CHAPTER 13

A Clustered Element-by-Element Iteration Method for Finite Element Computations†

J. Liou*
T. E. Tezduyar*

Abstract

The clustered element-by-element iteration method presented in this paper is a generalization of the standard element-by-element method. In this method, the elements are partitioned into clusters of elements, with a desirable number of elements in each cluster, and the iterations are performed in a cluster-by-cluster fashion. A number of preconditioners based on the clustered element-by-element approach is introduced to be used with the iterative solution technique employed. The proposed solution strategies are tested on model problems with regular and irregular meshes. Several benchmark computations are performed to compare these solution strategies based on the convergence rates achieved.

1. Introduction

In large-scale finite element computations, the cost involved in solving the linear equation systems arising from the finite element discretization of the problem is a major one. Especially in time-dependent and/or nonlinear cases, where such systems need to be solved many times, and for three-dimensional problems, the cost can become unbearable. Therefore it is essential that this cost is minimized by employing solution strategies which are as efficient as possible. We believe that no matter how competitive a sophisticated direct (noniterative) solution technique may appear to be for a test problem of certain size, one can always put that size to a level high enough so that the direct solution technique is no longer competitive relative to an efficient iterative solution technique. With well chosen preconditioners, iterative solution techniques such as the conjugate gradient method [1] (for equation systems with symmetric matrices) and the GMRES method [2] (for equation systems with symmetric and nonsymmetric matrices) can be efficiently used for a wide class of large-scale applications. Of course selection of the preconditioner is a major concern, and a substantial amount of research effort goes into this (see for example [3] and [4]).

The element-by-element (EBE) preconditioners, which are defined as series product of the element matrices, have proven successful for several classes of problems (see [5-9]). Algorithms based on EBE preconditioners are highly vectorizable and parallelizable [10].

As a generalization of the standard EBE method, we propose a clustered element-by-element (CEBE) method, in which the elements are merged into clusters of elements, the preconditioners are defined as series product of the cluster matrices, and the computations are performed in a cluster-by-cluster fashion. Any

† This research was sponsored by the National Science Foundation under grant MSM 8796352.

*Department of Aerospace Engineering and Mechanics and Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN 55455.

number of elements can be brought together to form a cluster, and this number should be viewed as an optimization parameter to minimize the computational cost. Each cluster matrix is formed by assembling together the element matrices corresponding to the elements in that cluster. To facilitate vectorization and parallel processing, just like the way it is done in the grouped element-by-element (GEBE) method [8], the clusters can be grouped in such a way that no two clusters in any group are neighbors. Furthermore, depending on the cluster size (i.e., the number of elements in the cluster), within each cluster elements can again be grouped in the same way. In the two limit cases, the CEBE method becomes equivalent to the GEBE (when the cluster size is equal to one) and direct solution (when the cluster size is equal to the total number of elements) methods.

In this paper we consider four different CEBE preconditioners: Crout, Gauss-Seidel, 2-Pass/Product, and 2-Pass/Average. To test these preconditioners, we use model problems governed by the Poisson equation, employ both regular and irregular meshes, and compare the convergence rates achieved.

2. The Clustered Element-by-Element (CEBE) Iteration Method

Consider the linear equation system

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{1}$$

We rewrite (1) in a scaled form

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tag{2}$$

where

$$\tilde{\mathbf{A}} = \mathbf{W}^{-\frac{1}{2}}\mathbf{A}\mathbf{W}^{-\frac{1}{2}}, \tag{3}$$

$$\tilde{\mathbf{x}} = \mathbf{W}^{\frac{1}{2}}\mathbf{x}, \tag{4}$$

$$\tilde{\mathbf{b}} = \mathbf{W}^{-\frac{1}{2}}\mathbf{b}. \tag{5}$$

The scaling matrix $\mathbf{W}$ is defined as

$$\mathbf{W} = diag\mathbf{A} \tag{6}$$

for a symmetric and positive definite $\mathbf{A}$. With this definition of $\mathbf{W}$, $diag\tilde{\mathbf{A}}$ becomes an identity matrix. Since both the square root and the inverse of $\mathbf{W}$ is needed for the scaling, when $\mathbf{A}$ is not symmetric and positive-definite, alternative definitions for $\mathbf{W}$ need to be used so that $\mathbf{W}$ is still positive-definite.

The CEBE preconditioners can be used with iterative solution techniques (such as the GMRES method) which are applicable to equation systems with nonsymmetric matrices (see [11]). However, since the test problems considered in this paper are all governed by the Poisson equation, we only cover the implementation of the CEBE preconditioners with the conjugate gradient method.

Remark:

1. The convergence rate of an iteration algorithm depends on the condition number of the matrix $\tilde{\mathbf{P}}^{-1}\tilde{\mathbf{A}}$, where $\tilde{\mathbf{P}}$ is the preconditioner. Therefore the target is to select a preconditioner that involves minimal inversion cost and provides, within the cost limitations, an optimal representation of $\tilde{\mathbf{A}}$. For example, the Jacobi conjugate gradient (JCG) method, in which $\tilde{\mathbf{P}} = diag\tilde{\mathbf{A}}(= \mathbf{I})$, involves a minimal cost for the inversion of $\tilde{\mathbf{P}}$; however this representation of $\tilde{\mathbf{A}}$ is rather a poor one, it usually takes too many iterations to converge.

Let ε denote the set of all elements resulting from the finite element discretization of the computational domain. The set ε is partitioned into subsets (clusters of neighboring elements) ε_J, $J = 1, 2, ..., N_{cl}$, where N_{cl} is the number of clusters, such that

$$\varepsilon = \bigcup_{J=1}^{N_{cl}} \varepsilon_J,$$

$$\emptyset = \bigcap_{J=1}^{N_{cl}} \varepsilon_J.$$

The global coefficient matrix $\tilde{\mathbf{A}}$ can then be expressed as

$$\tilde{\mathbf{A}} = \sum_{J=1}^{N_{cl}} \tilde{\mathbf{A}}_J$$

with the cluster matrix $\tilde{\mathbf{A}}_J$ defined as

$$\tilde{\mathbf{A}}_J = \sum_{e \in \varepsilon_J} \tilde{\mathbf{A}}^e,$$

where $\tilde{\mathbf{A}}^e$ is the element level matrix. Consider the following CEBE representations of $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}} \approx \prod_{J=1}^{N_{cl}} (\mathbf{I} + \tilde{\mathbf{B}}_J) \quad (1 - Pass/Forward),$$

$$\tilde{\mathbf{A}} \approx \prod_{J=N_{cl}}^{1} (\mathbf{I} + \tilde{\mathbf{B}}_J) \quad (1 - Pass/Backward),$$

$$\tilde{\mathbf{A}} \approx \prod_{J=1}^{N_{cl}} (\mathbf{I} + \frac{1}{2}\tilde{\mathbf{B}}_J) \prod_{J=N_{cl}}^{1} (\mathbf{I} + \frac{1}{2}\tilde{\mathbf{B}}_J) \quad (2 - Pass/Product),$$

where

$$\tilde{\mathbf{B}}_J = \tilde{\mathbf{A}}_J - \tilde{\mathbf{W}}_J.$$

We now define the four CEBE preconditioners used in this paper.

(i) *Crout (CR) CEBE preconditioner*

Consider the Crout factorization of the matrix $(\mathbf{I} + \tilde{\mathbf{B}}_J)$:

$$(\mathbf{I} + \tilde{\mathbf{B}}_J) = \hat{\mathbf{L}}_J \hat{\mathbf{D}}_J \hat{\mathbf{U}}_J, \quad J = 1, 2, ..., N_{cl}.$$

The CR CEBE preconditioner is defined as

$$\tilde{\mathbf{P}} = \prod_{J=1}^{N_{cl}} \hat{\mathbf{L}}_J \prod_{J=1}^{N_{cl}} \hat{\mathbf{D}}_J \prod_{J=N_{cl}}^{1} \hat{\mathbf{U}}_J.$$

(ii) *Gauss-Seidel (GS) CEBE preconditioner*

Consider the following lower and upper triangular decomposition of $\tilde{\mathbf{B}}_J$ (for $\mathbf{W} = diag\mathbf{A}$):

$$\tilde{\mathbf{B}}_J = \tilde{\mathbf{B}}_J^U + \tilde{\mathbf{B}}_J^L, \quad J = 1, 2, ..., N_{cl}.$$

The GS CEBE preconditioner is defined as

$$\tilde{\mathbf{P}} = \prod_{J=1}^{N_{cl}} (\mathbf{I} + \tilde{\mathbf{B}}_J^L) \prod_{J=N_{cl}}^{1} (\mathbf{I} + \tilde{\mathbf{B}}_J^U).$$

(iii) *2-Pass/Product (2P/P) CEBE preconditioner*

The 2P/P CEBE preconditioner is defined based on the representation of $\tilde{\mathbf{A}}$ as given by eq. (13):

$$\tilde{\mathbf{P}} = \prod_{J=1}^{N_{cl}} (\mathbf{I} + \frac{1}{2}\tilde{\mathbf{B}}_J) \prod_{J=N_{cl}}^{1} (\mathbf{I} + \frac{1}{2}\tilde{\mathbf{B}}_J). \tag{19}$$

(iv) *2-Pass/ Average (2P/A) CEBE preconditioner*

The inverse of the 2P/A CEBE preconditioner is defined as the average of the inverses of the 1-Pass forward and backward representations of $\tilde{\mathbf{A}}$ as given by eqs. (11) and (12):

$$\tilde{\mathbf{P}}^{-1} = \frac{1}{2}(\prod_{J=1}^{N_{cl}} (\mathbf{I} + \tilde{\mathbf{B}}_J)^{-1} + \prod_{J=N_{cl}}^{1} (\mathbf{I} + \tilde{\mathbf{B}}_J)^{-1}). \tag{20}$$

Remarks:

2. The convergence of the algorithm depends on the numbering of the clusters, but not on the numbering of the elements within each cluster. By treating each cluster as a super-element, we can identify the clustered element-by-element procedure as a generalization of the standard element-by-element method.

3. We have the option of storing the cluster matrices and their inverses or recomputing them as they are needed.

4. The GS CEBE preconditioner, unlike all others, does not involve the factorization of the cluster matrices. Therefore, if the cluster matrices and their inverses are being stored, the memory needed for this purpose by the GS CEBE preconditioner will be approximately half as much.

5. The computational cost associated with the backward and forward substitutions for the CR and GS CEBE preconditioners is approximately half of the cost associated with the 2P/P and 2P/A CEBE preconditioners.

3. Numerical Examples

We tested the preconditioners defined in Section 2 on several test problems governed by the Poisson equation. A convergence criterion of 10^{-7} was chosen for the residual ratio $||\mathbf{r}||/||\mathbf{r}_0||$. In all cases the initial guess for the solution vector is set to zero. We compared the performances of these preconditioners based on the number of iterations required by each preconditioner.

Test cases with uniform square meshes

Here we consider a steady-state version of the model problem used in [12]. The finite element meshes, generated over a 1 × 1 computational domain, have 16j × 16j elements, with j = 1,2,3,...,10. The function f in the governing equation

$$\nabla^2 \phi = f \tag{21}$$

is selected in such a way that the exact solution is of the form

$$\phi = x(1-x)y(1-y)e^{xy}. \tag{22}$$

We solved this problem with 2 × 1 clusters, m × m clusters with m = 2^i, i=1, 2, 3, and 4, and with the standard EBE method (i.e., with 16j × 16j clusters). The clusters are numbered horizontally. Note that for a 16 × 16 mesh with 16 × 16 clusters, the algorithm reduces to a standard EBE method. Figure 1 shows the number of iterations vs mesh size for different CEBE preconditioners and various numbers of clusters. The Jacobi conjugate gradient (JCG) method takes (29, 60, 91, 122, 152, 183, 214, 246, 277, 312) iterations to converge, respectively, for meshes with j = 1, 2,..., 10. The 2P/A preconditioner gives the best convergence performance, followed by the CR, 2P/P and GS preconditioners. The performance of each method increases with decreasing number of clusters. Of course, with decreasing number of clusters, the cluster matrix size increases and so does the computational cost for each iteration. The convergence rates for the 2P/P and GS preconditioners are less dependent on the number of clusters than for the 2P/A and CR preconditioners.

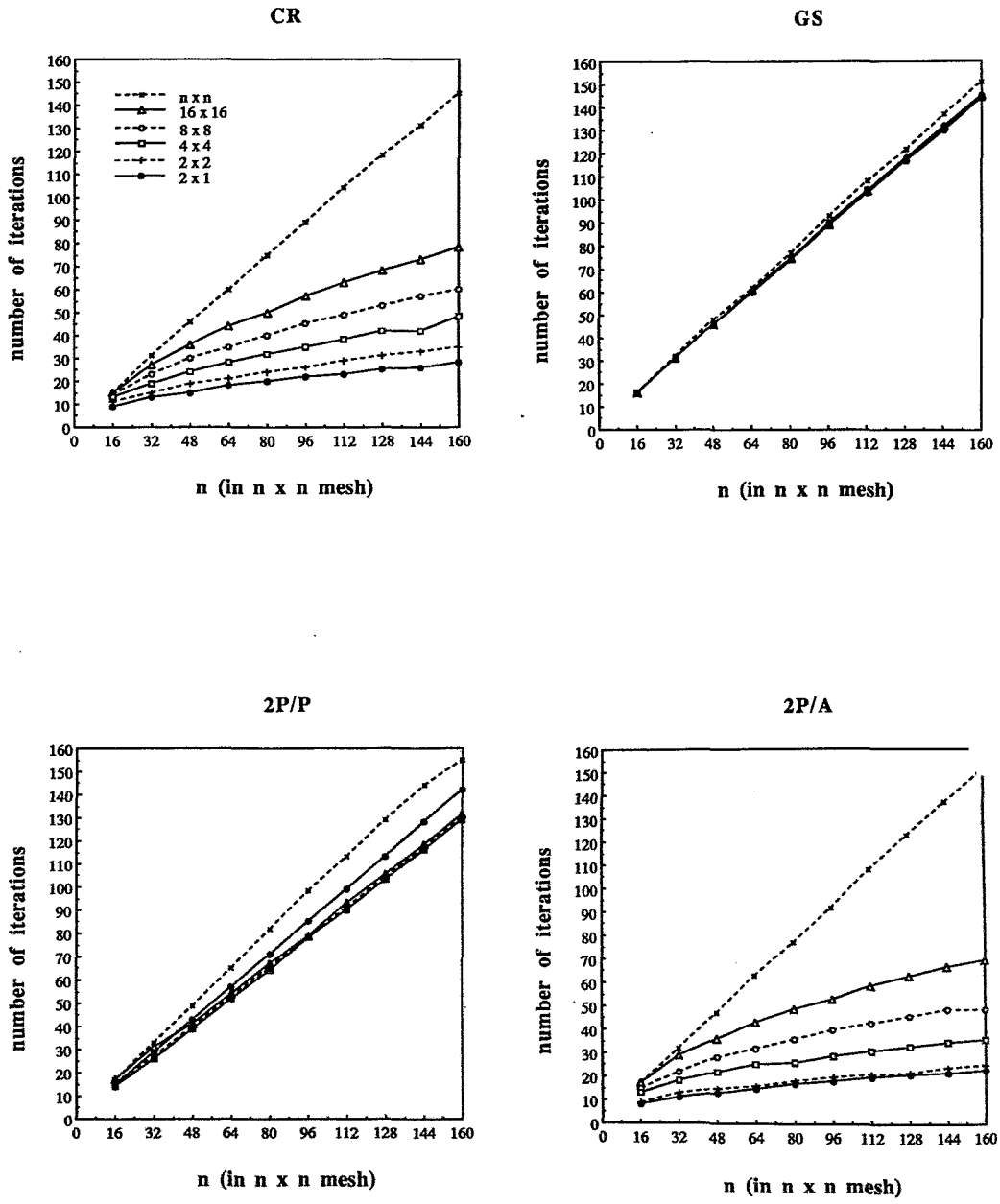


Figure 1. Test cases with uniform square meshes: number of iterations vs mesh size for different CEBE preconditioners and various numbers of clusters.

Test cases with irregular meshes

The problem configuration for the first set of test cases with irregular meshes is shown in Figure 2.

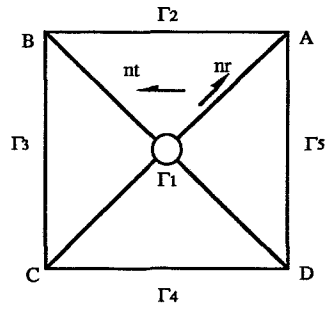


Figure 2. The problem configuration for the first set of test cases with irregular meshes.

The number of elements used is nr × nt × 4, with nr = nt = 8j, j =1, 2, 4, and 8, equally distributed in both directions. The problem is solved with $2^i \times 2^i \times 4$ clusters, with i = 0, 1, 2, 3 and 4. In the governing equation given by (21) $f = 0$.

First, as boundary conditions, we set $\phi = 1$ on Γ_1 and $\phi = 0$ at all other boundaries. Figure 3 shows the number of iterations vs mesh size for different CEBE preconditioners and various numbers of clusters. The JCG method takes (38, 78, 149, 292) iterations to converge, respectively, for meshes with j = (1,2,3,4).

Next, to make the problem no longer axially symmetric, we use the following set of nonsymmetric boundary conditions:

$$\phi = 1 \quad on\ \Gamma_1, \tag{23}$$

$$\phi = 0.25\frac{x - x_A}{x_B - x_A} \quad on\ \Gamma_2, \tag{24}$$

$$\phi = 0.25\frac{y - y_B}{y_C - y_B} + 0.25 \quad on\ \Gamma_3, \tag{25}$$

$$\phi = 0.25\frac{x - x_D}{x_C - x_D} \quad on\ \Gamma_4, \tag{26}$$

$$\phi = 0 \quad on\ \Gamma_5. \tag{27}$$

Figure 4 shows the number of iterations vs mesh size for different CEBE preconditioners and various numbers of clusters. The JCG method takes the same numbers of iterations as it does for the previous set of tests.

In this set of tests, the convergence performances of the CEBE preconditioners have the same relative trends as they have for the test cases with regular meshes. Furthermore, we observe that the 2P/P preconditioner is more sensitive to the boundary conditions specified than all the other preconditioners.

The second set of tests involve a "double-plate" problem with an irregular mesh (with 7,888 elements and 8,169 nodal points) as shown in Figure 5. In the governing equation given by (21) $f = 0$. The homogeneous Neumann type boundary condition is specified at the upper and lower boundaries. We set $\phi = 0.0$, 1.0, 1.5 and 2.0, respectively, at the left and right boundaries, and the upper and lower plates. Figure 6 shows the various types of element clustering and the number of iterations for different CEBE preconditioners. The JCG method takes 118 iterations to converge. The 2P/A CEBE preconditioner outperforms all the other preconditioners.

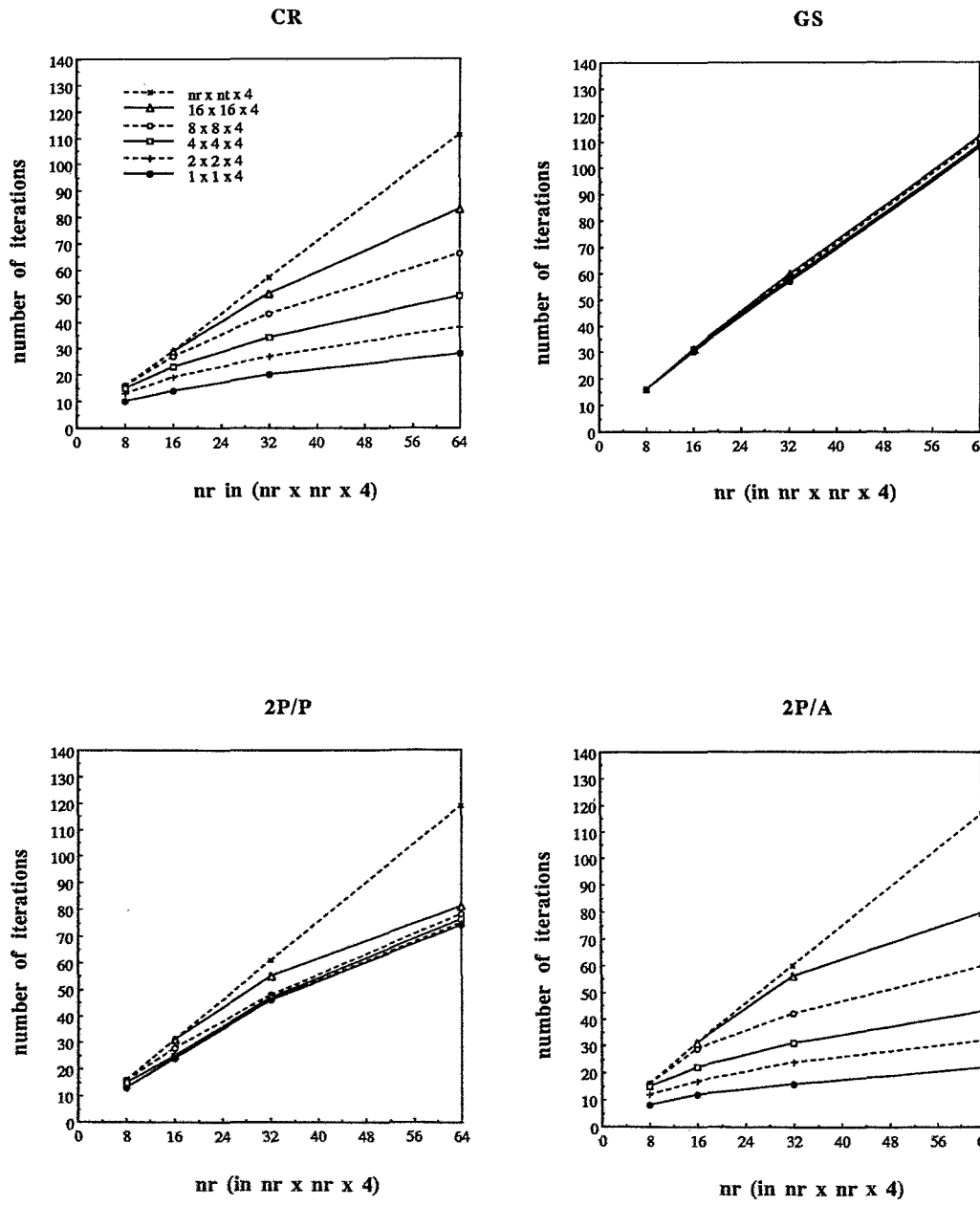


Figure 3. Test cases with irregular meshes and symmetric boundary conditions: number of iterations vs mesh size for different CEBE preconditioners and various numbers of clusters.

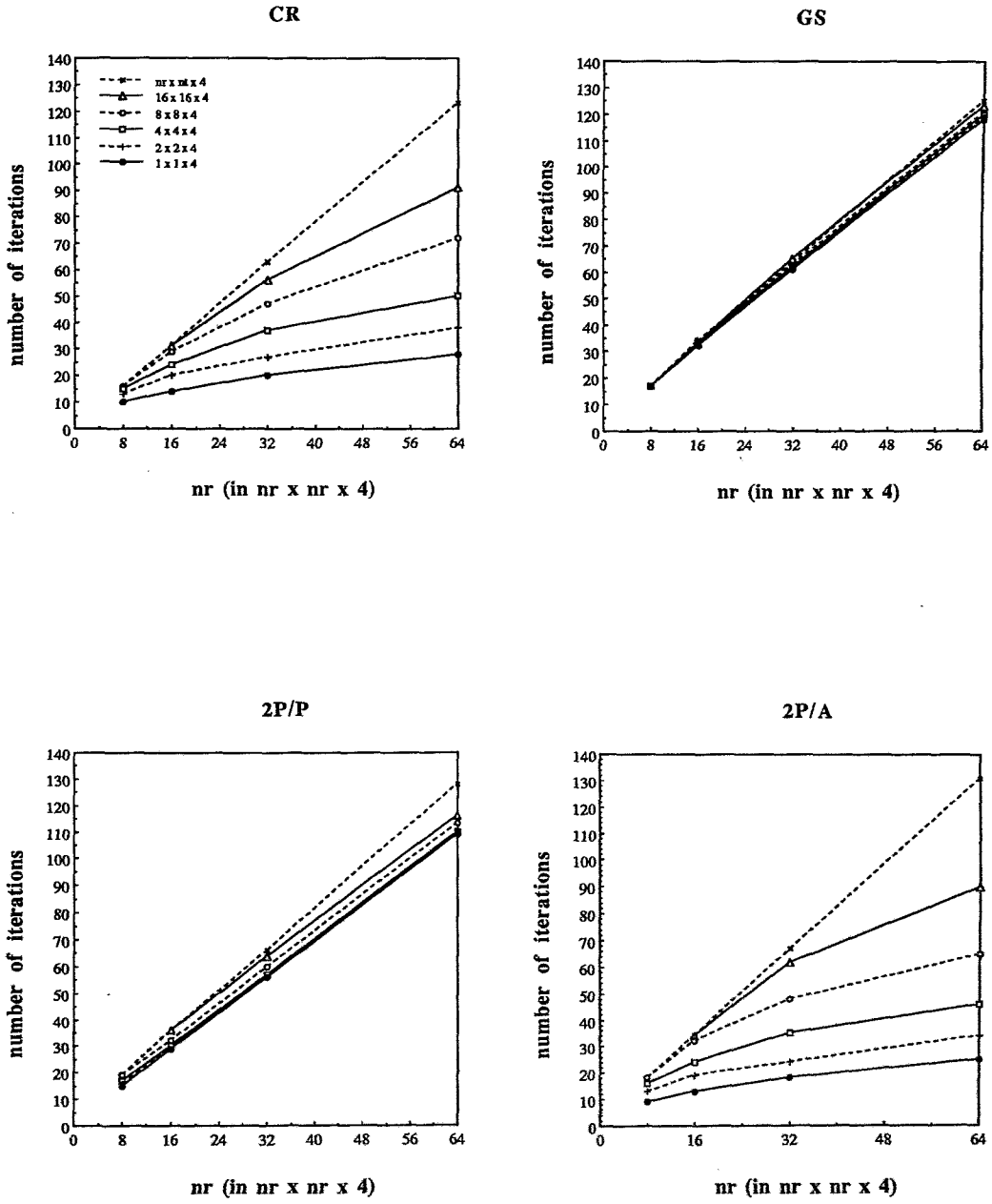


Figure 4. Test cases with irregular meshes and nonsymmetric boundary conditions: number of iterations vs mesh size for different CEBE preconditioners and various numbers of clusters.

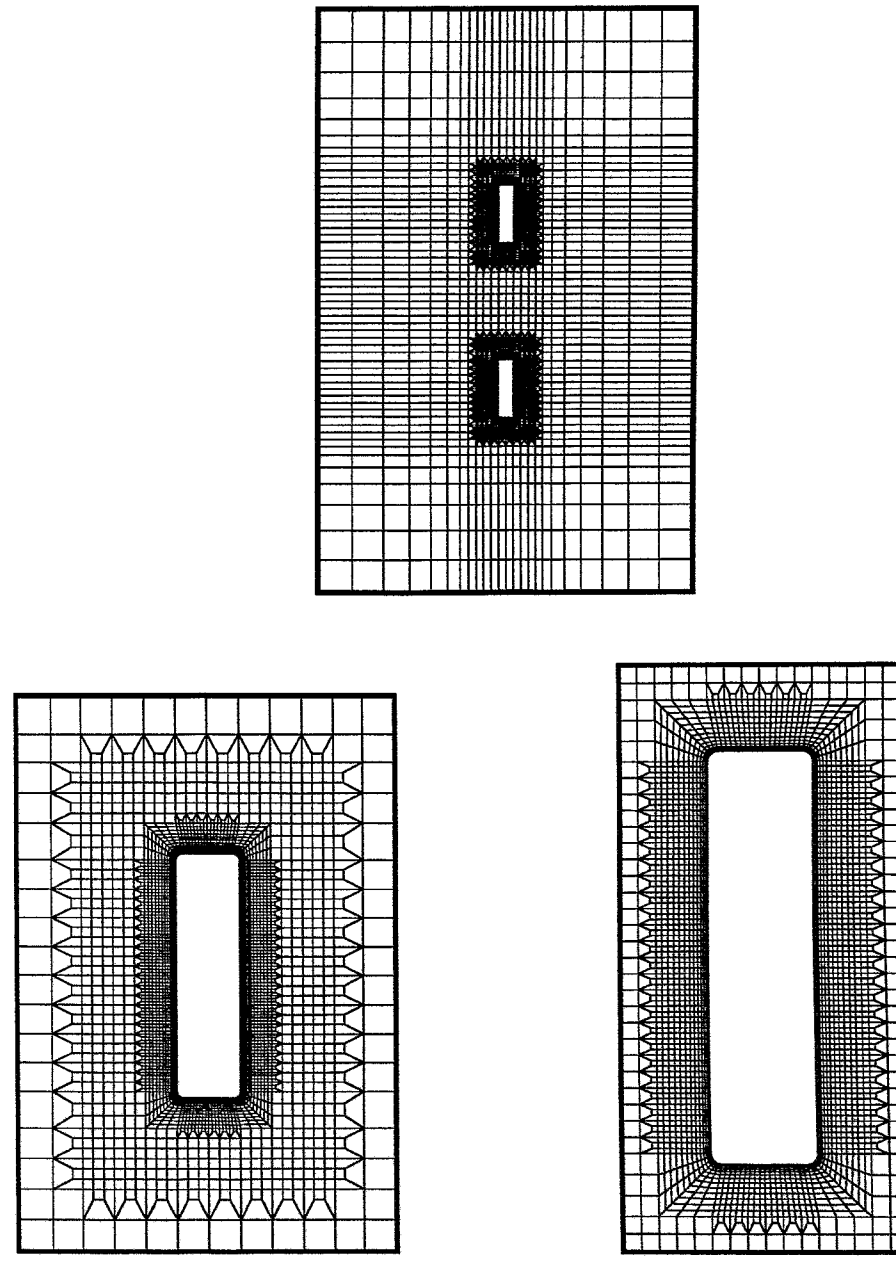
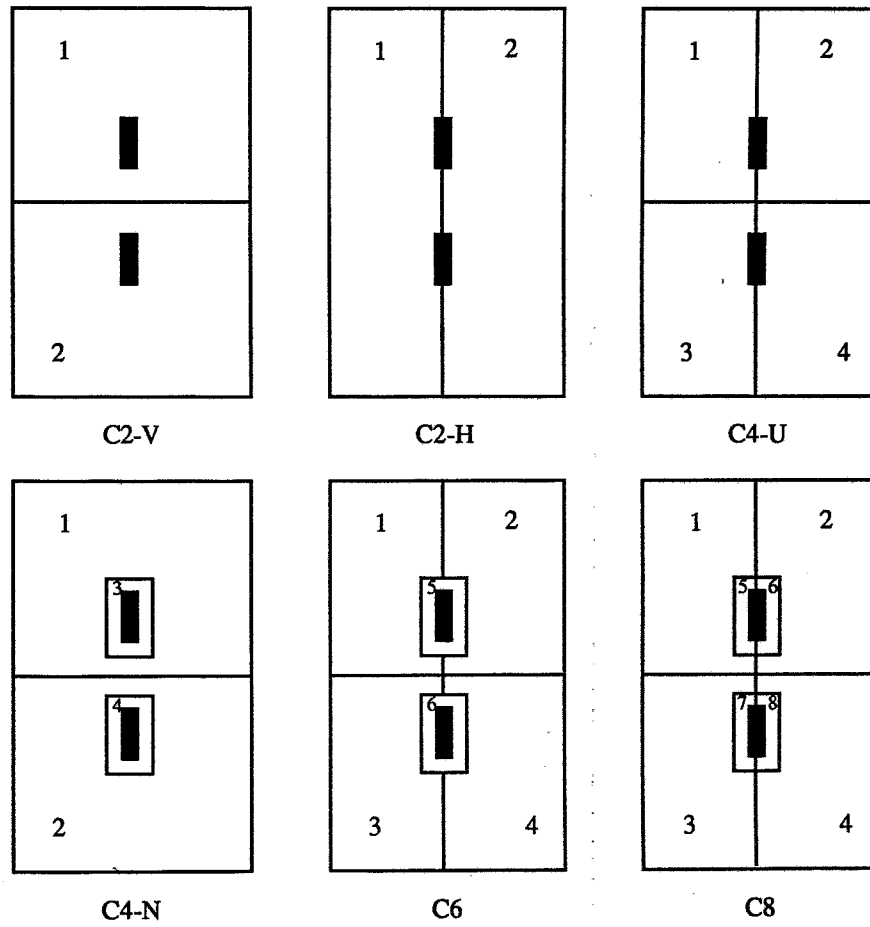


Figure 5. The finite element mesh (with 7,888 elements and 8,169 nodal points) for the double-plate problem.



	CR	GS	2P/P	2P/A
C2-V	12	62	58	11
C2-H	19	62	58	15
C4-U	20	62	58	16
C4-N	31	63	58	22
C6	32	63	59	24
C8	33	63	59	24
EBE	65	64	70	65

Figure 6. The double-plate problem: various types of element clustering and the number of iterations for different CEBE preconditioners.

References

1. R. Glowinski, *Numerical Methods for Nonlinear Variational Problems* , New York: Springer-Verlag, 1984.

2. Y. Saad and M. H. Schultz, " GMRES: a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", *Research Report YALEU/DCS/RR-254* , 1983.

3. Y. Kuznetsov, " New Algorithms for Approximate Realization of Implicit Difference Schemes", Sov. J. Numer. Anal. Math. Modelling v3 n^o 2, 1988.

4. M. O. Bristeau, R. Glowinski and J. Periaux, " Acceleration Procedures for the Numerical Simulation of Compressible and Incompressible Viscous Flows", Preprint, 1989.

5. T. J. R. Hughes, J. Winget, I. Levit, and T. E. Tezduyar , " New Alternating Direction Procedures in Finite Element Analysis Based upon EBE Approximate Factorizations", *Recent Developments in Computer Methods for Nonlinear Solid and Structural Mechanics*, S. Atluri and N. Perrone (eds.), ASME, 1983, pp. 75-109

6. T. J. R. Hughes, R. M. Ferencz, " Fully Vectorized EBE Preconditioners for Nonlinear Solid Mechanics: Applications to Large-Scale Three-Dimensional Continuum, Shell and Contact/Impact Problems", *First International Symposium on Domain Decomposition Methods for Partial Differential Equations* , R. Glowinski, G. H. Golub, G. A. Meurant, and J. Periaux (eds.), SIAM, 1988, pp. 261–280.

7. F. Shakib, T. J. R. Hughes and Z. Johan, " A Multi-element Group Preconditioned GMRES Algorithm for Nonsymmetric Systems Arising in Finite Element Analysis", to appear in *Computer Methods in Applied Mechanics and Engineering* .

8. T. E. Tezduyar and J. Liou, " Grouped Element-by-Element Iteration Schemes for Incompressible Flow Computations", *Computer Physics Communications* , 53 (1989), pp. 441–453.

9. J. Liou and T. E. Tezduyar," Combined AIE/EBE/GMRES Approach to Incompressible Flows", *Proceedings of the Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, T.F. Chan, R. Glowinski, J. Periaux, O.B. Widlund (eds.), SIAM, 1990, pp.462-480.

10. T. E. Tezduyar, J. Liou, T. Nguyen, and S. Poole," Adaptive Implicit-Explicit and Parallel Element-by-Element Iterative Schemes",*Proceedings of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations*, T.F. Chan, R. Glowinski, J. Periaux, O.B. Widlund (eds.), SIAM, 1989, pp.443-463.

11. J. Liou and T. E. Tezduyar," Computation of Compressible and Incompressible Flows with the Clustered Element-by-Element Method", University of Minnesota Supercomputer Institute Research Report UMSI 90/215, October, 1990.

12. G. Meurant, A Domain Decomposition Method for Parabolic Problems, Preprint, November 1989.