A NEW MIXED PRECONDITIONING METHOD FOR FINITE ELEMENT COMPUTATIONS[†]

T.E. Tezduyar, M. Behr, S.K. Aliabadi, S. Mittal and S.E. Ray
Department of Aerospace Engineering and Mechanics
Supercomputer Institute and
Army High-Performance Computing Research Center
University of Minnesota
1200 Washington Avenue South
Minneapolis, MN 55415, USA

Received 10 June 1991
Revised manuscript received 15 November 1991

Abstract

A new mixed clustered element-by-element (CEBE)/cluster companion (CC) preconditioning method for finite element computations is introduced. In the CEBE preconditioning, the elements are merged into clusters of elements, and the preconditioners are defined as series products of cluster level matrices. The CC preconditioning method, which is also introduced in this paper, shares a common philosophy with the multigrid methods. The CC preconditioners are based on companion meshes associated with different levels of clustering. For each level of clustering, we construct a CEBE preconditioner and an associated CC preconditioner. Because these two preconditioners in a sense complement each other, when they are used in a mixed way, they can be expected to give better performance. In fact, our numerical tests, for two and three-dimensional problems governed by the Poisson equation, demonstrate that the mixed CEBE/CC preconditioning results in convergence rates which are, in most cases, significantly better than the convergence rates obtained with the best of the CEBE and CC preconditioning methods.

1. Introduction

The element-by-element (EBE) preconditioners, which are constructed as series products of element level matrices, have been successfully applied to several classes of problems [1–4]. They can be used effectively with the conjugate-gradient and GMRES [5] methods, and are highly vectorizable and parallelizable (see [3, 6, 7]). They can also be used together with the implicit-explicit and adaptive implicit-explicit time-integration schemes [4, 7–9].

In clustered element-by-element (CEBE) preconditioning [10,11], the elements are merged into clusters of elements, and the preconditioners are constructed as series products of cluster level matrices. In [10], the CEBE preconditioning, together with the conjugate-gradient

[†]This research was sponsored by NASA-Johnson Space Center (under grant NAG 9-499) and by NSF (under grant MSM-8796352).

method, was used for solving problems with symmetric spatial operators (e.g., for problems governed by the Poisson equation). In [11], the CEBE preconditioning was employed, in conjunction with the GMRES method, to solve compressible and incompressible flow problems. Applications to the space-time finite element formulation of incompressible flows were included in [11]. To facilitate vectorization and parallel processing, as it is done in the grouped element-by-element (GEBE) method [6], the clusters can be grouped in such a way that no two clusters in any group are connected. Furthermore, depending on the number of elements in the cluster, within each cluster, elements can again be grouped in the same way. Each cluster matrix is formed by assembling together the element level matrices associated with the elements in that cluster. The number of elements in each cluster can be viewed as an optimization parameter that can be varied to minimize the computational cost. In fact, in [12], some of the unsteady incompressible flow computations were performed by using a space-time finite element formulation with a nearly optimal cluster size which was determined by numerical experimentation.

In this paper, we introduce the cluster companion (CC) preconditioning, which shares a common basis with the multi-grid methods. In the construction process of the CC preconditioners, we first start with a "primary" mesh with different levels of clustering. For each level of clustering in this primary mesh, we define a "companion" mesh, such that each cluster of the primary mesh forms an element of the companion mesh. We then define a CC preconditioner based on each companion mesh, such that there is a CC preconditioner associated with each CEBE preconditioner based on a certain level of clustering. This way, for each level of clustering, we obtain a CC preconditioner which we expect to have more inter-cluster coupling information then the associated CEBE preconditioner has. Conversely, the CEBE preconditioner can be expected to have more intra-cluster coupling information than the associated CC preconditioner has.

The mixed CEBE/CC preconditioning we propose in this paper is based on the belief that the CEBE and CC preconditioners complement each other, and therefore when they are mixed together they will result in better convergence rates. The mixed preconditioning can be implemented by using these two preconditioners alternately at each iteration of the conjugate gradient method or at each outer iteration of the GMRES method. Recently Saad [13] has formulated a new version of the GMRES algorithm which allows changing the preconditioner at every inner iteration. In fact, a GMRES subroutine, based on this new formulation and made available to us by Saad, is what we use to implement our mixed preconditioning.

The CEBE, CC and the mixed CEBE/CC preconditioning techniques are described, in detail, in Sections 2, 3 and 4. The numerical test results for problems governed by the Poisson equation are reported in Section 5.

2. CEBE (Clustered Element-by-Element) Preconditioning

Consider a linear equation system

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

encountered in finite element computation of a problem. Based on the finite element discretization of the problem domain Ω, the matrices $\mathbf{A}$ and $\mathbf{b}$ are formed by adding together

their element level constituents; i.e.,

$$\mathbf{A} = \sum_{e=1}^{n_{\text{el}}} \mathbf{A}^e, \tag{2}$$

$$\mathbf{b} = \sum_{e=1}^{n_{\text{el}}} \mathbf{b}^e, \tag{3}$$

where n_{el} is the number of elements.

REMARK 1
The domain Ω can also be a space-time domain, in which case the elements are space-time elements.

REMARK 2
The element level matrices $\mathbf{A}^e$ and $\mathbf{b}^e$ have the same dimensions as the global matrices $\mathbf{A}$ and $\mathbf{b}$, respectively; i.e., $n_{\text{eq}} \times n_{\text{eq}}$ and $n_{\text{eq}} \times 1$, where n_{eq} is the number equations. However, the only non-zero entries for these element level matrices are those corresponding to the nodes of element e, and this fact is taken into account in the implementation.

We assume that direct solution of (1) is not computationally feasible and that we would like to design a good preconditioner to maximize the efficiency of the iterative solution procedure. To achieve this, first we rewrite (1) in a scaled form

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tag{4}$$

where

$$\tilde{\mathbf{A}} = \mathbf{W}^{-1/2}\mathbf{A}\mathbf{W}^{-1/2}, \tag{5}$$

$$\tilde{\mathbf{b}} = \mathbf{W}^{-1/2}\mathbf{b}, \tag{6}$$

$$\tilde{\mathbf{x}} = \mathbf{W}^{1/2}\mathbf{x}. \tag{7}$$

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

$$\mathbf{W} = \text{diag } \mathbf{A}. \tag{8}$$

REMARK 3
This definition for the scaling matrix $\mathbf{W}$ is a good one when the matrix $\mathbf{A}$ is positive-definite. However, when $\mathbf{A}$ is not positive-definite, the following alternative definition [2] can be used:

$$\mathbf{W} = \text{lump } \mathbf{M}, \tag{9}$$

where lump $\mathbf{M}$ is the lumped version of the mass matrix $\mathbf{M}$. It is perhaps reasonable to look into defining a scaling matrix based on a combination of (8) and (9).

REMARK 4
In scaling a matrix, no matter what the level of that matrix is, the global scaling matrix $\mathbf{W}$ is the same. For example, the element level matrices $\mathbf{A}^e$ and $\mathbf{b}^e$ are scaled as

$$\tilde{\mathbf{A}}^e = \mathbf{W}^{-1/2}\mathbf{A}^e\mathbf{W}^{-1/2}, \tag{10}$$
$$\tilde{\mathbf{b}}^e = \mathbf{W}^{-1/2}\mathbf{b}^e. \tag{11}$$

The matrix $\mathbf{A}$ can be expressed as

$$\mathbf{A} = \mathbf{W} + \sum_{e=1}^{n_{\mathrm{el}}} (\mathbf{A}^e - \mathbf{W}^e). \tag{12}$$

In the scaled form this expression becomes

$$\tilde{\mathbf{A}} = \mathbf{I} + \sum_{e=1}^{n_{\mathrm{el}}} \tilde{\mathbf{B}}^e, \tag{13}$$

where

$$\tilde{\mathbf{B}}^e = \tilde{\mathbf{A}}^e - \tilde{\mathbf{W}}^e, \quad e = 1, 2, \ldots, n_{\mathrm{el}}. \tag{14}$$

The element-by-element (EBE) preconditioning is based on the approximation of (13) by a sequential product of element level matrices. Earlier implementations can be seen in [1, 2]. Various vectorized versions and applications to three-dimensional problems can be found in [3]. Parallel implementation of the method is achieved in [7] based on the grouped element-by-element (GEBE) approach [6], in which elements are ordered in groups with no inter-element coupling within each group. The number of groups is minimized to minimize the overhead associated with synchronization in parallel computations. Applications in conjunction with the implicit-explicit and adaptive implicit-explicit element grouping can be seen in [4, 7–9]. Depending on the form of matrix $\mathbf{A}$, the EBE type preconditioners can be used with the conjugate gradient, GMRES [5], or some other sophisticated search algorithm.

In the CEBE (clustered element-by-element) method the set of elements ε is partitioned into clusters of elements ε_J, $J = 1, 2, \ldots, N_{\mathrm{cl}}$. For example, Fig. 1 shows four different levels of clustering for a uniform 16×16 mesh. The cluster boundaries are marked with thick lines. In the first frame each cluster consists of one element, and therefore this would lead to an EBE method. In the last frame the cluster size is 8×8; the next level of clustering after that (i.e., level 5) would lead to a direct solution method. The global matrix $\mathbf{A}_J$ associated with cluster J is defined as

$$\mathbf{A}_J = \sum_{e \in \varepsilon_J} \mathbf{A}^e. \tag{15}$$

The matrix $\mathbf{A}$ can then be expressed, similar to (12), as

$$\mathbf{A} = \mathbf{W} + \sum_{J=1}^{N_{\mathrm{cl}}} (\mathbf{A}_J - \mathbf{W}_J). \tag{16}$$

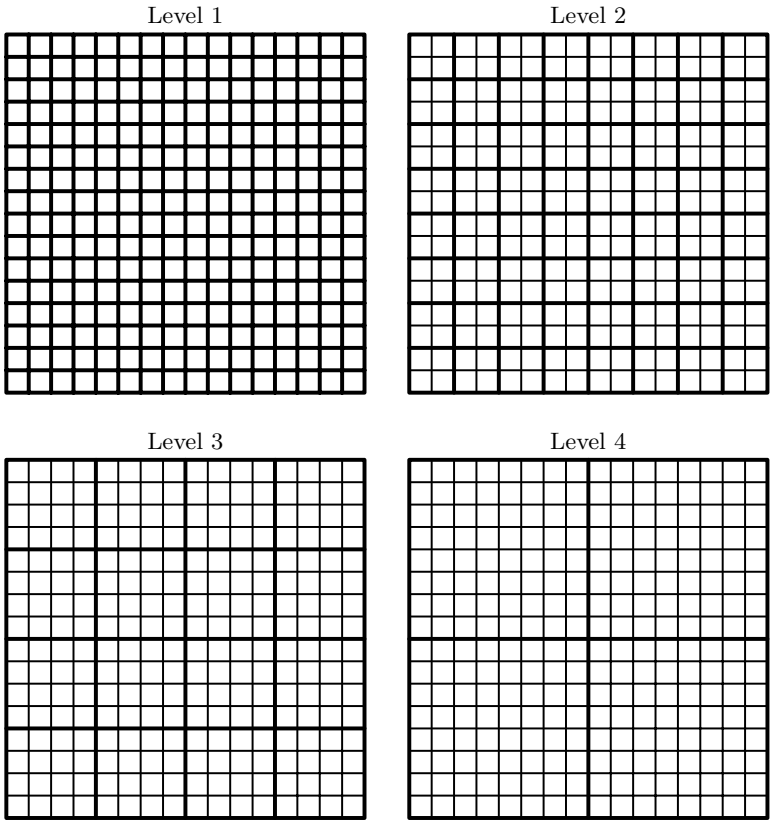


Figure 1. Four different levels of clustering for a uniform 16×16 mesh; in each frame the thick lines depict the cluster boundaries and the associated companion mesh.

In the scaled form this expression becomes

$$\tilde{\mathbf{A}} = \mathbf{I} + \sum_{J=1}^{N_{\mathrm{cl}}} \tilde{\mathbf{B}}_J, \tag{17}$$

where

$$\tilde{\mathbf{B}}_J = \tilde{\mathbf{A}}_J - \tilde{\mathbf{W}}_J, \quad J = 1, 2, \ldots, N_{\mathrm{cl}}. \tag{18}$$

The CEBE preconditioning is based on the approximation of (17) by a sequential product of cluster level matrices. Here we give two examples (see [10, 11]): 2-Pass CEBE preconditioner and Crout CEBE preconditioner. The 2-Pass CEBE preconditioner is defined as

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

and the Crout CEBE preconditioner is defined as

$$\tilde{\mathbf{P}}_C = \prod_{J=1}^{N_{\mathrm{cl}}} \hat{\mathbf{L}}_J \prod_{J=1}^{N_{\mathrm{cl}}} \hat{\mathbf{D}}_J \prod_{J=N_{\mathrm{cl}}}^{1} \hat{\mathbf{U}}_J, \tag{20}$$

where $\hat{\mathbf{L}}_J$, $\hat{\mathbf{D}}_J$ and $\hat{\mathbf{U}}_J$ are the matrices resulting from the following Crout factorization:

$$\mathbf{I} + \tilde{\mathbf{B}}_J = \hat{\mathbf{L}}_J \hat{\mathbf{D}}_J \hat{\mathbf{U}}_J, J = 1, 2, \ldots , N_{\mathrm{cl}}. \tag{21}$$

In [10], these types of preconditioning were used, in conjunction with the conjugate gradient method, for problems governed by the Poisson equation. In [11], they were used, together with the GMRES method, for compressible and incompressible flow problems. Of course the convergence rates depend on the cluster sizes. In [12], for the space-time finite element formulation of an incompressible flow problem, an optimal cluster size was determined by numerical experimentation and was used in the computations.

3. Cluster Companion (CC) Preconditioning

Let us consider a mesh with different levels of clustering. For each level of clustering in this "primary" mesh, we define a "companion" mesh, such that each cluster of the primary mesh forms an element of the companion mesh. For example, Fig. 1 can now also be seen as showing the companion meshes associated with four different levels of clustering in a 16×16 primary mesh. In each frame of Fig. 1, the thick lines not only mark the cluster boundaries for a certain level of clustering, but also depict the companion mesh associated with that level of clustering. In the first frame the companion mesh is the same as the primary mesh. In the last frame the companion mesh is a 2×2 mesh. In our notation, the level of clustering and the associated companion mesh will be identified by the same integer number; i.e., companion mesh l will be associated with clustering level l.

Because the companion mesh 1 is the same as the primary mesh, (1)–(3) can also be written as

$$(\mathbf{A})^1 (\mathbf{x})^1 = (\mathbf{b})^1 , \tag{22}$$

$$(\mathbf{A})^1 = \sum_{e=1}^{(n_{\mathrm{el}})^1} (\mathbf{A}^e)^1 , \tag{23}$$

$$(\mathbf{b})^1 = \sum_{e=1}^{(n_{\mathrm{el}})^1} (\mathbf{b}^e)^1 , \tag{24}$$

where the superscript "1" denotes the companion mesh number.

Let $(u)^1$ be the approximation of a displacement-like scalar field u over the companion mesh 1, such that

$$(u)^1 = \sum_{B=1}^{(n_{\mathrm{np}})^1} (N)^1_B (u)^1_B, \tag{25}$$

where $(n_{\mathrm{np}})^1$ is the number of nodal points in companion mesh 1, $(N)^1_B$is the shape function associated with node B, and $(u)^1_B$ is the value of $(u)^1$ at node B. A similar expression can be written to approximate u over the companion mesh 2:

$$(u)^2 = \sum_{B=1}^{(n_{\mathrm{np}})^2} (N)^2_B (u)^2_B, \tag{26}$$

Let $(\mathbf{u})^1 = \{(u)^1_B, B = 1, 2, \ldots, (n_{\rm np})^1\}$ and $(\mathbf{u})^2 = \{(u)^2_B, B = 1, 2, \ldots, (n_{\rm np})^1\}$. Given $(\mathbf{u})^2$, we would like to obtain an expression that approximates $(\mathbf{u})^1$. Based on least-squares minimization of the difference $(\mathbf{u})^1 - (\mathbf{u})^2$, we obtain

$$\mathbf{M}^{11}(\mathbf{u})^1 \simeq \mathbf{M}^{12}(\mathbf{u})^2, \tag{27}$$

where

$$\left(\mathbf{M}^{11}\right)_{AB} = \int_\Omega (N)^1_A (N)^1_B d\Omega, \quad A, B = 1, 2, \ldots, (n_{\rm np})^1, \tag{28}$$

$$\left(\mathbf{M}^{12}\right)_{AB} = \int_\Omega (N)^1_A (N)^2_B d\Omega, \quad A = 1, 2, \ldots, (n_{\rm np})^1, B = 1, 2, \ldots, (n_{\rm np})^2. \tag{29}$$

From (27), we can write

$$(\mathbf{u})^1 \simeq \mathbf{E}^{12}(\mathbf{u})^2, \tag{30}$$

with the "interpolation" matrix $\mathbf{E}^{12}$ defined as

$$\mathbf{E}^{12} = \left[\mathbf{M}^{11}\right]^{-1} \mathbf{M}^{12} \tag{31}$$

or

$$\mathbf{E}^{12} = \left[\text{lump } \mathbf{M}^{11}\right]^{-1} \mathbf{M}^{12}. \tag{32}$$

An expression similar to (30) can be written to obtain $(\mathbf{u})^2$ from $(\mathbf{u})^1$:

$$(\mathbf{u})^2 \simeq \mathbf{E}^{21}(\mathbf{u})^1, \tag{33}$$

with

$$\mathbf{E}^{21} = \left[\mathbf{M}^{22}\right]^{-1} \mathbf{M}^{21} \tag{34}$$

or

$$\mathbf{E}^{21} = \left[\text{lump } \mathbf{M}^{22}\right]^{-1} \mathbf{M}^{21}. \tag{35}$$

where

$$\left(\mathbf{M}^{22}\right)_{AB} = \int_\Omega (N)^2_A (N)^2_B d\Omega, \quad A, B = 1, 2, \ldots, (n_{\rm np})^2, \tag{36}$$

$$\left(\mathbf{M}^{21}\right)_{AB} = \int_\Omega (N)^2_A (N)^1_B d\Omega, \quad A = 1, 2, \ldots, (n_{\rm np})^2, B = 1, 2, \ldots, (n_{\rm np})^1. \tag{37}$$

More on the derivations related to $\mathbf{E}^{12}$ and $\mathbf{E}^{21}$ can be found in the Appendix A. Assuming that the Dirichlet type boundary conditions are somehow taken care of in the implementation, we can also use equations (30) and (33) to obtain $(\mathbf{x})^1$ and $(\mathbf{x})^2$ from each other. That is,

$$(\mathbf{x})^1 \simeq \mathbf{E}^{12}(\mathbf{x})^2, \tag{38}$$

$$(\mathbf{x})^2 \simeq \mathbf{E}^{21}(\mathbf{x})^1, \tag{39}$$

Furthermore, by assuming that $(\mathbf{b})^1$ and $(\mathbf{b})^2$ are force-like quantities and that the energy-like quantities expressed over the two companion meshes, $(\mathbf{b})^1(\mathbf{x})^1$ and $(\mathbf{b})^2(\mathbf{x})^2$, are equivalent, we can write

$$(\mathbf{b})^2 \simeq \mathbf{F}^{21}(\mathbf{b})^1, \tag{40}$$
$$(\mathbf{b})^1 \simeq \mathbf{F}^{12}(\mathbf{b})^2, \tag{41}$$

where

$$\mathbf{F}^{21} \simeq (\mathbf{E}^{12})^t, \tag{42}$$
$$\mathbf{F}^{12} \simeq (\mathbf{E}^{21})^t. \tag{43}$$

From (22), (38) and (40) we can write an approximate expression for $\left[(\mathbf{A})^1\right]^{-1}$:

$$\left[(\mathbf{A})^1\right]^{-1} \simeq \mathbf{E}^{12}\left[(\mathbf{A})^2\right]^{-1}\mathbf{F}^{21}. \tag{44}$$

This expression is the starting point for us to construct a companion preconditioner based on the companion mesh 2. The matrix $(\mathbf{A})^2$ can be computed either by using the definition of $\mathbf{A}$ over the companion mesh 2, or by using the following expression:

$$(\mathbf{A})^2 \simeq \mathbf{F}^{21}(\mathbf{A})^1\mathbf{E}^{12}. \tag{45}$$

We note, for implementational purposes, that (45) is equivalent to

$$(\mathbf{A})^2 \simeq \sum_{e=1}^{(n_{\mathrm{el}})^1} \mathbf{F}^{21}(\mathbf{A})^1\mathbf{E}^{12}. \tag{46}$$

We also note that, if $(\mathbf{A})^1$ is symmetric and positive-definite, so is $(\mathbf{A})^2$ given by the expression (45). However, we cannot say the same thing for $\left[(\mathbf{A})^1\right]^{-1}$ given by the expression (44). Therefore, to define our companion preconditioner, we use a regularization similar to the one used in (12). The cluster companion preconditioner based on companion mesh 2 is then defined as

$$\left[(\mathbf{P}_{CC})^{121}\right]^{-1} = \mathbf{W}^{-1} + \mathbf{E}^{12}\left(\left[(\mathbf{A})^2\right]^{-1} - \mathbf{E}^{21}\mathbf{W}^{-1}\mathbf{F}^{12}\right)\mathbf{F}^{21}. \tag{47}$$

In scaled form, (47) can be rewritten as follows:

$$\left[\left(\tilde{\mathbf{P}}_{CC}\right)^{121}\right]^{-1} = \mathbf{I} + \mathbf{W}^{1/2}\mathbf{E}^{12}\left(\left[(\mathbf{A})^2\right]^{-1} - \mathbf{E}^{21}\mathbf{W}^{-1}\mathbf{F}^{12}\right)\mathbf{F}^{21}\mathbf{W}^{1/2}. \tag{48}$$

We also experimented with the following modified version of (47):

$$\left[(\mathbf{P}_{CC})^{121}\right]^{-1} = \mathbf{W}^{-1} + \mathbf{E}^{12}\left[(\mathbf{A})^2\right]^{-1}\mathbf{F}^{21}, \tag{49}$$

which can be written in scaled form as

$$\left[\left(\tilde{\mathbf{P}}_{CC}\right)^{121}\right]^{-1} = \mathbf{I} + \mathbf{W}^{1/2}\mathbf{E}^{12}\left[(\mathbf{A})^2\right]^{-1}\mathbf{F}^{21}\mathbf{W}^{1/2}. \tag{50}$$

We can repeat the expressions given by (48) and (50) for the cluster companion preconditioner based on the companion mesh 3:

$$\left[\left(\tilde{\mathbf{P}}_{CC}\right)^{131}\right]^{-1} = \mathbf{I} + \mathbf{W}^{1/2}\mathbf{E}^{13}\left(\left[(\mathbf{A})^3\right]^{-1} - \mathbf{E}^{31}\mathbf{W}^{-1}\mathbf{F}^{13}\right)\mathbf{F}^{31}\mathbf{W}^{1/2}, \tag{51}$$

$$\left[\left(\tilde{\mathbf{P}}_{CC}\right)^{131}\right]^{-1} = \mathbf{I} + \mathbf{W}^{1/2}\mathbf{E}^{13}\left[(\mathbf{A})^3\right]^{-1}\mathbf{F}^{31}\mathbf{W}^{1/2}. \tag{52}$$

Here $\mathbf{E}^{13}$ and $\mathbf{E}^{31}$ can be computed either by using definitions similar to those given by equations (31), (32), (34) and (35), or by using the following relations:

$$\mathbf{E}^{13} = \mathbf{E}^{12}\mathbf{E}^{23}, \tag{53}$$
$$\mathbf{E}^{31} = \mathbf{E}^{32}\mathbf{E}^{21}. \tag{54}$$

In any case, the matrices $\mathbf{F}^{31}$ and $\mathbf{F}^{13}$ are defined as

$$\mathbf{F}^{31} = (\mathbf{E}^{13})^t, \tag{55}$$
$$\mathbf{F}^{13} = (\mathbf{E}^{31})^t. \tag{56}$$

REMARK 5

It is quite clear that the philosophy behind this type of preconditioning is similar to the philosophy behind multigrid iteration methods.

REMARK 6
One could also incorporate the idea of "companion" meshes in conjunction with formulations employing higher-order elements. For example, for a mesh using bi-quadratic elements only the nodes at the corners of the higher-order elements will form the companion mesh at level 2. To go to level 3 one could cluster elements in the mesh at level 2 and so on.

4. Mixed CEBE/CC Preconditioning

It is reasonable to expect that the CEBE preconditioner has more intra-cluster coupling information than the CC preconditioner has. It is also reasonable to expect that the CC preconditioner has more inter-cluster coupling information than the CEBE preconditioner has. Therefore, because these two preconditioners in a sense complement each other, it is reasonable to hope that when they are mixed together they lead to better convergence rates.

Initially our plan was to use these two preconditioners alternately at each iteration of the conjugate gradient method or at each outer iteration of the GMRES method. However, it was recently brought to our attention that Saad [13] has formulated a new version of the GMRES algorithm which allows one to change the preconditioner at every inner iteration. A GMRES subroutine based on this new formulation was made available to us by Saad, and we simply use this subroutine to implement our mixed preconditioning.

In our notation, CEBE-l will represent the CEBE preconditioning based on clustering level l, CC-l will represent the CC preconditioning based on companion mesh l associated

with clustering level l, and CEBE/CC-l will represent the mixing of the two. For example, CC-1 would lead to a direct solution method, and therefore we will normally start our test computations with $l = 2$ or higher.

REMARK 7
We note that as l increases, the cost associated with CEBE-l increases and the cost associated with CC-l decreases.

5. Numerical Tests

We tested the preconditioners defined in Sections 2, 3 and 4 on several test problems governed by the Poisson equation:

$$\nabla^2 \phi = f. \tag{57}$$

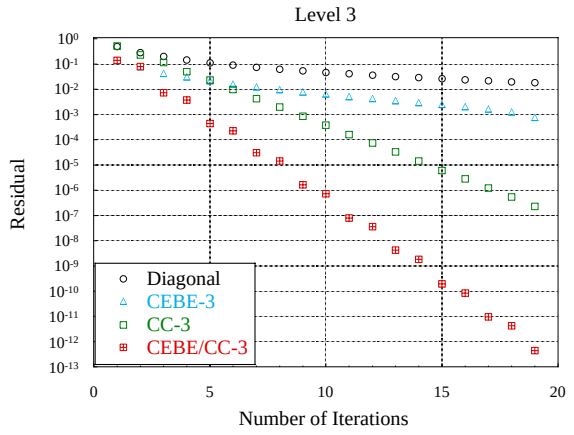
Some of these problems are similar to the ones solved in [10] using the CEBE preconditioners. For all the results reported here a Krylov space of dimension 20 was used and the initial guess for the solution vector was set to zero. The matrices $(\mathbf{A})^l$ for all the cases were computed directly over the corresponding companion meshes, and the companion preconditioners were based on (50). To compare the performance of the preconditioners used, we monitored the scaled residual (normalized by the initial residual) during the inner iteration loop (number of inner iterations is the same as the size of Krylov space) for one outer iteration.

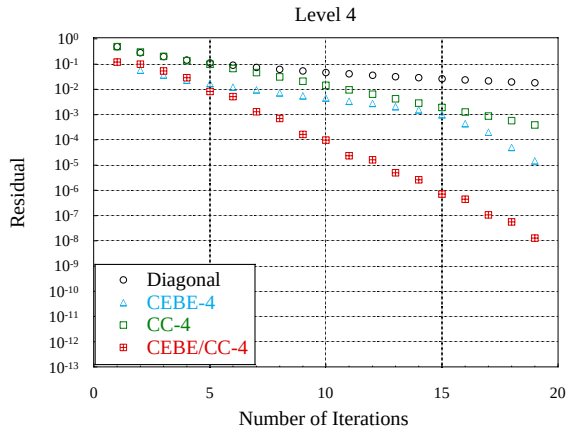
5.1. Tests with two-dimensional uniform square meshes

We solved (57) over a unit square computational domain for two different test cases.

CASE 1
For this case $f = 0$, and homogeneous Dirichlet type boundary conditions are imposed at three sides of the square domain. A parabolic boundary condition with a maximum value of $\phi = 1$ at the center is imposed at the fourth side of the domain. Fig. 2 shows, for a primary mesh with 64×64 elements, the convergence behavior of the diagonal, CEBE-l, CC-l and CEBE/CC-l preconditioners for $l = 3$, 4 and 5. We expect the performance of the CEBE method to improve for higher levels of clustering and this is clearly seen in Fig. 2. Also visible is the faster convergence of the CC method for lower levels of companion grids. At levels 3 and 4 the performance of CEBE and CC preconditioners is closely matched, and a relatively high rate of convergence of the mixed CEBE/CC scheme is easily observed.

Results obtained for a primary mesh with 128×128 elements for $l = 4$, 5 and 6 are shown in Fig. 3. It is interesting to note that the absolute rate of convergence of CEBE/CC-4 and CEBE/CC-5 preconditioners on the 64×64 mesh is very similar to the rate for the same preconditioner on the 128×128 mesh. However the performance of the CEBE/CC preconditioners relative to their stand-alone CEBE and CC counterparts is similar between a given companion mesh level on the 64×64 grid and the next (higher) companion mesh level on the 128×128 grid.





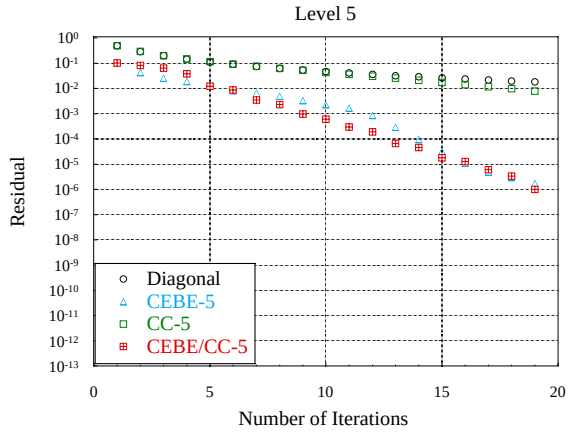
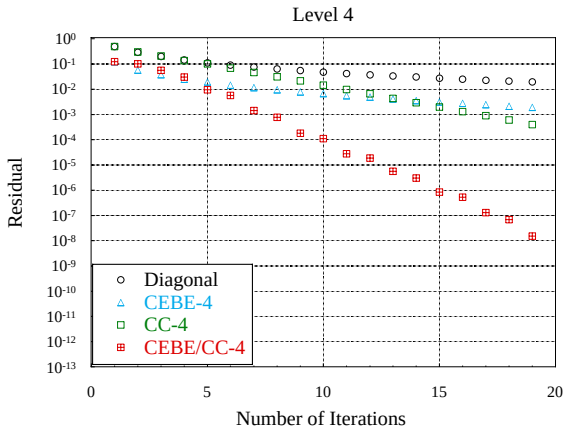
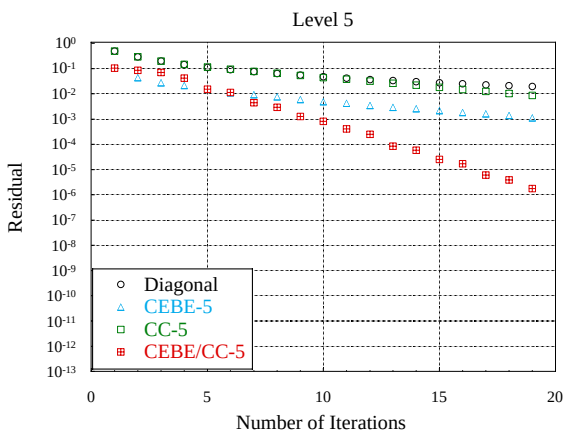


Figure 2. Convergence histories for a two-dimensional uniform mesh with 64×64 elements: Case 1.





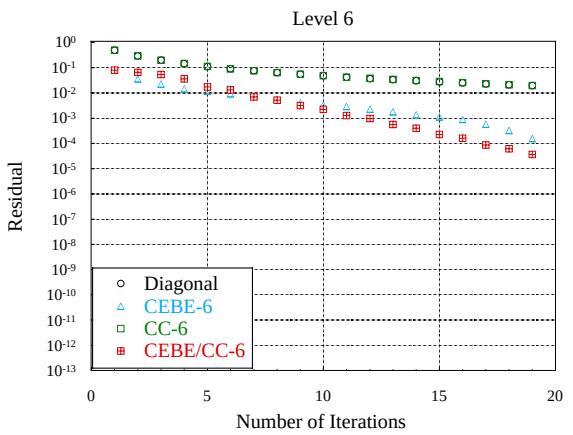


Figure 3. Convergence histories for a two-dimensional uniform mesh with 128×128 elements: Case 1.

CASE 2

In this case the function f in equation (57) and the boundary conditions are chosen in such a way that the exact solution is of the form

$$\phi = xy(1-x)(1-y)\mathrm{e}^{xy}. \tag{58}$$

This problem is a steady-state version of the one used in [14]. The convergence results for this case are shown in Fig. 4. These results are similar to the ones described earlier for the previous case.

5.2. Tests with two-dimensional non-uniform meshes

In this problem a cylinder of unit radius is located at the center of a 32×32 square, as shown in Fig. 5. The primary mesh consists of $(4 \times nt) \times nr$ elements, where $nr = nt = 32$ or 64. In the governing equation given by (57) we set $f = 0$ and the following set of non-symmetric

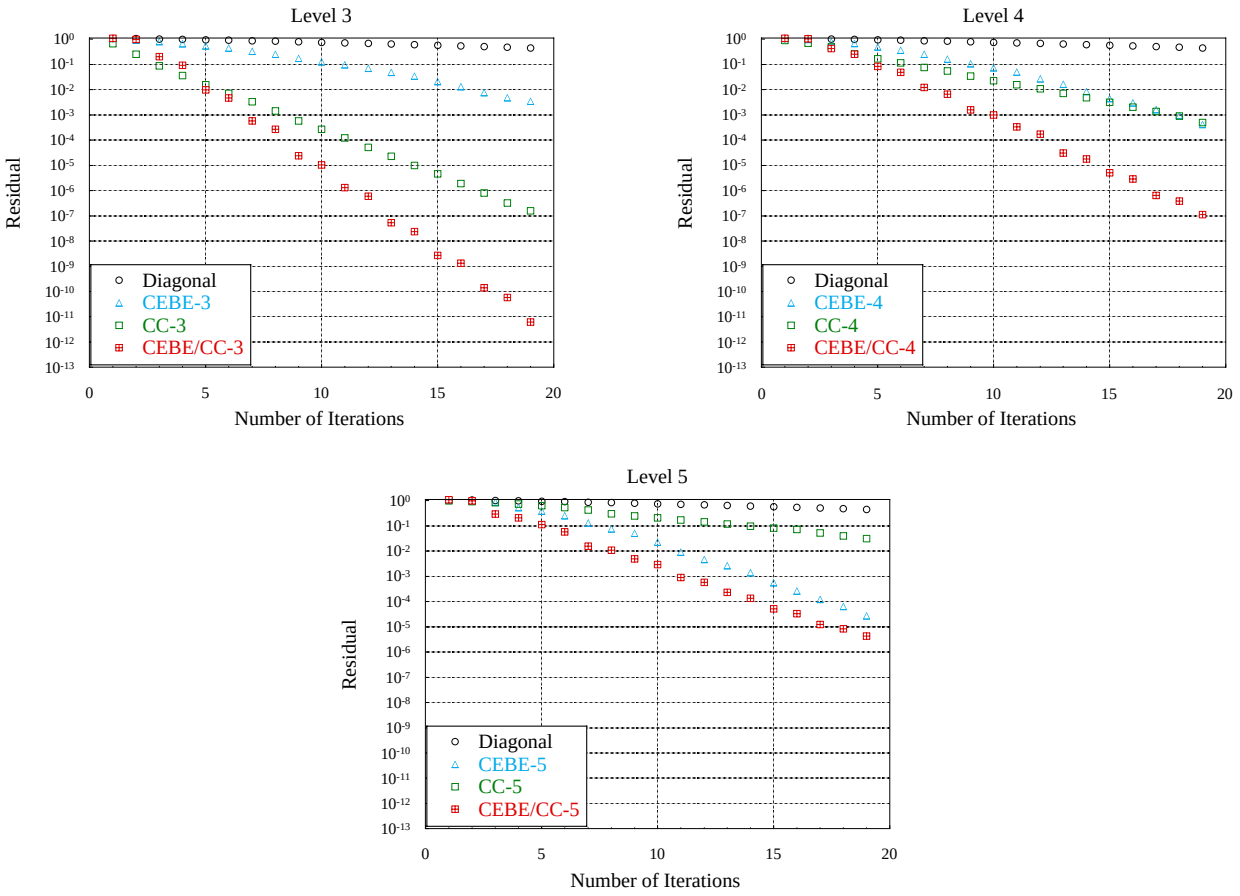


Figure 4. Convergence histories for a two-dimensional uniform mesh with 64×64 elements: Case 2.

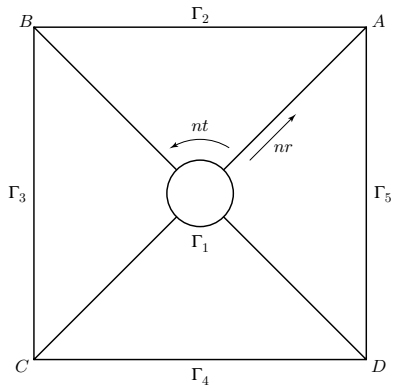


Figure 5. Problem configuration for the test cases with non-uniform meshes.

boundary conditions are used:

$$\phi = 1 \qquad \text{on } \Gamma_1, \tag{59}$$

$$\phi = 0.25(x - x_A)/(x_B - x_A) \qquad \text{on } \Gamma_2, \tag{60}$$

$$\phi = 0.25(y - y_A)/(y_B - y_A) + 0.25 \qquad \text{on } \Gamma_3, \tag{61}$$

$$\phi = 0.50(x - x_D)/(x_C - x_D) \qquad \text{on } \Gamma_4, \tag{62}$$

$$\phi = 0 \qquad \text{on } \Gamma_5, \tag{63}$$

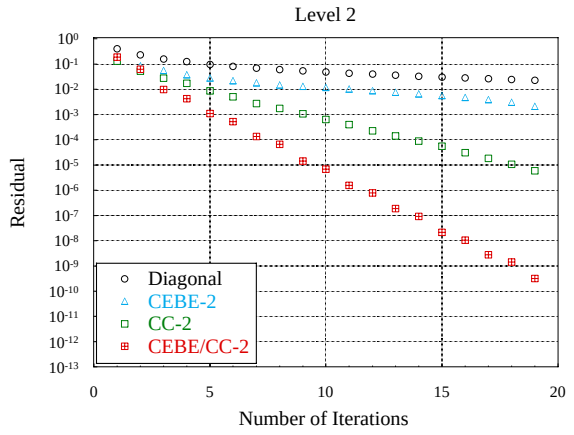
Figure 6 shows the convergence histories for the various (l =2, 3 and 4) preconditioners used on a 128×32 primary mesh. Again, the CEBE/CC preconditioners outperform the others. Results obtained for a 256×64 primary mesh for $l = 3$, 4 and 5 are shown in Fig. 7.

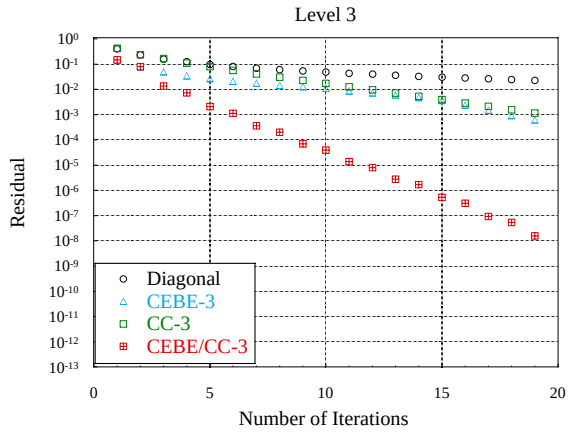
5.3. Tests with three-dimensional uniform meshes

This case is a three-dimensional extension of the first test with uniform square mesh. The domain is a rectangular parallelepiped, discretized by trilinear brick elements. For this case $f = 0$, and homogeneous Dirichlet type boundary conditions are imposed on five faces of the domain. A boundary condition of bi-quadratic form, with a maximum value of $\phi = 1$ at the center and $\phi = 0$ at the edges is imposed on the sixth face. In Fig. 8 we show, for a primary mesh with $128 \times 128 \times 64$ elements, the convergence behavior of the diagonal, CEBE-l, CC-l and CEBE/CC-l preconditioners for $l = 4$. At this level the performances of the CEBE and CC preconditioners are comparable, and the CEBE/CC scheme achieves much higher rate of convergence than either of its two components.

REMARK 8

In all two-dimensional test cases we observe that the mixed CEBE/CC preconditioning scheme performs well for all levels of companion meshes down to, and including, a companion mesh consisting of 8×8 elements for uniform meshes, and 32×8 elements for non-uniform meshes.





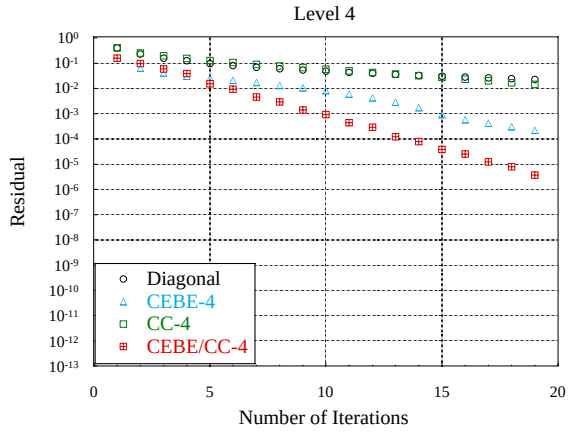
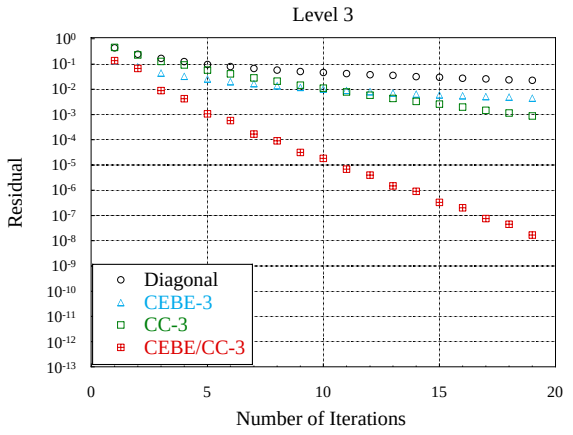
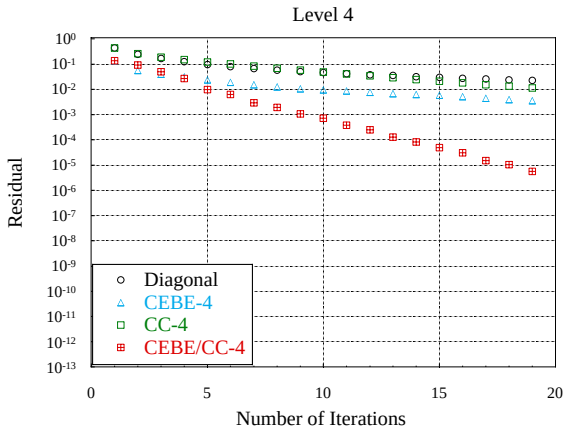


Figure 6. Convergence histories for a two-dimensional non-uniform mesh with 128×32 elements.





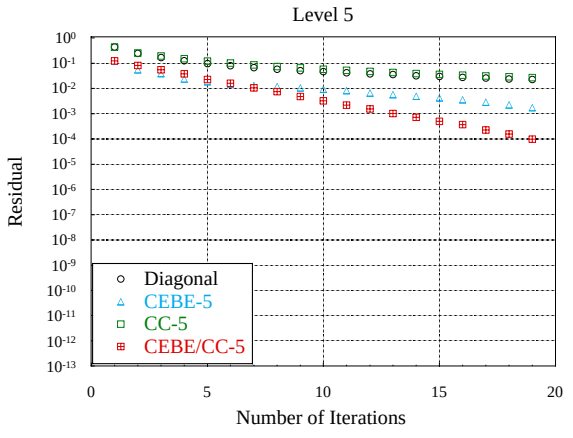


Figure 7. Convergence histories for a two-dimensional non-uniform mesh with 256×64 elements.

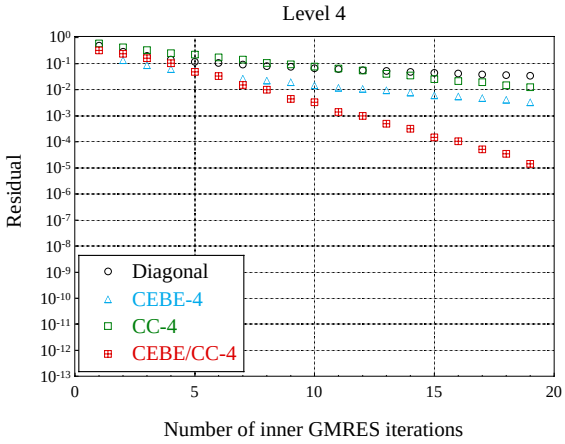


Figure 8. Convergence histories for a three-dimensional uniform mesh with $128 \times 128 \times 64$ elements.

Acknowledgement

We are thankful to Professor Youcef Saad for making his GMRES subroutine available to us and for contributing valuable comments.

Appendix: Derivations Related to $\mathbf{E}^{12}$ and $\mathbf{E}^{21}$

Let $(\mathbf{N})^1 = \{(N)^1_B, B = 1, 2, \ldots, (n_{\text{np}})^1\}$ and $(\mathbf{N})^2 = \{(N)^2_B, B = 1, 2, \ldots, (n_{\text{np}})^2\}$. Furthermore, let $(\mathbf{V})^{21} = [(V^{21})_{AB}, A = 1, 2, \ldots, (n_{\text{np}})^2,\ B = 1, 2, \ldots, (n_{\text{np}})^1]$, where

$$(V^{21})_{AB} = (N)^2_A((y)^1_B),\ A = 1, 2, \ldots, (n_{\text{np}})^2,\ B = 1, 2, \ldots, (n_{\text{np}})^1. \tag{A.1}$$

Here $(y)^1_B$ is the coordinate of the node associated with shape function $(N)^1_B$. If we assume that the value of $(N)^2_B$ at $(y)^1_B$ is defined and that the components of $(\mathbf{N})^2$ can be represented by linear combinations of the components of $(\mathbf{N})^1$, we can write

$$(\mathbf{N})^2 = \mathbf{V}^{21}(\mathbf{N})^1. \tag{A.2}$$

Equation (A.2) can also be written as

$$(\mathbf{N})^2 = (\mathbf{N})^1\mathbf{V}^{12}, \tag{A.3}$$

where

$$\mathbf{V}^{12} = (\mathbf{V}^{21})^t. \tag{A.4}$$

From (29), (36), (37), (A.2), and (A.3) we obtain

$$\mathbf{M}^{12} = \mathbf{M}^{11}\mathbf{V}^{12}, \tag{A.5}$$

$$\mathbf{M}^{21} = \mathbf{V}^{21}\mathbf{M}^{11}, \tag{A.6}$$

$$\mathbf{M}^{22} = \mathbf{V}^{21}\mathbf{M}^{11}\mathbf{V}^{12}. \tag{A.7}$$

Then, from (31), (34), (A.5), (A.6) and (A.7) we can write

$$\mathbf{E}^{12} = \mathbf{V}^{12}, \tag{A.8}$$

$$\mathbf{E}^{21} = \left[\mathbf{V}^{21}\mathbf{M}^{11}\mathbf{V}^{12}\right]^{-1}\mathbf{V}^{21}\mathbf{M}^{11}. \tag{A.9}$$

From (A.8) and (A.9) we can show that

$$\mathbf{E}^{21}\mathbf{E}^{12} = \left[\mathbf{V}^{21}\mathbf{M}^{11}\mathbf{V}^{12}\right]^{-1}\left[\mathbf{V}^{21}\mathbf{M}^{11}\mathbf{V}^{12}\right] = \mathbf{I}^{22}, \tag{A.10}$$

where $\mathbf{I}^{22}$ is an identity matrix with dimensions $(n_{\mathrm{np}})^2 \times (n_{\mathrm{np}})^2$.

References

[1] T.J.R. Hughes, I. Levit, and J. Winget, "An element-by-element solution algorithm for problems of structural and solid mechanics", *Computer Methods in Applied Mechanics and Engineering*, **36** (1983) 241–254.

[2] 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", in S.N. Atluri and N. Perrone, editors, *Recent Developments in Computer Methods for Nonlinear Solid and Structural Mechanics*, 75–109, ASME, 1983.

[3] T.J.R. Hughes and R.M. Ferencz, "Fully vectorized EBE preconditioners for nonlinear solid mechanics: Applications to large-scale three-dimensional continuum, shell and contact/impact problems", in R. Glowinski et al., editor, *Domain Decomposition Methods for Partial Differential Equations*, 261–280, SIAM, Philladelphia, PA, 1988.

[4] T.E. Tezduyar and J. Liou, "Element-by-element and implicit-explicit finite element formulations in computational fluid dynamics", in R. Glowinski et al., editor, *Domain Decomposition Methods for Partial Differential Equations*, 281–300, SIAM, Philladelphia, PA, 1988.

[5] Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems", *SIAM Journal of Scientific and Statistical Computing*, **7** (1986) 856–869.

[6] T.E. Tezduyar and J. Liou, "Grouped element-by-element iteration schemes for incompressible flow computations", *Computer Physcis Communications*, **53** (1989) 441–453.

[7] T.E. Tezduyar, J. Liou, T. Nguyen, and S. Poole, "Adaptive implicit-explicit and parallel element-by-element factorization schemes", in T.F. Chan et al., editor, *Domain Decomposition Methods*, 443–463, SIAM, Philladelphia, PA, 1989.

[8] S. Shakib, T.J.R. Hughes, and Z. Johan, "A multi-element group preconditionined GMRES algorithm for nonsymmetric systems arising in finite element analysis", *Computer Methods in Applied Mechanics and Engineering*, **75** (1989) 415–456.

[9] J. Liou and T.E. Tezduyar, "Iterative adaptive implicit-explicit methods for flow problems", *International Journal for Numerical Methods in Fluids*, **11** (1990) 867–880.

[10] J. Liou and T.E. Tezduyar, "A clustered element-by-element iteration method for finite element computations", in R. Glowinski et al., editor, *Domain Decomposition Methods for Partial Differential Equations*, Chapter 13, 140–150, SIAM, 1991.

[11] J. Liou and T.E. Tezduyar, "Computation of compressible and incompressible flows with the clustered element-by-element method", Research Report UMSI 90/215, University of Minnesota Supercomputer Institute, 1200 Washington Avenue South, Minneapolis, MN 55415, October 1990.

[12] T.E. Tezduyar and S. Mittal, "Finite element computation of incompressible flows", Research Report UMSI 91/152, University of Minnesota Supercomputer Institute, 1200 Washington Avenue South, Minneapolis, MN 55415, 1991.

[13] Y. Saad, "A flexible inner-outer preconditioned GMRES algorithm", *SIAM Journal on Scientific Computing*, **14** (1993) 461–469.

[14] G. Meurant, "A domain decomposition method for parabolic problems", preprint, 1989.