engrg., 32 (1991), pp. 1205–1227.

[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 (1998), pp. 129–151.

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

[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 (1998), pp. 836–867.

[8] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition*, 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. Linear Algebra Appl., to appear.

[10] G. Reese, M. Bhardwaj, D. Segalman, A. Joseph, F. Kenneth, and B. Driessen, *Salinas—User's Notes*, Tech. report SAND99-2801, Sandia National Laboratories, Albuquerque, NM, 1999. Available online at http://infoserve.sandia.gov/sand_doc/1999/992801.pdf.

[11] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK User's Guide*, 3rd ed., SIAM, Philadelphia, 1999.

[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 (2000), pp. 1199–1219.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[14] B. Hendrickson and R. Leland, *The Chaco User's Guide Version* 2.0, Tech. report SAND95-2344, Sandia National Laboratories, Albuquerque, NM, 1995. Available online at http://infoserve.sandia.gov/sand_doc/1995/952344.pdf.