Computer Methods in Applied Mechanics and Engineering 104 (1993) 397–422
North-Holland

CMA 342

SUPG finite element computation of compressible flows with the entropy and conservation variables formulations[†]

G.J. Le Beau
Computational Fluid Dynamics Group, Navigation, Control and Aeronautics Division, NASA-Johnson Space Center, Houston, TX 77058, USA

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

Received 9 March 1992

SUPG-stabilized finite element formulations of compressible Euler equations based on the conservation and entropy variables are investigated and compared. The formulation based on the conservation variables consists of the formulation introduced by Tezduyar and Hughes plus a shock capturing term. The formulation based on the entropy variables is the same as the one by Hughes, Franca and Mallet, which has a shock capturing term built in. These formulations are tested on several subsonic, transonic and supersonic compressible flow problems. It is shown that the stabilized formulation based on the conservation variables gives solutions which are just as good as those obtained with the entropy variables. Furthermore, the solutions obtained using the two formulations are very close and in some cases almost indistinguishable. Consequently, it can be deduced that the relative merits of these two formulations will continue to remain under debate.

1. Introduction

The physics and dynamics of problems involving compressible flows in aerospace applications are not yet fully understood. Particularly, better understanding of dynamical, thermal, and chemical aspects of the reentry conditions is needed. While the prediction of aerodynamic and heating loads is very important for designing space vehicles [5], the conditions, such as altitude and speed, under which space vehicles operate makes the simulation by ground test facilities extremely difficult. Therefore, computational predictions based on the solution of the appropriate governing equations have become crucial. The finite element method permits the

Correspondence to: Tayfun E. Tezduyar, Minnesota Supercomputer Institute, University of Minnesota, 1200 Washington Avenue South, Minneapolis, MN 55415, USA.

[†] This research was sponsored by NASA-Johnson Space Center under grant NAG 9-449 and by NSF under grant MSM-8796352. Partial support for this work has also come from the Army Research Office contract number DAAL03-89-C-0038 with the Army High-Performance Computing Research Center at the University of Minnesota.

use of unstructured grids which are often needed to effectively solve problems with complicated geometries. While it is hoped that the inherent flexibility of the finite element method will enable engineers to model flows past such vehicles as the Space Shuttle and the Aerobraking Flight Experiment, this work has concentrated on developing effective strategies for the solution of compressible flows in two dimensions, which can then be extended to more complex three-dimensional problems.

Among the difficulties associated with numerical simulation of compressible flows is the treatment of shocks and sharp layers. For high-speed flows, the magnitude of such sharp variations in the flow field becomes very large, making the computation even more challenging. It is well known that classical Galerkin finite element (or classical central difference) schemes result in spurious oscillations for advection-dominated flows, especially in the presence of sharp layers in the solution. To minimize such oscillations, the streamline-upwind/Petrov–Galerkin (SUPG) formulation is employed. The SUPG formulation was first introduced, in the context of incompressible flows, by Hughes and Brooks [6]. A comprehensive description of the formulation, together with various numerical examples, can be found in [7].

The SUPG stabilization for hyperbolic systems, in particular compressible Euler equations, was first introduced in a 1982 NASA report by Tezduyar and Hughes [1]. This report includes a stability and accuracy analysis and several 1D and 2D examples. The method was also presented in a 1983 AIAA paper [8]. The journal version of the NASA report was published in 1984 with some additional numerical examples [9]. The stabilization techniques introduced in [1] constitutes pilot work for compressible flows. For example, the Taylor–Galerkin stabilization method promoted by Donea [10] is very similar (under certain conditions identical) to one of the stabilization methods introduced in [1].

The SUPG stabilization used in [1] was achieved in the context of conservation variables. Following that, similar stabilization techniques were employed in the context of entropy variables [2–4]. Together with a shock capturing term added to the formulation, it was shown that the solutions obtained with the entropy variables were more accurate than those obtained with the conservation variables. However, the SUPG formulation introduced in [1] did not involve any shock capturing term. Recently, it was shown in [11] that this SUPG formulation based on conservation variables, supplemented with a shock capturing term, gives solutions which are just as good as those obtained with the entropy variables. The main objective of this work is to investigate and compare the finite element solutions of compressible flows using the conservation and entropy variables formulations.

Computations reported in this work are performed by an implicit implementation of inviscid impermeable wall boundary conditions. In this implementation, the normal and tangential components of the momentum are kept track of, rather than the global Cartesian components. The normal component of the momentum can then be set to zero as a Dirichlet boundary condition. Transformation between the normal-tangential components frame and the global Cartesian components frame is accomplished by using a local transformation rule, dependent on the geometry of the boundary.

2. The governing equations

Let Ω be a domain in $R^{n_{sd}}$ and T be a positive real number, where n_{sd} is the number of spatial dimensions. The spatial and temporal coordinates are denoted by $x \in \bar{\Omega}$ and $t \in [0, T]$,

where a superposed bar indicates the set closure. The boundary of the domain Ω is denoted by Γ. The dynamics which governs multi-dimensional inviscid, compressible fluid flow is described by the Euler equations. These equations, written in terms of conservation variables $\boldsymbol{U} = (\rho, \rho u_1, \ldots, \rho u_{n_{sd}}, \rho e)$, are

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_i}{\partial x_i} = 0 \quad \text{on } \Omega \times (0, T), \tag{1}$$

where $\boldsymbol{F}_1, \ldots, \boldsymbol{F}_{n_{sd}}$ are the Euler fluxes, and repeated indices imply summation over the range of spatial dimension n_{sd}. Alternatively, (1) can be written as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{A}_i \frac{\partial \boldsymbol{U}}{\partial x_i} = 0 \quad \text{on } \Omega \times (0, \mathrm{T}), \tag{2}$$

where

$$\boldsymbol{A}_i = \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{U}}, \quad i = 1, 2, \ldots, n_{sd}. \tag{3}$$

We assume that associated with (2), the following boundary and initial conditions are given:

$$\boldsymbol{B}\boldsymbol{U} = \boldsymbol{G} \qquad \text{on } \Gamma \times (0, T), \tag{4}$$

$$\boldsymbol{U}(\boldsymbol{x}, 0) = \boldsymbol{U}_0(\boldsymbol{x}) \quad \text{on } \Omega, \tag{5}$$

where $\boldsymbol{B}$ denotes a general boundary condition operator, and $\boldsymbol{G}$ and $\boldsymbol{U}_0$ are given functions.

By performing a transformation of variables $\tilde{\boldsymbol{U}} = \tilde{\boldsymbol{U}}(\boldsymbol{U})$, we can obtain the Euler equations in terms of the entropy variables [2] such that

$$\tilde{\boldsymbol{A}}_0 \frac{\partial \tilde{\boldsymbol{U}}}{\partial t} + \tilde{\boldsymbol{A}}_i \frac{\partial \tilde{\boldsymbol{U}}}{\partial x_i} = 0 \quad \text{on } \Omega \times (0, T), \tag{6}$$

where

$$\tilde{\boldsymbol{A}}_0 = \frac{\partial \boldsymbol{U}}{\partial \tilde{\boldsymbol{U}}} \quad \text{and} \quad \tilde{\boldsymbol{A}}_i = \boldsymbol{A}_i \tilde{\boldsymbol{A}}_0. \tag{7a,b}$$

In this form, $\tilde{\boldsymbol{A}}_0$ is a symmetric and positive-definite matrix, and the $\tilde{\boldsymbol{A}}_i$'s are symmetric matrices. Also, associated with (6) is a general boundary condition operator $\tilde{\boldsymbol{B}}$ and a given function $\tilde{\boldsymbol{G}}$, such that

$$\tilde{\boldsymbol{B}}\tilde{\boldsymbol{U}} = \tilde{\boldsymbol{G}} \quad \text{on } \Gamma \times (0, T). \tag{8}$$

The transformation $\tilde{\boldsymbol{U}} = \tilde{\boldsymbol{U}}(\boldsymbol{U})$, given in [2], can be written, in two dimensions, as

$$\tilde{\boldsymbol{U}}(\boldsymbol{U}) = \frac{1}{\rho i} \begin{bmatrix} -U_4 + \rho i(\gamma + 1 - s + s_0) \\ U_2 \\ U_3 \\ -U_1 \end{bmatrix}, \tag{9}$$

where the internal energy ρi and the entropy s can be expressed as

$$\rho i = U_4 - \frac{U_2^2 + U_3^2}{2U_1}\,, \tag{10}$$

$$s = \ln\left[\frac{(\gamma - 1)\rho i}{U_1^{\gamma}}\right], \tag{11}$$

and γ is the ratio of the specific heats.

Because the Euler equations neglect the viscous terms of the compressible Navier–Stokes equations, the flow is allowed to slip at a solid surface. Thus at each point on the surface, the solution of an Eulerian flow must satisfy

$$u_i n_i = 0\,, \tag{12}$$

where n_i, $i = 1, 2, \ldots, n_{\mathrm{sd}}$, are the components of the unit normal vector.

3. The finite element formulation

Consider a finite element discretization of the domain Ω into subdomains (elements) Ω^e, $e = 1, 2, \ldots, n_{\mathrm{el}}$, where n_{el} is the number of elements. Throughout, we shall assume that on Γ the trial solutions $\boldsymbol{U}$ and $\tilde{\boldsymbol{U}}$ satisfy $\boldsymbol{BU} = \boldsymbol{G}$ and $\tilde{\boldsymbol{B}}\tilde{\boldsymbol{U}} = \tilde{\boldsymbol{G}}$, respectively. Furthermore, on Γ, the weighting functions $\boldsymbol{W}$ and $\tilde{\boldsymbol{W}}$ satisfy $\boldsymbol{BW} = \boldsymbol{0}$ and $\tilde{\boldsymbol{B}}\tilde{\boldsymbol{W}} = \boldsymbol{0}$, respectively. We define the following finite-dimensional function spaces.

For conservation variables:

$$\boldsymbol{S}^h = \{\boldsymbol{U}^h \mid \boldsymbol{U}^h \in [H^{1h}(\Omega)]^{n_{\mathrm{dof}}},\ \boldsymbol{U}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\mathrm{dof}}},\ \boldsymbol{BU}^h \doteq \boldsymbol{G} \text{ on } \Gamma\}\,, \tag{13}$$

$$\boldsymbol{V}^h = \{\boldsymbol{W}^h \mid \boldsymbol{W}^h \in [H^{1h}(\Omega)]^{n_{\mathrm{dof}}},\ \boldsymbol{W}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\mathrm{dof}}},\ \boldsymbol{BW}^h \doteq \boldsymbol{0} \text{ on } \Gamma\}\,; \tag{14}$$

For entropy variables:

$$\tilde{\boldsymbol{S}}^h = \{\tilde{\boldsymbol{U}}^h \mid \tilde{\boldsymbol{U}}^h \in [H^{1h}(\Omega)]^{n_{\mathrm{dof}}},\ \tilde{\boldsymbol{U}}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\mathrm{dof}}},\ \tilde{\boldsymbol{B}}\tilde{\boldsymbol{U}}^h \doteq \tilde{\boldsymbol{G}} \text{ on } \Gamma\}\,, \tag{15}$$

$$\tilde{\boldsymbol{V}}^h = \{\tilde{\boldsymbol{W}}^h \mid \tilde{\boldsymbol{W}}^h \in [H^{1h}(\Omega)]^{n_{\mathrm{dof}}},\ \tilde{\boldsymbol{W}}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\mathrm{dof}}},\ \tilde{\boldsymbol{B}}\tilde{\boldsymbol{W}}^h \doteq \boldsymbol{0} \text{ on } \Gamma\}\,; \tag{16}$$

where $n_{\mathrm{dof}} = n_{\mathrm{sd}} + 2$.

The finite element formulation of the Euler equations in conservation variables is written as follows: Find $\boldsymbol{U}^h \in \boldsymbol{S}^h$ such that $\forall \boldsymbol{W}^h \in \boldsymbol{V}^h$

$$\int_{\Omega} \boldsymbol{W}^h \cdot \left(\frac{\partial \boldsymbol{U}^h}{\partial t} + \boldsymbol{A}_i \frac{\partial \boldsymbol{U}^h}{\partial x_i}\right) \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{T}_j \frac{\partial \boldsymbol{W}^h}{\partial x_j} \cdot \left(\frac{\partial \boldsymbol{U}^h}{\partial t} + \boldsymbol{A}_i \frac{\partial \boldsymbol{U}^h}{\partial x_i}\right) \mathrm{d}\Omega$$

$$+ \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{D} \frac{\partial \boldsymbol{W}^h}{\partial x_i} \cdot \frac{\partial \boldsymbol{U}^h}{\partial x_i} \mathrm{d}\Omega = 0\,, \tag{17}$$

where

$$\boldsymbol{T}_j = \tau \boldsymbol{A}_j^{\mathrm{T}} , \tag{18}$$

$$\boldsymbol{D} = \nu_d \boldsymbol{I} . \tag{19}$$

In this formulation, τ and ν_d are defined as

$$\tau = \max_{1 \le i \le n_{\mathrm{sd}}} \frac{\alpha h_i}{\rho_i} \boldsymbol{I} , \tag{20}$$

$$\nu_d = \left[\frac{\boldsymbol{A}_i \dfrac{\partial \boldsymbol{U}^h}{\partial x_i} \cdot \tilde{\boldsymbol{A}}_0^{-1} \boldsymbol{A}_j \dfrac{\partial \boldsymbol{U}^h}{\partial x_j}}{\dfrac{\partial \xi_l}{\partial x_j} \dfrac{\partial \boldsymbol{U}^h}{\partial x_j} \cdot \tilde{\boldsymbol{A}}_0^{-1} \dfrac{\partial \xi_l}{\partial x_k} \dfrac{\partial \boldsymbol{U}^h}{\partial x_k}} \right]^{1/2} , \tag{21}$$

where $\boldsymbol{I}$ is the identity tensor, h_i is the element length in the x_i direction, α is a parameter controlling the stability and accuracy of the time-integration algorithm, ρ_i is the spectral radius of $\boldsymbol{A}_i$, and ξ_l, $l = 1, 2, \ldots, n_{\mathrm{sd}}$ are the components of the element (local) coordinate.

The finite element formulation of the Euler equations in entropy variables, on the other hand, is written as follows: Find $\tilde{\boldsymbol{U}}^h \in \tilde{\boldsymbol{S}}^h$ such that $\forall \tilde{\boldsymbol{W}}^h \in \tilde{\boldsymbol{V}}^h$

$$\int_\Omega \tilde{\boldsymbol{W}}^h \cdot \left(\tilde{\boldsymbol{A}}_0 \frac{\partial \tilde{\boldsymbol{U}}^h}{\partial t} + \tilde{\boldsymbol{A}}_i \frac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_i} \right) \mathrm{d}\Omega + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \tilde{\boldsymbol{T}}_j \frac{\partial \tilde{\boldsymbol{W}}^h}{\partial x_j} \cdot \left(\tilde{\boldsymbol{A}}_0 \frac{\partial \tilde{\boldsymbol{U}}^h}{\partial t} + \tilde{\boldsymbol{A}}_i \frac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_i} \right) \mathrm{d}\Omega$$

$$+ \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \tilde{\boldsymbol{D}} \frac{\partial \tilde{\boldsymbol{W}}^h}{\partial x_i} \cdot \frac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_i} \mathrm{d}\Omega = 0 , \tag{22}$$

where

$$\tilde{\boldsymbol{T}}_j = \tilde{\tau} \tilde{\boldsymbol{A}}_j , \tag{23}$$

$$\tilde{\boldsymbol{D}} = \max(0, (\tilde{\nu}_d - \tilde{\nu}_\tau)) \tilde{\boldsymbol{A}}_0 . \tag{24}$$

In this formulation, $\tilde{\tau}$, $\tilde{\nu}_d$ and $\tilde{\nu}_\tau$ are defined as

$$\tilde{\tau} = \left(\frac{\partial \xi_i}{\partial x_j} \tilde{\boldsymbol{A}}_j \tilde{\boldsymbol{A}}_0^{-1} \frac{\partial \xi_i}{\partial x_k} \tilde{\boldsymbol{A}}_k \right)^{-1/2} , \tag{25}$$

$$\tilde{\nu}_d = \left[\frac{\tilde{\boldsymbol{A}}_i \dfrac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_i} \cdot \tilde{\boldsymbol{A}}_0^{-1} \tilde{\boldsymbol{A}}_j \dfrac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_j}}{\dfrac{\partial \xi_l}{\partial x_j} \dfrac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_j} \cdot \tilde{\boldsymbol{A}}_0 \dfrac{\partial \xi_l}{\partial x_k} \dfrac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_k}} \right]^{1/2} , \tag{26}$$

$$\tilde{\nu}_\tau = \left[\frac{\tilde{\boldsymbol{A}}_i \dfrac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_i} \cdot \tilde{\tau} \tilde{\boldsymbol{A}}_j \dfrac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_j}}{\dfrac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_j} \cdot \tilde{\boldsymbol{A}}_0 \dfrac{\partial \tilde{\boldsymbol{U}}^h}{\partial x_j}} \right] . \tag{27}$$

REMARK 1. In (17) and (22), the first global integral represents the Galerkin variational formulation of the problem. The first series of element level integrals are the SUPG stabilization terms added to the variational formulation. The second series of element level integrals are the shock capturing terms.

REMARK 2. Note that, in the entropy variables formulation, the contribution of the SUPG operator is subtracted from the shock capturing operator to prevent excessive diffusion.

4. Numerical examples and observations

In order to evaluate the aforementioned methods, several test problems are considered, for the purpose of comparing the performances of these formulations in modeling subsonic, transonic and supersonic flows. The first test problem is a reflected shock considered by Mallet [12]. The two other problems are those of flow past a circular cylinder and flow past a parabolic arc bump.

4.1. Reflected shock

This problem has several features which make it a popular test problem. First, it can be solved analytically, allowing for a check of the accuracy of our solutions. The problem also involves oblique shocks and a reflection point, both of which will more severely test the robustness of our methods. Finally, as this problem was considered in the entropy variables work of Mallet [12], it allows for comparison of our solutions with those obtained with the entropy variables.

There are three regions of flow in this problem, partitioned by shocks. In the steady-state solution of this problem, all three regions have constant values for the flow variables, which, as given in [12], are

Region 1	Region 2	Region 3
$M = 2.9$	$M = 2.3781$	$M = 1.94235$
$\rho = 1.0$	$\rho = 1.7$	$\rho = 2.6872838$
$u_1 = 2.9$	$u_1 = 2.619334$	$u_1 = 2.401499$
$u_2 = 0.0$	$u_2 = -0.5063$	$u_2 = 0.0$
$p = 0.714286$	$p = 1.52819$	$p = 2.93398$
$c = 1.0$	$c = 1.121833$	$c = 1.23633$

The computational domain for this problem has the dimensions 4.1 and 1.0 units in the x_1 and x_2 directions, respectively. The elements are of uniform size, with 60 elements in the x_1 direction, and 20 in the x_2 direction, making for a total of 1200 elements and 1281 nodes in our computational domain. The left and upper boundaries have Dirichlet type conditions for all four primitive variables, density, both velocities and temperature. The lower boundary is taken as a frictionless wall, and therefore, no-penetration conditions imposed along this boundary. At the right boundary, none of the flow variables are specified as this is a supersonic exit boundary.

The density at $x_2 = 0.25$ and contour plots, also of density, are shown, for the conservation variables solution in Fig. 1, and for the entropy variables solution in Fig. 2. The solutions have no oscillations around the shocks, and the line plots show that the shocks are captured within only a few elements, even though the mesh lines are skew to the shocks.

The effects of the shock capturing operators, defined by (19) and (24), were also examined by computing this problem without these terms. Without the shock capturing terms, for both formulations, instead of being captured within 4 or 5 elements, the shocks are spread out over almost a dozen elements and there are large overshoots and undershoots associated with each shock.

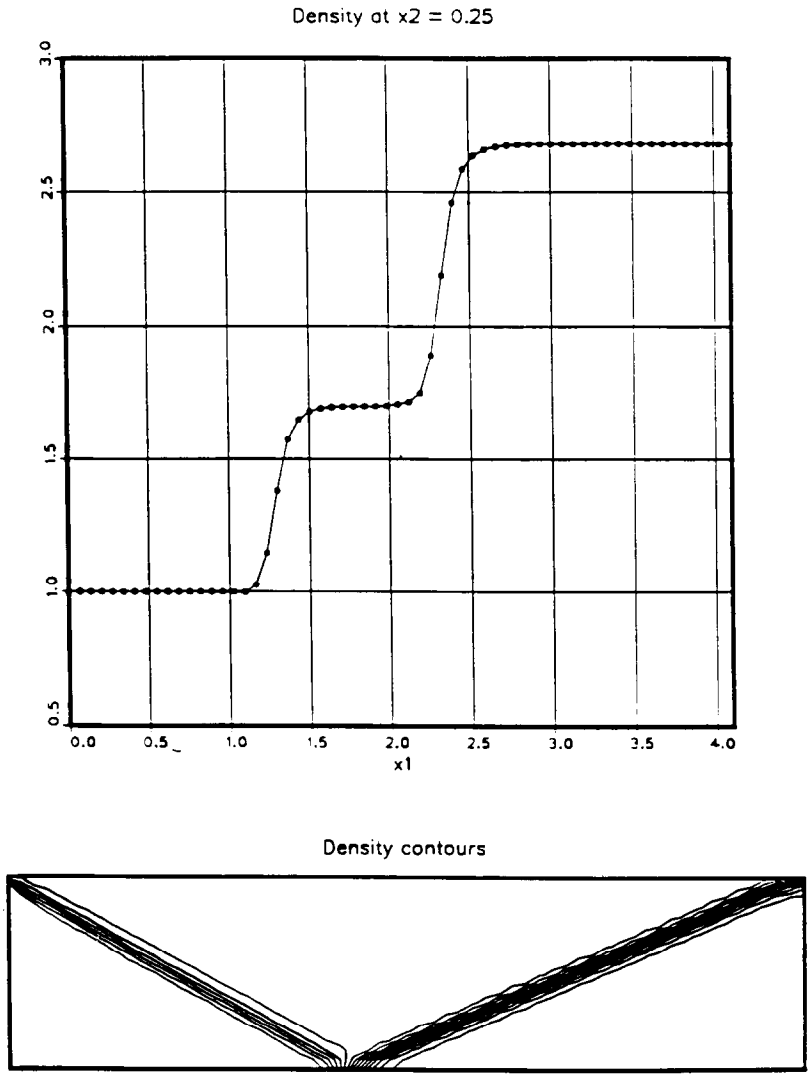

Fig. 1. Reflected shock problem: solution obtained by using the conservation variables formulation.

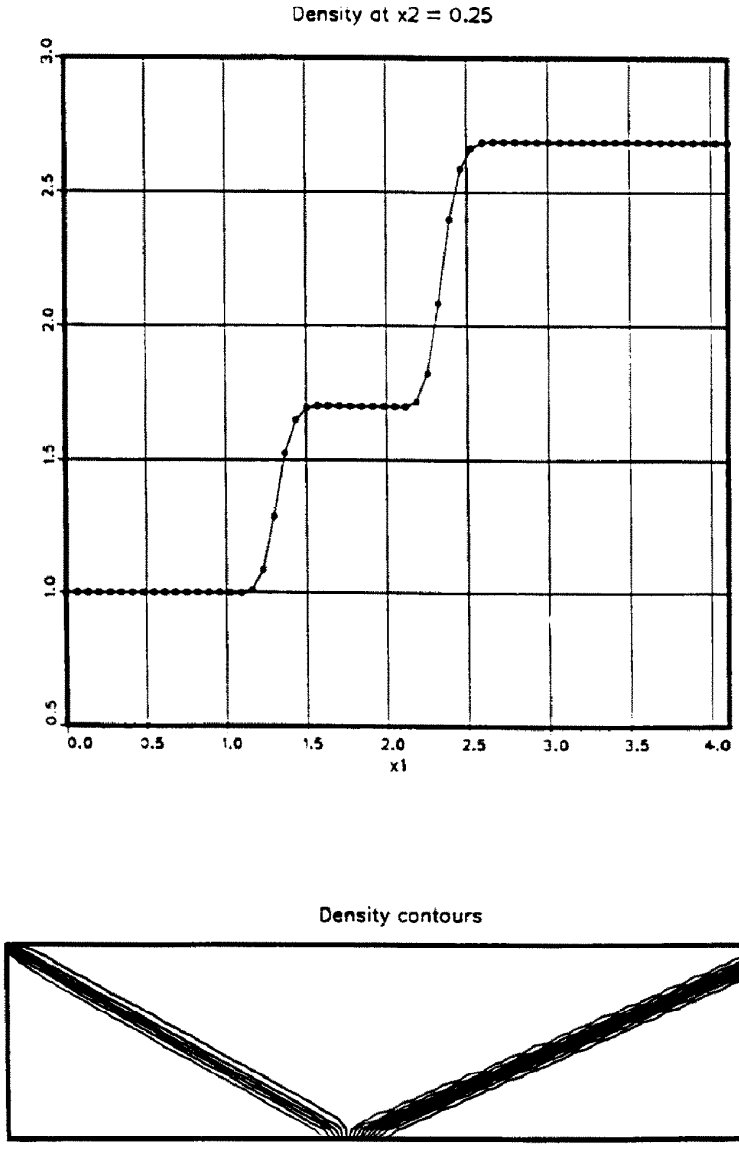

Fig. 2. Reflected shock problem: solution obtained by using the entropy variables formulation.

4.2. *Flow past a parabolic arc bump*

This problem is that of a parabolic arc bump placed on the floor of a frictionless wind tunnel. The arc bounding the bump is described by

$$x_2 = b[1 - (2x_1)^2], \tag{28}$$

where b is the ratio of the maximum arc thickness to the length. The bump lies in the center of

the bottom boundary of the domain, which extend L units in front and behind the bump, and H units above, with both H and L normalized by the arc length. The left boundary has Dirichlet boundary conditions for ρ, u_1 and u_2 (and e, if the inflow is supersonic). The bottom boundary satisfies the no-penetration condition, as does the top boundary in the supersonic case, while it has all four variables set to their free-stream values in the subsonic and transonic cases.

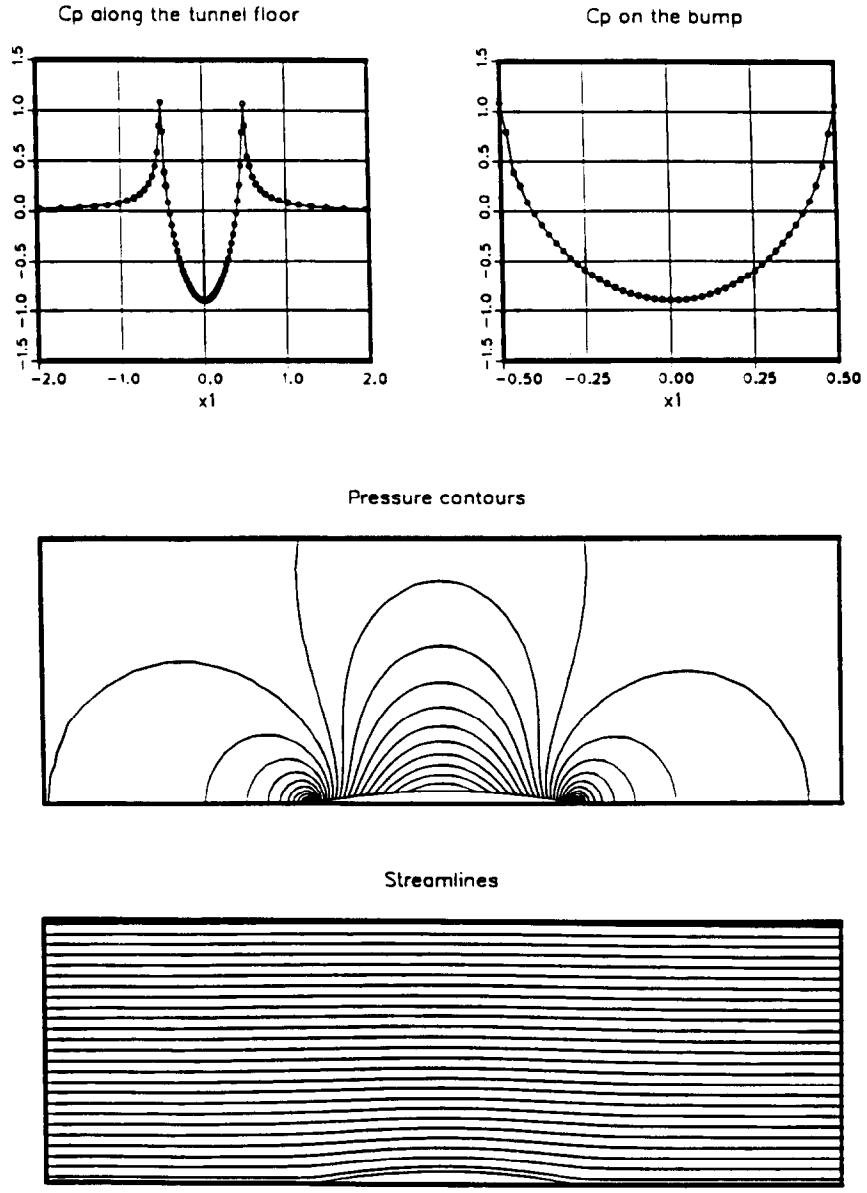

Fig. 3. Flow past a parabolic arc bump at Mach number 0.5: solution obtained by using the conservation variables formulation (without the discontinuity capturing operator).

4.2.1. At Mach number 0.5

The first case we examine is that of subsonic flow past the parabolic arc bump. The free-stream Mach number is 0.5, and the subsonic nature of the flow forces the computational domain to be extended many chord lengths upstream to minimize any upstream boundary effects. The domain parameters are set as follows: $L = 8.0$, $H = 3.0$ and $b = 5\%$. The mesh has 22 elements in the x_2 direction and 94 in the x_1 direction, giving a total of 2068 elements and 2185 nodal points. Figures 3 and 4 were generated without employing the shock capturing

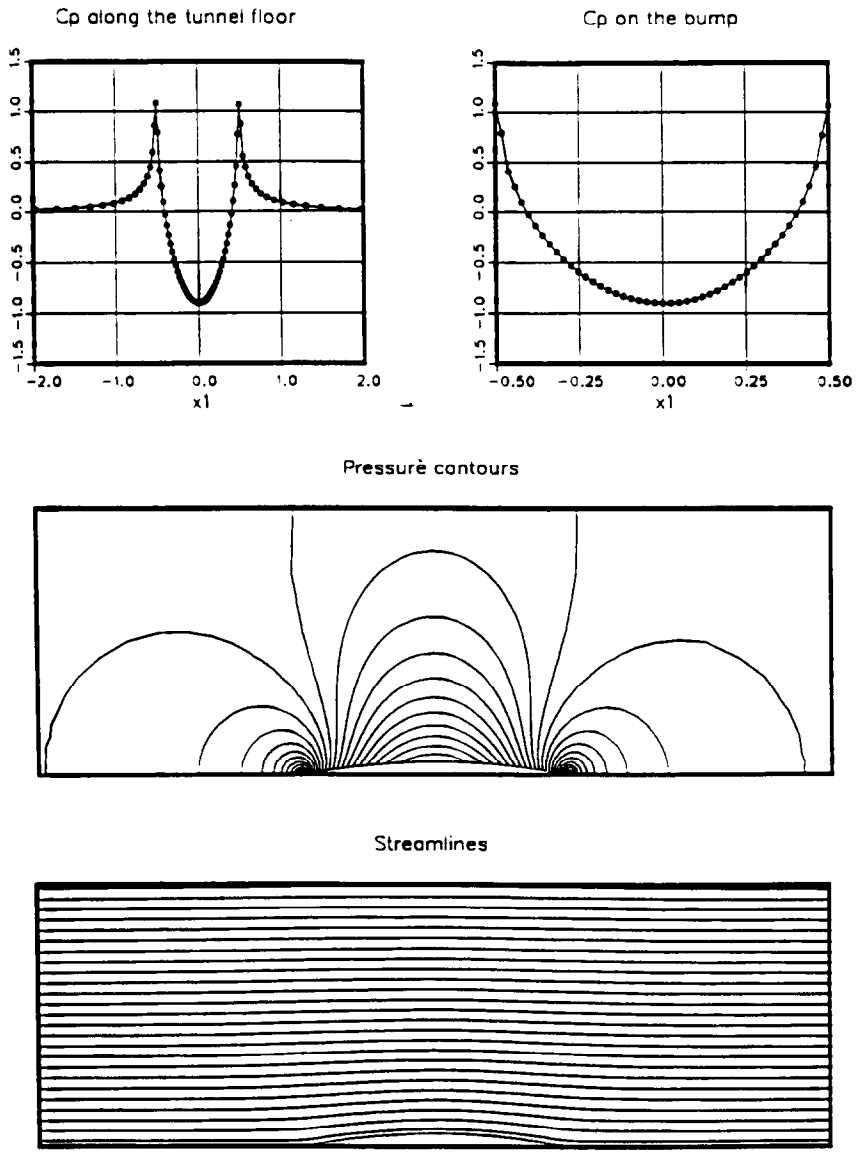

Fig. 4. Flow past a parabolic arc bump at Mach number 0.5: solution obtained by using the entropy variables formulation (without the discontinuity capturing operator).

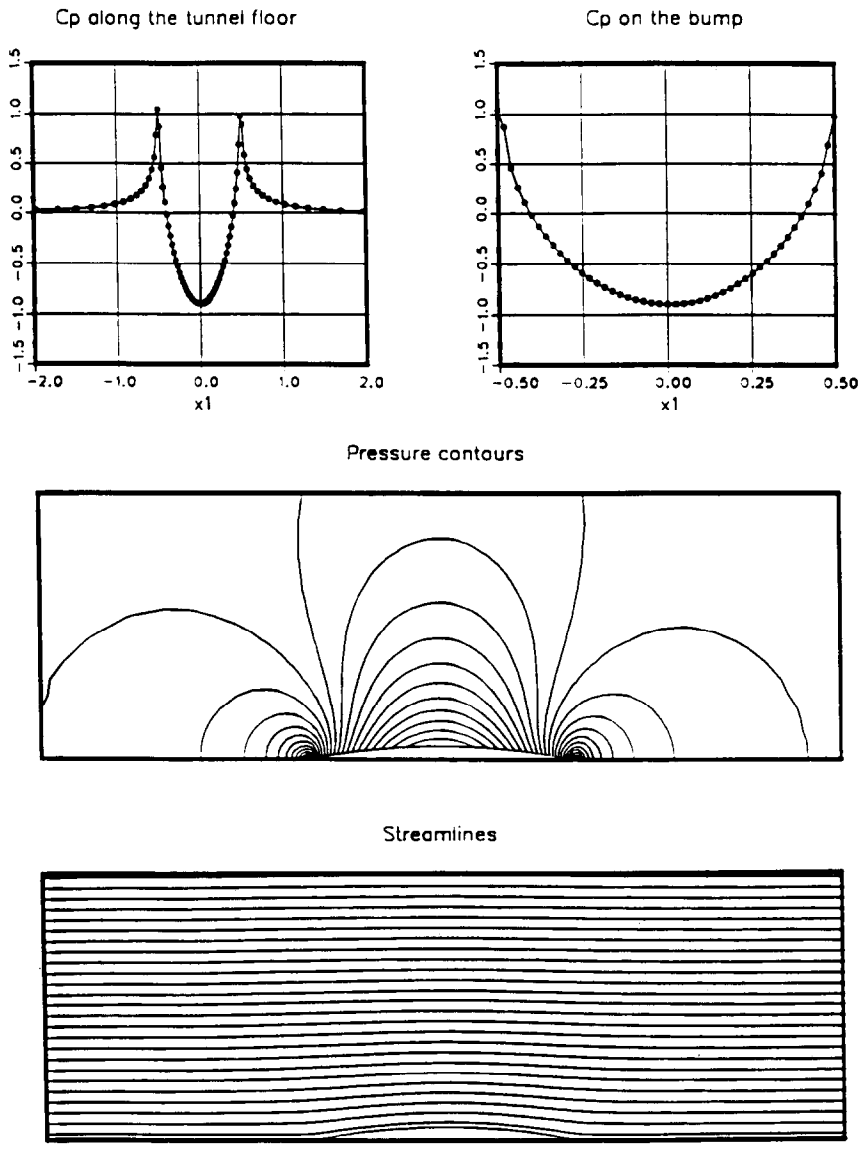

Fig. 5. Flow past a parabolic arc bump at Mach number 0.5: solution obtained by using the conservation variables formulation.

operators, and show the pressure coefficient along the tunnel floor, together with the pressure contours and the streamlines. Figures 5 and 6 show the solutions generated with the shock capturing operator employed; these solutions are only slightly different from those obtained without the shock capturing operators.

4.2.2. At Mach number 0.84

A slightly more challenging case for these formulations is the transonic flow past the parabolic arc bump. A portion of the domain becomes supersonic and a shock wave forms on

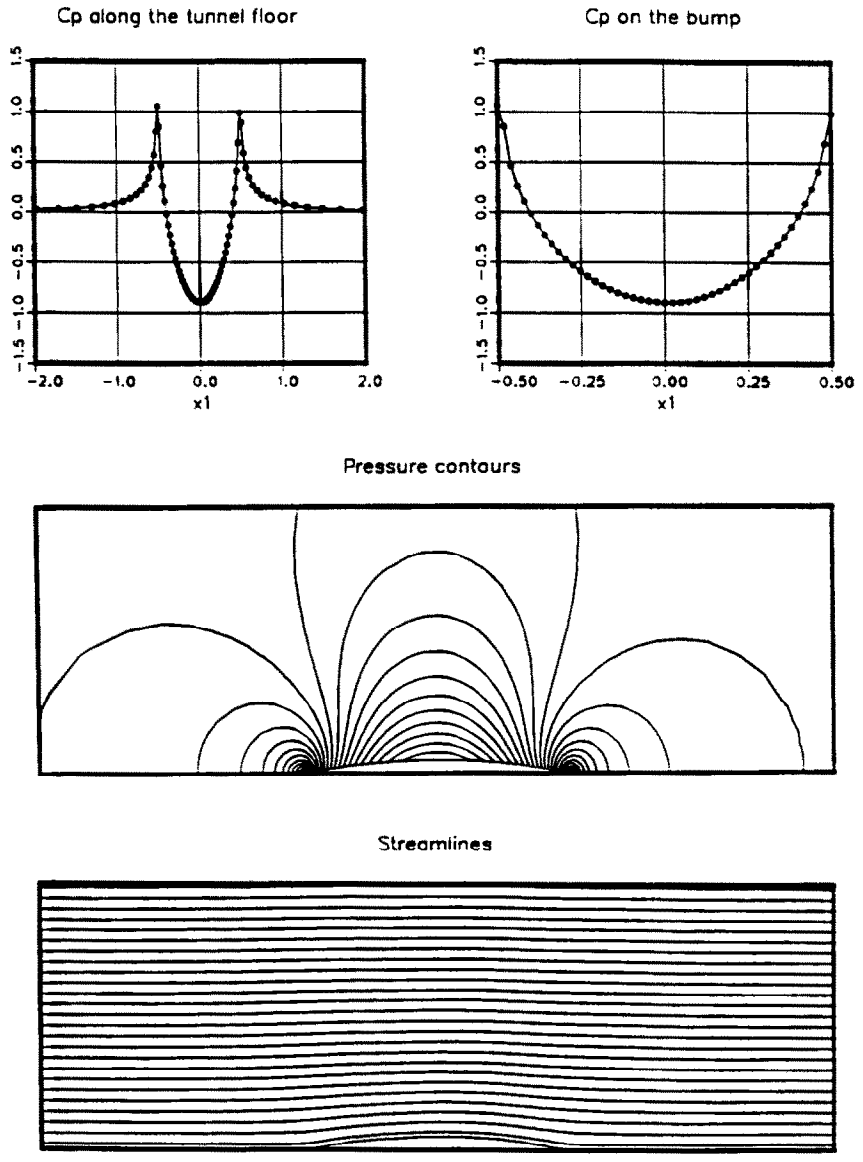

Fig. 6. Flow past a parabolic arc bump at Mach number 0.5: solution obtained by using the entropy variables formulation.

the surface of the bump. The computational domain and mesh are identical to that for the subsonic case, and the free-stream Mach number is set to 0.84.

Figures 7 and 8, generated using the shock capturing terms, show the pressure coefficient along the symmetry axis, together with pressure contours around the bump, and Mach contours in the supersonic region. Figure 7 shows these for the conservation variables formulation, while Fig. 8 shows the same for the entropy variables formulation. Both figures show the formation of the shock at around $x_1 = 0.3$. This shock is captured by both

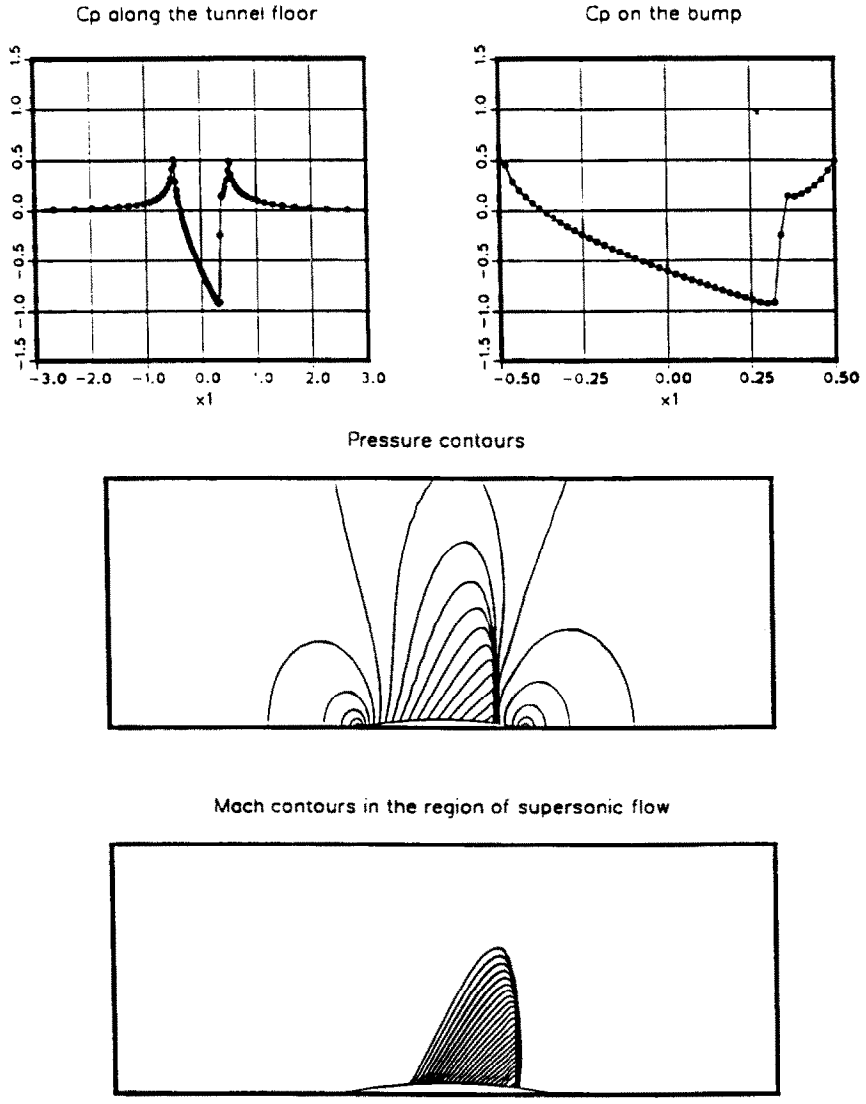

Fig. 7. Flow past a parabolic arc bump at Mach number 0.84: solution obtained by using the conservation variables formulation.

formulations within only two elements, without the overshoots and undershoots often seen with such shocks.

Once again this problem was also computed with both formulations and without the respective shock capturing terms; oscillations appear in the neighborhood of the shock in each solution.

4.2.3. At Mach number 1.4

The case of supersonic flow past the parabolic arc bump poses a more challenging case for the methods presented. The free-stream Mach number is set to 1.4. The domain is also

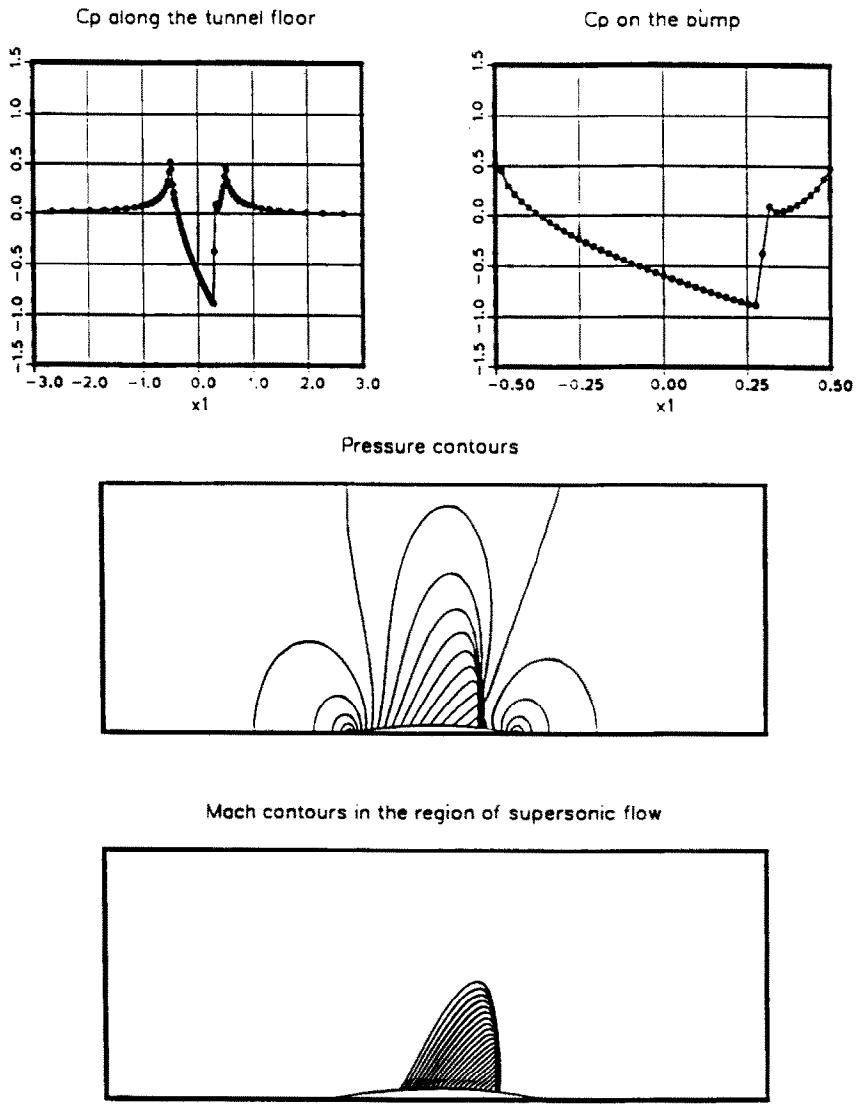

Fig. 8. Flow past a parabolic arc bump at Mach number 0.84: solution obtained by using the entropy variables formulation.

decreased in size, because no information propagates upstream; L and H are now both set to 1.0, and b is set to 4%. The mesh, shown in Fig. 9 has 30 elements in the x_2 direction and 92 in the x_1 direction, giving 2760 elements and 2883 nodal points. Figure 9 also shows, for the conservation variables formulation, the density at $x_2 = 1.0$ and $x_2 = 0.5$, and pressure contours in the neighborhood of the bump. Figure 10 shows the same for the entropy variables formulation. The shock interaction which occurs behind the bump is captured well by both formulations. This interaction is the result of the leading edge shock reflecting off the upper boundary, crossing the trailing edge shock, reflecting again and finally merging with the

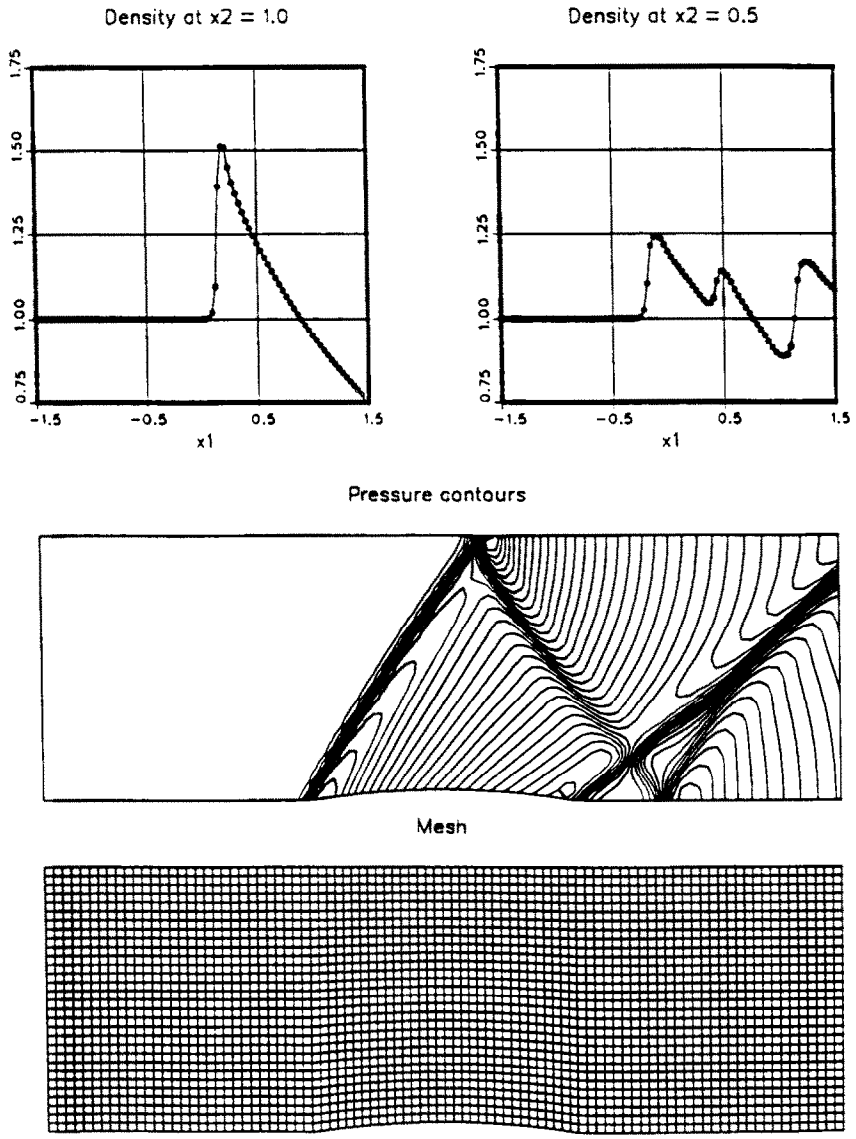

Fig. 9. Flow past a parabolic arc bump at Mach number 1.4: solution obtained by using the conservation variables formulation.

trailing edge shock. Both formulations also capture well the expansion waves caused by the curvature in the arc, and the decrease in strength of the reflected shock, a result of the expansion waves. These two solutions were generated using the shock capturing operators. Solutions were also generated without these terms; and as was the case for the reflected shock and the transonic flow over an arc bump, overshoots and undershoots appear near the shocks. Figure 11 shows this same problem computed using the entropy variables formulation with shock capturing, on a mesh that has twice the number of elements in both the x_1 and x_2

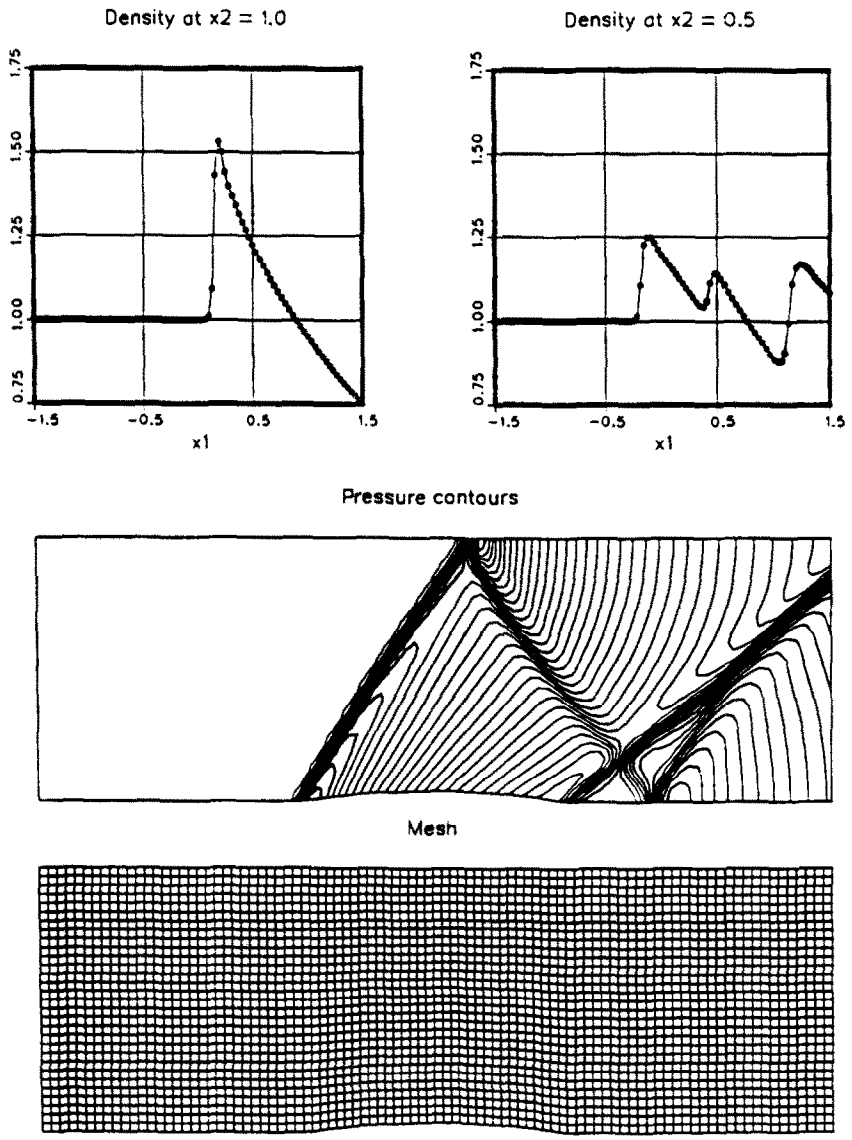

Fig. 10. Flow past a parabolic arc bump at Mach number 1.4: solution obtained by using the entropy variables formulation.

directions, for a total of 11 040 elements and 11 285 nodes. Much better resolution of the shocks compared to the coarser mesh is evident, suggesting that solving this type of problem can be achieved more efficiently by using adaptive meshing algorithms.

4.3. *Flow past a circular cylinder*

In this case, we consider subsonic and supersonic flow past a circular cylinder. Due to the symmetry of the solutions in both cases, only half of the cylinder is modelled. The domain is a

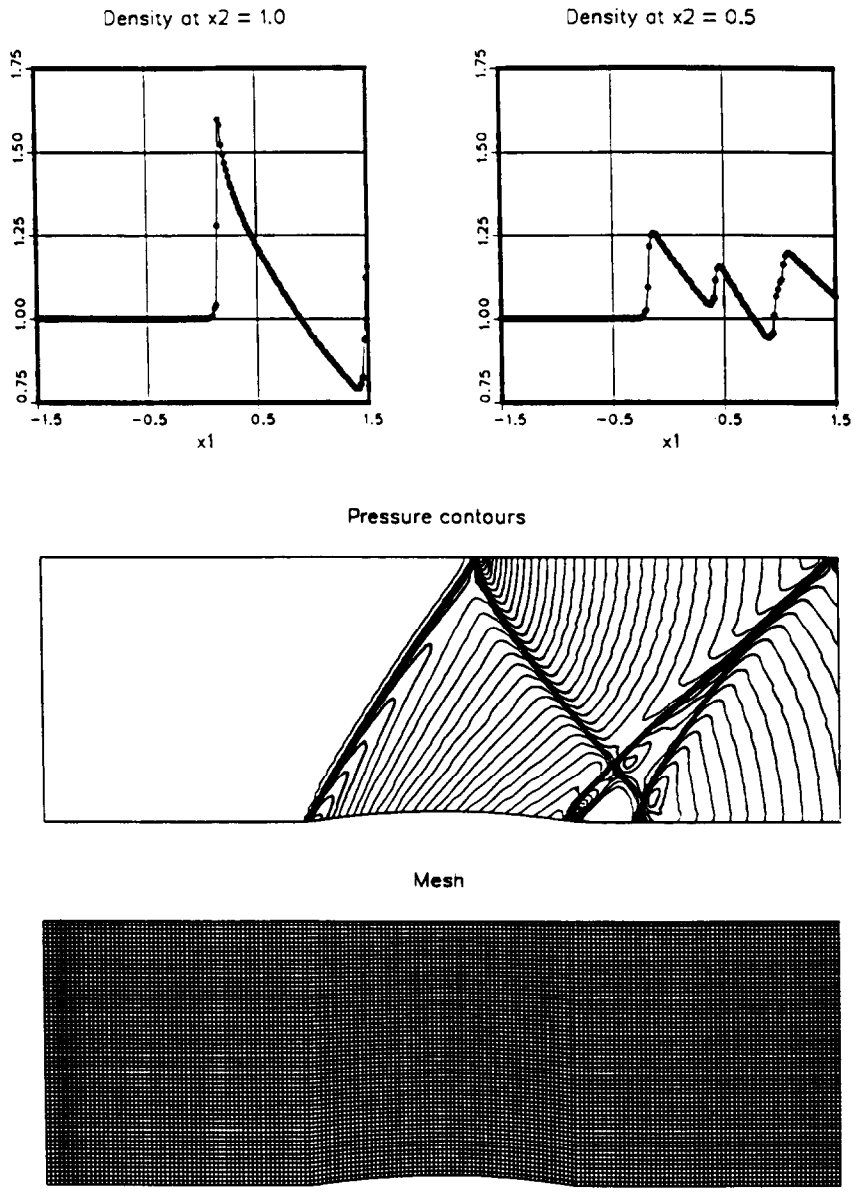

Fig. 11. Flow past a parabolic arc bump at Mach number 1.4: solution obtained by using the entropy variables formulation (with a more refined mesh).

rectangle of height $L/2$ and length L, with L normalized by the cylinder radius, and the half cylinder is located at the center of the bottom boundary. The upstream boundary has Dirichlet boundary conditions for ρ, u_1 and u_2 (and e, if the inflow is supersonic), while the bottom boundary satisfies no-penetration, and the downstream boundary is free. The upper boundary is free in the supersonic case, but all four variables are Dirichlet type and set to their free-stream values in the subsonic case.

4.3.1. At Mach number 0.3

The case of subsonic flow at free-stream Mach number 0.3 is considered first. The computational domain extends 15 radii upstream, downstream and above the cylinder. The mesh has 45 elements in the radial direction and 80 elements in the tangential direction, giving 3600 elements and 3726 nodes. The no-penetration condition is imposed on the cylinder surface using the algorithm described in Appendix A.

Figure 12 shows, for the conservation variables formulation, the pressure coefficient and

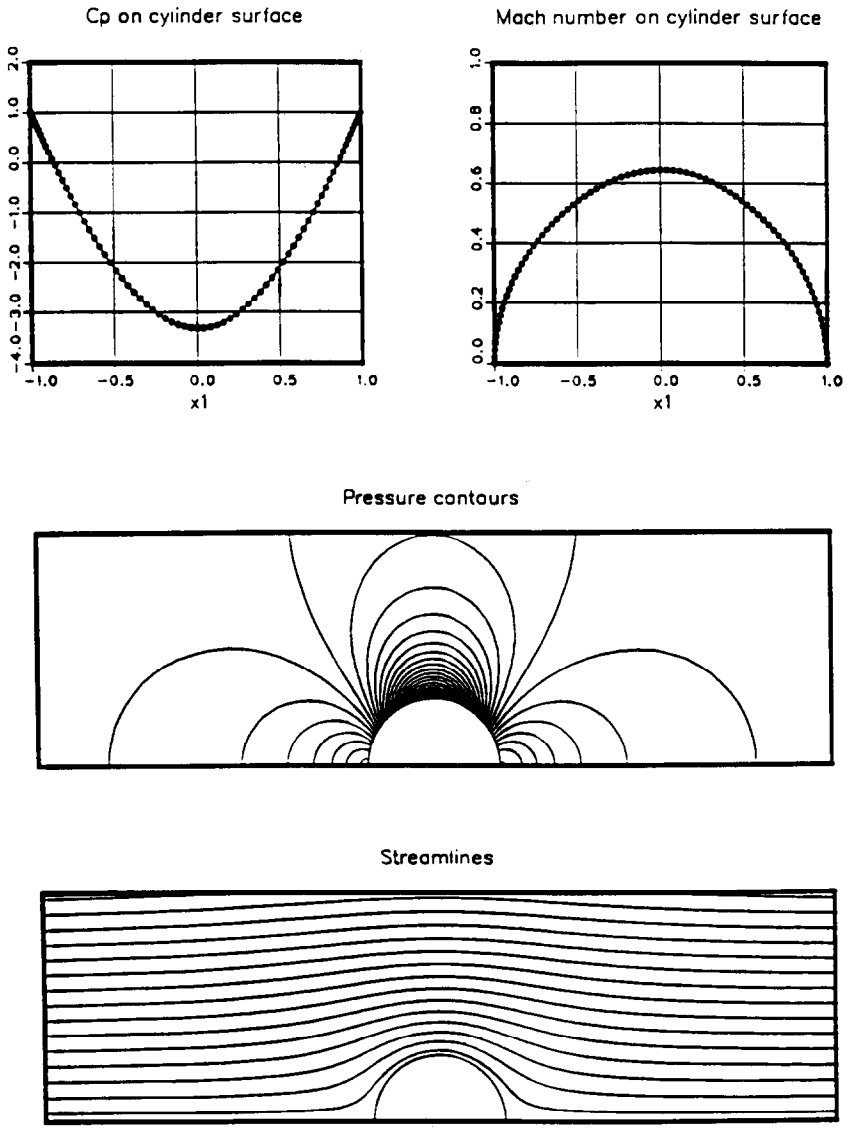

Fig. 12. Flow past a circular cylinder at Mach number 0.3: solution obtained by using the conservation variables formulation (without the discontinuity capturing operator).

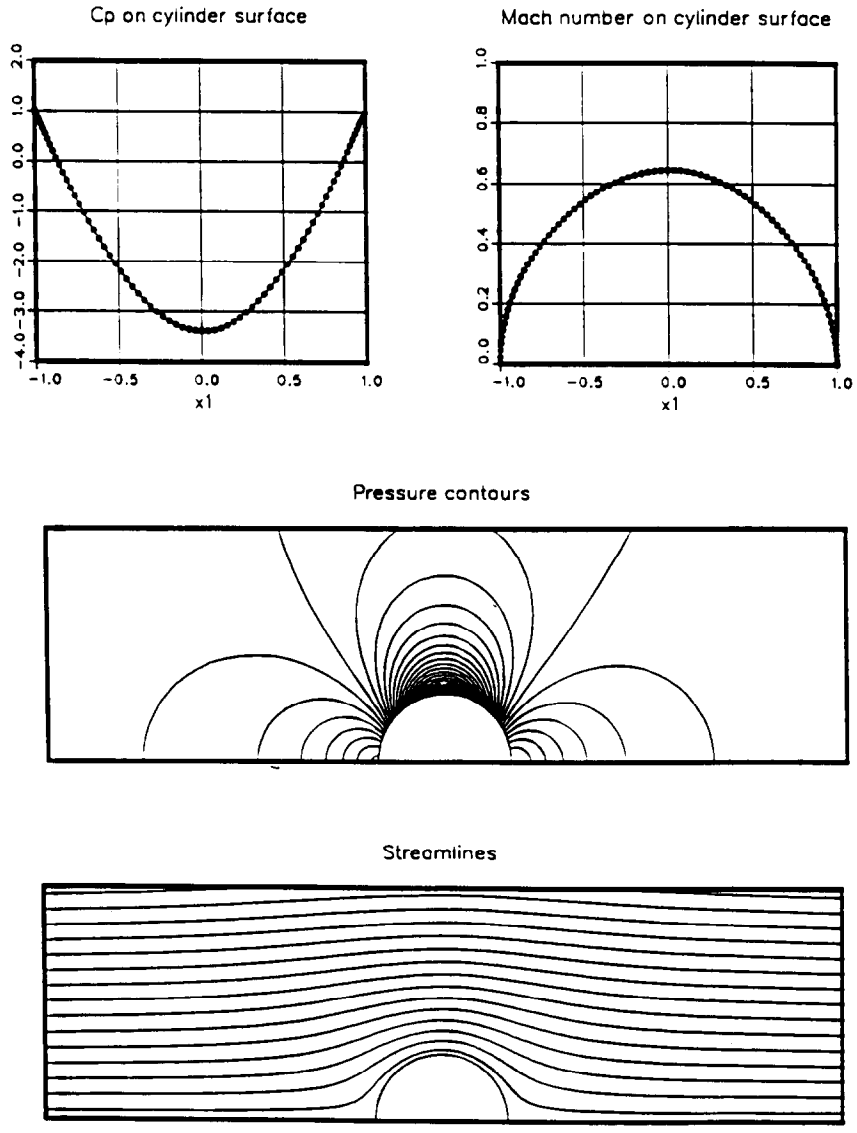

Fig. 13. Flow past a circular cylinder at Mach number 0.3: solution obtained by using the entropy variables formulation (without the discontinuity capturing operator).

Mach number on the cylinder surface, as well as pressure contours and streamlines near the cylinder. Figure 13 shows the same for the entropy variables formulation. Figures 14 and 15 show the previous cases with the shock capturing terms, with only slight changes in the solutions obtained.

4.3.2. At Mach number 3.0

The last problem considered is another challenging one, the case of supersonic flow past the cylinder at free-stream Mach number 3.0. The computational domain again extends 15 radii

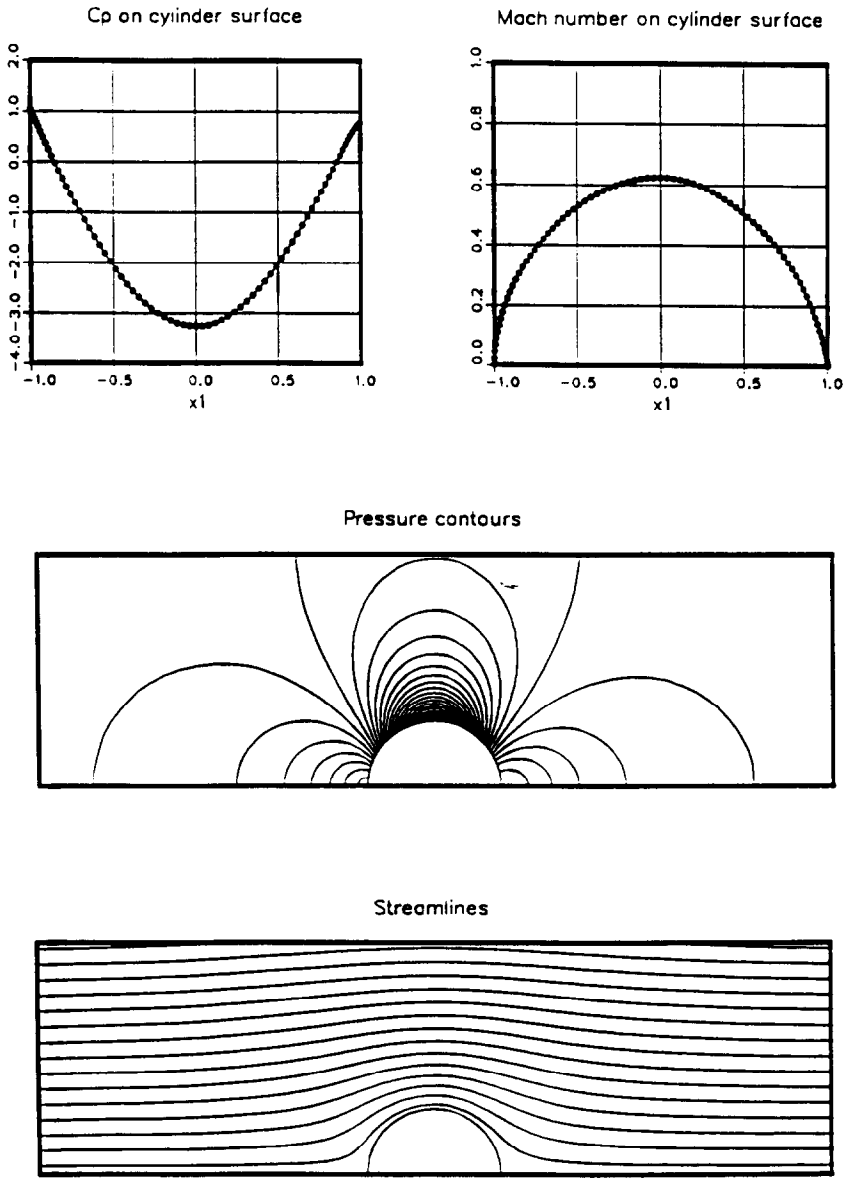

Fig. 14. Flow past a circular cylinder at Mach number 0.3: solution obtained by using the conservation variables formulation.

upstream, downstream and above the cylinder, and the mesh has 60 elements in the radial direction and 80 in the tangential direction, giving 4800 elements and 4941 nodes. Along the upstream boundary, all four primitive variables, density, both velocity components and energy, are specified, and the bottom boundary is a symmetry line except on the cylinder, where the no-penetration condition is imposed. The remaining boundaries are left free. The

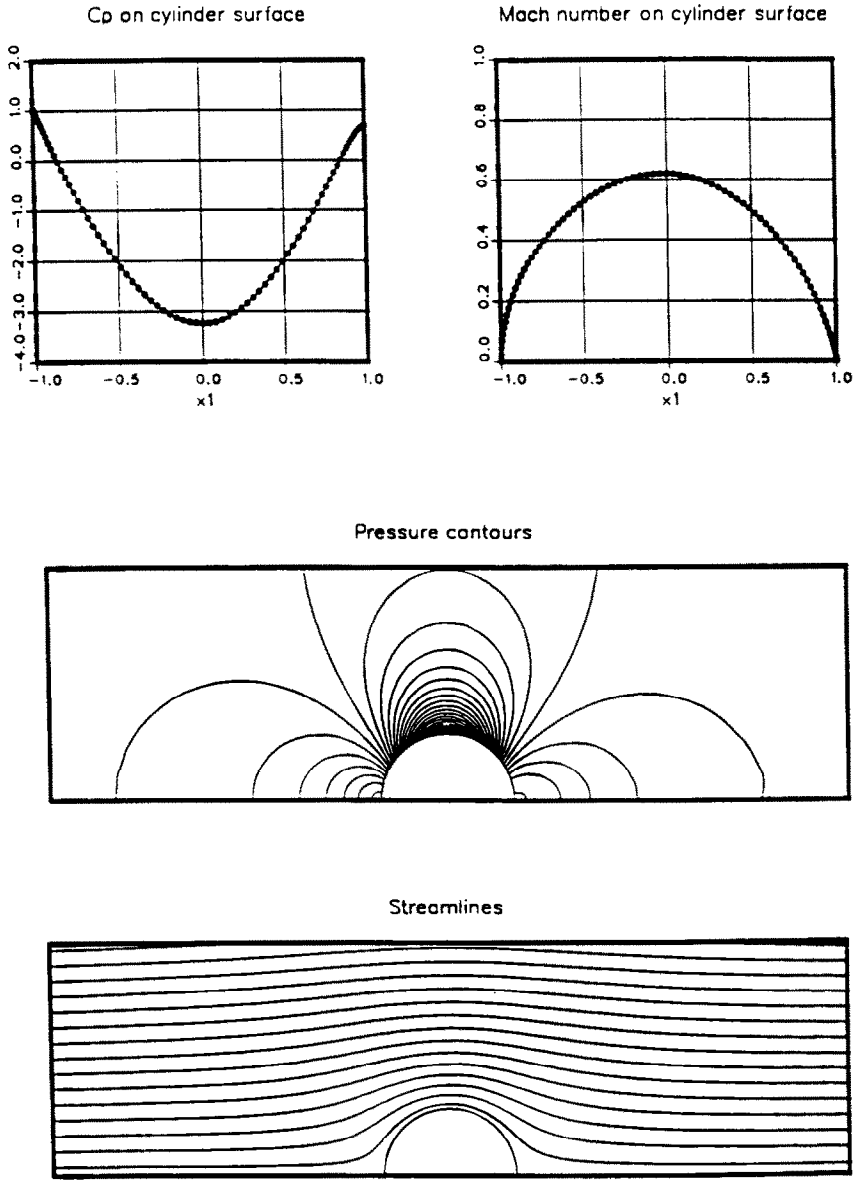

Fig. 15. Flow past a circular cylinder at Mach number 0.3: solution obtained by using the entropy variables formulation.

steady-state solution generated by the conservation variables formulation with shock capturing is shown in Fig. 16, while the same obtained with the entropy variables formulation is shown in Fig. 17. In both cases, the stagnation values and the stand-off distances of the shocks are in good agreement with experimental and theoretic[illegible] values. Solutions could not be obtained by either formulation without using the shock capturing operators.

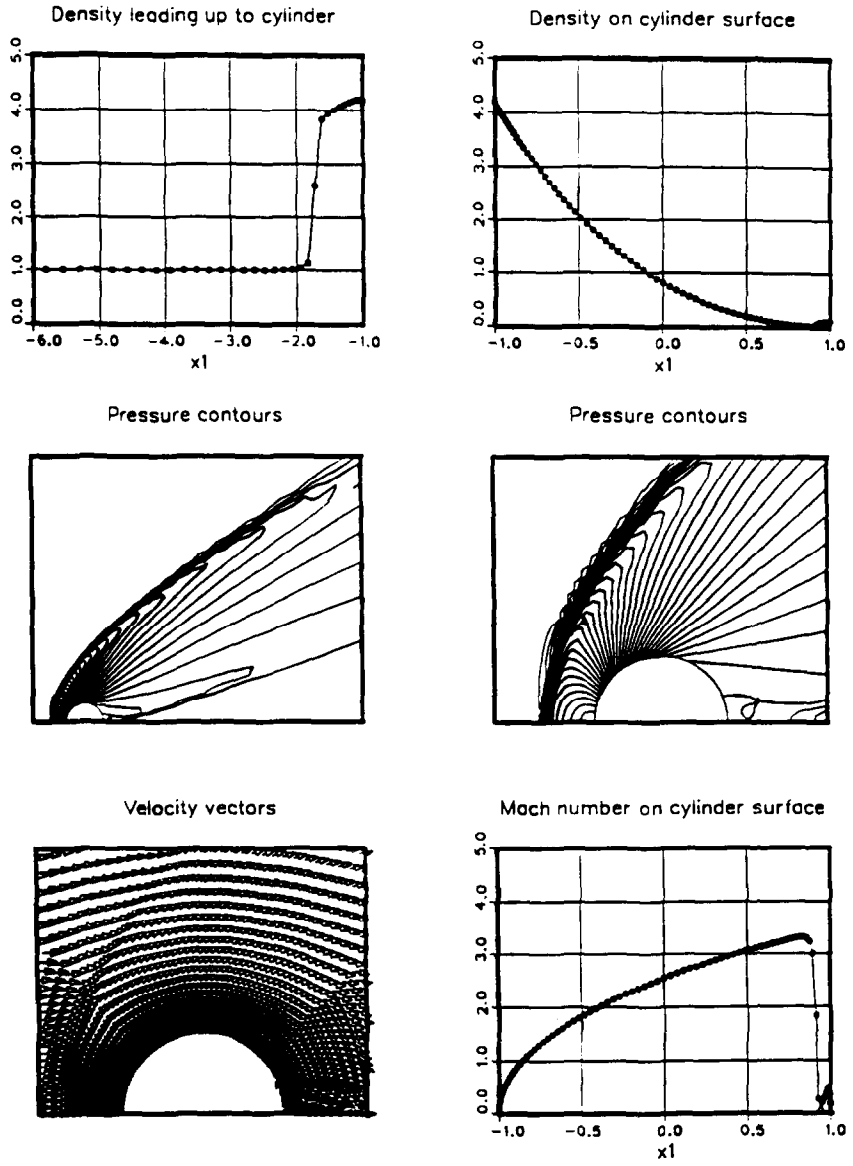

Fig. 16. Flow past a circular cylinder at Mach number 3.0: solution obtained by using the conservation variables formulation.

5. Concluding remarks

The main objective of this work was to investigate and compare the SUPG-stabilized finite element formulations of compressible Euler equations based on the conservation and entropy variables. The formulation based on the conservation variables is essentially the same as the one introduced by Tezduyar and Hughes [1] in 1982, supplemented with a shock capturing term. The formulation based on the entropy variables is the same as the one in [2–4], which

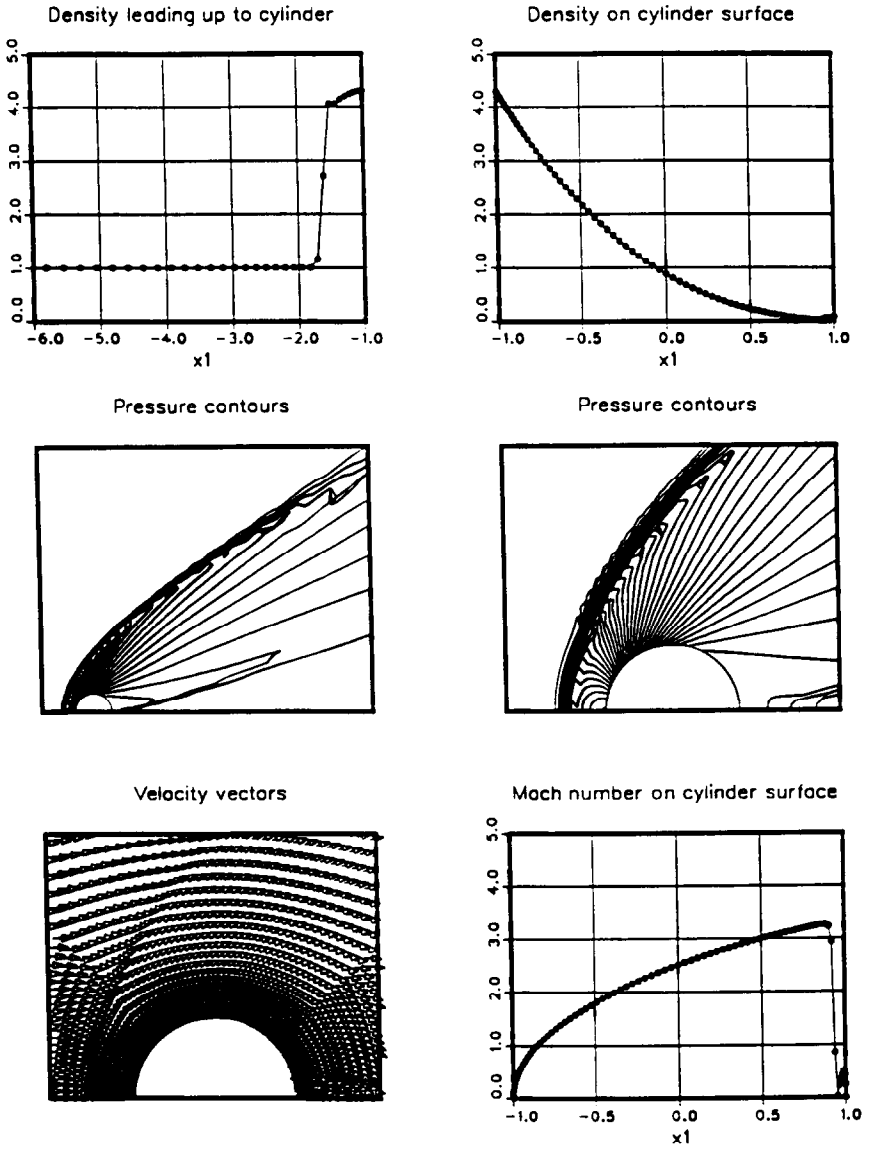

Fig. 17. Flow past a circular cylinder at Mach number 3.0: solution obtained by using the entropy variables formulation.

has a shock capturing term built in. We tested these formulations on several compressible flow problems in subsonic, transonic and supersonic ranges. We were able to show that not only the stabilized formulation based on the conservation variables gives just as good results as those obtained with the entropy variables, but the solutions obtained using the two formulations are very close and in some cases almost indistinguishable. Consequently, we believe that the relative merits of these two formulations will continue to remain under debate, and therefore it is not yet definitive to us if one formulation should be abandoned in favor of the other one.

Appendix A. Implementation of the impermeable wall boundary condition

In the past, typical implementation of the impermeable boundary condition given by (12) involved manipulation of the velocity vectors such that the velocity normal to the surface is zero after each iteration. This was accomplished by simply nullifying the normal component of the velocity at the surface, or by modifying the velocity vector tangent to the surface in an effort to conserve the kinetic energy at each point. These methods of satisfying the boundary condition have been successfully implemented for quite some time, yet it should be realized that this is an explicit treatment of the boundary condition. Thus even when using an implicit method, the stability of the algorithm will be influenced by this explicit treatment of the boundary condition. Herein is a description of the method used for handling the impermeable wall for the case of the conservation variables. This implementation is based on the treatment of the solid surface as a Dirichlet boundary which, by definition of the function spaces used, allows us to dictate that the normal component of the velocity on a stationary surface must be zero. This implicit treatment of the boundary condition has been used to improve the convergence rates of both explicit and implicit algorithms and has also allowed the use of very large Courant numbers to reach the steady-state solutions of Eulerian flows.

In the finite element discretization, the quantity $\boldsymbol{U}^h$ is interpolated within each element by the nodal values $\boldsymbol{U}_B$ and a set of interpolation (shape) functions $\boldsymbol{N}_B$ such that

$$\boldsymbol{U}^h = \sum_{B=1}^{n_{en}} N_B \boldsymbol{U}_B \,, \tag{A.1}$$

where n_{en} is the number of element nodes. In the case of a two-dimensional quadrilateral element, (A.1) can be written as

$$\boldsymbol{U}^h = \begin{bmatrix} (U_1)_1 \\ (U_2)_1 \\ (U_3)_1 \\ (U_4)_1 \end{bmatrix} N_1 + \begin{bmatrix} (U_1)_2 \\ (U_2)_2 \\ (U_3)_2 \\ (U_4)_2 \end{bmatrix} N_2 + \begin{bmatrix} (U_1)_3 \\ (U_2)_3 \\ (U_3)_3 \\ (U_4)_3 \end{bmatrix} N_3 + \begin{bmatrix} (U_1)_4 \\ (U_2)_4 \\ (U_3)_4 \\ (U_4)_4 \end{bmatrix} N_4 \,, \tag{A.2}$$

where $U_1 = \rho$, $U_2 = \rho u_1$, $U_3 = \rho u_2$, $U_4 = \rho e$, and the second subscript associates the nodal vector of unknowns to its corresponding shape function.

Equation (A.2) uses a nodal vector of unknowns based on a global Cartesian coordinate system. Such a system does not directly allow enforcement of the requirement that the normal component of the velocity on a surface be zero because such a requirement has to be imposed on a geometric combination of the x_1 and x_2 components of the velocity at the surface node. Therefore a new nodal vector of unknowns for surface nodes is defined where the second and third components of this new vector represent components of the velocity in the tangential (s) and normal (n) directions, respectively. This new vector is defined as

$$\boldsymbol{d} = \begin{bmatrix} \rho \\ \rho u_s \\ \rho u_n \\ \rho e \end{bmatrix} . \tag{A.3}$$

Having the vector of unknowns in a local coordinate system allows the third degree of freedom to be specified as a Dirichlet boundary condition.

Although introduction of a rotated coordinate system permits the direct enforcement of a no-penetration condition, there may be a need for many such coordinate systems depending on the shape and resolution of the surface. It then becomes obvious that it will not be possible to obtain a solution over a spatial domain where, conceivably, each node may use a separate rotated coordinate system. Thus each of these s–n systems must be rotated to one uniform system so that the proper governing equations can be used. In this work the flux vectors on solid surfaces are rotated to a global Cartesian coordinate system as defined in the problem statement. For example, in considering a typical boundary element with nodes 3 and 4 lying on the boundary, (A.2) would be modified for the boundary nodes, such that

$$\boldsymbol{U}^h = \begin{bmatrix} (U_1)_1 \\ (U_2)_1 \\ (U_3)_1 \\ (U_4)_1 \end{bmatrix} N_1 + \begin{bmatrix} (U_1)_2 \\ (U_2)_2 \\ (U_3)_2 \\ (U_4)_2 \end{bmatrix} N_2 + \boldsymbol{R}_3 \begin{bmatrix} (d_1)_3 \\ (d_2)_3 \\ (d_3)_3 \\ (d_4)_3 \end{bmatrix} N_3 + \boldsymbol{R}_4 \begin{bmatrix} (d_1)_4 \\ (d_2)_4 \\ (d_3)_4 \\ (d_4)_4 \end{bmatrix} N_4 , \tag{A.4}$$

where $\boldsymbol{R}_3$ and $\boldsymbol{R}_4$ are rotation matrices which rotate the normal and tangential components for nodes 3 and 4, to the global Cartesian system. Such nodal rotation matrices are defined as

$$\boldsymbol{R}_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_B & -\sin\theta_B & 0 \\ 0 & \sin\theta_B & \cos\theta_B & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} , \tag{A.5}$$

where θ_B is the angle measured from the x_1 axis to the s basis vector for the surface at node B. In general, this can be written for all elements as

$$\boldsymbol{U}^h = \sum_{B=1}^{n_{en}} N_B \boldsymbol{R}_B \boldsymbol{d}_B , \tag{A.6}$$

where the s and n components of $\boldsymbol{d}$ for those nodes not on a solid surface would correspond to the global Cartesian components, and the rotation matrices would simply be the identity matrix. The same type of rotation must also take place on the test functions;

$$\boldsymbol{W}^h = \sum_{A=1}^{n_{en}} N_A \boldsymbol{R}_A \boldsymbol{c}_A . \tag{A.7}$$

The computational cost associated with using the preceding rotation matrices is not insignificant (multiplication of three 4×4 matrices for each nodal interpolation and integration point of each element), thus it is a worthwhile endeavor to look for the most efficient manner of implementing this method. For this work, two schemes have been used to minimize the cost. The first one involves the formation of an array which keeps track of boundary nodes so that the rotation multiplications only occur when needed. The second strategy recognizes that each of the terms used in the formation of the element level matrices need not be rotated during formation, but instead the entire element level matrix can be rotated after the

formation has been completed. This is a favorable strategy in that it maintains the ability to vectorize element level program loops.

This implicit handling of the no-penetration condition can also be implemented, in a similar manner, in the entropy variables formulation [13].

References

[1] T.E. Tezduyar and T.J.R. Hughes, Development of time-accurate finite element techniques for first-order hyperbolic systems with particular emphasis on the compressible Euler equations, Report prepared under NASA-Ames University Consortium Interchange, No. NCA2-OR745-104, 1982.

[2] 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, Comput. Methods Appl. Mech. Engrg. 54 (1986) 223–234.

[3] 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, Comput. Methods Appl. Mech. Engrg. 58 (1986) 305–328.

[4] T.J.R. Hughes and M. Mallet, A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems, Comput. Methods Appl. Mech. Engrg. 58 (1986) 329–336.

[5] C.P. Li, Computation of three-dimensional flow about aerobrake configurations, AIAA Paper 86-0566.

[6] T.J.R. Hughes and A.N. Brooks, A multi-dimensional upwind scheme with no crosswind diffusion, in: T.J.R. Hughes, ed., Finite Element Methods for Convection Dominated Flows, AMD Vol. 34 (ASME, New York, 1979) 19–35.

[7] A.N. Brooks and T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.

[8] T.E. Tezduyar and T.J.R. Hughes, Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations, Proc. AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, AIAA Paper 83-0125 (1983).

[9] T.J.R. Hughes and T.E. Tezduyar, Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations, Comput. Methods Appl. Mech. Engrg. 45 (1984) 217–284.

[10] J. Donea, A Taylor–Galerkin method for convective transport problems, Internat. J. Numer. Methods Engrg. 20 (1984) 101–120.

[11] G.J. Le Beau and T.E. Tezduyar, Finite element computation of compressible flows with the SUPG formulation, in: M.N. Dhaubhadel, M.S. Engelman and J.N. Reddy, eds., Advances in Finite Element Analysis in Fluid Dynamics, FED Vol. 123 (ASME, New York, 1991) 21–27.

[12] M. Mallet, A finite element method for computational fluid dynamics, Ph.D. Thesis, Department of Civil Engineering, Stanford University, 1985.

[13] F. Shakib, Finite element analysis of the compressible Euler and Navier–Stokes equations, Ph.D. Thesis, Department of Mechanical Engineering, Standford University, 1988.