Computational Mechanics 23 (1999) 151–157 © Springer-Verlag 1999

Parallel computation of unsteady compressible flows with the EDICT

S. Mittal, S. Aliabadi, T. Tezduyar

Abstract Recently, the Enhanced-Discretization Interface-Capturing Technique (EDICT) was introduced for simulation of unsteady flow problems with interfaces such as two-fluid and free-surface flows. The EDICT yields increased accuracy in representing the interface. Here we extend the EDICT to simulation of unsteady viscous compressible flows with boundary/shear layers and shock/expansion waves. The purpose is to increase the accuracy in selected regions of the computational domain. An error indicator is used to identify these regions that need enhanced discretization. Stabilized finite-element formulations are employed to solve the Navier-Stokes equations in their conservation law form. The finite element functions corresponding to enhanced discretization are designed to have two components, with each component coming from a different level of mesh refinement over the same computational domain. The primary component comes from a base mesh. A subset of the elements in this base mesh are identified for enhanced discretization by utilizing the error indicator. A secondary, more refined, mesh is constructed by patching together the second-level meshes generated over this subset of elements, and the second component of the functions comes from this mesh. The subset of elements in the base mesh that form the secondary mesh may change from one time level to other depending on the distribution of the error in the computations.

S. Mittal
Department of Aerospace Engineering,
Indian Institute of Technology, Kanpur, UP 208 016, India

S. Aliabadi
Department of Engineering and Army HPC Research Center,
Clark Atlanta University, 223 James P. Brawley Dr. S.W.,
Atlanta, GA 30314

T. Tezduyar
Mechanical Engineering and Materials Science,
Army HPC Research Center, Rice University-MS 321
6100 Main Street,
Houston, TX 77005

Correspondence to: T. Tezduyar

This work is sponsored by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-00-08. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. CRAY time was provided in part by the Minnesota Supercomputer Institute.

Using a parallel implementation of this EDICT-based method, we apply it to test problems with shocks and boundary layers, and demonstrate that this method can be used very effectively to increase the accuracy of the base finite element formulation.

1 Introduction

Recently, Tezduyar et al. [1] introduced the Enhanced-Discretization Interface-Capturing Technique (EDICT) for simulation of unsteady flow problems with interfaces such as two-fluid and free-surface flows. The starting point for the EDICT is the volume of fluid (VOF) method [2]. In the EDICT, the Navier-Stokes equations are solved over a non-moving mesh and an interface function, with two distinct values that serves as a marker identifying the two fluids, is transported with a time-dependent advection equation. To increase the accuracy in representing the interface, function spaces corresponding to enhanced discretization are used at and near the interface.

In this article we extend the EDICT to unsteady compressible flows with shock/expansion waves, boundary/shear layers, and their interactions. Our target is to increase the accuracy in selected regions of the computational domain. These regions are identified by an error indicator. The Navier-Stokes equations of compressible flows in their conservation law form are solved using a stabilized finite element formulation based on conservation variables. The streamline-upwind/Petrov-Galerkin (SUPG) stabilization technique is employed to stabilize the computations against potential numerical oscillations in advection dominated flows [3, 4, 5]. A shock-capturing term is added to the formulation to provide stability to the computations in the presence of discontinuities and large gradients in the flow. The interaction between the shock-capturing term and the viscous effects and SUPG term was addressed earlier in [6]. With the error indicator employed to provide an estimate of the error in the computations, the regions identified for enhanced discretization usually coincide with those that are associated with large gradients in the flow variables. The finite element functions corresponding to enhanced discretization are designed to have two components. Each component comes from a different level of mesh refinement over the same computational domain. The first component comes from a base mesh. A subset of the elements in this base mesh are identified based on the error estimated by the error indicator. A second mesh is constructed by patching together the second-level meshes generated over this subset of ele-

ments, and the second component of the functions comes from this mesh. The elements in this subset will change from one time level to other, depending on the distribution of the error in the computations. An element which is in this subset at the current time step may be out of it some time later, and come back in again some time after that. For each element in this subset there will be a unique second-level mesh. If an automatic mesh generator is used to generate the second-level mesh, the mesh will be generated only once and stored, to be used later if that element needs a second-level mesh again.

The second-level mesh will not be re-defined for every time step but frequently enough to cover the zones identified by an error larger than a predetermined value. Several error indicators have been suggested in the literature for time-dependent calculations (for example, see articles by Lohner [7] and Oden et al. [8]). Our aim here is not to compare the different error indicators reported in the literature, but to use one of them to demonstrate the implementation of our methodology. The error indicator we use in the present work is the one proposed by Lohner [7].

To reduce the memory requirements associated with large-scale computations, a matrix-free iteration technique is employed. This vector-based technique [9, 10] totally eliminates the need to compute or store any coefficient matrices, even at the element level. This method has been implemented on the shared-memory parallel computing platforms of the SGI multi-processor systems, including a 20-processor ONYX and a 12-processor POWER CHALLENGE.

In Section 2, we review the governing equations of compressible flows. The stabilized formulation is described in Section 3. We provide a brief description of the error indicator in Section 4 and more details on the concept of enhanced discretization in Section 5. In Section 6 we present our numerical examples, and end with concluding remarks in Section 7.

2 The governing equations

Let $\Omega \subset \mathbb{R}^{n_{sd}}$ and $(0, T)$ be the spatial and temporal domains respectively, where n_{sd} is the number of space dimensions, and let Γ denote the boundary of Ω. The spatial and temporal coordinates are denoted by x and t. The Navier-Stokes equations of compressible flows in conservation law form are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad \text{on } \Omega \text{ for } (0, T) \ , \tag{1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla p - \nabla \cdot \mathbf{T} = 0 \quad \text{on } \Omega \text{ for } (0, T) \ , \tag{2}$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) + \nabla \cdot (p\mathbf{u}) - \nabla \cdot (\mathbf{T}\mathbf{u}) + \nabla \mathbf{q} = 0 \quad \text{on } \Omega \text{ for } (0, T) \ . \tag{3}$$

Here $\rho, \mathbf{u}, p, \mathbf{T}, e$ and $\mathbf{q}$ are the density, velocity, pressure, viscous stress tensor, total energy per unit mass, and the heat flux vector, respectively. The viscous stress tensor is defined as

$$\mathbf{T} = \mu((\nabla \mathbf{u}) + (\nabla \mathbf{u}^{\mathrm{T}})) + \lambda(\nabla \cdot \mathbf{u})\mathbf{I} \ , \tag{4}$$

where μ and λ are the viscosity coefficients. It is assumed that μ and λ are related by

$$\lambda = -\frac{2}{3}\mu \ . \tag{5}$$

Pressure is related to the other variables via the equation of state. For ideal gases, the equation of state assumes the special form

$$p = (\gamma - 1)\rho i \ , \tag{6}$$

where γ is the ratio of specific heats and i is the internal energy per unit mass. The internal energy is related to the total energy and kinetic energy as

$$i = e - \frac{1}{2}\|u\|^2 \ . \tag{7}$$

The heat flux vector is defined as

$$\mathbf{q} = -\kappa \nabla \theta \ , \tag{8}$$

where κ is the heat conductivity and θ is the temperature. The temperature is related to the internal energy with the following expression:

$$\theta = \frac{\gamma - 1}{\mathrm{R}} i \ , \tag{9}$$

where R is the ideal gas constant. Prandtl number (Pr), assumed to be specified, relates the heat conductivity to the fluid viscosity according to

$$\kappa = \frac{\gamma \mathrm{R} \mu}{(\gamma - 1)\mathrm{Pr}} \ . \tag{10}$$

Equations (1), (2), and (3) can be written in conservation variables as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \frac{\partial \mathbf{E}_i}{\partial x_i} = 0 \quad \text{on } \Omega \text{ for } (0, T) \ , \tag{11}$$

where for 2D flows $\mathbf{U} = (\rho, \rho u_1, \rho u_2, \rho e)$ is the vector of conservation variables, and $\mathbf{F}_i$ and $\mathbf{E}_i$ are, respectively, the Euler and viscous flux vectors defined as

$$\mathbf{F}_i = \begin{pmatrix} u_i \rho \\ u_i \rho u_1 + \delta_{i1} p \\ u_i \rho u_2 + \delta_{i2} p \\ u_i (\rho e + p) \end{pmatrix} \tag{12}$$

$$\mathbf{E}_i = \begin{pmatrix} 0 \\ \tau_{i1} \\ \tau_{i2} \\ -q_i + \tau_{ik} u_k \end{pmatrix} \ . \tag{13}$$

Here u_i, q_i and τ_{ik} are the components of the velocity, heat flux, and viscous stress tensor, respectively. Equation (11) can be re-written as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j}\right) = 0 \quad \text{on } \Omega \text{ for } (0, T) \ , \tag{14}$$

where

$$\mathbf{A}_i = \frac{\partial \mathbf{F}_i}{\partial \mathbf{U}} \tag{15}$$

is the Euler Jacobian matrix, and $\mathbf{K}_{ij}$ is the diffusivity matrix satisfying

$$\mathbf{K}_{ij}\frac{\partial \mathbf{U}}{\partial x_j} = \mathbf{E}_i \ . \tag{16}$$

Corresponding to Eq. (14), the boundary and initial conditions are represented as

$$\mathbf{U} = \mathbf{g} \quad \text{on } \Gamma_g \text{ for } (0,T) \ , \tag{17}$$

$$\mathbf{n}\cdot\mathbf{E} = \mathbf{h} \quad \text{on } \Gamma_h \text{ for } (0,T) \ , \tag{18}$$

$$\mathbf{U}(x,0) = \mathbf{U}_0 \quad \text{on } \Omega_0 \ . \tag{19}$$

3
Finite element formulation

Consider a finite element discretization of Ω into subdomains Ω^e, $e = 1, 2, ..., n_{\text{el}}$, where n_{el} is the number of elements. Based on this discretization, we define the finite element trial function space $\mathscr{S}_n^h$ and weighting function space $\mathscr{V}_n^h$. The superscript h implies that these are finite-dimensional function spaces, while the subscript n denotes that corresponding to different time levels we may have different spatial discretizations. We will give more precise definition of these function spaces in Section 5. The stabilized finite element formulation of Eq. (14) is written as follows: given $\mathbf{U}_n^h$, find $\mathbf{U}_{n+1}^h \in \mathscr{S}_{n+1}^h$ such that $\forall \mathbf{W}_{n+1}^h \in \mathscr{V}_{n+1}^h$,

$$\begin{aligned}
&\int_\Omega \mathbf{W}_{n+1}^h \cdot \left(\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i}\right) d\Omega \\
&\quad + \int_\Omega \left(\frac{\partial \mathbf{W}_{n+1}^h}{\partial x_i}\right) \cdot \left(\mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j}\right) d\Omega \\
&\quad + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \boldsymbol{\tau} \left(\mathbf{A}_k^h\right)^{\mathrm{T}} \left(\frac{\partial \mathbf{W}_{n+1}^h}{\partial x_k}\right) \cdot \\
&\qquad \left[\frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j}\right)\right] d\Omega \\
&\quad + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \delta \left(\frac{\partial \mathbf{W}_{n+1}^h}{\partial x_i}\right) \cdot \left(\frac{\partial \mathbf{U}^h}{\partial x_i}\right) d\Omega \\
&\qquad = \int_{\Gamma_h} \mathbf{W}_{n+1}^h \cdot \mathbf{h}^h \, d\Gamma \ .
\end{aligned} \tag{20}$$

In the variational formulation given by Eq. (20), the first two terms and the right-hand-side constitute the Galerkin formulation of the problem. The first series of element-level integrals in Eq. (20) are the SUPG stabilization terms added to the variational formulation to stabilize the computations against node-to-node oscillations in the advection-dominated range. The second series of element level integrals in the formulation are the shock-capturing terms that stabilize the computations in the presence of sharp gradients. The stabilization coefficients δ and $\boldsymbol{\tau}$ are the ones used by Mittal [5, 11] (which were based on those introduced in Aliabadi and Tezduyar [6]):

$$\boldsymbol{\tau} = \max[0, \boldsymbol{\tau}_a - \boldsymbol{\tau}_\delta] \ . \tag{21}$$

$$\boldsymbol{\tau}_a = \left(\left(\frac{2(c+\|u\|)}{h}\right)^2 + \left(\frac{12\nu}{h^2}\right)^2\right)^{-\frac{1}{2}} \mathbf{I} \ , \tag{22}$$

$$\boldsymbol{\tau}_\delta = \frac{\delta}{2(c+\|u\|)^2}\mathbf{I} \ , \tag{23}$$

$$\delta = \left[\frac{\left\|\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i}\right\|_{\mathbf{A}_0^{-1}}}{\left\|J_{1i}\frac{\partial \mathbf{U}}{\partial x_i}\right\|_{\mathbf{A}_0^{-1}} + \left\|J_{2i}\frac{\partial \mathbf{U}}{\partial x_i}\right\|_{\mathbf{A}_0^{-1}}}\right]^{\frac{1}{2}} \ , \tag{24}$$

where c is the wave speed, h is the element length, J_{jk} are the components of the Jacobian transformation matrix from physical to the local coordinates, and $\mathbf{A}_0^{-1}$ is the inverse of Reimannian metric tensor related to the transformation between the conservation and entropy variables [12]. As shown in Eq. (21), $\boldsymbol{\tau}_\delta$ is subtracted from $\boldsymbol{\tau}_a$ to account for the shock-capturing.

Time discretization of the variational formulation given by Eq. (20) is achieved with the generalized trapezoidal rule. For unsteady computations, temporally we employ a second-order accurate procedure.

4
Error indicator

An error indicator is employed to identify the elements in the primary mesh that need enhanced discretization for higher accuracy. The error indicator we use here is the one proposed by Lohner [7]:

$$E^I = \sqrt{\frac{\sum_{k=1}^{n_{\text{sd}}}\sum_{l=1}^{n_{\text{sd}}}\left(\int_\Omega N_{,k}^I U_{,l} d\Omega\right)^2}{\sum_{k=1}^{n_{\text{sd}}}\sum_{l=1}^{n_{\text{sd}}}\left[\int_\Omega |N_{,k}^I|\left(|U_{,l}| + \epsilon\sum_{J=1}^{n_{\text{en}}}\left(|N_{,l}^J||U_J|\right)\right)d\Omega\right]^2}} \ , \tag{25}$$

where N^I denotes shape-function associated with the node I, U is the error indicator variable chosen from the set $\mathbf{U}$, and n_{er} is the number of nodes in each element. In our computations we set $\epsilon = 0.05$ and U represents the density. First the error is calculated for all the nodes in the base mesh. Then, it is transferred to the element level by taking the average of the nodal values for each element. If the error for an element exceeds a certain value (0.1 in our computations), that element is identified as one that needs enhanced discretization.

5
Construction of function spaces – enhanced discretization

We begin with a base mesh which we call Mesh-1. The set of elements and nodal points in Mesh-1 are ϵ^1 and η^1, respectively. At a time level n, a subset $(\epsilon^1)_n^2$ from this set of elements is identified, based on an error indicator, requiring enhanced discretization for higher accuracy. A more refined mesh, Mesh-2, is constructed over the set of elements $(\epsilon^1)_n^2$ by subdividing each of these elements into

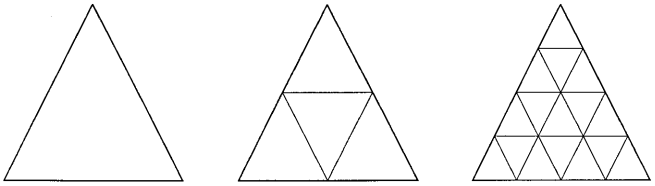

Fig. 1. A representative element from Mesh-1 (left) and Mesh-2 with each element in set $(\epsilon^1)_n^2$ divided into four (center) or sixteen elements (right)

smaller elements. The subscript n implies that this subset might change from one time level to other. For each element in ϵ^1, there is a unique second level mesh. Therefore, if an automatic mesh generator is being used to generate these meshes, the cost for that mesh generation will be a one-time cost. The set of elements and nodal points for Mesh-2 are denoted by ϵ_n^2 and η_n^2, respectively.

We will limit ourselves to linear elements in 2D and 3D, as well as bilinear and trilinear elements, respectively, in 2D and 3D.

We decompose $\mathbf{U}_n^h$ as follows:

$$\mathbf{U}_n^h = \mathbf{U}_h^1 + \mathbf{U}_n^2 \ . \tag{26}$$

The function $\mathbf{U}_n^1$ comes from a space of functions with the basis set consisting of the shape functions associated with all the nodes in η^1, excluding those "surrounded" by the elements in $(\epsilon^1)_n^2$. The function $\mathbf{U}_n^1$ also needs to satisfy the Dirichlet-type boundary conditions, except at those nodes that have been surrounded at the boundary of Ω.

The function $\mathbf{U}_n^2$ comes from a space of functions with the basis set consisting of the shape functions associated with all the nodes in η_n^2, excluding those at the boundaries of the zones covered by the elements in ϵ_n^2. However, the nodes at the boundary of Ω are included unless they coincide with the nodes in η_n^1 that have not been surrounded.

The weighting functions are decomposed in a similar fashion:

$$\mathbf{W}_n^h = \mathbf{W}_n^1 + \mathbf{W}_n^2 \ , \tag{27}$$

The components of each weighting function are defined in the same way as was done for the trial functions, except that the weighting functions need to satisfy the homogeneous form of the Dirichlet-type boundary conditions.

In our implementation we have the option to divide each of the elements in the base mesh belonging to the set $(\epsilon^1)_n^2$ into either four or sixteen smaller elements as shown in Fig. 1. For the computations reported in this article, each of the elements in the primary mesh that need enhanced discretization are subdivided into sixteen elements.

6 Examples

Shock-reflection problem. The steady-state solution for this 2D inviscid flow problem involves three flow regions separated by an oblique shock and its reflection from a wall as shown in Fig. 2.

The computational domain is a rectangular region of dimensions 4.1 in the x_1 direction and 1.0 in the x_2 direction. At the left boundary, flow data corresponding to Mach 2.9 is prescribed:

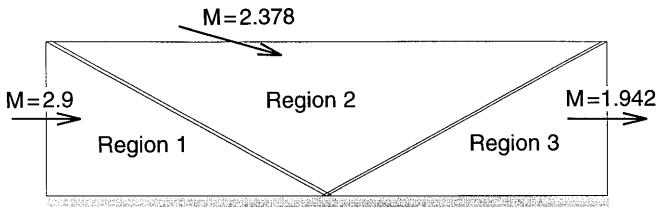

Fig. 2. Shock-reflection problem: problem description

$$\text{Region 1} \begin{cases} M &= 2.9 \\ \rho &= 1 \\ u_1 &= 1 \\ u_2 &= 0 \\ \theta &= 0 \end{cases} . \tag{28}$$

At the top boundary, the flow conditions that are specified correspond to Mach 2.3781 and an incident shock angle of 29°:

$$\text{Region 2} \begin{cases} M &= 2.3781 \\ \rho &= 1.7 \\ u_1 &= 0.9033 \\ u_2 &= -0.1746 \\ \theta &= 0.07685 \end{cases} . \tag{29}$$

At the lower boundary, the component of velocity normal to the wall is set to zero. The computations begin with a uniform Mach 2.9 flow in the domain. The base mesh consists of 1281 nodes and 2400 triangular elements. Mesh-2 is obtained by subdividing each element in $(\epsilon^1)_n^2$ into sixteen elements, in a manner as shown in Figure 1. The time step in the computation is 0.01 and $(\epsilon^1)_n^2$ is redefined at every 25 time steps. Figure 3 shows Mesh-1 together with Mesh-2 and density at non-dimensional time $t = 0.5, 1.0, 1.5, 2.0, 2.5, 3.0$. It can be observed that the shocks are quite crisp and that the regions identified by the error indicator for enhanced discretization closely follow the shock. At $t = 3.0$ the solution reaches steady-state.

Mach 3, Re 10^5 flow past a circular cylinder. A circular cylinder is placed in a Mach 3.0 flow. The Reynolds number based on the diameter of the cylinder and the free-stream values of the velocity and kinematic viscosity is 10^5. The cylinder wall is assumed to be adiabatic and the no-slip condition is specified for the velocity on the surface of the cylinder. All the variables are specified at the upstream boundary. At the downstream boundary, we specify a Neumann-type boundary condition for the momentum and energy equations that correspond to zero viscous stress and heat flux vectors. The computations are initiated with the free-stream conditions in the entire domain.

Figure 4 shows the base mesh, Mesh-1, which consists of 2396 nodes and 4694 triangular elements. Mesh-2 is obtained by subdividing each element in $(\epsilon^1)_n^2$ into sixteen elements, in a manner as shown in Figure 1. The time step in the computation is 0.02 and $(\epsilon^1)_n^2$ is redefined at every 20 time steps. Figure 5 shows Mesh-1 together with Mesh-2 and density at non-dimensional time t =1.6, 3.2, 4.8, 6.4, 8.0, 9.6. It can be observed that in this computation the regions identified for enhanced discretization are the ones associated with the boundary layer and shock waves.

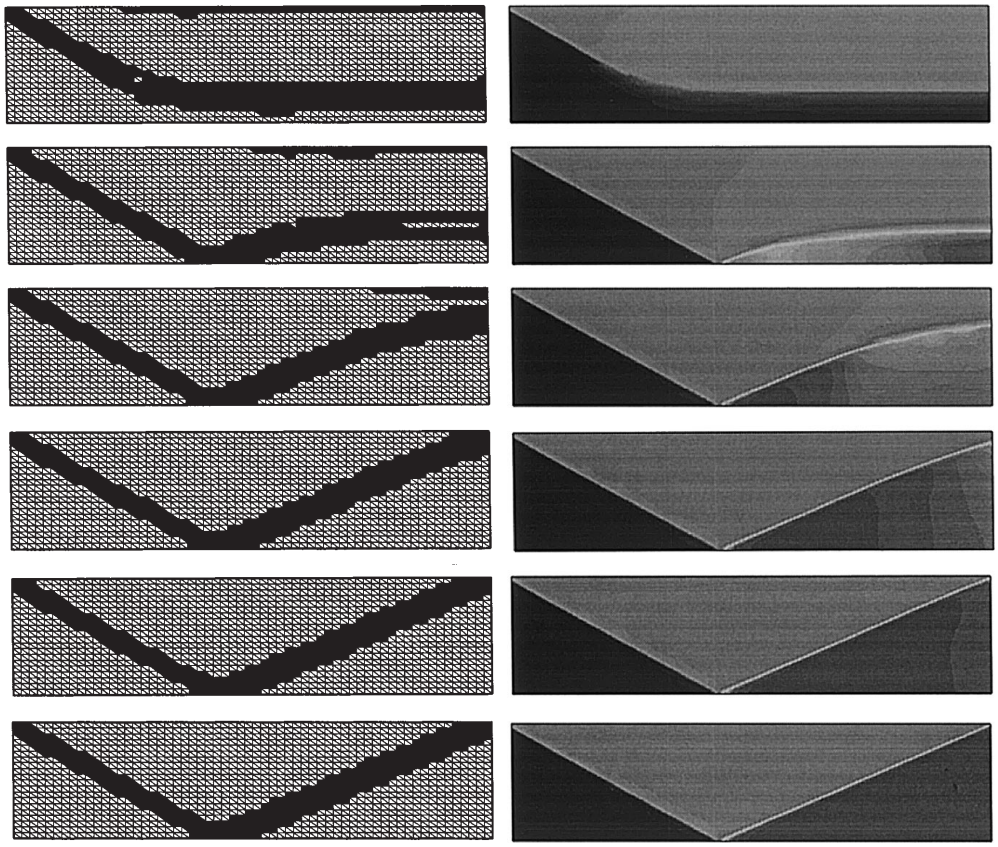

Fig. 3. Shock-reflection problem: Mesh-2 shown on top of Mesh-1 and density at $t = 0.5, 1.0, 1.5, 2.0, 2.5, 3.0$

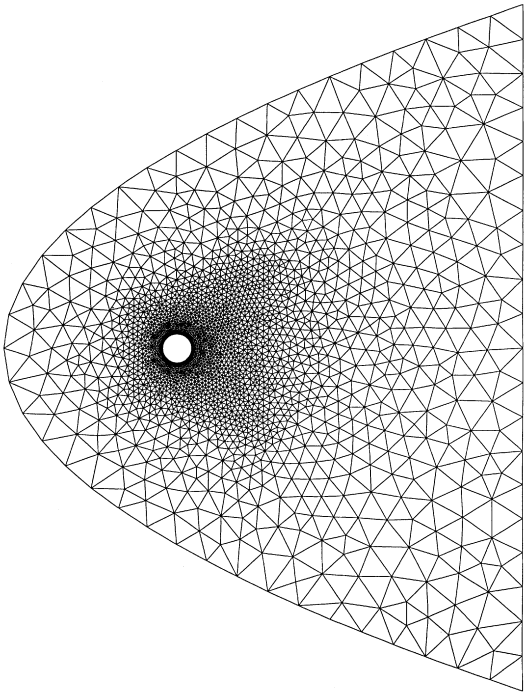

Fig. 4. Mach 3, Re 10^5 flow past a circular cylinder: Mesh-1 (2396 nodes and 4694 triangular elements)

7 Concluding remarks

In this paper we have extended the EDICT to unsteady viscous compressible flows with boundary/shear layers and shock/expansion waves. The Navier-Stokes equations are solved using stabilized finite element formulations that are known to possess good stability and accuracy properties. To further increase the accuracy of the computations an error indicator is employed to identify the regions in the computational domain that need enhanced discretization. The finite element functions corresponding to enhanced discretization are designed to have two components, with each component coming from a different level of mesh refinement over the same computational domain. The first component of the functions for the flow variables comes from the base mesh. A subset of the elements in this mesh, for which the error indicator shows the error larger than a certain threshold value, are identified for enhanced discretization. A more refined mesh is constructed by patching together the second-level meshes generated over this subset of elements. This subset of element may change from one time level to other depending on the error distribution in the computations. The second component of the functions for the flow variables comes from this more refined mesh. However, the second-level mesh is not redefined at every time level, but frequently enough to keep the zones, with error larger than a predetermined threshold, covered with these higher level meshes. The methodology has been implemented on a shared-memory parallel computer and results from test computations for flows involving shock waves and boundary layers demonstrate that the EDICT can be utilized to increase accuracy of solutions that involve crisp shocks and layers.

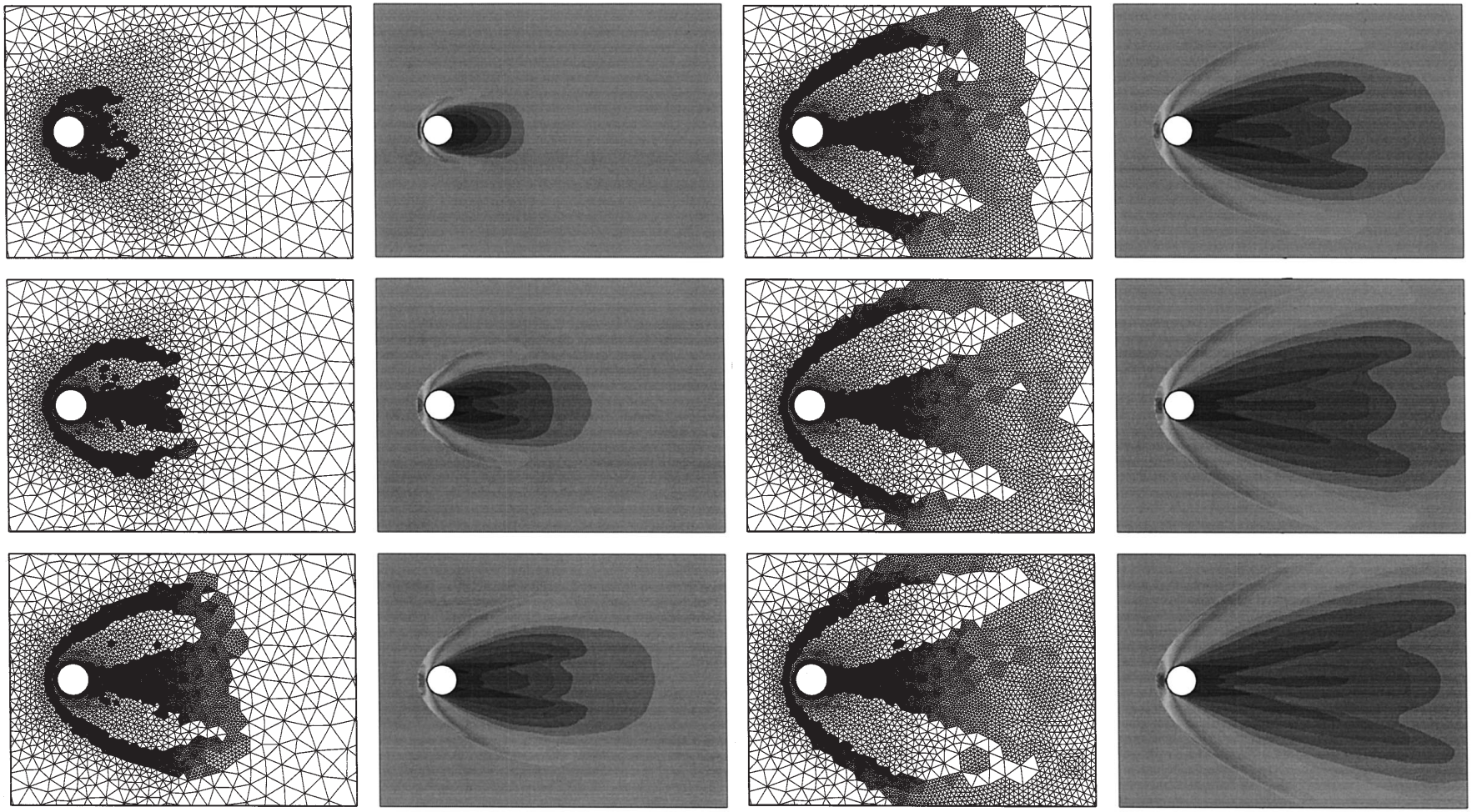

Fig. 5a, b. **a** Mach 3, Re 10^5 flow past a circular cylinder: Mesh-2 shown on top of Mesh-1 and density at $t = 1.6, 3.2, 4.8$; **b** Mach 3, Re 10^5 flow past a circular cylinder: Mesh-2 shown on top of Mesh-1 and density at $t = 6.4, 8.0, 9.6$

References

[1] **Tezduyar TE, Aliabadi S, Behr M** (1997) Enhanced-discretization interface-capturing technique. In Proceedings of the ISAC' 97 High Performance Computing on Multiphase Flows, Tokyo, Japan

[2] **Hirt CW, Nichols BD** (1981) Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Physics 39:201–225

[3] **Tezduyar TE, Hughes TJR** (1983) Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations. In Proceedings of AIAA 21st Aerospace Sciences Meeting, AIAA Paper 83–0125, Reno, Nevada

[4] **Hughes TJR, Tezduyar TE** (1984) Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. Comp. Meth. Appl. Mech. Eng. 45:217–284

[5] **Mittal S** (1997) Finite element computation of unsteady viscous compressible flows. To appear in Comp. Meth. Appl. Mech. Eng.

[6] **Aliabadi S, Tezduyar TE** (1993) Space-time finite element computation of compressible flows involving moving boundaries and interfaces. Comp. Meth. Appl. Mech. Eng. 107:209–223

[7] **Lohner R** (1987) An adaptive finite element scheme for transient problems in CFD. Comp. Meth. Appl. Mech. Eng. 61:323–338

[8] **Oden JT, Demkowicz L, Racowicz W, Wastermann TA** (1990) A posteriori error analysis in finite elements: The element residual method for symmetrizable problems with applications to compressible Euler and Navier-Stokes equations. Comp. Meth. Appl. Mech. Eng. 82:183–203

[9] **Johan Z, Hughes TJR, Shakib F** (1991) A globally convergent matrix-free algorithm for implicit time-marching schemes arising in finite element analysis in fluids. Comp. Meth. Appl. Mech. Eng. 87:281–304

[10] **Aliabadi SK, Tezduyar TE** (1995) Parallel fluid dynamics computations in aerospace applications. Int. J. Num. Meth. Fluids 21: 783–805

[11] **Mittal S** (1997) Finite element computation of unsteady viscous compressible flows past stationary airfoils. To appear in Computat. Mech.

[12] **Hughes TJR, Mallet M** (1986) A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems. Comp. Meth. Appl. Mech, Eng. 58:329–339