Combined AIE/EBE/GMRES Approach to Incompressible Flows*

J. Liou†
T.E. Tezduyar†

Abstract

We present the combined adaptive implicit-explicit (AIE)/grouped element-by-element (GEBE)/generalized minimum residuals (GMRES) solution techniques for incompressible flows. In this approach, the GEBE and GMRES iteration methods are employed to solve the equation systems resulting from the implicitly treated elements, and therefore no direct solution effort is involved. The benchmarking results demonstrate that this approach can substantially reduce the CPU time and memory requirements in large-scale flow problems. Although the description of the concepts and the numerical demonstrations are based on the incompressible flows, the approach presented here is applicable to a larger class of problems in computational mechanics.

1. Introduction

In this paper we present the adaptive implicit-explicit (AIE) procedures [1] which are employed in combination with iteration techniques such as the grouped element-by-element (GEBE) [2] and generalized minimum residual (GMRES) [3] methods. The solution procedures are described in the context of vorticity-stream function formulation of the time-dependent incompressible Navier-Stokes equations.

The AIE method is based on dynamic (adaptive) grouping of the elements into implicit and explicit subsets. The selection of the implicit elements is made, at a given instant in time, depending on the element level Courant number and some measure of the local variations in the solution. Since the computational cost associated with the explicit

*This research was sponsored by NASA-Johnson Space Center under contract NAS-9-17892 and by NSF under grant MSM-8796352.

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

elements is much smaller than that of the implicit elements, by placing the implicit elements only where and when they are needed, substantial savings in the CPU time and memory can be achieved. However something still needs to be done for the cost associated with the implicitly treated elements.

In the GEBE method computations are performed in an element group-by-element group fashion, and this makes the method highly vectorizable and parallelizable. This is because this method is based on arrangement of the elements into groups with the condition that no two elements in the same group can share a common node. In the GEBE iteration method the preconditioning matrix is chosen to be a sequential product of the element group matrices; this approach is a variation of the one taken in the regular EBE methods employed in computational fluid dynamics [4-6] and solid mechanics [7]. The GEBE approach eliminates the need for the formation, storage, and factorization of large global matrices. The element level matrices can be either stored or recomputed; in the case they are stored, the storage needed is still only linearly proportional to the number of elements. To minimize the overhead associated with the synchronization involved in moving from one group to another, we attempt to minimize the number of groups. Furthermore, to increase the vector efficiency of the computations performed, within each group the elements are processed in packets of 128 elements.

The GMRES method [3] is based on the minimization of the residual norm over a Krylov space. This method, with a properly chosen preconditioner, can improve the convergence rate of the iteration algorithms for nonsymmetric systems substantially. Applications of the GMRES method to various fluid dynamics problems, including compressible and incompressible flows, can be found in [6,8].

The GEBE and GMRES iteration methods can be utilized solve the equation systems resulting from the implicitly treated elements. This strategy gives us the iterative versions of the AIE method [9]. This way the cost associated with the implicitly treated elements is also minimized. It is important to note that this approach leads to an iterative AIE scheme which involves no direct solution effort at all.

We demonstrate the performance and efficiency of these techniques by solving three numerical example problems: flow past a circular cylinder at Re=100, driven cavity flow at Re=1,000, and plane jet impinging on a wedge at Re=250. The benchmarking results are presented for the cylinder problem.

2. The Formulation

Consider a two-dimensional spatial domain Ω and a time interval (0,T) with $\mathbf{x}$ and t representing the coordinates associated with Ω and (0,T). In two-dimensional space the vorticity-stream function formulation of the incompressible Navier-Stokes equations consists of a time-dependent convection-diffusion equation for the vorticity ω:

$$\partial \omega / \partial t + \mathbf{u} \cdot \nabla \omega - \nu \nabla^2 \omega = 0 \qquad \text{on } \Omega \times (0,T), \tag{1}$$

and a Poisson's equation for the stream function ψ:

$$\nabla^2 \psi + \omega = 0 \qquad \text{on } \Omega \times (0,T), \tag{2}$$

where

$$\mathbf{u} = \{ \partial \psi / \partial x_2 \, , \, - \partial \psi / \partial x_1 \} \tag{3}$$

is the velocity and ν is the kinematic viscosity. The boundary conditions associated with (1) and (2) are rather involved; we refer the interested reader to [10]. We note that the convection-diffusion equation governing the transport of a passive contaminant is a special case of (1) in which the velocity field is known.

A proper finite element formulation of the problem results in the following equation system for the incremental values of the nodal unknown vectors:

$$\begin{vmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{vmatrix} \begin{vmatrix} \Delta\omega_* \\ \Delta\psi \\ \Delta\omega_G \end{vmatrix} = \begin{vmatrix} R_1 \\ R_2 \\ R_3 \end{vmatrix} \quad . \tag{4}$$

The vectors ψ, ω_*, and ω_G represent the unknown nodal values of the stream function, vorticity at the interiors, and vorticity at the boundaries, respectively.

Solution of the coupled system (4) by a direct method such as Gaussian elimination places a prohibitive burden in terms of the CPU time and memory for large-scale problems. Alternatively we can consider a block-iteration scheme in which the following uncoupled equation systems are solved iteratively until a predetermined convergence condition is met:

$$A_{11}\ \Delta\omega_* = R_1 \quad , \qquad \text{(Block 1)} \tag{5}$$

$$A_{22}\ \Delta\psi = R_2 \quad , \qquad \text{(Block 2)} \tag{6}$$

$$A_{33}\ \Delta\omega_G = R_3 \quad . \qquad \text{(Block 3)} \tag{7}$$

Remarks:

1. Although under certain conditions the matrix A_{11} can be symmetric and positive-definite [10], we assume that, in general, this is not the case.

2. A_{22} is symmetric and positive-definite.

3. With a proper implementation A_{33} can be of tri-diagonal form; this makes the solution of this block essentially as easy as the solution of a one-dimensional problem.

3. The Solution Techniques

In our block-iteration procedure, at every iteration we need to solve three equation systems : (5), (6) and (7). The cost involved in (7) is quite minor (see Remark 3) and therefore we solve this equation with a direct method. Our main objective here is to minimize the computational cost associated with solving equations (5) and (6), which, after hiding all the subscripts, can be rewritten in the following general form:

$$A\ x = b \quad . \tag{8}$$

For completeness we first briefly describe the AIE, GEBE, and GMRES methods.

The Adaptive Implicit-Explicit (AIE) Method

Let $\mathcal{E}$ be the set of all elements, e=1,2,...,n_{el} where n_{el} is the number of elements. The assembly of the global matrix A can be expressed as

$$A = \sum_{e \in \mathcal{E}} A^e \quad , \tag{9}$$

where A^e is the component of A contributed by element e.

The AIE method is based on partitioning of the set of elements into the subsets $\mathcal{E}_I$ and $\mathcal{E}_E$ such that

$$\mathcal{E} = \mathcal{E}_I \cup \mathcal{E}_E \quad , \tag{10}$$

$$\varnothing = \mathcal{E}_I \cap \mathcal{E}_E \quad . \tag{11}$$

We then replace with

$$A_{AIE} = \sum_{e \in \mathcal{E}_I} A^e + \sum_{e \in \mathcal{E}_E} (A^e)_E \tag{12}$$

in which

$$(A^e)_E = \text{lump}(M^e) \qquad \text{for Block 1} \tag{13}$$

and

$$(A^e)_E = \text{diag}(A^e) \qquad \text{for Block 2} \ . \tag{14}$$

Here lump(M^e) is the lumped version of the mass matrix for element e and diag(A^e) is the diagonal of A^e. The matrix A_{AIE} has a skyline profile which is typically as shown in Figure 1.

In Block 1 we use a direct method to solve (5), whereas in Block 2 we use A_{AIE} as the preconditioner for the conjugate gradient method [1] employed to solve (6). Because the bandwidth for the parts of A_{AIE} corresponding to the explicit regions is substantially reduced, the method leads to savings in CPU time and memory.

The grouping given by (10) and (11) is achieved dynamically (adaptively) based on element level stability and accuracy considerations.

The stability criterion is given in terms of the element Courant number, $C_{\Delta t}$, which is defined as

$$C_{\Delta t} = \frac{\| \mathbf{u} \| \Delta t}{h} \quad , \tag{15}$$

where h is the "element length" [11]. Any element with Courant number greater than the stability limit of the explicit method needs to belong to the implicit group $\mathcal{E}_I$.

For accuracy considerations we want to use a test parameter σ_ε^e which is a measure of the local variations in the solution. One possible way is to define this test parameter based on the element level L^2 - norm of the residual r; we borrow this idea from the adaptive mesh refinement techniques given in [12]. That is

$$\sigma_\varepsilon^e = \frac{\| r \|_{\Omega^e}^0 - \min_e (\| r \|_{\Omega^e}^0)}{\max_e (\| r \|_{\Omega^e}^0) - \min_e (\| r \|_{\Omega^e}^0)} , \tag{16}$$

where

$$\| r \|_{\Omega^e}^0 = (\int_{\Omega^e} r^2 \, d\Omega)^{1/2} . \tag{17}$$

Elements with σ_ε^e greater than a predetermined value belong to group $\mathcal{E}_I$. For other choices for σ_ε^e see [1].

Implementation of the AIE scheme is quite straightforward; compared to adaptive schemes based on grid-moving or element-subdividing it involves minimal bookkeeping and no geometric constraints.

The Grouped Element-by-Element (GEBE) Method

In this method the elements are arranged into N_{pg} groups with the provision that no two elements within a group can share a common node. This way, within each group,

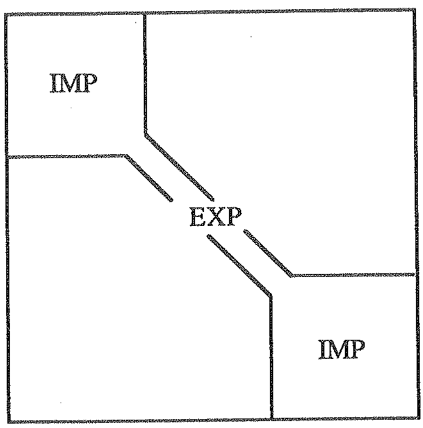

Figure 1. Typical skyline profile of the matrix A_{AIE} .

computations performed in element-by-element fashion can be done in parallel. In parallel computations we would like to minimize the synchronization overhead associated with finishing with one group and starting with another one. For this purpose the element grouping algorithm described in [2] tries to minimize the number of groups. We note that this grouping is a static (one-time) kind and therefore the computational cost involved in achieving it is a one-time cost. Furthermore, within each group the elements are processed in packets of 128 (or an appropriate size) elements. This increases the vector efficiency of the computations.

Based on the grouping, the matrix A can be written as

$$A = \sum_{K=1}^{N_{pg}} A_K \ , \tag{18}$$

with the "group matrices" defined as

$$A_K = \sum_{e \in \mathcal{E}_K} A^e \ , \qquad K = 1,2,\ldots, N_{pg} \ . \tag{19}$$

where $\mathcal{E}_K$ is the set of elements which belong to group K.

We start with the following scaled version of (8):

$$\tilde{A}\ \tilde{x} = \tilde{b} \ , \tag{20}$$

where

$$\tilde{A} = W^{-1/2} A\, W^{-1/2} \ , \tag{21}$$

$$\tilde{x} = W^{1/2} x \ , \tag{22}$$

$$\tilde{b} = W^{-1/2} b \ , \tag{23}$$

and W is the scaling matrix. Selection of this scaling matrix depends on the properties of A ; the two choices we have considered are lump(M) and diag(A).

In the preconditioned iteration method, at iteration m, the following equation system is solved for $\Delta\tilde{y}_m$:

$$\tilde{P}\ \Delta\tilde{y}_m = \tilde{r}_m \ , \tag{24}$$

where $\tilde{P}$ is the preconditioning matrix, and the residual vector $\tilde{r}_m$ is defined as

$$\tilde{r}_m = \tilde{b} - \tilde{A}\ \tilde{x}_m \ . \tag{25}$$

If A is symmetric and positive-definite then the vector $\tilde{x}_m$ is updated by using a conjugate-gradient method; otherwise we update this vector according to the expression

$$\tilde{x}_{m+1} = \tilde{x}_m + s\,\Delta\tilde{y}_m \ , \tag{26}$$

for which the search parameter s is determined with the formula

$$s = \frac{(\tilde{A}\,\Delta\tilde{y}_m)\bullet\tilde{r}_m}{\|\tilde{A}\,\Delta\tilde{y}_m\|^2} \quad ; \tag{27}$$

this formula is obtained by minimizing $\|\tilde{r}_{m+1}\|^2$ with respect to s .

Remark:

4. In the evaluation of the residual vector (25), the matrix-vector multiplication is performed in element-by-element fashion as shown below:

$$\tilde{A}\,\tilde{x} = \sum_{e=1}^{n_{el}} \tilde{A}^e\,\tilde{x}\,. \tag{28}$$

Therefore the residual vector computations are highly vectorizable. In our computations we choose to store the element level matrices.

In our GEBE approach, for Block 1 (i.e. equation (5)) we use the 2-Pass GEBE preconditioner, whereas for Block 2 (i.e. equation (6)) we use the GEBE preconditioner based on Crout factorization. We give a brief description of these two preconditioners.

<u>The 2-Pass GEBE preconditioner (2P-GEBE)</u>

This preconditioning matrix, in its scaled form, is defined as

$$\tilde{P} = \prod_{K=1}^{N_{pg}} \tilde{E}_K \prod_{K=N_{pg}}^{1} \tilde{E}_K \quad , \tag{29}$$

where

$$\tilde{E}_K = I + \frac{1}{2}\,\tilde{B}_K \quad , \qquad K = 1,2,\ldots,N_{pg}\ , \tag{30}$$

with

$$\tilde{B}_K = \tilde{A}_K - \tilde{W}_K \ , \qquad K = 1,2,\ldots,N_{pg} \tag{31}$$

and

$$\tilde{W}_K = W^{-1/2}\,(\,W_K\,)\,W^{-1/2} \qquad K = 1,2,\ldots,N_{pg}\ . \tag{32}$$

The definition given by (35) leads to "Winget regularization"; we have also been experimenting with the an alternative definition given as

$$\tilde{B}_K = \tilde{A}_K \quad , \qquad K = 1,2,\ldots,N_{pg}\ . \tag{33}$$

Remark :

5. Since there is no inter-element coupling within each group, $\tilde{E}_K$ can also be written as:

$$\tilde{E}_K = \prod_{e \in \varepsilon_K} \left(I + \frac{1}{2} \tilde{B}^e \right) , \qquad K = 1,2,\ldots,N_{pg} . \tag{34}$$

The GEBE preconditioner based Crout factorization (Crout-GEBE)

Consider the following Crout factorization:

$$I + \tilde{B}_K = \hat{L}_K \hat{D}_K \hat{U}_K , \qquad K = 1,2,\ldots,N_{pg} . \tag{35}$$

The Crout-GEBE preconditioner, in its scaled form, is defined as

$$\tilde{P} = \prod_{K=1}^{N_{pg}} \hat{L}_K \prod_{K=1}^{N_{pg}} \hat{D}_K \prod_{K=N_{pg}}^{1} \hat{U}_K . \tag{36}$$

Details on vectorization and parallel processing of the GEBE method can be found in [13].

The Generalized Minimum Residual (GMRES) Method

For Block 1, we also have the option to employ the GMRES method; an outline of the version of the GMRES method used is given below.

Given x_0 ,

set $m = 0$.

(i) Calculate the residual:

$$r_m = W^{-1} (A x_m - b) . \tag{37}$$

(ii) Construct the Krylov space :

$$e^{(1)} = r_m / \| r_m \| , \tag{38}$$

$$f^{(j)} = W^{-1} A e^{(j-1)} - \sum_{i=1}^{j-1} (W^{-1} A e^{(j-1)}, e^{(i)}) e^{(i)} , \qquad 2 \le j \le k , \tag{39}$$

$$e^{(j)} = f^{(j)} / \| f^{(j)} \| , \tag{40}$$

where k is the dimension of the Krylov space.

(iii) Update the unknown vector:

$$x_{m+1} = x_m + \sum_{j=1}^{k} s_j e^{(j)} , \tag{41}$$

where $s = \{s_j\}$ is the solution of the equation system

$$\mathbf{Q}\,\mathbf{s} = \mathbf{z} \tag{42}$$

with

$$\mathbf{Q} = [(\mathbf{W}^{-1}\mathbf{A}\,\mathbf{e}^{(i)}, \mathbf{W}^{-1}\mathbf{A}\,\mathbf{e}^{(j)})] \quad , \qquad 1 \leq i,j \leq k \quad , \tag{43}$$

$$\mathbf{z} = \{(\mathbf{W}^{-1}\mathbf{A}\,\mathbf{e}^{(i)}, -\mathbf{r}_m\,)\} \quad , \qquad 1 \leq\, i \leq k \ . \tag{44}$$

(iv) Go to the next iteration:

$n \leftarrow n+1$ and go to (37) .

The iterations continue until $\| \mathbf{r}_m \|$ becomes less than a predetermined value. We note that the matrix $\mathbf{Q}$ is symmetric and positive-definite. Again, for the scaling matrix $\mathbf{W}$ we consider the choices lump(M) and diag(A). For other kinds of preconditioners see [7,8].

Combinations of the AIE and Iterative Methods

Within the framework of our AIE scheme we can employ an iterative technique, such as the GEBE or GMRES method, to solve the equation systems resulting from the implicitly treated elements. This way those elements which need to be treated implicitly are treated so, and yet the scheme involves no direct solution effort. In the rest of this section we describe two of the several possible combinations.

AIE/GEBE on Block 1 and GEBE on Block 2

In this method, for Block 1 we use the AIE technique with the 2P-GEBE method employed to solve the equation system resulting from the implicitly treated elements; for Block 2 we use the Crout-GEBE preconditioned conjugate-gradient method.

The element grouping concept still applies in this method. In the implicit zones we still have groups within which the elements have no common nodal points. We do not need to redo the element grouping every time the distribution of the implicit zones is changed. The implicit elements are selected from the entire set of elements which are already grouped. The parallel nature of the GEBE method therefore is not affected by mixing with the AIE scheme.

AIE/GMRES on Block 1 and GEBE on Block 2

This time, for Block 1 we use the AIE technique with the GMRES method employed to solve the equation system resulting from the implicitly treated elements; for Block 2 we use, again, the Crout-GEBE preconditioned conjugate-gradient method.

In the AIE/GMRES algorithm we initialize the unknown vector as

$$\mathbf{x}^0 = (\mathrm{lump}(\mathbf{M}))^{-1}\,\mathbf{b} \quad . \tag{45}$$

This way, for explicitly treated equations the corresponding part of $\mathbf{x}^0$ is accepted as the solution; whereas for the remaining equations the corresponding part of $\mathbf{x}^0$ is used as the initial guess for the GMRES iterations. The element grouping concept remains in effect and facilitates the vectorization and potential parallel processing.

4. Numerical Examples and Benchmarking

All computations were performed on the Minnesota Supercomputer Center CRAY-2 (4 CPUs, 512 Megawords of memory, 4.1 ns clock, and UNICOS 4.0 operating system).

Flow past a circular cylinder

For this benchmark problem we used three different finite element meshes. Mesh A consists of 1,310 elements and 1,365 nodes; around the cylinder there are 29 elements in the radial and 40 elements in the circumferential directions. Mesh B involves 5,220 elements and 5,329 nodes with 58 and 80 elements in the radial and circumferential directions. Mesh C contains 19,836 elements and 20,046 nodes with 116 and 156 elements in the radial and circumferential directions. The dimensions of the computational domain, normalized by the cylinder diameter, are 30.5 and 16.0 in the flow and cross-flow directions respectively. The free stream velocity is 0.125, and the initial value of the vorticity is zero everywhere in the domain. Reynolds number based on the uniform free stream velocity and the cylinder diameter is 100.

The critical values for the test parameters $C_{\Delta t}$ and σ_{ε}^{e} are 1.0 and 10^{-5}, respectively.

For the GMRES method the dimension of the Krylov space is 5. The convergence limit for the iterative solvers is 10^{-7} for Block 1 and 10^{-6} for Block 2. We tested seven methods: implicit, block iteration, AIE, GEBE (2P-GEBE on Block 1 and Crout-GEBE on Block 2), GMRES (GMRES on Block 1 and Crout-GEBE on Block 2), AIE/GEBE (AIE/GEBE on Block 1 and Crout-GEBE on Block 2), and AIE/GMRES (AIE/GMRES on Block 1 and Crout-GEBE on Block 2). The results for the benchmarking based on the CPU time and memory requirements are shown in Tables 3 and 4.

MESH	IMP	BLOCK	AIE	GEBE	GMRES	AIE/GEBE	AIE/GMRES
A	1.0	0.478	0.542	0.165	0.191	0.167	0.193
B	1.0	0.578	0.423	0.118	0.159	0.139	0.160
C	1.0	0.797	0.349	0.159	0.169	0.165	0.166

Table 3. The results for the benchmarking based on the CPU time for various methods applied to flow past a circular cylinder.

MESH	IMP	BLOCK	AIE	GEBE	GMRES	AIE/GEBE	AIE/GMRES
A	1.0	0.399	0.084	0.172	0.148	0.104	0.123
B	1.0	0.407	0.080	0.085	0.073	0.051	0.061
C	1.0	0.410	0.077	0.042	0.037	0.025	0.030

Table 4. The results for the benchmarking based on the memory needed for the coefficient matrices for various methods applied to flow past a circular cylinder.

For this problem the solutions obtained with the AIE method can be found in [1].

Plane jet impinging on a wedge

In this problem we illustrate how the iterative AIE method works; we employ the AIE/GEBE method (AIE/GEBE on Block 1 and Crout-GEBE on Block 2). The computational domain is an 80×80 square, and the distance between the jet and the leading tip of the wedge is 7.5. The single mesh employed contains 10,566 nodal points and 10,296 elements (see Figure 2). The jet inlet consists of a parabolic velocity profile with both the width and the mean value set to unity; Reynolds number based on these values is 250. The computation is performed with a time step size of 0.05. The critical values for the test parameters $C_{\Delta t}$ and σ_{ε}^{e} are 1.0 and 10^{-5}, respectively. The convergence limit for the iterative solvers is 10^{-7} for Block 1 and 10^{-6} Block 2. Figures 3-5 show, at various time steps, the distribution of the implicit elements, the vorticity and the streamlines.

Driven cavity flow

In this test a 64×64 uniform mesh is used on a unit-square computational domain. The lid of the cavity has unit velocity; based on this velocity and the dimensions of the cavity the Reynolds number is 1,000. We employ the AIE/GMRES method (AIE/GMRES on Block 1 and Crout-GEBE on Block 2). The value of the time step used results in an estimated maximum element Courant number of 3.0. The critical values for the test parameters $C_{\Delta t}$ and σ_{ε}^{e} are 1.0 and 10^{-6}, respectively. The convergence limit for the iterative solvers is 10^{-7} for Block 1 and 10^{-6} for Block 2. Figures 6 and 7 show, at various time steps, the distribution of the implicit elements, the vorticity and the streamlines.

5. Conclusion

We have presented the combined adaptive implicit-explicit (AIE)/grouped element-by-element (GEBE)/generalized minimum residuals (GMRES) solution techniques for incompressible flows. In this approach, the GEBE and GMRES iteration methods are employed to solve the equation systems resulting from the implicitly treated elements, and therefore no direct solution effort is involved. We have applied these techniques to three numerical examples from incompressible flows: flow past a circular cylinder at Re=100, driven cavity flow at Re=1,000, and plane jet impinging on a wedge at Re=250.The benchmarking results for the cylinder problem demonstrate that this approach can substantially reduce the CPU time and memory requirements in large-scale flow problems. Although the description of the concepts and the numerical demonstrations are based on the incompressible flows, the approach presented here is applicable to a larger class of problems in computational mechanics.

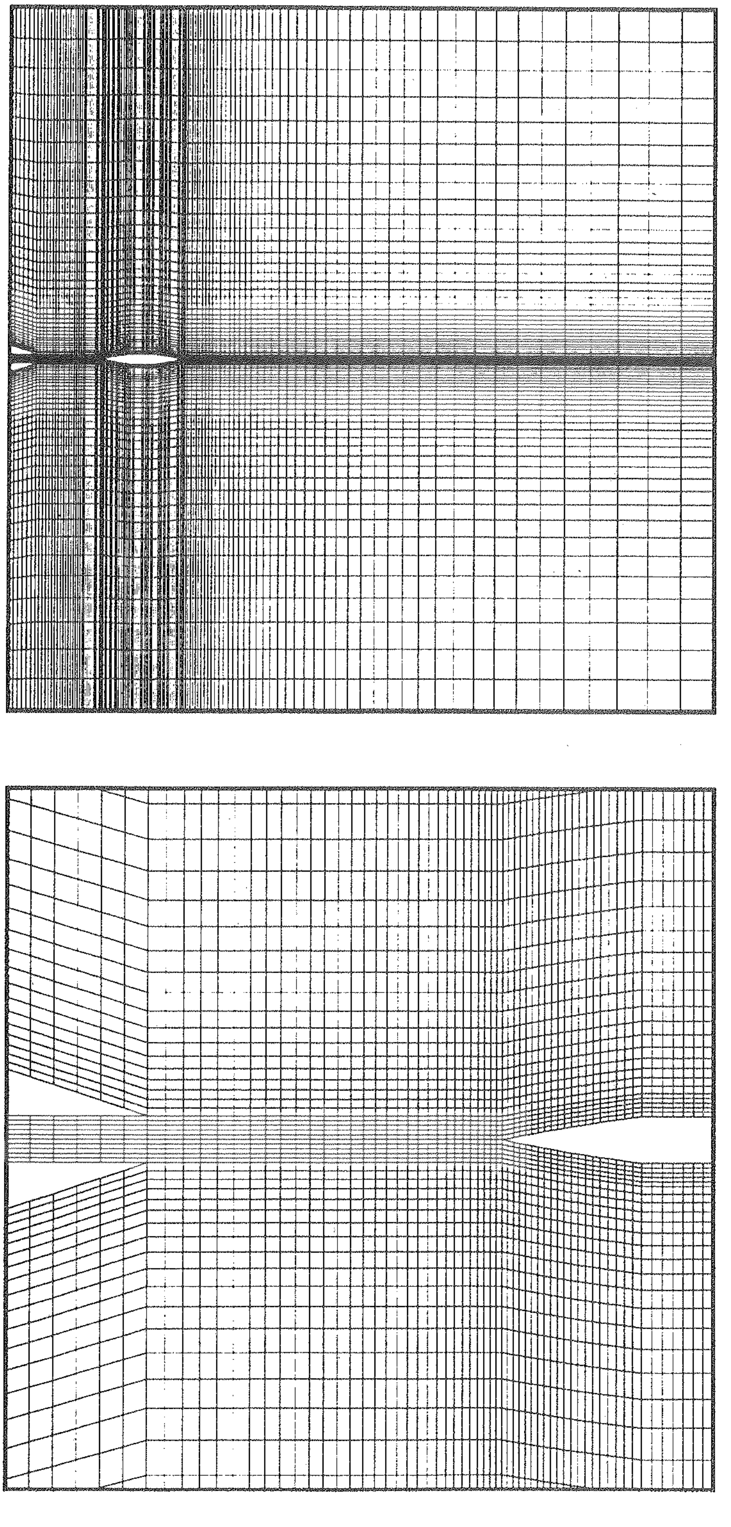

Figure 2. Plane jet impinging on a wedge : the finite element mesh (10,296 elements, 10,566 nodes).

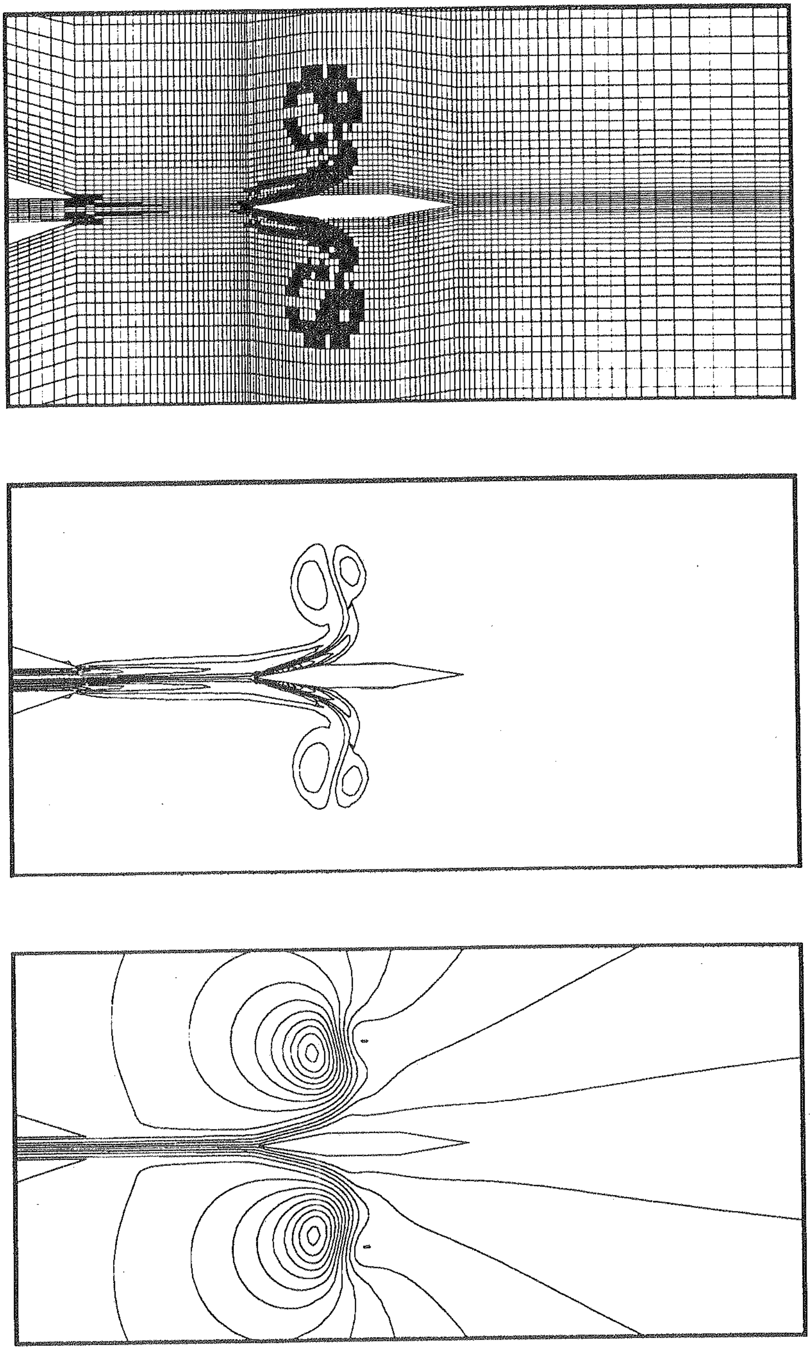

Figure 3. Plane jet impinging on a wedge at Reynolds number 250 : solution obtained by the AIE/GEBE method at t = 28.75; from top to bottom : distribution of the implicit elements, the vorticity and the streamlines.

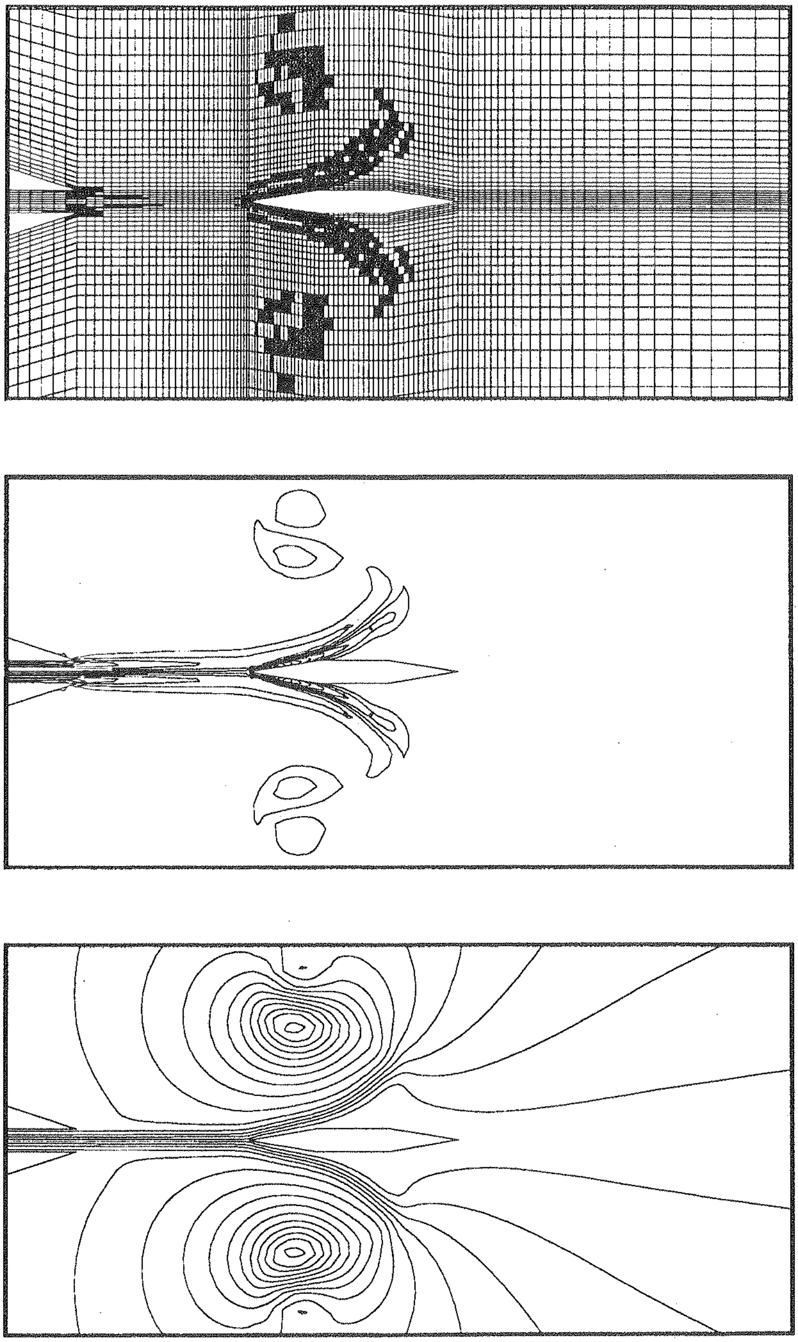

Figure 4. Plane jet impinging on a wedge at Reynolds number 250 : solution obtained by the AIE/GEBE method at t = 37.50; from top to bottom : distribution of the implicit elements, the vorticity and the streamlines.

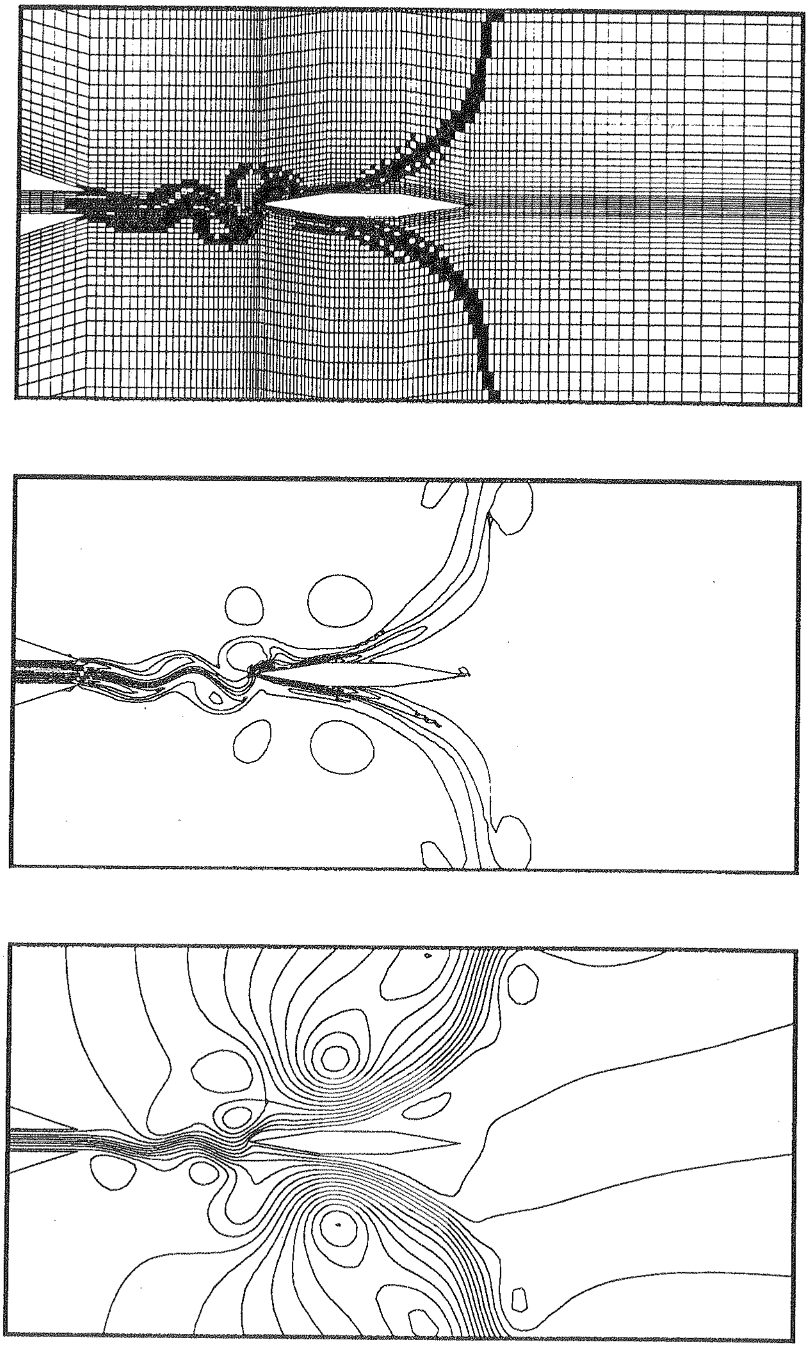

Figure 5. Plane jet impinging on a wedge at Reynolds number 250 : solution obtained by the AIE/GEBE method at t = 63.75; from top to bottom : distribution of the implicit elements, the vorticity and the streamlines.

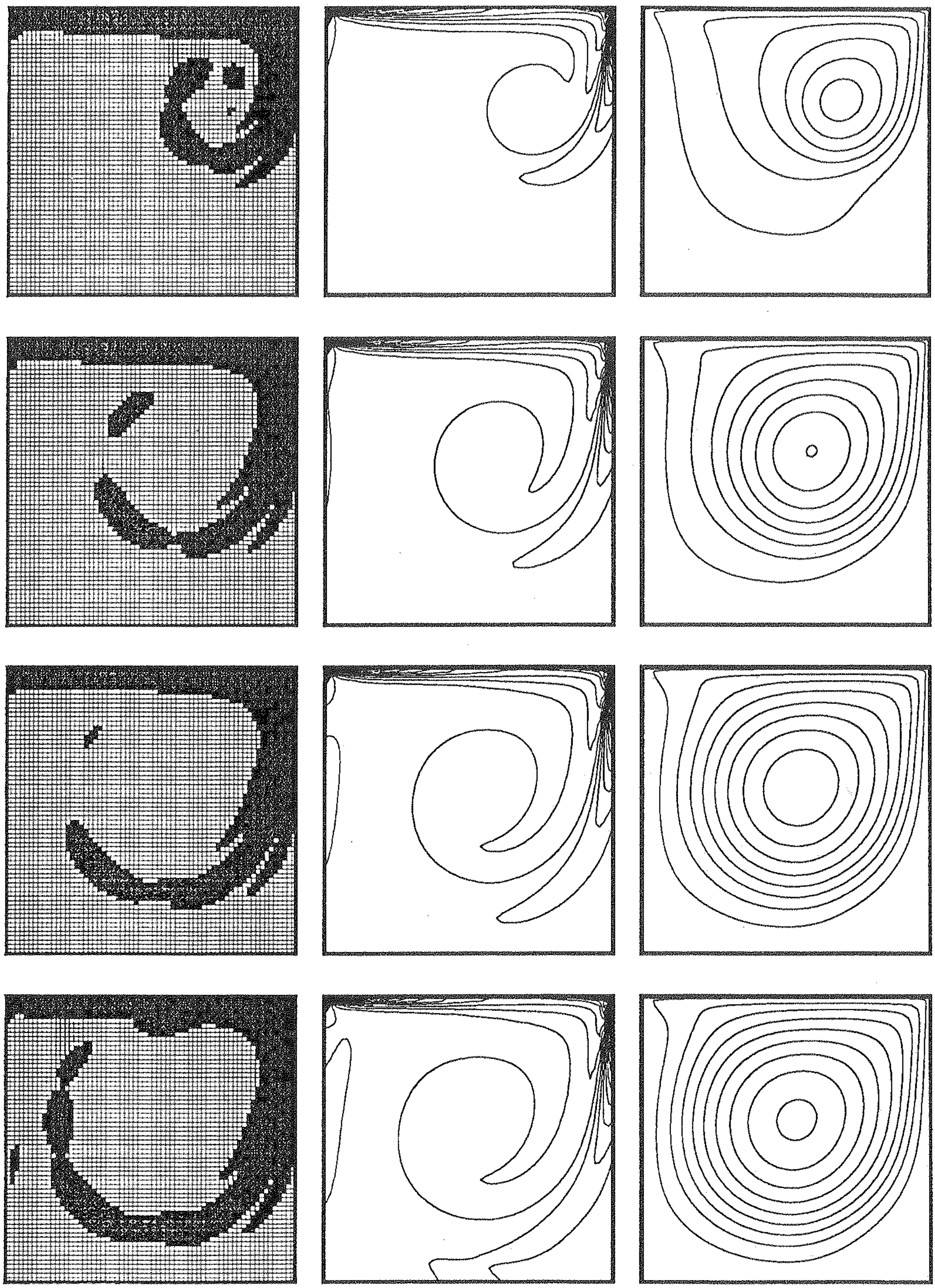

Figure 6. Driven cavity flow at Reynolds number 1,000: solution obtained by the AIE/GMRES method; distribution of the implicit elements, the vorticity and the stream function at t = 4.69, 9.38, 14.06, and 18.75.

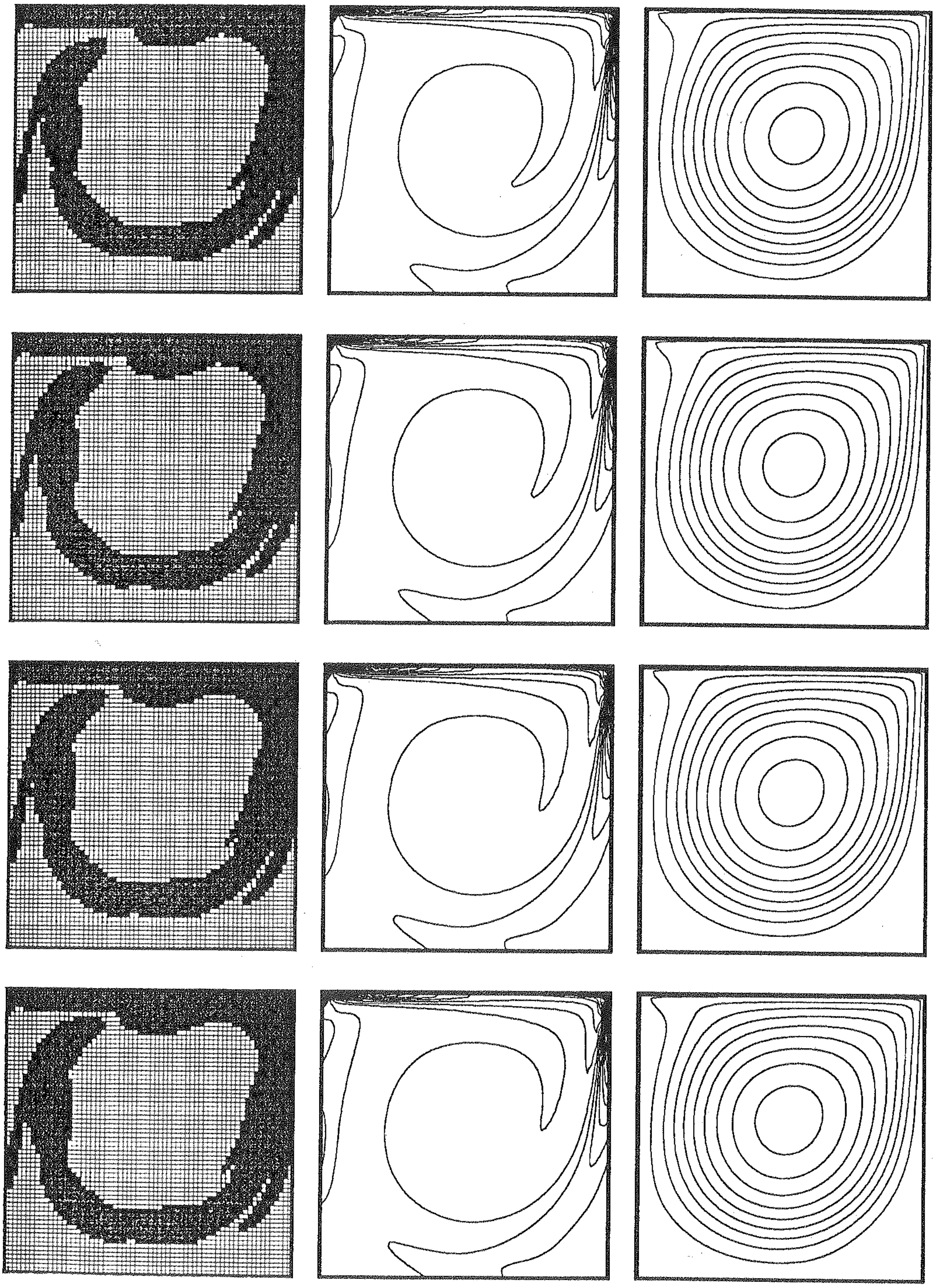

Figure 7. Driven cavity flow at Reynolds number 1,000: solution obtained by the AIE/GMRES method; distribution of the implicit elements, the vorticity and the stream function at t = 23.44, 28.13, 32.81, and 37.50.

References

1. T.E. Tezduyar and J. Liou , " Adaptive Implicit-Explicit Finite Element Algorithms for Fluid Mechanics Problems", to appear in *Computer Methods in Applied Mechanics and Engineering.*

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

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

4. 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", *Computer Methods for Nonlinear Solids and Mechanics*, S.N. Atluri and N. Perrone (eds.), AMD Vol. 54, ASME, New York, 1983, pp. 75-110.

5. T.E. Tezduyar and J. Liou, "Element-by-Element and Implicit-Explicit Finite Element Formulations for Computational Fluid Dynamics", *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. 281-300.

6. 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.*

7. 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.

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

9. J. Liou and T.E. Tezduyar, "Iterative Adaptive Implicit-Explicit Methods for Flow Problems", Minnesota Supercomputer Institute Report UMSI 89/146, October 1989.

10. T.E. Tezduyar, R. Glowinski and J. Liou, " Petrov-Galerkin Methods on Multi-Connected Domains for the Vorticity - Stream Function Formulation of the Incompressible Navier-Stokes Equations", *International Journal of Numerical Methods in Fluids*, 8 (1988), pp.1269-1290.

11. T.E. Tezduyar and D.K. Ganjoo," Petrov-Galerkin Formulations with Weighting Functions Dependent upon Spatial and Temporal Discretization: Application to Transient Convection-Diffusion Problems", *Computer Methods in Applied Mechanics and Engineering*, 59 (1986), pp. 47-71.

12. G.F. Carey and J.T. Oden , *Finite Elements: Computational Aspects*, Vol 3, Prentice-Hall, New Jersey, 1984.

13. T.E. Tezduyar, J. Liou, T. Nguyen, and S. Poole, "Adaptive Implicit-Explicit and Parallel Element-By-Element Iteration Schemes", Chapter 34 in *Domain Decomposition Methods*, T.F. Chan, R. Glowinski, J. Periaux and O.B. Widlund (eds), SIAM, 1989, pp.443-463.