Parallel Computational Fluid Dynamics (ed. H.D. Simon), MIT Press, Cambridge, Massachusetts (1992) 167-187.

9 Clustered Element-by-Element Computations for Fluid Flow

J. Liou and T. E. Tezduyar[1]

Abstract. It is shown that the clustered element-by-element method, together with the preconditioned generalized minimal residual method, can be effectively used to solve compressible and incompressible flow problems. Discretizations of the governing equations of these flow problems involve the entropy variable formulation for compressible flows and the space-time formulation for incompressible flows. The clustered element-by-element method is a generalized version of the standard element-by-element method. In this method, the elements are partitioned into clusters of elements, with a desired number of elements in each cluster, and the iterations are performed in a cluster-by-cluster fashion. The method is highly vectorizable and parallelizable if used with proper clustering and element-grouping schemes. To demonstrate the effectiveness of the proposed methods, and to evaluate the convergence rates achieved, several test computations are performed on model problems with regular and irregular meshes.

9.1 Introduction

The clustered element-by-element method (CEBE) was first proposed in [Liou91] to be used with the conjugate gradient method for solving problems with symmetric spatial operators (e.g., for problems governed by the Poisson equation). In this chapter, we use the CEBE method in conjunction with the generalized minimal residual (GMRES) method [SaSch86] to solve compressible and incompressible flow problems. These problems involve numerical difficulties such as nonlinearities, nonsymmetric spatial operators, shocks, and the incompressibility constraint. In the CEBE method the elements are merged into clusters of elements. Any number of elements can be brought together to form a cluster, and the number should be viewed as an optimization parameter to minimize the computational cost. The CEBE preconditioners are defined as series products of the cluster matrices, and the computations are performed in a cluster-by-cluster fashion. 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 [Tez89], the

[1]Department of Aerospace Engineering and Mechanics, Army-High Performance Computing Research Center, and Minnesota Supercomputer Institute University of Minnesota, Minneapolis, MN 55415. This research was sponsored by NASA-Johnson Space Center under grant NAG 9-449, and by NSF under grant MSM-8796352.

clusters can be grouped in such a way that no two clusters in any group have shared nodes. Furthermore, depending on the cluster size (i.e., the number of elements in the cluster), elements within each cluster can again be grouped in the same way. In the two limit cases, the CEBE method becomes equivalent to the GEBE method (when the cluster size is equal to one) and the direct solution method (when the cluster size is equal to the total number of elements).

The space-time finite element formulation has been successfully used for various problems (see, for example,[Shak88, HanSze90, Hugh88]). In this chapter we employ this formulation for discretization of the equations governing incompressible flows. Equal-order interpolation functions are used for velocity and pressure, and the variational formulation includes Galerkin/least-squares stabilization [Hugh89]. Because the finite element interpolation functions are discontinuous in time, the fully discrete equations can be solved one space-time slab at a time. However, the memory needed for the global matrices involved in this method is very substantial, and this could be a major obstacle for the extension of the method to practical large-scale problems. For example, in two-dimensions, the memory needed for the space-time formulation (with piecewise linear interpolation functions) of a problem is approximately four times more compared to using the finite element method only for spatial discretization. We employ the CEBE iteration scheme to reduce the cost involved in solving the linear equation systems arising from the space-time finite element discretization.

For compressible flow problems, we solve the Euler equations by using the entropy variables formulation [Hugh86a]. With properly designed streamline-upwind/Petrov-Galerkin (SUPG) and shock capturing operators, this formulation has been successfully applied to various challenging problems (see, for example, [Shak88, Mall85]). Explicit computations, though economical in memory requirements, are limited by stability conditions, and thus involve constraints in terms of the time step size that can be chosen for the computation. Implicit computations, on the other hand, allow larger time steps, but need much more memory for the global matrices. Furthermore, the implicit computations take much more computing time per time step. By using the CEBE iteration method to solve the equation systems involved, we can essentially keep the desirable stability properties of the implicit method while reducing the memory and computing time needed.

In section 9.2, the governing equations and the finite element formulations will be discussed for incompressible and compressible flow computations. In section 9.3, implementation of the CEBE method with the GMRES method will be discussed. In section 9.4, the proposed methods are tested on several compressible and incompressible flow problems. Benchmark computations are performed to compare the

convergence rates achieved. The concluding remarks are given in section 9.5.

9.2 The Governing Equations and the Finite Element Formulation

In this section we discuss the discretization techniques for the governing equations of incompressible and compressible flows.

9.2.1 Incompressible Flows

Let Ω and (0,T) denote the spatial and temporal domains, with $\mathbf{x}$ and t representing the coordinates associated with Ω and (0,T). We consider the velocity-pressure formulation of the Navier-Stokes equations:

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) - \nabla \cdot \sigma = 0 \quad \text{on } \Omega \times (0,T), \tag{9.1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{on } \Omega \times (0,T), \tag{9.2}$$

where ρ is the density, and σ is the stress tensor defined as

$$\sigma = -p\mathbf{I} + 2\mu\epsilon(\mathbf{u}), \tag{9.3}$$

with

$$\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T). \tag{9.4}$$

Here μ represents the viscosity while $\mathbf{I}$ denotes the identity tensor. Both the Dirichlet and Neumann type boundary conditions are taken into account:

$$\mathbf{u} = g \quad \text{on } \Gamma_g \times (0,T), \tag{9.5}$$

$$\mathbf{n} \cdot \sigma = h \quad \text{on } \Gamma_h \times (0,T), \tag{9.6}$$

where Γ_g and Γ_h are complementary subsets of the boundary Γ. The initial condition consists of a divergence-free velocity field specified over the entire domain:

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0 \quad \text{on } \Omega. \tag{9.7}$$

In the space-time finite element formulation, we partition the time interval (0,T) into subintervals $I_n = (t_n, t_{n+1})$, where t_n and t_{n+1} belong to an ordered series of time levels $0 = t_0 < t_1 < ... < t_N = T$. We define the space-time slab Q_n as the domain enclosed by the surfaces Ω_n, Ω_{n+1}, and P_n, where P_n is the surface

described by the boundary Γ as t traverses I_n. P_n can be decomposed into $(P_n)_g$ and $(P_n)_h$ with respect to the type of boundary conditions being applied. The finite element interpolation functions are discontinuous in time, and the fully discretized equations are solved one space-time slab at a time.

Let ε denote the set of elements resulting from the finite element discretization of the space-time slab. For each slab we define the following finite element function spaces:

$$H^{1h}(Q_n) = \{\phi^h | \phi^h \in C^0(Q_n), \phi^h|_{Q_n^e} \in P^1(\Omega) \times P^i(t), i = 0 \text{ or } 1, \ \forall Q_n^e \in \varepsilon\}, \quad (9.8)$$

$$\mathcal{S}_{\mathbf{u}}^h = \{\mathbf{u}^h | \mathbf{u}^h \in [H^{1h}(Q_n)]^{n_{sd}}, \mathbf{u}^h \doteq g \text{ on } (P_n)_g\}, \quad (9.9)$$

$$\mathcal{V}_{\mathbf{u}}^h = \{\mathbf{w}^h | \mathbf{w}^h \in [H^{1h}(Q_n)]^{n_{sd}}, \mathbf{w}^h \doteq \mathbf{0} \text{ on } (P_n)_g\}, \quad (9.10)$$

$$\mathcal{S}_p^h = \mathcal{V}_p^h = \{q^h | q^h \in H^{1h}(Q_n)\}, \quad (9.11)$$

where n_{sd} is the number of space dimensions. In time domain, we can use piecewise constant ($i = 0$) or piecewise linear ($i = 1$) functions.

The space-time finite element formulation of 9.1 - 9.7 can be written as follows: given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in \mathcal{S}_{\mathbf{u}}^h$ and $p^h \in \mathcal{S}_p^h$, such that , $\forall \mathbf{w}^h \in \mathcal{V}_{\mathbf{u}}^h \quad \forall q^h \in \mathcal{V}_p^h$

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h) \ d\Omega dt + \int_{Q_n} \epsilon(\mathbf{w}^h) : \sigma(p^h, \mathbf{u}^h) \ d\Omega dt \\
&+ \sum_{e=1}^{n_{el}} \int_{Q_n^e} \tau \ [\rho(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h) - \nabla \cdot \sigma(q^h, \mathbf{w}^h)] \\
&\cdot \ [\rho(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h) - \nabla \cdot \sigma(p^h, \mathbf{u}^h)] \ d\Omega dt \\
&+ \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h \ d\Omega dt + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot ((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-) \ d\Omega \\
&= \int_{(P_n)_h} \mathbf{w}^h \cdot h \ dP, \qquad (9.12)
\end{aligned}$$

where n_{el} is the number of elements in the space-time slab, and

$$(\mathbf{u}^h)_n^{\pm} = lim_{\delta \to 0} \mathbf{u}^h(t_n \pm \delta). \quad (9.13)$$

The formulation is applied sequentially to all space-time slabs $Q_1, Q_2, ..., Q_{n-1}$. The computations start with

$$(\mathbf{u}^h)_0^- = \mathbf{u}_0^h. \tag{9.14}$$

The coefficient τ used in 9.12 is obtained by a multi-dimensional generalization of the optimal τ given in [Hugh86b] for the space-time formulation of a one-dimensional advection-diffusion problem:

$$\tau = ((\frac{2 \parallel \mathbf{u}^h \parallel}{h})^2 + (\frac{4\nu}{h^2})^2)^{-1/2}. \tag{9.15}$$

Here h is the spatial "element length", and ν is the kinematic viscosity.
Remarks:

1. It can be shown that with functions piecewise constant in time, and with the definition of τ as given by equation 9.15, the steady-state solution is independent of the time step size. Furthermore, as the time step size goes to infinity, the steady-state solution obtained with functions piecewise linear in time becomes identical to the one obtained with functions piecewise constant in time. Therefore, one can use very large time steps to reach the steady-state solution.

2. The last term on the left-hand-side of equation 9.12 enforces, weakly, the continuity of the solution in time.

9.2.2 Compressible Flows

For the equations given in the remaining part of this section, repeated indices imply summation over the range of the spatial dimension (n_{sd}). The dynamics which governs two-dimensional, inviscid, compressible flows are described by the Euler equations. In terms of conservation variables, $\mathbf{U} = (\rho, \rho u_1, \rho u_2, \rho e)^T$, these equations are

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} = 0 \quad \text{on } \Omega \times (0,T), \tag{9.16}$$

where $\mathbf{F}_i$'s are the Euler fluxes. Alternatively, equation 9.16 can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} = 0 \quad \text{on } \Omega \times (0,T), \tag{9.17}$$

where

$$\mathbf{A}_i = \frac{\partial \mathbf{F}_i}{\partial \mathbf{U}}, \quad i = 1, ..., n_{sd}. \tag{9.18}$$

The following boundary and initial conditions are assumed to be given:

$$\mathbf{BU} = \mathbf{G} \quad \text{on } \Gamma_g \times (0, T), \tag{9.19}$$

$$\mathbf{U}(\mathbf{x}, 0) = \mathbf{U}_0(\mathbf{x}) \quad \text{on } \Omega, \tag{9.20}$$

where $\mathbf{B}$ is a general boundary condition operator (see [LeBeau] for the implicit treatment of impermeable boundary conditions), and $\mathbf{G}$ and $\mathbf{U}_0(\mathbf{x})$ are given functions.

Based on a transformation of variables, $\mathbf{U} = \mathbf{U}(\mathbf{V})$, one can write the Euler equations in terms of entropy variables [Hugh86a]:

$$\tilde{\mathbf{A}}_0 \frac{\partial \mathbf{V}}{\partial t} + \tilde{\mathbf{A}}_i \frac{\partial \mathbf{V}}{\partial x_i} = 0 \quad \text{on } \Omega \times (0, T), \tag{9.21}$$

where

$$\tilde{\mathbf{A}}_0 = \mathbf{U}_{,\mathbf{V}} \quad \text{and} \quad \tilde{\mathbf{A}}_i = \mathbf{A}_i \tilde{\mathbf{A}}_0, \quad i = 1, ..., n_{sd}. \tag{9.22}$$

$\tilde{\mathbf{A}}_0$ is a symmetric and positive-definite matrix, and $\tilde{\mathbf{A}}_i$'s are symmetric matrices.

For the finite element formulation of the entropy variables form of the Euler equations, we define the following function spaces:

$$\mathcal{S}^h = \{\mathbf{V}^h | \mathbf{V}^h \in [H^{1h}(\Omega)]^{n_{dof}}, \mathbf{V}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{dof}}, \mathbf{q}(\mathbf{V}^h) = g(t) \text{ on } \Gamma_g\}, \tag{9.23}$$

$$\mathcal{V}^h = \{\mathbf{W}^h | \mathbf{W}^h \in [H^{1h}(\Omega)]^{n_{dof}}, \mathbf{W}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{dof}}, \mathbf{q}(\mathbf{W}^h) = \mathbf{0} \text{ on } \Gamma_g\}, \tag{9.24}$$

where n_{dof} is the number of degrees of freedom and $\mathbf{q}(\mathbf{V}^h)$ is the boundary condition operator for the entropy variables [Shak88]. The finite element formulation of the Euler equations can be written as follows: find $\mathbf{V}^h \in \mathcal{S}^h$, such that, $\forall \mathbf{W}^h \in \mathcal{V}^h$

$$\begin{aligned} \int_\Omega \quad & \mathbf{W}^h \cdot (\tilde{\mathbf{A}}_0 \frac{\partial \mathbf{V}^h}{\partial t} + \tilde{\mathbf{A}}_i \frac{\partial \mathbf{V}^h}{\partial x_i}) \; d\Omega \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau \; \tilde{\mathbf{A}}_i \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot (\tilde{\mathbf{A}}_0 \frac{\partial \mathbf{V}^h}{\partial t} + \tilde{\mathbf{A}}_j \frac{\partial \mathbf{V}^h}{\partial x_j}) \; d\Omega \\ + \quad & \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot (\nu_d - \nu_\tau) \tilde{\mathbf{A}}_0 \frac{\partial \mathbf{V}^h}{\partial x_i} \; d\Omega = 0, \end{aligned} \tag{9.25}$$

where τ (a symmetric and positive-definite matrix) is the generalized SUPG operator [Hugh86b], defined as follows:

$$\tau = (\frac{\partial \xi_i}{\partial x_j}\tilde{\mathbf{A}}_j\tilde{\mathbf{A}}_0^{-1}\frac{\partial \xi_i}{\partial x_k}\tilde{\mathbf{A}}_k)^{-1/2}. \tag{9.26}$$

The term $(\nu_d - \nu_\tau)$ is the shock capturing diffusivity [Mall85], defined as follows:

$$\nu_d = (\frac{\tilde{\mathbf{A}}_i\frac{\partial \mathbf{V}^h}{\partial x_i}\cdot\tilde{\mathbf{A}}_0^{-1}\tilde{\mathbf{A}}_j\frac{\partial \mathbf{V}^h}{\partial x_j}}{\frac{\partial \xi_k}{\partial x_m}\frac{\partial \mathbf{V}^h}{\partial x_m}\cdot\tilde{\mathbf{A}}_0\frac{\partial \xi_k}{\partial x_n}\frac{\partial \mathbf{V}^h}{\partial x_n}})^{1/2}, \tag{9.27}$$

$$\nu_\tau = (\frac{\tilde{\mathbf{A}}_i\frac{\partial \mathbf{V}^h}{\partial x_i}\cdot\tau\ \tilde{\mathbf{A}}_j\frac{\partial \mathbf{V}^h}{\partial x_j}}{\frac{\partial \mathbf{V}^h}{\partial x_k}\cdot\tilde{\mathbf{A}}_0\frac{\partial \mathbf{V}^h}{\partial x_k}})^{1/2}. \tag{9.28}$$

Note that the contribution of the generalized SUPG operator is subtracted from the shock capturing operator to avoid excessive diffusion around the shocks.

9.3 The Clustered Element-by-Element Method

After linearization of the fully discretized equations, the following equation system needs be solved for the nodal values of the unknowns:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{9.29}$$

We rewrite 9.29 in a scaled form

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tag{9.30}$$

where

$$\tilde{\mathbf{A}} = \mathbf{W}^{-\frac{1}{2}}\mathbf{A}\mathbf{W}^{-\frac{1}{2}}, \tag{9.31}$$

$$\tilde{\mathbf{x}} = \mathbf{W}^{\frac{1}{2}}\mathbf{x}, \tag{9.32}$$

$$\tilde{\mathbf{b}} = \mathbf{W}^{-\frac{1}{2}}\mathbf{b}. \tag{9.33}$$

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

$$\mathbf{W} = diag\mathbf{A}. \tag{9.34}$$

With this definition of $\mathbf{W}, diag\tilde{\mathbf{A}}$ becomes an identity matrix.

For the flow problems considered in this chapter, the matrix $\mathbf{A}$ is not in general symmetric and positive-definite. Therefore, the proposed CEBE preconditioner

will be used in conjunction with the GMRES method. For completeness, we briefly describe the GMRES method used for solving 9.30:
0. Set the iteration counter m = 0, and start with an initial guess $\tilde{\mathbf{x}}_0$.

1. Calculate the residual scaled with the preconditioner matrix $\tilde{\mathbf{P}}$:

$$\tilde{\mathbf{r}}_m = \tilde{\mathbf{P}}^{-1}(\tilde{\mathbf{A}}\tilde{\mathbf{x}}_m - \tilde{\mathbf{b}}). \tag{9.35}$$

2. Construct the Krylov vector space :

$$\mathbf{e}^{(1)} = \frac{\tilde{\mathbf{r}}_m}{\| \tilde{\mathbf{r}}_m \|}, \tag{9.36}$$

$$\mathbf{f}^{(j)} = \tilde{\mathbf{P}}^{-1}\tilde{\mathbf{A}}\mathbf{e}^{(j-1)} - \sum_{i=1}^{j-1}(\tilde{\mathbf{P}}^{-1}\tilde{\mathbf{A}}\mathbf{e}^{(j-1)}, \mathbf{e}^{(i)})\mathbf{e}^{(i)}, \quad 2 \leq j \leq n_{kg}, \tag{9.37}$$

$$\mathbf{e}^{(j)} = \frac{\mathbf{f}^{(j)}}{\| \mathbf{f}^{(j)} \|}, \tag{9.38}$$

where n_{kg} is the dimension of the Krylov space, and $\mathbf{e}^{(i)}, i = 1, 2, ..., n_{kg}$, are the basis vectors.

3. Update the unknown vector:

$$\tilde{\mathbf{x}}_{m+1} = \tilde{\mathbf{x}}_m + \sum_{j=1}^{n_{kg}} s_j \mathbf{e}^{(j)}, \tag{9.39}$$

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

$$\mathbf{Q}\mathbf{s} = \mathbf{z}, \tag{9.40}$$

with

$$\mathbf{Q} = [Q_{ij}] = [(\tilde{\mathbf{P}}^{-1}\tilde{\mathbf{A}}\mathbf{e}^{(i)}, \tilde{\mathbf{P}}^{-1}\tilde{\mathbf{A}}\mathbf{e}^{(j)})], \quad 1 \leq i, j \leq n_{kg}, \tag{9.41}$$

$$\mathbf{z} = \{z_i\} = \{(\tilde{\mathbf{P}}^{-1}\tilde{\mathbf{A}}\mathbf{e}^{(i)}, -\tilde{\mathbf{r}}_m)\}, \quad 1 \leq i \leq n_{kg}. \tag{9.42}$$

4. For next iteration, set $m \leftarrow m + 1$ and go to 1.

The iterations continue until the ratio $\| \tilde{\mathbf{r}}_m \| / \| \tilde{\mathbf{r}}_0 \|$ falls below a predetermined value. It should be noted that the matrix $\mathbf{Q}$ is symmetric and positive-definite.
Remark:

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

With the CEBE method, the set of elements ε is partitioned into subsets (clusters of elements) $\varepsilon_J, J = 1, 2, ..., N_{cl}$, where N_{cl} is the number of clusters, such that

$$\varepsilon = \bigcup_{J=1}^{N_{cl}} \varepsilon_J, \tag{9.43}$$

$$\emptyset = \bigcap_{J=1}^{N_{cl}} \varepsilon_J. \tag{9.44}$$

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

$$\tilde{\mathbf{A}} = \sum_{J=1}^{N_{cl}} \tilde{\mathbf{A}}_J, \tag{9.45}$$

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

$$\tilde{\mathbf{A}}_J = \sum_{e \in \varepsilon_J} \tilde{\mathbf{A}}^e, \tag{9.46}$$

where $\tilde{\mathbf{A}}^e$ is the element level matrix.

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

$$(\mathbf{I} + \tilde{\mathbf{B}}_J) = \hat{\mathbf{L}}_J \hat{\mathbf{U}}_J, \quad J = 1, 2, ..., N_{cl}, \tag{9.47}$$

where

$$\tilde{\mathbf{B}}_J = \tilde{\mathbf{A}}_J - \tilde{\mathbf{W}}_J, \tag{9.48}$$

and $\hat{\mathbf{L}}_J$ and $\hat{\mathbf{U}}_J$ are the lower and upper triangular factors of $(\mathbf{I} + \tilde{\mathbf{B}}_J)$. The CEBE preconditioner is defined as

$$\tilde{\mathbf{P}} = \prod_{J=1}^{N_{cl}} \hat{\mathbf{L}}_J \prod_{J=N_{cl}}^{1} \hat{\mathbf{U}}_J. \tag{9.49}$$

Remarks:

4. 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.

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

9.4 Numerical Examples

We tested the proposed CEBE/GMRES method on several two-dimensional model problems governed by the incompressible Navier-Stokes and compressible Euler equations. We compared the convergence rates achieved in computations using different numbers of GMRES basis vectors. In the space-time formulation of all incompressible flow problems considered, for velocity and pressure we use functions which are piecewise linear in time. Also for all incompressible flow problems, a very large time step (100,000.0) is used to reach the steady-state solution.

9.4.1 Lid-driven cavity flow at Reynolds number 400 and 1000

In both of these problems the cavity has a square shape, and the Reynolds number is based on the dimension of the cavity and the velocity of the lid. For a square mesh with n $\times$ n elements and m $\times$ m clusters, the memory needed for the global coefficient matrices is approximately $36n^3/m$. We note that there are six unknowns associated with each node in space. Figure 9.1 shows (for Re = 400 and a mesh with 32 $\times$ 32 elements) the residual ratio vs the number of outer iterations for various cluster sizes and different numbers of GMRES basis vectors (n_{kg}). It can be seen from Figure 9.1 that, as expected, the larger the cluster size is, the better the convergence rate achieved. Of course, when the cluster size is equal to the number of elements, the method reduces to a fully implicit one, and only one iteration is needed to reach the solution. When the cluster size is equal to one, on the other hand, the method becomes a standard element-by-element method. Regardless of the cluster size, the CEBE preconditioner gives better convergence rates than the Jacobi preconditioner.

Figures 9.2 and 9.3 show the computed flow field for Re = 400 and 1000, respectively. In both cases, we use a mesh with 64 $\times$ 64 elements and 4 $\times$ 4 clusters.

Both figures show the velocity components along the vertical and horizontal center lines, pressure, vorticity, and stream function.

9.4.2 Incompressible flow past a circular cylinder at Reynolds number 100

With this test problem, we evaluate the performance of the CEBE method on irregular meshes. 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. Reynolds number based on the uniform free-stream velocity and the cylinder diameter is 100. Symmetry conditions are imposed on the upper and lower boundaries, and the traction-free condition is imposed at the outflow boundary. First, to study the convergence rates achieved, we use a finite element mesh with 1280 elements and 1360 nodes (see Figure 9.4). Figure 9.5 shows the residual ratio vs the number of outer iterations for various cluster sizes and different numbers of GMRES basis vectors. Figure 9.6 shows, for a more refined mesh (with 5400 elements, 5510 nodes, and 72 clusters), pressure, vorticity, stream function and stationary stream function.

9.4.3 Compressible flow past a circular cylinder at Mach number 3.0

In this problem, we impose symmetry conditions along the horizontal axis of the cylinder, and use a computational domain containing only half of the cylinder. The dimensions of the computational domain, normalized by the cylinder diameter, are 16 and 8 in the horizontal and vertical directions, respectively. The mesh consists of 4800 elements and 4941 nodes. With four unknowns associated with each node, the number of equations for this mesh is 19,478. Impermeable boundary conditions are specified on the cylinder. Figure 9.7 shows, at a typical time step, the residual ratio vs the number of iterations for various cluster sizes and different numbers of GMRES basis vectors. Figure 9.8 shows the mesh, and pressure and density contours for the steady-state solution.

9.5 Concluding Remarks

We have demonstrated that the clustered element-by-element preconditioners can be effectively used with the generalized minimal residual method to solve large-scale compressible and incompressible flow problems. We employed these iteration methods in conjunction with discretizations involving the entropy variable formulation for compressible flows and the space-time formulation for incompressible flows. In the CEBE method, the elements are partitioned into clusters of elements, with

a desired number of elements in each cluster, and the iterations are performed in a cluster-by-cluster fashion. The clustering concept is very similar, in philosophy, to the domain decomposition concept. With this approach, we can select an algorithm anywhere in the spectrum of algorithms ranging from the direct solution technique to the standard element-by-element iteration method. The method is highly vectorizable and parallelizable.

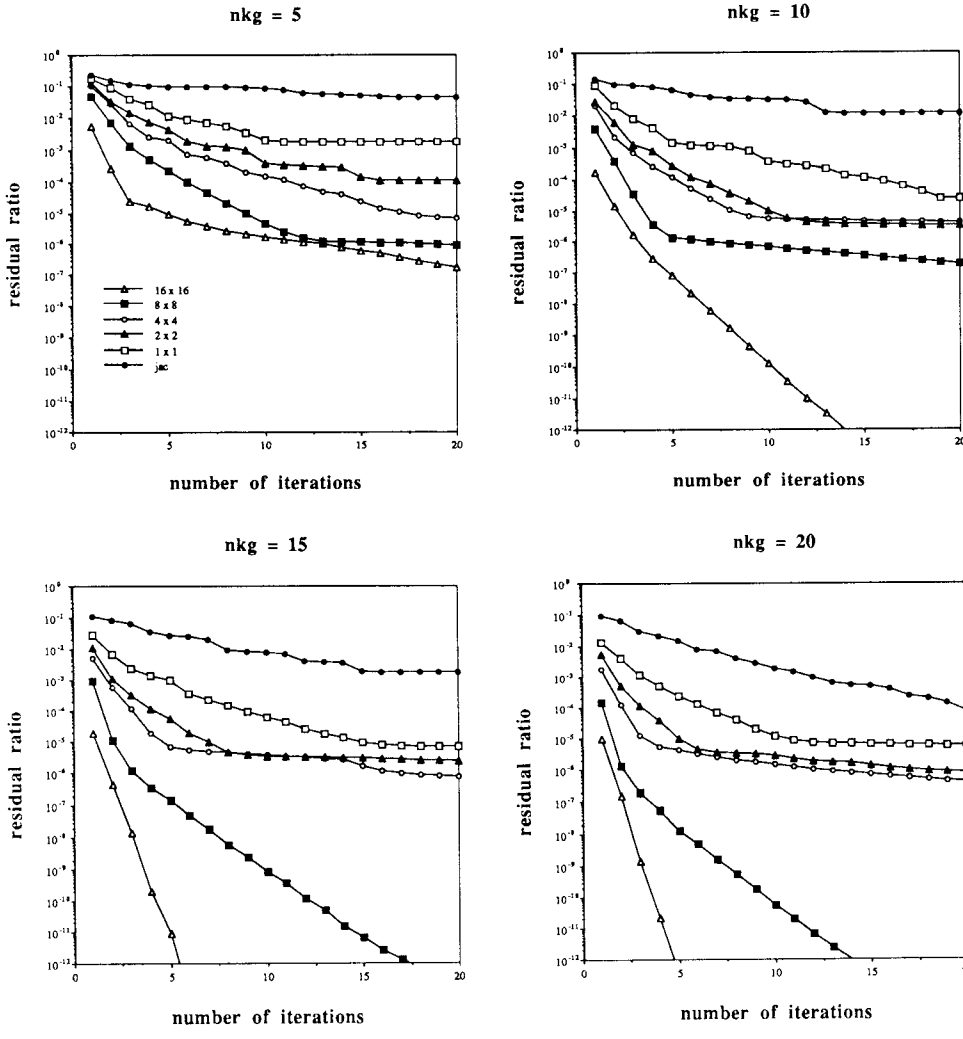

Figure 9.1
Lid-driven cavity flow at Reynolds number 400: residual ratio versus number of outer iterations for various cluster sizes and different numbers of GMRES basis vectors.

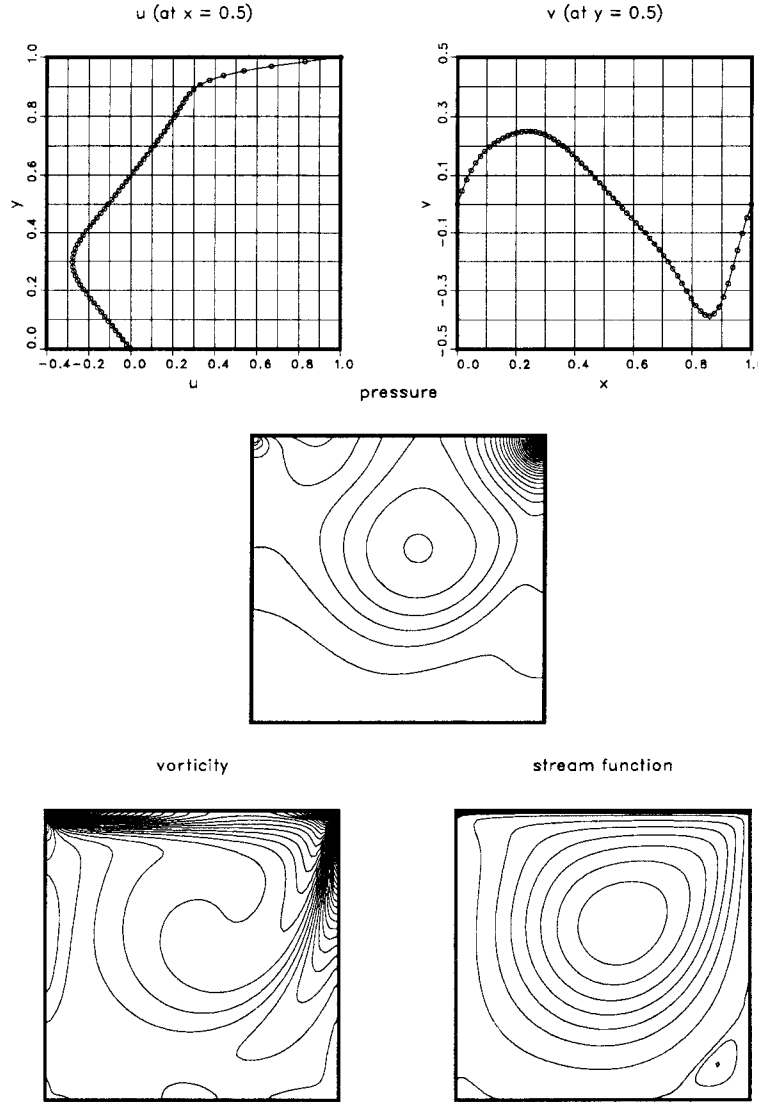

Figure 9.2
Lid-driven cavity flow at Reynolds number 400: velocity components along the vertical and horizontal center lines, pressure, vorticity, and stream function.

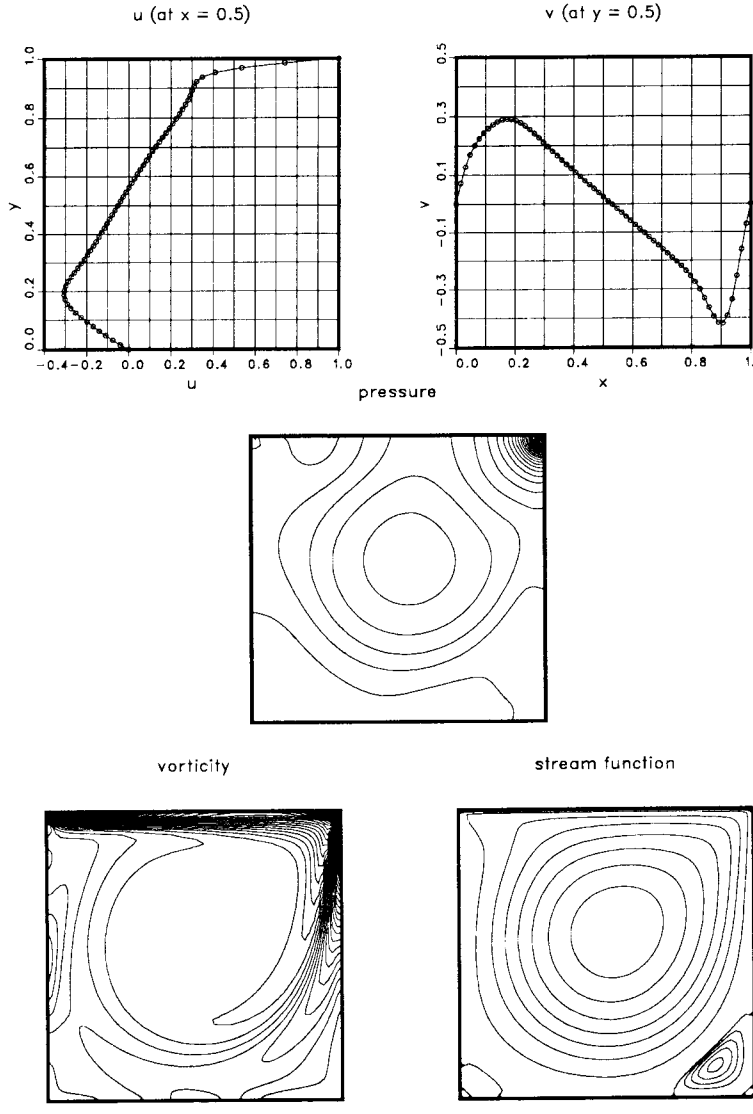

Figure 9.3
Lid-driven cavity flow at Reynolds number 1000: velocity components along the vertical and horizontal center lines, pressure, vorticity, and stream function.

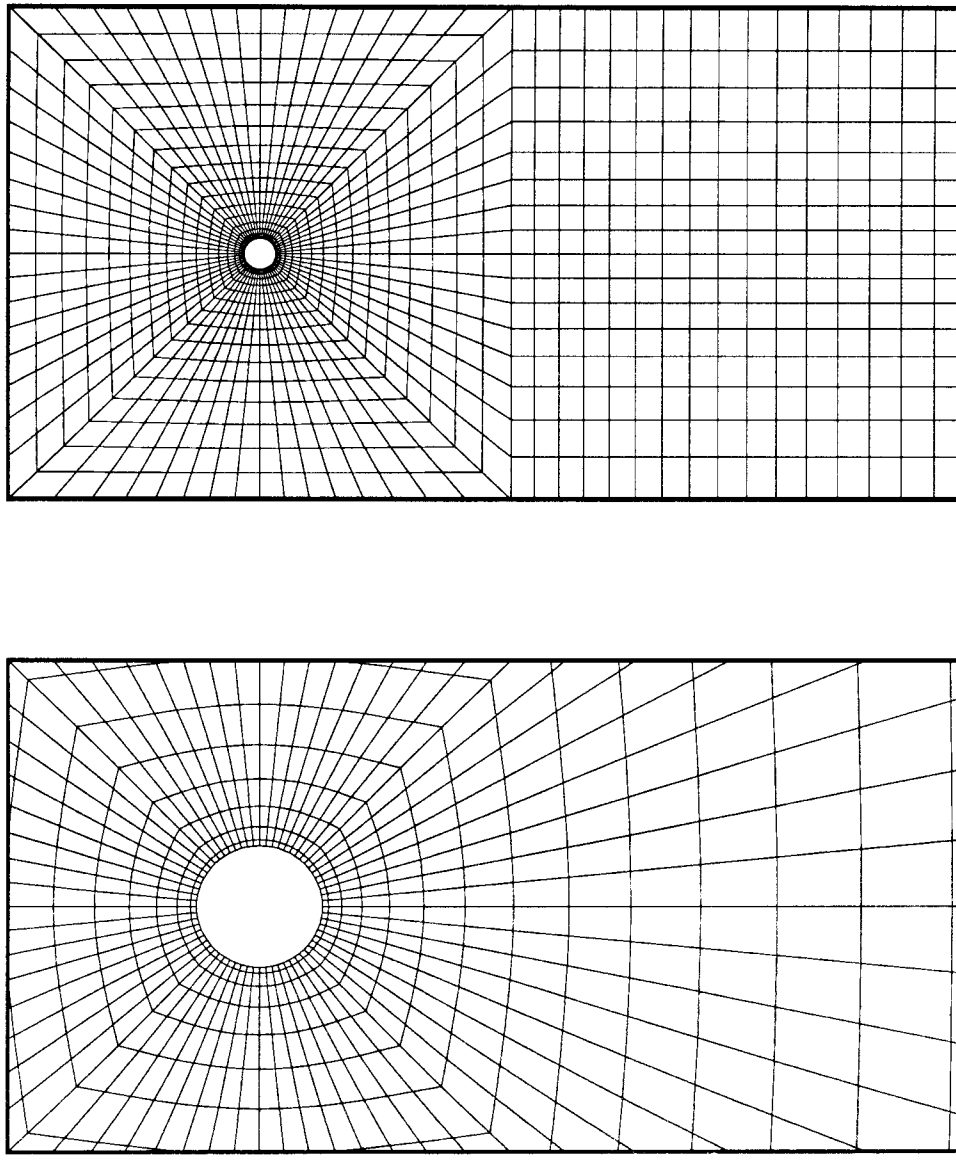

Figure 9.4
Incompressible flow past a circular cylinder at Reynolds number 100: the finite element mesh used in the benchmark computations (1280 elements and 1360 nodes).

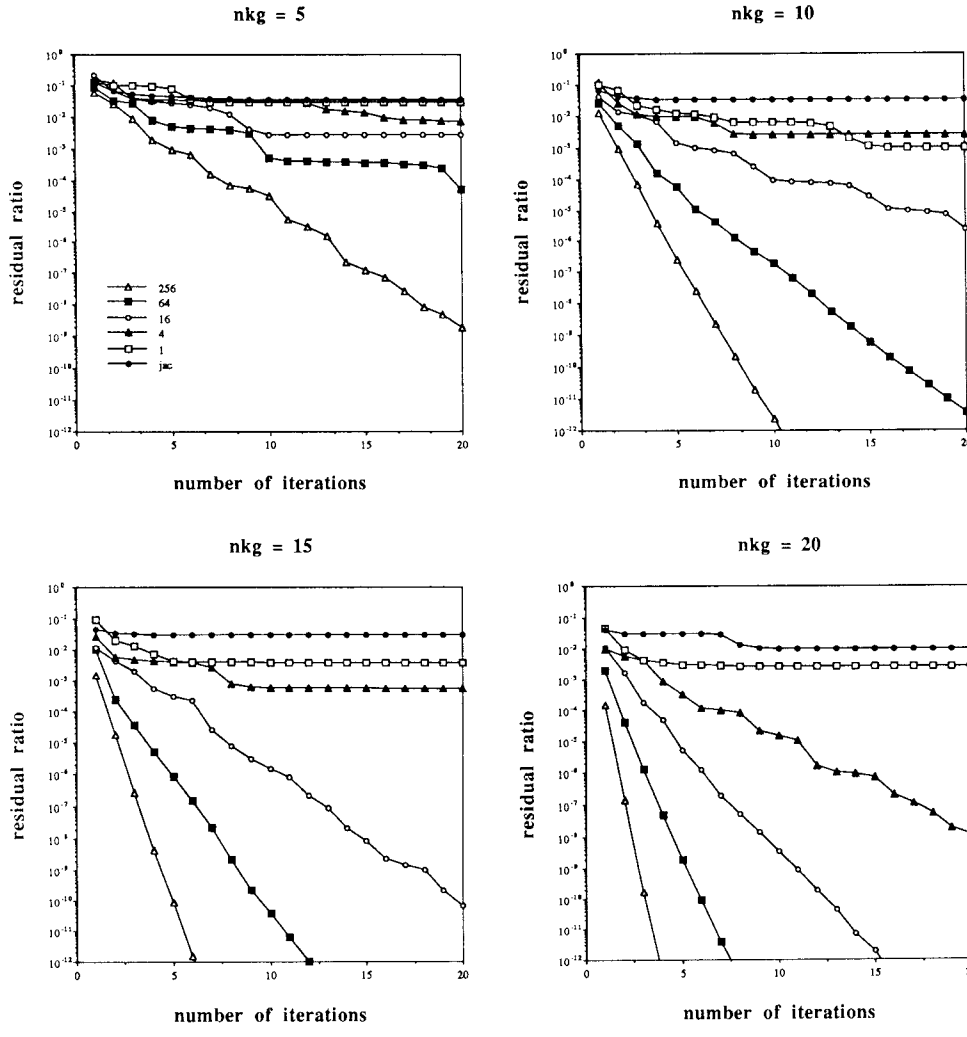

Figure 9.5
Incompressible flow past a circular cylinder at Reynolds number 100: residual ratio versus number of outer iterations for various cluster sizes and different numbers of GMRES basis vectors.

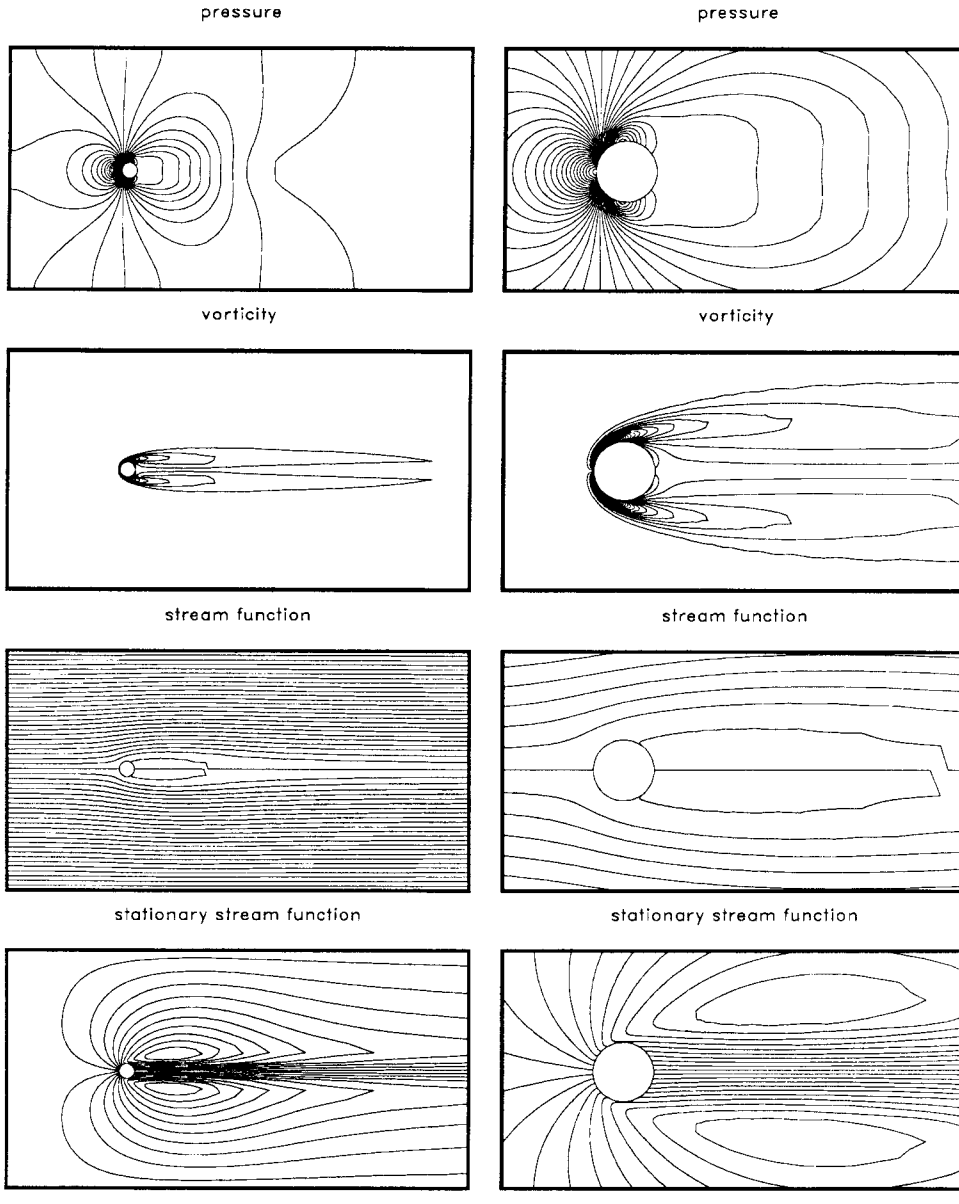

Figure 9.6
Incompressible flow past a circular cylinder at Reynolds number 100: pressure, vorticity, stream function, and stationary stream function.

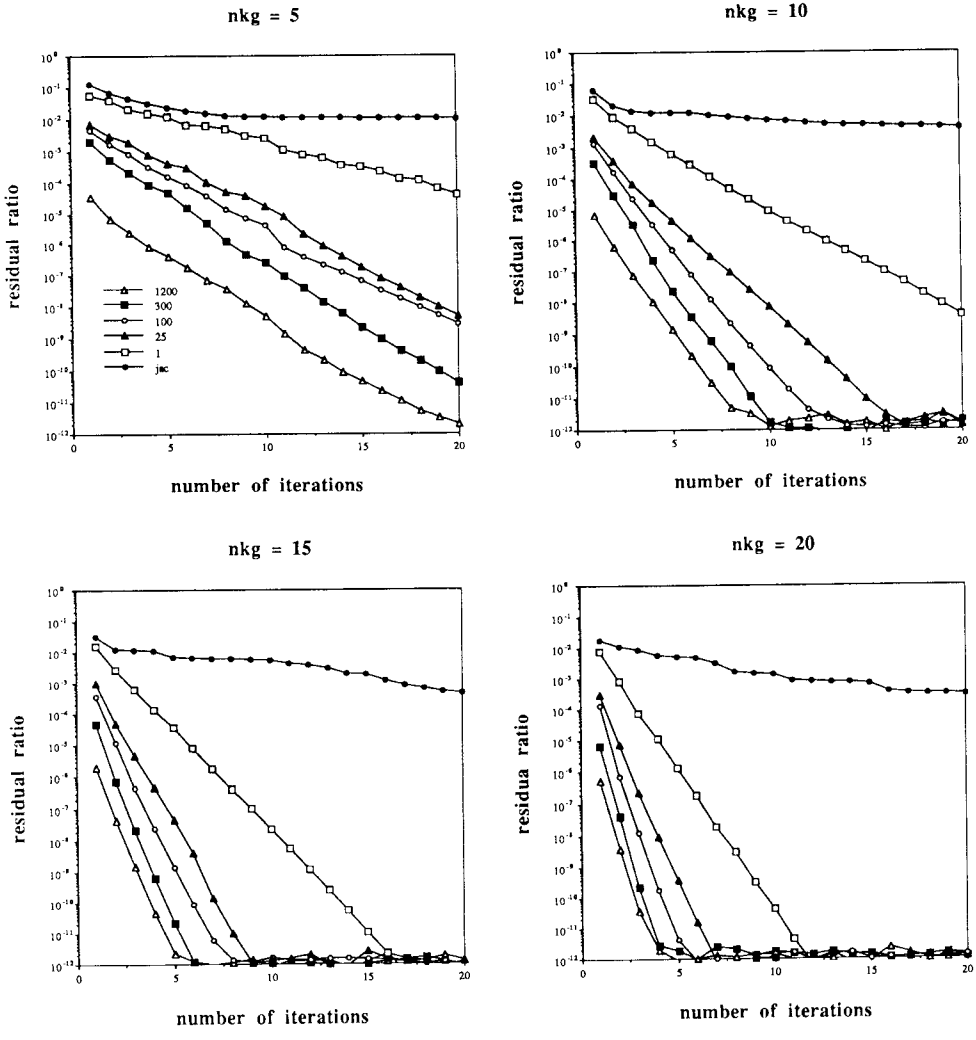

Figure 9.7
Compressible flow past a circular cylinder at Mach number 3.0: residual ratio versus number of outer iterations for various cluster sizes and different numbers of GMRES basis vectors.

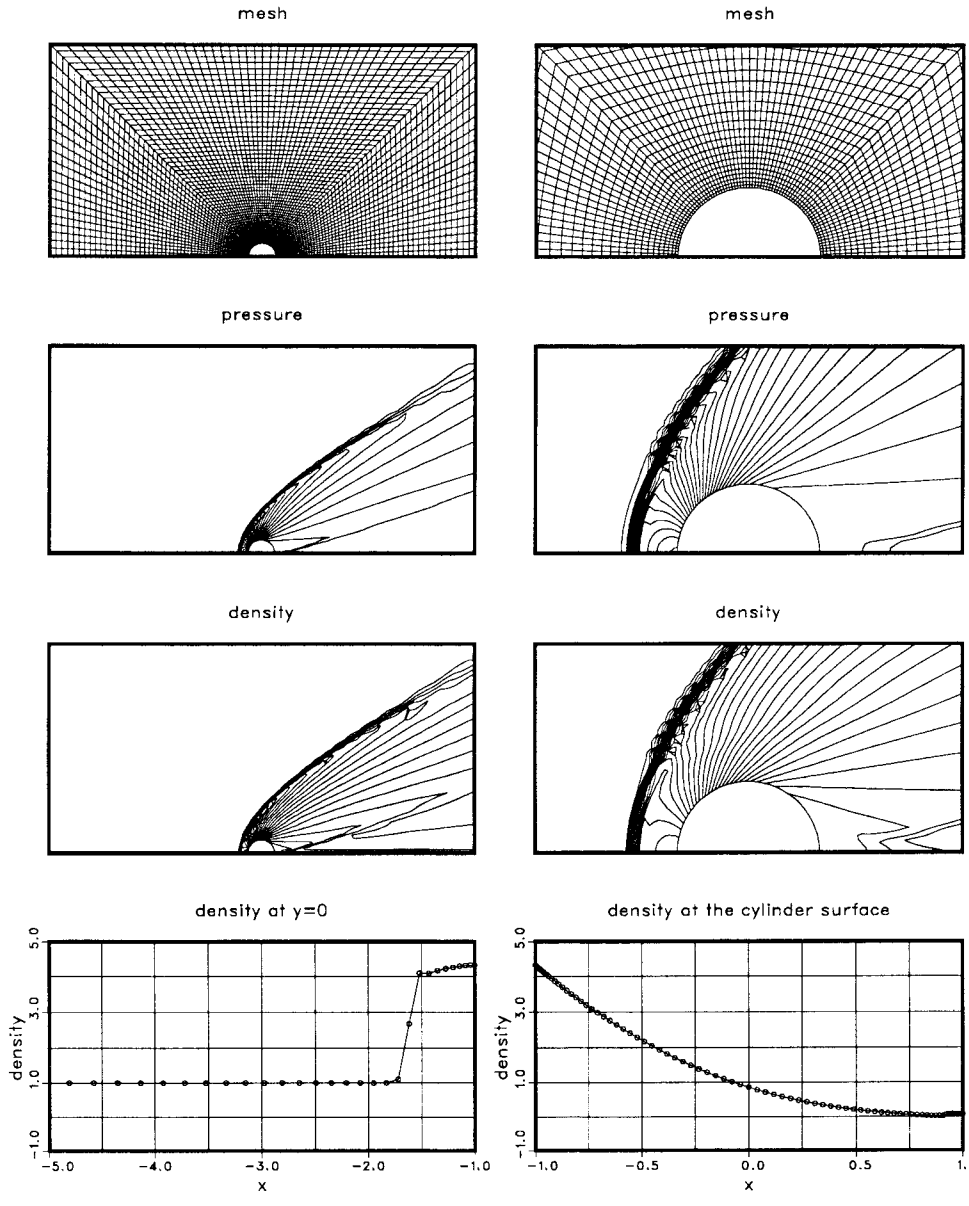

Figure 9.8
Compressible flow past a circular cylinder at Mach number 3.0: finite element mesh (4800 elements and 4941 nodes), pressure and density.

9.6 Bibliography

[HanSze90] P. Hansbo and A. Szepessy. A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. and Eng.*, 84:175 – 192, 1990.

[Hugh89] T.J.R. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least squares method for advective diffusive equations. *Comp. Meth. in Appl. Mech. and Eng.*, 73:173 – 189, 1989.

[Hugh86a] T.J.R. Hughes, L. P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: I. symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Comp. Meth. in Appl. Mech. and Eng.*, 54:223 - 234, 1986.

[Hugh86b] T.J.R. Hughes and M. Mallet. A new finite element formulation for computational fluid dynamics: III. the generalized streamline operator for multidimensional advective-diffusive systems. *Comp. Meth. in Appl. Mech. and Eng.*, 58:305 – 328, 1986.

[Hugh88] T.J.R. Hughes and G. M. Hulbert. Space-time finite element methods for elastodynamics: Formulations and error estimates. *Comp. Meth. in Appl. Mech. and Eng.*, 66:339 – 363, 1988.

[LeBeau] J. LeBeau. The finite element computation of compressible flows. Master's thesis, University of Minnesota, 1991.

[Liou91] J. Liou and T. E. Tezduyar. A clustered element-by-element iteration method for finite element computations. Chapter 13 in *Domain Decomposition Methods for Partial Differential Equations*, editors R. Glowinski et al., SIAM, 140 -150, 1991.

[Mall85] M. Mallet. *A Finite Element Method for Computational Fluid Dynamics.* PhD thesis, Stanford University, 1985.

[SaSch86] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Scient. Stat. Comput.*, 7:856 – 869, 1986.

[Shak88] F. Shakib. *Finite Element Analysis of the Compressible Euler and Navier- Stokes Equations.* PhD thesis, Stanford University, 1988.

[Tez89] T. E. Tezduyar and J. Liou. Grouped element-by-element iteration schemes for incompressible flow computations. *Computer Physics Comm.*, 53:441 – 453, 1989.