

A PETROV-GALERKIN TECHNIQUE FOR  
THE SOLUTION OF  
TRANSONIC AND SUPERSONIC FLOWS

C. E. BAUMANN†

M. A. STORTI‡

S. R. IDELSOHN †

Computational Mechanics Department

(CONICET - UNL)

Güemes 3450 - 3000 Santa Fe - Argentina

ABSTRACT

This paper is both the description of a streamline-upwind/Petrov-Galerkin (SUPG) formulation and the documentation of the development of a code for the finite element solution of transonic and supersonic flows. The aim of this work is to present a formulation to be able to treat domains of any configuration and to use the appropriate physical boundary conditions, which are the major stumbling blocks of the finite difference schemes.

The implemented code has the following features : The Hughes' SUPG-type formulation with an oscillation-free shock-capturing operator, adaptive refinement, explicit integration with local time-step and hourglassing control. An automatic scheme for dealing with slip boundary conditions and a boundary-augmented lumped mass matrix for speeding up convergence.

In Section 1 we will describe briefly the theoretical background of the SUPG formulation. In Section 2 it is described how the foregoing formulation was used in the finite element code and which are the appropriate boundary conditions to be used. Finally in Section 3 we will show some results obtained with this code.

---

† Graduate Reserch Assistant

‡ Professor and Scientific Researcher

## SECTION 1

### Streamline-Upwind Petrov-Galerking formulations

We begin analyzing the formulations for the one-dimensional compressible Euler equations and the bidimensional scalar advective equation. Afterwards it is explained what was used for dealing with the multidimensional compressible Euler equations.

### One-dimensional compressible Euler equations :

The compressible Euler equations constitute a first-order hyperbolic system that can be written as follows

$$U_{,t} + F_{,x} + G = 0,$$

where the vector  $F$  is referred to as the flux vector and  $G$  is a source term. This vector equation stands for the conservation of mass, momentum, and energy in the flow field. Using the Jacobian matrix  $A = \partial F / \partial U$ , we can also write

$$U_{,t} + AU_{,x} + G = 0. \quad (1)$$

In what follows, we will consider a system of equations like system 1 without the source term.

Using Taylor's theorem we can write

$$U^{n+1} = U^n + U_{,t}^n \Delta t + U_{,tt}^n \frac{\Delta t^2}{2!} + \alpha(\Delta t^2),$$

and the substitution of eq 1 in the above equation leads to

$$U^{n+1} = U^n - AU_{,x}^n \Delta t + A^2 U_{,xx}^n \frac{\Delta t^2}{2!} + \alpha(\Delta t^2), \quad (2)$$

where it was assumed a constant advection matrix. Because only the steady state is of our concern, we can neglect the terms of higher order of the series (see [1] for further details).

The Jacobian matrix  $A$  can be diagonalized (see [12]), therefore we can write

$$A = SAS^{-1},$$

and making the change of variables  $V = S^{-1}U$  we transform eq 2 obtaining

$$V^{n+1} = V^n - \Lambda V_{,x}^n \Delta t + \Lambda^2 V_{,xx}^n \frac{\Delta t^2}{2!}, \quad (3)$$

which is a system of uncoupled scalar equations.

If eq 3 were to converge to a steady state we would have the problem solved. Now it is important to emphasize that the temporal integration is only a way to reach the steady state, and that this procedure can be regarded as a relaxation process.

At this stage we can make a spatial discretization of eq 3 with linear finite elements, which yield central differences in space, and investigate the behavior of the resulting scheme. For an assemblage of elements of uniform length we get

$$V_j^{n+1} = V_j^n - \Delta t \frac{\Lambda}{2h} (V_{j+1}^n - V_{j-1}^n) + \frac{\Delta t^2 \Lambda^2}{2h^2} (V_{j+1}^n - 2V_j^n + V_{j-1}^n). \quad (4)$$

The stability of this scheme can be assessed using the von Neumann analysis, based on Fourier analysis. The Fourier decomposition of the continuous solution is ( summation on  $l$  is assumed )

$$V^n = H^n(k_l) e^{ik_l x},$$

and that of the discretized case is

$$V_j^n = H^n(k_l) e^{ik_l j h}. \quad (5)$$

Here,  $i$  is the imaginary unit used to represent the sinusoidal functions with wave numbers  $k_l$ , and  $H^n(k_l)$  is the amplitude of the particular wave component  $k_l$ .

Substitution of eq 5 in eq 4 and the use of  $C = \left(\frac{\Delta t}{h}\right) \Lambda$  gives

$$V_j^{n+1} = \left[ 1 - \frac{C}{2} (e^{ik_l h} - e^{-ik_l h}) + \frac{C^2}{2} (e^{ik_l h} - 2 + e^{-ik_l h}) \right] H^n(k_l) e^{ik_l j h},$$

$$V_j^{n+1} = G(C) V_j^n,$$

where  $C$  is a diagonal matrix in which the diagonal elements are the CFLNs of the eigenmodes. The latter equation gives the evolution of each Fourier component ( interpreted either as a part of the solution or as a perturbation error ).

The *norm condition*

$$\|G(C)\| \leq 1$$

is sufficient for the stability. Because  $G(C)$  is a symmetric (diagonal) matrix, if we use the norm  $\| \cdot \|_2$ , there follows that

$$\rho(G(C)) = \|G(C)\|_2,$$

and the *norm condition* is satisfied if

$$|\mu_i| \leq 1 \quad \forall i,$$

in which  $\mu_i$  represents the eigenvalues of the amplification matrix  $G(C)$ . The analysis of the diagonal entries of  $G(C)$  gives

$$|\mu_i| \leq 1 \Rightarrow C_i^2 \leq 1 \quad \forall i.$$

As a result, we can conclude that

- |    |           |   |
|----|-----------|---|
| if | $C_i > 1$ | the iterates blows up,  |
|    | $C_i = 1$ | exact nodal values are obtained,                              |
|    | $C_i < 1$ | spurious oscillations develops, as will shortly become clear. |

A foregone conclusion is that the only stable way of treating the system of eq 4 is to integrate it with a time-step based on the greatest eigenvalue, but obviously in that case spurious oscillations will appear in those eigenmodes integrated with a  $CFLN < 1$ . It is thus because the steady state is reached when the sequence

$$V_j^{n+1} = V_j^n - \frac{C}{2} [(V_{j+1}^n - V_{j-1}^n) - C(V_{j+1}^n - 2V_j^n + V_{j-1}^n)] \quad (6)$$

converges, but for reaching convergence the term within the square brackets must vanish, i.e.

$$V_{j+1}^n - V_{j-1}^n = C (V_{j+1}^n - 2V_j^n + V_{j-1}^n),$$

now, considering the case in which the CFLN tends to zero, we can see the source of the oscillations because  $(-1)^j$  is a solution for the uniform mesh.

To avoid this drawback, we can think of a scheme in which the  $C_i$  is substituted for  $sgn(\lambda_i)$  in every component within the square brackets of eq 6. Introducing this modification in the original equation, we obtain the new difference equation

$$V_j^{n+1} = V_j^n - \frac{\Delta t}{2h} A (V_{j+1}^n - V_{j-1}^n) + \frac{\Delta t}{2h} |A| (V_{j+1}^n - 2V_j^n + V_{j-1}^n).$$

Replacing  $V_j$  by  $S^{-1}U_j$  and premultiplying by  $S$ , we obtain the final formulation

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{2h} A (U_{j+1}^n - U_{j-1}^n) + \frac{\Delta t}{2h} |A| (U_{j+1}^n - 2U_j^n + U_{j-1}^n). \quad (7)$$

The finite element discretization both in space and time of the previous formulation is the following:

The boundary  $\Gamma$  of the domain  $\Omega$  is assumed to be decomposed as follows

$$\Gamma = \overline{\Gamma_{u_i} \cup \Gamma_{f_i}}, \quad \emptyset = \Gamma_{u_i} \cap \Gamma_{f_i}, \quad (i = 1, 2, 3),$$

here,  $\Gamma_{u_i}$  refers to that part of the boundary on which a Dirichlet-type b.c. is specified for the  $i$ -th component of the primitive variables (i.e.,  $\rho, u, p$ ), and  $\Gamma_{f_i}$ , to that part on which no b.c. is specified for the  $i$ -th component. There exists another b.c. which is imposed on the slip part of  $\Gamma$ , but it does not make sense in the one-dimensional case.

Let  $V^i$  and  $S^i$  denote the finite-dimensional subsets of  $H^1(\Omega)$  satisfying the following conditions

$$N_i \in V^i \Rightarrow N_i(x) \doteq 0 \quad \text{only when } x \in \{\Gamma_{inflow} \text{ with Mach} > 1\}$$

and

$$U_i(u_1, u_2, u_3) \in S^i \Rightarrow u_i(x) \doteq \bar{u}_i(x) \quad \forall x \in \Gamma_{u_i},$$

where  $N_i$  is the typical finite element weighting function,  $U_i$  the  $i$ -th component of the trial solutions in conservation variables, and the function  $\tilde{u}_i(x)$ , the Dirichlet b.c. for the  $i$ -th component of the primitive variables.

We assume that both subsets consist of the typical  $C^0$  finite element interpolation functions, and that the so-called *group approximation* of the flux vector  $F$  is employed so that its components are also piecewise bilinear functions (for bilinear form functions) determined by their values at element nodes. This finite approximation leads to

$$U = \sum_{j=1}^{numnp} N^j U^j, \quad F = \sum_{j=1}^{numnp} N^j F^j,$$

where  $numnp$  denotes the total number of nodes in the discretization,  $N^j = \text{diag}(N_1^j, N_2^j, N_3^j)$  are the global piecewise bilinear basis functions, and  $U^j, F^j$  are the values of  $U, F$  at node  $j$ .

We have now all the elements to give the space-time finite element formulation equivalent to the difference equation 7 when forward Euler differencing is used for representing the time derivative term. The formulation is the following

$$\left( \int_{\Omega} N^j U_{,t} d\Omega \right)_{t_{nmp}} + \int_{\Omega} \left( N^j + N^j_{,x} \left( \frac{h}{2} \right) \text{sgn } A \right) A U_{,x} d\Omega = 0$$

$$\forall N_i^j \in V^i.$$

It is important to recognize in this formulation a weighted residual method, that is, consistency is insured. Also this formulation is conservative, it is thus because in every point of  $\Omega$

$$\sum_{j=1}^{NodElem} N^j = 1, \quad \text{and therefore} \quad \sum_{j=1}^{NodElem} N^j_{,x} = 0,$$

then

$$\sum_{j=1}^{NodElem} \int_{\Omega^e} \left( N^j + N^j_{,x} \left( \frac{h}{2} \right) \text{sgn } A \right) A U_{,x} d\Omega = \sum_{j=1}^{NodElem} \int_{\Omega^e} N^j A U_{,x} d\Omega;$$

and integrating by parts the right-hand-side, we get

$$\begin{aligned} \sum_{j=1}^{NodElem} \left( \int_{\Gamma^e} N^j A U n_x d\Gamma - \int_{\Omega^e} N^j_{,x} A U d\Omega \right) &= \\ &= \sum_{j=1}^{NodElem} \int_{\Gamma^e} N^j A U_n d\Gamma = \int_{\Gamma^e} F_n d\Gamma, \end{aligned}$$

therefore, the formulation is conservative (this proof, with some modifications, holds for the multidimensional case).

Two-dimensional scalar linear advective equation :

The governing differential equation can be written as follows

$$a_i u_{,i} = 0, \quad (8)$$

where  $u$  is the unknown scalar field and  $a_i$  is the  $i$ -th component of the flow velocity. Eq 8 together with the appropriate boundary conditions define a well posed physical problem (see [2] and [9] for a comprehensive description).

The residual formulation, however, has to take into account the nature of the physical process. In the advective process, the value of the scalar field downstream is that one resulting from the verification of the advective equation upstream. The foregoing statement is not unlike the following : the value of the scalar field in a nodal point of the discretized problem has to be that one which minimizes the residue  $R = a_i u_{,i}$  upstream from that nodal point in a given weighted form.

Let  $V$  and  $S$  denote the finite-dimensional subsets of  $H^1(\Omega)$  satisfying the following conditions

$$N^j \in V \Rightarrow N^j(x) \doteq 0 \quad \forall x \in \Gamma_u (\Gamma_u \equiv \Gamma_{inflow})$$

and

$$u \in S \Rightarrow u(x) \doteq \tilde{u}(x) \quad \forall x \in \Gamma_u,$$

where  $N^j$  are the typical finite element weighting functions, and  $u$  the trial solutions.

We assume that both subsets consist of the typical  $C^0$  finite element interpolation functions.

The weighted residual formulation is the following

$$\int_{\Omega} N^j(x, y) R(x - \Delta x, y - \Delta y) d\Omega = 0 \quad \forall N^j \in V, \quad (9)$$

with

$$\Delta x_i = a_i \tau = \frac{h}{\alpha} \frac{a_i}{|\mathbf{a}|},$$

where  $h$  represents the length of the element side, and  $\tau$  is a characteristic time that is a function of the element size and of the flow velocity.

Using a truncated Taylor expansion, we obtain the following approximation to eq 9

$$\int_{\Omega} N^j(x, y) (R(x, y) - R_{,x_i} \Delta x_i) d\Omega = 0,$$

integrating by parts, we obtain

$$\int_{\Omega} (N^j R + N^j_{,x_i} R \Delta x_i) d\Omega - \int_{\Gamma_{outflow}} N^j R \Delta x_i n_i d\Gamma = 0,$$

and without considering the contour integral, we have the final formulation

$$\int_{\Omega} \left( N^j + N^j_{,x_i} \frac{h}{\alpha} \frac{a_i}{|\mathbf{a}|} \right) (a_i u_{,i}) d\Omega = 0 \quad \forall N^j \in V. \quad (10)$$

The value of  $\alpha$  can be chosen so that in the unidimensional case the solution will have nodal exactness.

In the unidimensional case this formulation reduces to

$$\int_{\Omega} \left( N^j + N_{,\xi}^j \left( \frac{2}{\alpha} \right) \operatorname{sgn}(a) \right) (a u_{,x}) d\Omega = 0, \quad (11)$$

where  $\xi$  refers to the natural coordinate system  $[-1,1]$ . Due to the form of the resulting weighting functions when  $\alpha$  takes the value 2 (see Figure 1), nodally exact solutions are obtained no matter how much different in length the elements may be. Therefore the optimal value of  $\alpha$  to be used in eq 10 is 2.

With the insight gained in the one-dimensional case, we can see that for the two-dimensional case, the following formulation

$$b_i = a_j \frac{\partial \xi_i}{\partial x_j}, \quad |b| = (b_i b_i)^{1/2},$$

$$\int_{\Omega} \left( N^j + N_{,\xi_i}^j \frac{b_i}{|b|} \right) (a_i u_{,i}) d\Omega = 0 \quad \forall N^j \in V, \quad (12)$$

in which  $\xi_i$  refers to the  $i$ -th natural coordinate, will give nodally exact solutions for those flows parallel to the mesh directions, no matter how much different the sides of the elements may be. In any other situation, the approximation will be much better than the one of eq 10.

When adaptive refinement is used, neither eq 10 nor eq 12 is optimal for the irregular nodes if in the assemblage process the contribution of an irregular node is one-half for each one of its neighbors (as is advocated in [8]). When there are irregular nodes in the one-dimensional case, nodal exactness is obtained if the following weighting functions are used for all those *elements* that share an irregular node

$$\bar{N}^j = (N^j + 2 N_{,\xi}^j \operatorname{sgn}(a)) \quad (j = 1, 2).$$

In Figures 2-a, 2-b, and 2-c we have the sketches of the form functions for the nodes  $i-1$ ,  $i$ , and  $i+1$  respectively, where the node  $i$  is irregular. In Figures 2-d and 2-e are represented the compound weighting functions of the nodes  $i-1$  and  $i+1$  after the normal assemblage process, that is, one-half of the  $i$ -th weighting function for each one of its neighbors.

With regard to these modified form functions, we can see that using their counterpart in the two-dimensional case (only for the irregular node and its two neighbors) a good improvement is obtained.

#### Multidimensional compressible Euler equations :

A Petrov-Galerkin formulation will be presented and will be shown how it reduces exactly to the already known formulations of both the scalar case and the case of systems, in the bidimensional and one-dimensional cases respectively.

Considering

$$U_{,i} = \frac{\partial \xi_j}{\partial x_i} U_{,k_j}$$

and

$$A_i U_{,i} = B_j U_{,k_j}$$

there follows that

$$B_j = A_i \frac{\partial \xi_j}{\partial x_i}$$

Let  $V^i$  and  $S^i$  denote the finite-dimensional subsets of  $H^1(\Omega)$  satisfying the following conditions

$$N_i \in V^i \Rightarrow N_i(x) \doteq 0 \quad \text{only when } x \in \{\Gamma_{inflow} \text{ with Mach} > 1\}$$

and

$$U_i(u_1, u_2, u_3, u_4) \in S^i \Rightarrow u_i(x) \doteq \bar{u}_i(x) \quad \forall x \in \Gamma_{u_i}$$

where  $N_i$  is the typical finite element weighting function,  $U_i$  the  $i$ -th component of the trial solutions in conservation variables, and the function  $\bar{u}_i(x)$ , the Dirichlet b.c. for the  $i$ -th component of the primitive variables.

We assume that both subsets consist of the typical  $C^0$  finite element interpolation functions.

The proposed Petrov-Galerkin formulation is the following (without the contour terms, see [10])

$$\int_{\Omega} \left( N_i^j + N_{,k_j}^j, B_j^i (B^T B)^{-1/2} \right) (A_i U_{,i}) d\Omega = 0 \quad \forall N_i^j \in V^i, \quad (13)$$

where

$$N^j = \text{diag}(N_1^j, N_2^j, N_3^j, N_4^j),$$

and

$$B = \begin{pmatrix} B_1^i \\ \vdots \\ B_{n,d}^i \end{pmatrix}$$

**Verifications:**

(1) One-dimensional *symmetric* advective systems.

The Euler equations of Gas Dynamics do not constitute a *symmetric* system when they are written in terms of *conservation variables*. For many physical systems of equations, however, a change of variables exists so that they can be written in symmetric form, see [3,4,5].

In this code, the *entropy variables* (as the variables resulting from the symmetrizing change of variables are referred to) were used only at the element sub-routine level, whereas primitive variables were used at global level.



From the condition of symmetry,

$$\mathbf{A} = \mathbf{A}' = \Phi \Lambda \Phi^{-1} = \Phi \Lambda \Phi',$$

then

$$\begin{aligned} \mathbf{B}'_j (\mathbf{B}_i \mathbf{B}'_i)^{-1/2} &= \mathbf{A}' (\mathbf{A} \mathbf{A}')^{-1/2} = \\ &= \mathbf{A} (\Phi \Lambda^2 \Phi')^{-1/2} = \mathbf{A} (\Phi |\Lambda|^{-1} \Phi') = \text{sgn } \mathbf{A}. \end{aligned}$$

Using

$$N^j_{,x} = \frac{h}{2} N^j_{,x},$$

we obtain the already known formulation, i.e.

$$\int_{\Omega} (N^j + N^j_{,x} \frac{h}{2} (\text{sgn } \mathbf{A})) (\mathbf{A} \mathbf{U}_{,x}) d\Omega = 0 \quad \forall N^j \in V^i.$$

(2) Bidimensional scalar case.

Using

$$(\mathbf{B}' \mathbf{B})^{-1/2} = (b_i b_i)^{-1/2} = |b|^{-1},$$

we obtain once more the formulation sought, i.e.

$$\int_{\Omega} (N^j + N^j_{,x} b_i |b|^{-1}) (a_i u_{,i}) d\Omega = 0 \quad \forall N^j \in V.$$

Note :

Considering a two-dimensional system that could be simultaneously diagonalized, we can see that each equation of the decoupled system is a two-dimensional scalar advective equation. Therefore, we can apply the latter verification to each component, and thus, we have verified once more the comprehensiveness of this formulation.

## Shock Capturing Concept

### Bidimensional scalar advective equation:

The SUPG formulation for the bidimensional scalar advective equation was written as follows

$$\int_{\Omega} (N^j + N^j_{,t} b_t |b|^{-1}) (a; u, i) d\Omega = 0 \quad \forall N^j \in V.$$

The one-dimensional case has the bidimensional characteristic  $\nabla_{\xi} u \parallel b$ , but in the bidimensional case we have  $b = b_{\parallel} + b_{\perp}$ , and only  $b_{\perp}^t \nabla_{\xi} u = 0$ . There follows that the foregoing formulation can be written in the following way

$$\int_{\Omega} [N^j b^t \nabla_{\xi} u + (\nabla_{\xi} N^j)^t b_{\perp} \frac{1}{|b|} b_{\parallel}^t \nabla_{\xi} u + (\nabla_{\xi} N^j)^t b_{\parallel} \frac{1}{|b|} b_{\parallel}^t \nabla_{\xi} u] d\Omega = 0,$$

the third term is not the optimal value because the one-dimensional analogy calls for the value

$$b_{\parallel} \frac{1}{|b_{\parallel}|} b_{\parallel}^t.$$

A way of adding only the necessary artificial diffusivity is to introduce the so called shock capturing term

$$b_{\parallel} b_{\parallel}^t \left( \frac{1}{|b_{\parallel}|} - \frac{1}{|b|} \right),$$

after the introduction of the above term, the final formulation is the following

$$\int_{\Omega} \left\{ [N^j + (\nabla_{\xi} N^j)^t b \frac{1}{|b|}] (b^t \nabla_{\xi} u) + (\nabla_{\xi} N^j)^t b_{\parallel} \left( \frac{1}{|b_{\parallel}|} - \frac{1}{|b|} \right) (b^t \nabla_{\xi} u) \right\} d\Omega = 0$$

$$\forall N^j \in V.$$

A comprehensive description of the shock capturing concept for the linear scalar advection-diffusion equation and the multidimensional advective-diffusive systems can be found in [9] and [11] respectively.

### Multidimensional first-order systems of hyperbolic equations:

As was made for the scalar case, the Jacobian matrices are split in the following way

$$\mathbf{A}_i = \mathbf{A}_{i\parallel} + \mathbf{A}_{i\perp}, \quad (i = 1, \dots, nsd.),$$

so that

$$\mathbf{A}_i \mathbf{U}_{,i} = \mathbf{A}_{i\parallel} \mathbf{U}_{,i}$$

and

$$\mathbf{A}_{i\perp} \hat{\mathbf{U}}_i = 0 \quad \forall \hat{\mathbf{U}} / \hat{\mathbf{U}}' \nabla \mathbf{U} = 0,$$

there follows that  $\mathbf{A}_{\parallel}$  is an operator of rank 1 that acts only in the direction of the gradient.

We can define the operator  $\mathbf{A}_{\parallel}$  as follows

$$\mathbf{A}_{i\parallel} = (\mathbf{A}_j \mathbf{U}_{,j}) \frac{\mathbf{U}_{,i}}{|\nabla \mathbf{U}|^2},$$

and following the development in natural coordinates, we define

$$\mathbf{B}_{i\parallel} = \frac{\partial \xi_i}{\partial x_j} \mathbf{A}_{j\parallel},$$

from which, the shock capturing part of the formulation is the following

$$\int_{\Omega} \mathbf{N}_{,k} \mathbf{B}_{i\parallel}^t (\mathbf{B}_{\parallel}^t \mathbf{B}_{\parallel})^{-1/2} \mathbf{B}_{i\parallel} \mathbf{U}_{,k} d\Omega,$$

where

$$\mathbf{B}_{\parallel} = \begin{pmatrix} \mathbf{B}_{1\parallel}^t \\ \vdots \\ \mathbf{B}_{nsd\parallel}^t \end{pmatrix}$$

Because  $\mathbf{B}_{\parallel}^t \mathbf{B}_{\parallel}$  has rank 1, the negative square root of it is defined in its non-degenerate subspace, namely

$$(\mathbf{B}_{\parallel}^t \mathbf{B}_{\parallel})^{-1/2} = |\lambda|^{-1} \phi \phi^t,$$

where  $\lambda$  and  $\phi$  are the result of the following eigenproblem

$$(\mathbf{B}_{\parallel}^t \mathbf{B}_{\parallel} - \lambda^2 \mathbf{I}) \phi = 0.$$

Substituting

$$\mathbf{B}_{i\parallel} = \frac{\partial \xi_i}{\partial x_j} \mathbf{A}_{j\parallel} = (\mathbf{A}_k \mathbf{U}_{,k}) \frac{\mathbf{U}_{,j}}{|\nabla \mathbf{U}|^2} \frac{\partial \xi_i}{\partial x_j} = (\mathbf{A}_k \mathbf{U}_{,k}) \tilde{\mathbf{U}}_j^t$$

in the eigenproblem, we get

$$(\mathbf{A} \cdot \nabla \mathbf{U} |\tilde{\mathbf{U}}|^2 (\mathbf{A} \cdot \nabla \mathbf{U})^t - \lambda^2 \mathbf{I}) \phi = 0,$$

the solution to this eigenproblem is

$$\phi = \mathbf{A} \cdot \nabla \mathbf{U} \quad \text{and} \quad \lambda^2 = |\tilde{\mathbf{U}}|^2 |\mathbf{A} \cdot \nabla \mathbf{U}|^2,$$

there follows that

$$(\mathbf{B}_{\parallel}^t \mathbf{B}_{\parallel})^{-1/2} = (\mathbf{A} \cdot \nabla \mathbf{U}) \frac{1}{|\tilde{\mathbf{U}}| |\mathbf{A} \cdot \nabla \mathbf{U}|} (\mathbf{A} \cdot \nabla \mathbf{U})^t,$$

and now, the shock capturing part of the formulation is

$$\int_{\Omega} N_{,i} B_{j\parallel}^t \left( \frac{|\mathbf{A} \cdot \nabla \mathbf{U}|}{|\tilde{\mathbf{U}}|} \right) \mathbf{A} \cdot \nabla \mathbf{U} \, d\Omega.$$

Again, as was made for the two-dimensional scalar case, we must subtract from the above expression a quantity equal to the contribution of the plain SUPG in the direction of  $\nabla \mathbf{U}$ .

From eq 13, the contribution of the plain SUPG is

$$\begin{aligned} \mathbf{B}_{\parallel} (\mathbf{B} \mathbf{B}^t)^{-1/2} \mathbf{A} \cdot \nabla \mathbf{U} &= \tilde{\mathbf{U}} (\mathbf{A} \cdot \nabla \mathbf{U})^t (\mathbf{B} \mathbf{B}^t)^{-1/2} (\mathbf{A} \cdot \nabla \mathbf{U}) = \\ &= \tilde{\mathbf{U}} (\mathbf{A} \cdot \nabla \mathbf{U})^t \alpha (\mathbf{A} \cdot \nabla \mathbf{U}) = \mathbf{B}_{\parallel} \alpha (\mathbf{A} \cdot \nabla \mathbf{U}), \end{aligned}$$

where

$$\alpha = \frac{(\mathbf{A} \cdot \nabla \mathbf{U})^t (\mathbf{B} \mathbf{B}^t)^{-1/2} (\mathbf{A} \cdot \nabla \mathbf{U})}{|\mathbf{A} \cdot \nabla \mathbf{U}|^2},$$

therefore, the final expression for the shock capturing operator is the following

$$\int_{\Omega} N_{,i} B_{j\parallel}^t \left( \frac{|\mathbf{A} \cdot \nabla \mathbf{U}|}{|\tilde{\mathbf{U}}|} - \frac{(\mathbf{A} \cdot \nabla \mathbf{U})^t (\mathbf{B} \mathbf{B}^t)^{-1/2} (\mathbf{A} \cdot \nabla \mathbf{U})}{|\mathbf{A} \cdot \nabla \mathbf{U}|^2} \right) \mathbf{A} \cdot \nabla \mathbf{U} \, d\Omega.$$

## SECTION 2

In this section we make a description of some important aspects concerning the implementation and use of the method we are dealing with.

Two-dimensional weighted residual formulation for the compressible Euler equations :

The Euler equations can be written in conservation form as follows

$$U_{,j} + F_{j,j} = 0 \quad \text{on } \Omega \subset \mathbb{R}^2,$$

where

$$F_j = \begin{pmatrix} u_j \rho \\ u_j \rho u_1 + \delta_{1j} p \\ u_j \rho u_2 + \delta_{2j} p \\ u_j (\rho e + p) \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho e \end{pmatrix},$$

$$e = \epsilon + \frac{u_i u_i}{2},$$

here,  $e$  is the total energy and  $\epsilon$  is the internal energy per unit mass.

In order to complete the system we must specify an equation of state  $p = p(\rho, \epsilon)$ . Any equation of this kind may be used, but the equation of a perfect gas is currently used for transonic and supersonic calculations, i.e.

$$p = (\gamma - 1) \rho \epsilon,$$

where  $\gamma$  is the ratio of specific heats.

The flux vector  $F_j(U)$  is a homogeneous function of degree one in the conservative variables  $U$ , it follows that (see [12])

$$F_j(U) = A_j U$$

and

$$F_{j,j} = A_j U_{,j}.$$

Let  $n = (n_1, n_2)$  be the outward unit normal vector on the boundaries and let  $F_j$  be split in the following way

$$F_j = F_j^{(1)} + F_j^{(2)} = \begin{pmatrix} 0 \\ \delta_{1j} p \\ \delta_{2j} p \\ u_j p \end{pmatrix} + \begin{pmatrix} u_j \rho \\ u_j \rho u_1 \\ u_j \rho u_2 \\ u_j \rho e \end{pmatrix}.$$

Later we will make reference to

$$F_n^{(2)} = F_j^{(2)} n_j = \begin{pmatrix} u_n \rho \\ u_n \rho u_1 \\ u_n \rho u_2 \\ u_n \rho e \end{pmatrix}.$$

Consider a discretization of  $\Omega$  into element subdomains  $\Omega^e$ ,  $e = 1, \dots, n_{el}$ , where  $n_{el}$  is the number of elements. We assume

$$\bar{\Omega} = \bigcup_{e=1}^{nel} \bar{\Omega}^e, \quad \emptyset = \bigcap_{e=1}^{nel} \Omega^e.$$

Also let  $\Gamma^e$  be the whole boundary of element  $e$ ,  $\Gamma$  the boundary of  $\Omega$ , and  $\Gamma_{int}$  the following set

$$\Gamma_{int} = \left( \bigcup_{e=1}^{nel} \Gamma^e \right) - \Gamma.$$

The boundary  $\Gamma$  of the domain  $\Omega$  is assumed to be decomposed as follows

$$\begin{aligned} \Gamma &= \overline{\Gamma_{u_i} \cup \Gamma_{f_i} \cup \Gamma_{slip}}, & \emptyset &= \Gamma_{u_i} \cap \Gamma_{f_i}, \\ \emptyset &= (\Gamma_{u_i} \cup \Gamma_{f_i}) \cap \Gamma_{slip}, & (i &= 1, 2, 3, 4.), \end{aligned}$$

where  $\Gamma_{u_i}$  refers to that part of the boundary on which a Dirichlet-type b.c. is specified for the  $i$ -th component of the primitive variables (i.e.,  $\rho, u_1, u_2, p$ ),  $\Gamma_{f_i}$  to that part on which no b.c. is specified for the  $i$ -th component, and  $\Gamma_{slip}$  to that part on which the natural b.c.  $F_n^{(2)} = 0$  is specified.

The only natural b.c. is  $F_n^{(2)} = 0$  on  $\Gamma_{slip}$ , because on the inflow/outflow part of the boundary only Dirichlet b.c. are specified for a number of primitive variables (i.e.,  $\rho, u_1, u_2, p$ ) according to the nature of the boundary (inflow/outflow) and to the Mach number. This point will be explained later in this section.

Let  $V^i$  and  $S^i$  denote the finite-dimensional subsets of  $H^1(\Omega)$  satisfying the following conditions

$$N_i \in V^i \Rightarrow N_i(x) \doteq 0 \quad \text{only when } x \in \{\Gamma_{inflow} \text{ with Mach} > 1\}$$

and

$$U_i(u_1, u_2, u_3, u_4) \in S^i \Rightarrow u_i(x) \doteq \tilde{u}_i(x) \quad \forall x \in \Gamma_{u_i},$$

where  $N_i$  is the typical finite element weighting function,  $U_i$  the  $i$ -th component of the trial solutions in conservation variables, and the function  $\tilde{u}_i(x)$ , the Dirichlet b.c. for the  $i$ -th component of the primitive variables.

We assume that both subsets consist of the typical  $C^0$  finite element interpolation functions, and that the so-called *group approximation* of the flux vector  $F_i$  is employed so that its components are also piecewise bilinear functions (for bilinear form functions) determined by their values at element nodes. This finite approximation leads to

$$U = \sum_{j=1}^{numnp} N^j U^j, \quad F_i = \sum_{j=1}^{numnp} N^j F_i^j,$$

where  $numnp$  denotes the total number of nodes in the discretization,  $N^j = \text{diag}(N_1^j, N_2^j, N_3^j, N_4^j)$  are the global piecewise bilinear basis functions, and  $U^j, F_i^j$  are the values of  $U, F_i$  at node  $j$ .

In Section 1 it was demonstrated how the SUPG formulation is cast in a weighted residual form, in which the weighting functions are modified by the addition of  $C^{-1}$  perturbations, and now, we make use of that by writing the variational equation for the compressible Euler equations in the Euler-Lagrange form

$$0 = \sum_i \int_{\Omega^e} (N^j + \tilde{P}^j) (U_{,t} + F_{i,i}) d\Omega - \int_{\Gamma_{slip}} N^j F_n^{(2)} d\Gamma - \int_{\Gamma_{int}} N^j [F_n] d\Gamma$$

$$\forall N_i^j \in V^i,$$

in which  $N^j = \text{diag}(N_1^j, N_2^j, N_3^j, N_4^j)$ , and the Euler-Lagrange equations are the following:

$$\begin{aligned} U_{,t} + F_{j,j} &= 0 \quad \text{on } \Omega && \text{governing equation,} \\ F_n^{(2)} &= 0 \quad \text{on } \Gamma_{slip} && \text{null flux condition,} \\ [F_n] &= 0 \quad \text{on } \Gamma_{int} && \text{continuity condition.} \end{aligned}$$

In the latest equation, the square brackets represents the jump of  $F_n$  across the interelement boundary. In fact, this equation is automatically verified because  $F_n$  has  $C^0$  continuity. Integrating by parts, we obtain the weak form of the weighted residual equation

$$0 = \sum_i \int_{\Omega^e} \tilde{P}^j (U_{,t} + F_{i,i}) d\Omega + \sum_i \int_{\Omega^e} (N^j U_{,t} - N_{,i}^j F_i) d\Omega + \sum_i \int_{\Gamma^e} N^j F_n d\Gamma - \int_{\Gamma_{slip}} N^j F_n^{(2)} d\Gamma - \int_{\Gamma_{int}} N^j [F_n] d\Gamma$$

$$\forall N_i^j \in V^i.$$

Using the following splitting

$$\sum_i \int_{\Gamma^e} N^j F_n d\Gamma = \int_{\Gamma_{int}} N^j [F_n] d\Gamma + \int_{\Gamma_{in/outflow}} N^j F_n d\Gamma + \int_{\Gamma_{slip}} N^j F_n d\Gamma,$$

we can write

$$0 = \sum_i \int_{\Omega^e} (N^j + \tilde{P}^j) U_{,t} d\Omega + \sum_i \int_{\Omega^e} (\tilde{P}^j F_{i,i} - N_{,i}^j F_i) d\Omega + \int_{\Gamma_{slip}} N^j F_n^{(1)} d\Gamma + \int_{\Gamma_{in/outflow}} N^j F_n d\Gamma \quad \forall N_i^j \in V^i.$$

Making use of the forward Euler scheme in the time discretization, we can write the complete formulation in matrix form as follows

$$0 = M \Delta b - (\Delta t) R,$$

where  $M$  is the consistent mass matrix,  $R$  the residue, and  $\Delta b$  the vector of nodal variations of the conservation variables. The use of the consistent mass matrix is not the one consistent with the developments of Section 1, and on the other hand, it is more CPU-time consuming, therefore, a lumped mass matrix was used instead.

Any variation in the conservation variables ( $\Delta \bar{b}$ ) is related to the variation of the primitive variables ( $\Delta \bar{a}$ ) by a very known triangular matrix, i.e.

$$\Delta \bar{a} = D^{-1} \Delta \bar{b}.$$

Now considering the nodal vector of primitive variables ( $a$ ), it is updated after each iteration as follows

$$a_{n+1}^j = a_n^j + \bar{D}_j^{-1} \Delta b^j \quad (j = 1, \dots, \text{numnp.}),$$

where  $\bar{D}_j^{-1}$  is the triangular matrix obtained from  $D_j^{-1}$  by filling with zeros the  $i$ -th row if the  $i$ -th component of the primitive variables was specified as a Dirichlet b.c. for the  $j$ -th node.

#### Stability analysis :

The SUPG formulation for the scalar case and for a rectangular mesh with element sides of uniform length  $h$  gives the following result for the  $j$ -th generic node

$$u_{n+1}^j = u_n^j - \alpha \frac{\Delta t}{h^2} \left\{ \int_{\Omega} \left( N^j + N_{,s}^j \frac{h}{2} \frac{a_i}{|a|} \right) (a_i u_{,t}) d\Omega \right\},$$

where  $\alpha$  is equal to 1 for the interior nodes and to 2 for the boundary nodes.

Using  $v_i = a_i / |a|$  and  $C = \Delta t |a| / h$ , we can rewrite the above equation as follows

$$u_{n+1}^j = u_n^j - \alpha C \left\{ \int_{\Omega} \left( \frac{1}{h} N^j + N_{,s}^j \frac{1}{2} v_i \right) (v_i u_{,t}) d\Omega \right\}.$$

Replacing in the above equation the following field

$$u_n^j = e^{i(k_x j_x h + k_y j_y h)} = e^{i(\alpha_x h + \alpha_y h)},$$

where  $i$  is the imaginary unit and  $k_x, k_y$  are the wave numbers in the  $x, y$  directions respectively, we can obtain two equations, one for the interior nodes and the other for the boundary nodes, of the following general form

$$u_{n+1}^j = G(C, \alpha_x, \alpha_y, v_x, v_y) u_n^j,$$



in which the function  $G$  is the amplification factor.

The amplification factor  $G$  will be  $\leq 1$  if  $C \leq .90$  for the internal nodes and  $C \leq .50$  for the boundary nodes. When only plane waves are introduced, which are compatible with one-dimensional problems, the limit of stability for the internal nodes is reached with  $C = 1.0$ , but the corresponding to boundary nodes remains the same.

The unidimensional Euler equations can be analyzed as a system of decoupled scalar equations integrated with the same  $\Delta t$ , therefore, the previous stability analysis leads to the following stability conditions:

$$\Delta t \leq \frac{h}{\bar{c} + |a|} \quad \text{for internal nodes (CFLN} = 1.0)$$

and

$$\Delta t \leq \frac{h}{2(\bar{c} + |a|)} \quad \text{for boundary nodes (CFLN} = 0.5),$$

where  $\bar{c}$  is the local sound speed,  $|a|$  the absolute value of the velocity, and  $\bar{c} + |a|$  the greatest eigenvalue of the system.

It is interesting to note that in subsonic conditions, the eigenvalues have different signs, therefore, although an instability were generated only at the boundary, it propagates inwards rapidly.

The two-dimensional Euler equations can not be analyzed as a system of decoupled scalar equations, this is thus because there is not a similarity transformation that diagonalizes both Jacobians simultaneously. Nevertheless, we had not any instability running the program with  $CFLN \leq .90$ .

Several points should be considered with regard to this formulation :

1. Since the objective is simply to obtain a steady state as soon as possible, the order of accuracy used to evaluate the transient state is not important at all. This allows the use of schemes selected mainly for their properties of stability and damping. In this regard we used the forward Euler integration scheme, which stems from a Taylor's expansion of the vector of conservation variables as was seen in Section 1. For using another scheme, an analysis of stability is necessary.

2. It appears from the formulation that the natural b.c. of null flux on slip boundaries would have to be verified, in the weighted form, in the same way that the flow of heat is where null flow is specified as a natural b.c. of a heat transfer analysis. However, this proved to be a most unstable b.c., not being verified at all and spoiling the solution. A large number of schemes for the analysis of inviscid compressible flows appear to have the same shortcoming (see [13], page 335).

The code avoids this shortcoming evaluating automatically, for each node of the declared slip boundaries, a unit vector  $\tilde{n}^j$ , that take into account the orientations and lengths of the elements' sides that converge to the  $j$ -th node and that are part of the slip boundary, i.e.

$$\tilde{n}_i^j = \int_{\Gamma_{s,i,j}} N^j n_i d\Gamma \left( \sum_{l=1}^2 \left( \int_{\Gamma_{s,i,j}} N^l n_l d\Gamma \right)^2 \right)^{-1/2}$$

Then, after each iteration, the velocities are modified as follows

$$V^j = V_{iter}^j - (\bar{u}^j \cdot V_{iter}^j) \bar{u}^j,$$

where  $V_{iter}^j$  is the velocity in the  $j$ -th node obtained from  $a_{n+1}^j$  and  $V^j$  is the new value of this velocity to be assigned to  $a_{n+1}^j$ .

3. If we consider that the rate of convergence is given by the CFLN and that the meshes have in general highly variable element sizes, it is understood why the convergence is speeded up when the optimum time step is used for each node. This code automatically uses a nodal time step that is in accordance with a specified CFLN, we usually specify CFLN = .9. This CFLN is reduced for those nodes that are on the boundary because of stability, the reduction is indirectly accomplished by using an augmented lumped mass matrix.

4. Because the steady state is our target, we can use a sequence of meshes. The coarser a mesh, the cheaper is to obtain an approximated solution. Therefore, we begin with a coarse mesh and when the rate of convergence decays an automatic switch is made to a finer mesh.

With regard to the automatic refinements you can choose between an overall refinement or a localized (adaptive) refinement. As a matter of fact, the first ones are ever overall refinements, while the adaptive ones are used in the final stages of refinement.

5. Any type of upwind introduces artificial diffusivity. The diffusivity acts in the zones of high gradients, no matter which is the origin of such gradients, as a result, there may be zones in which spurious generation of entropy occurs (e.g., the stagnation zone generated by a blunt body). The straightforward procedure for avoiding such errors is to use adaptive refinement in those zones.

#### Boundary Conditions :

The number of primitive variables to be specified on the inflow/outflow part of the boundary depends upon the local Mach number.

The b.c. for the inflow/outflow part is introduced in our formulation through the integral term

$$\int_{\Gamma_{in/outflow}} N F_n d\Gamma,$$

where

$$F_n = F_i n_i = (A_i n_i) U = A_n U$$

here,  $A_n = \partial F_n / \partial U$  is the Jacobian matrix (see [12]).

The matrix  $A_n$  has a complete set of real eigenvalues for any flow condition. Therefore,  $A_n$  can be written as follows

$$A_n = \Phi \Lambda \Phi^{-1},$$

where  $\Lambda$  is a diagonal matrix with entries

$$\begin{aligned}\lambda_1 &= \lambda_2 = \lambda_3 = u_i n_i, \\ \lambda_4 &= \lambda_1 + a, \\ \lambda_5 &= \lambda_1 - a,\end{aligned}$$

here,  $a$  stands for the local sound speed.

Considering

$$\Lambda^\pm = \frac{1}{2}(\Lambda \pm |\Lambda|),$$

$$A_n^\pm = \Phi \Lambda^\pm \Phi^{-1},$$

$$A_n = A_n^+ + A_n^-,$$

$$F_n = A_n^+ U + A_n^- U,$$

here, the Jacobian  $A_n^+$  ( $A_n^-$ ) has only positive (negative) eigenvalues which represent the speeds of those signals propagating outside (inside) the control volume.

With regard to the appropriate Dirichlet b.c., we can see that those variables which represent the far-field conditions propagating inside the control volume must be specified whereas the remaining are left free.

Running the tests, we specified the essential boundary conditions as is shown in the following table

$M_\infty$ Number	Inflow	Outflow
< 1	$u_1, u_2, \rho$	$p$
> 1	$u_1, u_2, \rho, p$	-

### SECTION 3

#### Evaluation of an Oblique Shock Wave

To begin with we present a problem of which we know the analytical solution, it consists in the evaluation of an oblique shock wave originated when a flow incides over a wedge (see Figure 3). This problem has already been used to test several schemes [6].

As a result of the obliqueness of the shock with the mesh, this test enables us to check the capability of this scheme to evaluate this kind of shocks.

The mesh consists of  $20 \times 20$  elements homogeneously distributed over the domain (a unit square).

At the inflow [A-B-C] all variables were specified ( $M > 1$ ). The wall [D-A] was specified as slip boundary so that the code could rectify the velocities of all those nodes lying on that boundary. For this domain we could have imposed the null flux on [D-A] by restraining the corresponding d.o.f ( $u_2 = 0$ ), but for general curved surfaces this solution is not practical, and you necessarily have to rely on the declaration of slip boundary.

No variable was fixed neither on the outflow (C-D) nor on the slip boundary [D-A]. The boundary condition to be imposed in the node A is not unique, but anyway, the values of the state variables after the shock and the angle of the shock itself will not be affected.

$$\text{inflow } (M = 2) \begin{cases} \rho & = 1. \\ u_1 & = 0.98481 \\ u_2 & = -0.17365 \\ p & = 0.178596 \end{cases}$$

The result is the following

$$\text{outflow } (M = 1.64) \begin{cases} \rho & = 1.458 \\ u_1 & = 0.887 \\ u_2 & = 0.000 \\ p & = 0.304 \end{cases}$$

Figure 4 shows this result in the form of density elevation (in Figure 3 is indicated the observation point). We can see that a sharp shock without oscillations was obtained. With regard to the numerical values, we can say that there was complete agreement between the numerical and the analytical values.

#### Calculations for the NACA0012 airfoil

Two test cases were chosen, both are lifting flows. The first of them is the ubiquitous case  $M_\infty = 0.80$  with an *angle of attack* = 1.25 deg, and the second test has  $M_\infty = 1.20$  and an *angle of attack* = 7.00 deg.

These are two of the cases considered in the AGARD Fluid Dynamics Panel Working Group 07 (see [7]).

## ACKNOWLEDGEMENTS

The authors wish to express their gratitude to CONICET for its financial support. They also wish to thank Nestor Aguilera and Norberto Nigro for helpful comments, and to Karl for his outstanding job in typing the manuscript.

## REFERENCES

- [1] J. Donéa, 'A Taylor-Galerkin method for convective transport problems, *Internat. J. Numer. Meths. Engrg.* 20 (1) (1984) 199-259.
- [2] T.J.R. Hughes, L.P. Franca and G.M. Hulbert, 'A new finite element method for computational fluid dynamics: VIII. The Galerkin/Least-Squares method for Advective-Diffusive equations', *Compt. Meths. Appl. Mech. Engrg.* 73 (1989) 173-189.
- [3] A. Harten, 'On the symmetric form of systems of conservation laws with entropy', *J. Comput. Phys.* 49 (1983) 151-164.
- [4] E. Tadmor, 'Skew-selfadjoint forms for systems of conservation law', *J. Math. Anal. Appl.* 103 (1984) 428-442.
- [5] T.J.R. Hughes, L.P. Franca and M. Mallet, 'A new finite element method for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics', *Compt. Meths. Appl. Mech. Engrg.* 54 (1986) 223-234.
- [6] S.F. Davis, 'A Rotationally Biased Upwind Difference Scheme for the Euler Equations', *J. Comput. Phys.* 56 (1984) 65-92.
- [7] T.H. Pulliam and J.T. Barton, 'Euler Computations of AGARD Working Group 07 Airfoil Test Cases', *AIAA Paper 85-0018* (1985).
- [8] P. Devloo, J.T. Oden and T. Strouboulis, 'Implementation of an Adaptive Refinement Technique for the SUPG Algorithm', *Compt. Meths. Appl. Mech. Engrg.* 61 (1987) 339-358.
- [9] T.J.R. Hughes, M. Mallet and M. Mizukami, 'A new finite element method for computational fluid dynamics: II. Beyond SUPG', *Compt. Meths. Appl. Mech. Engrg.* 54 (1986) 341-355.
- [10] T.J.R. Hughes and M. Mallet, 'A new finite element method for computational fluid dynamics: III. The generalized streamline operator for multidimensional advection-diffusion systems', *Compt. Meths. Appl. Mech. Engrg.* 58 (1986) 305-328.

## Meshes

Figure 5 shows the final mesh for the first test and Figure 9 that one of the second test. Each one was obtained from an initial coarse C-mesh that was automatically refined.

In the coarser C-meshes, the convergence was very fast and the evaluation of the residue very little time consuming. As the C-meshes became finer, the convergence became slower and the evaluation of the residue more time consuming.

Only the final stages of refinement were of adaptive type.

## Initial and Boundary Conditions

Each case was initialized with a uniform freestream flow at the prescribed Mach number and angle of attack.

For the first case the initial conditions are  $\rho = 1.$ ,  $u_1 = \cos(1.25)$ ,  $w_2 = \sin(1.25)$ , and  $p = 1.11607$  everywhere.

The boundary conditions are the following: (1) Slipping boundary condition on the airfoil. (2) Imposition of  $\rho$ ,  $u_1$ , and  $u_2$  on the inflow part of the domain. (3) Imposition of  $p$  on the outflow part.

For the second case the initial conditions are  $\rho = 1.$ ,  $u_1 = \cos(7.0)$ ,  $w_2 = \sin(7.0)$ , and  $p = 0.49603$  everywhere.

The boundary conditions are the following: (1) Slipping boundary condition on the airfoil. (2) Imposition of  $\rho$ ,  $u_1$ ,  $u_2$ , and  $p$  on the inflow part of the domain.

## Numerical Results

Figures 6-8 show pressure contours, Mach contours and  $C_p$  distribution for the first case, whereas for the second case, the pressure and Mach contours are shown in Figures 10 and 11 respectively.

The numerical results are in good agreement with those reported in [7]. Only in the first test case there is a slight difference in the positions of the shock waves, the positions given by this code are slightly downstream when compared with those given in [7].

## CONCLUSIONS

The numerical solution for the wedge problem and the airfoil calculations show that this SUPG version gives very good shock resolution and values of the state variables for steady-state computations.

With regard to the CPU time, we acknowledge that this finite element code is much slower than a finite difference code provided that a structured grid could be fitted. When a structured grid can not be fitted, this code still can solve the problem, no matter how complicated the domain or the boundary conditions may be.

- [11] T.J.R. Hughes and M. Mallet, 'A new finite element method for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems', *Compt. Meths. Appl. Mech. Engrg.* 54 (1986) 223-234.
- [12] T.J.R. Hughes and T.E. Tezduyar, 'Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations', *Compt. Meths. Appl. Mech. Engrg.* 45 (1984) 217-284.
- [13] J.T. ODEN, T. STROUBOULIS AND P. DEVLOO, 'Adaptive finite element methods for the analysis of inviscid compressible flow: Part I. Fast refinement/unrefinement and moving mesh methods for unstructured meshes', *Compt. Meths. Appl. Mech. Engrg.* 56 (1986) 327-362.

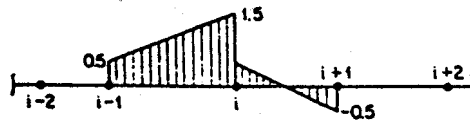


Figure 1

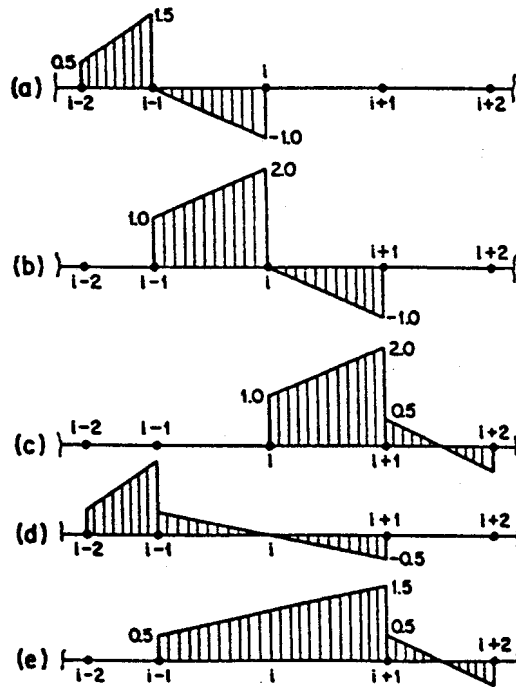


Figure 2



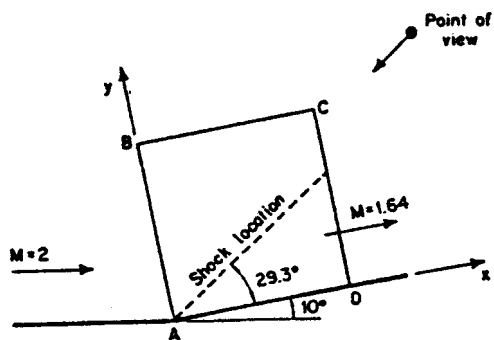


Figure 3

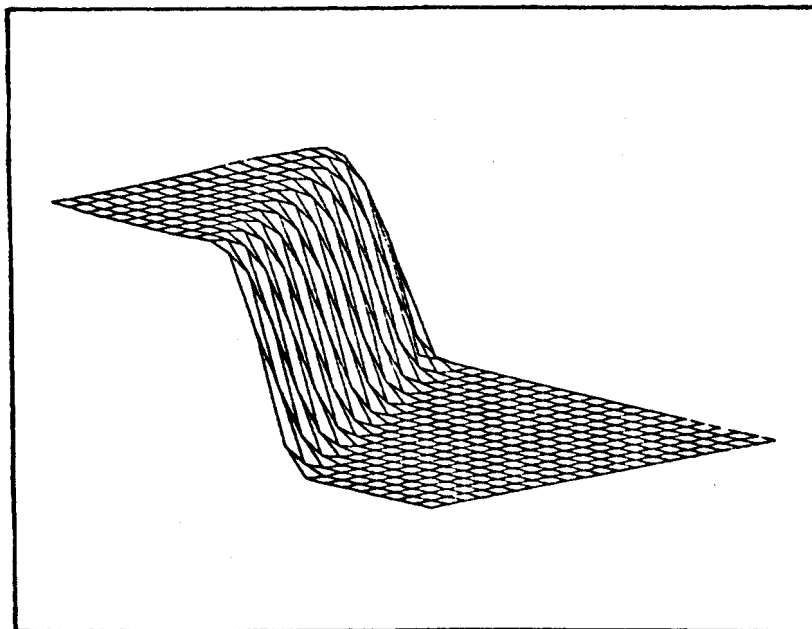


Figure 4

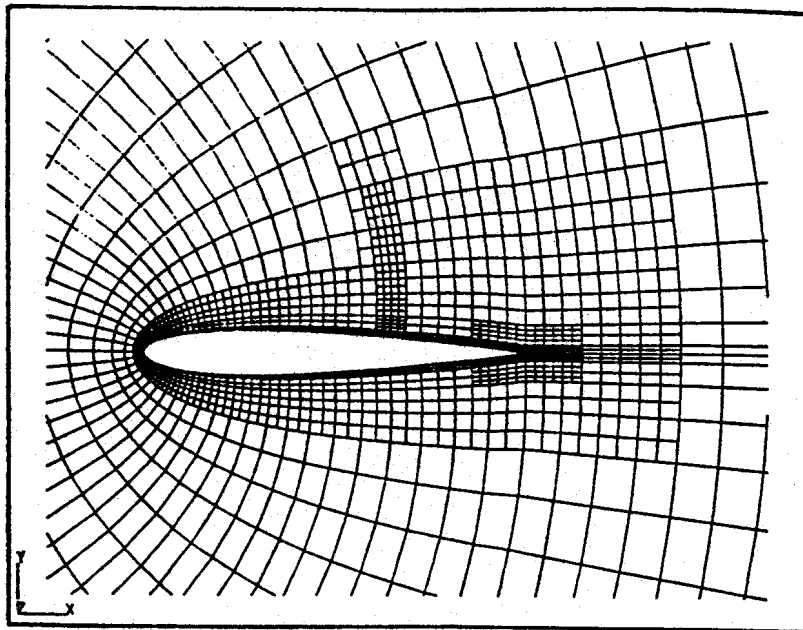


Figure 5

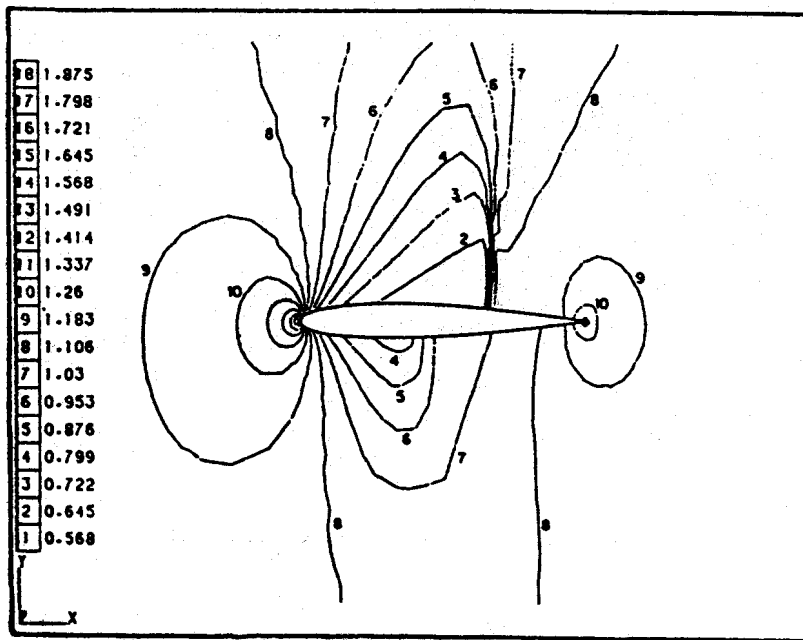


Figure 6

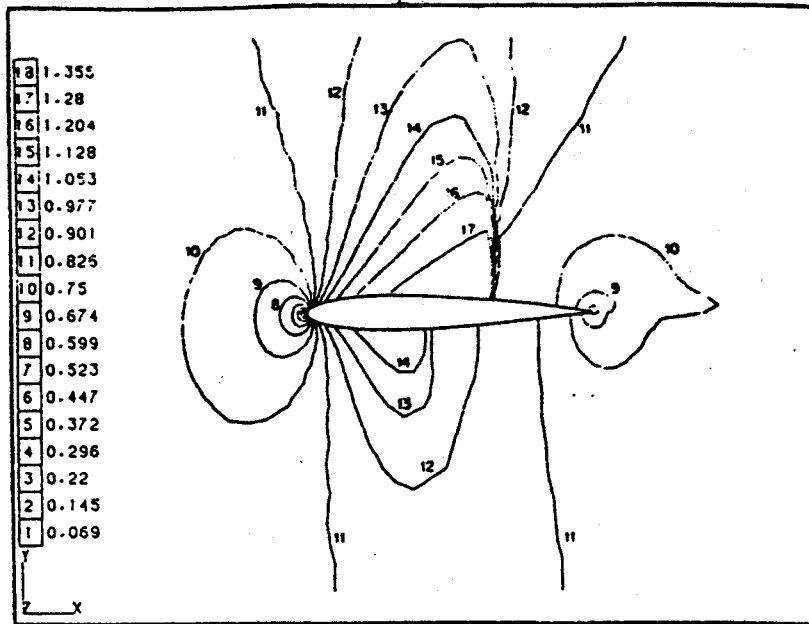


Figure 7

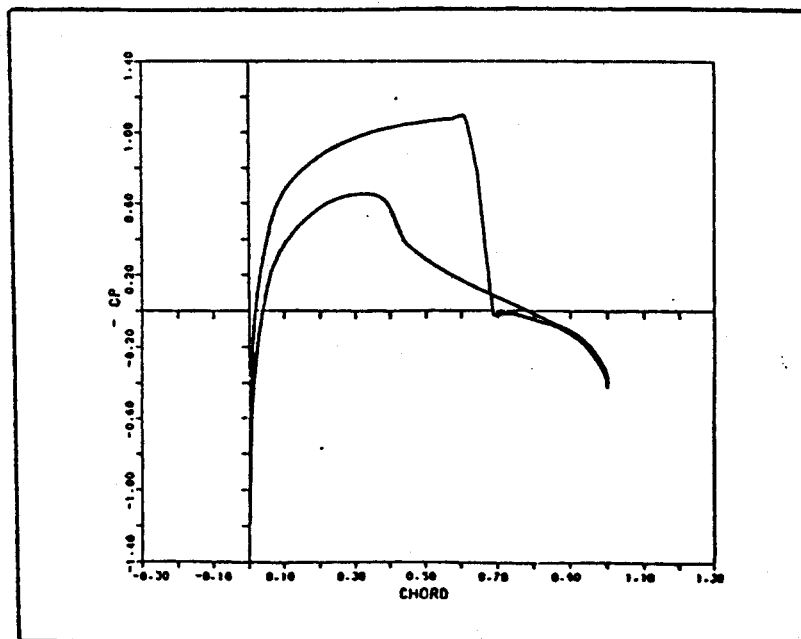


Figure 8

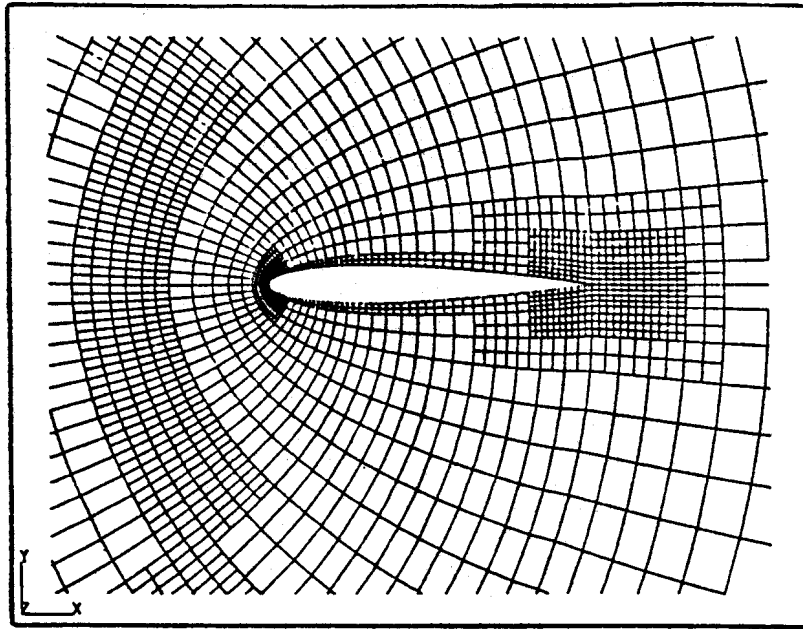


Figure 9

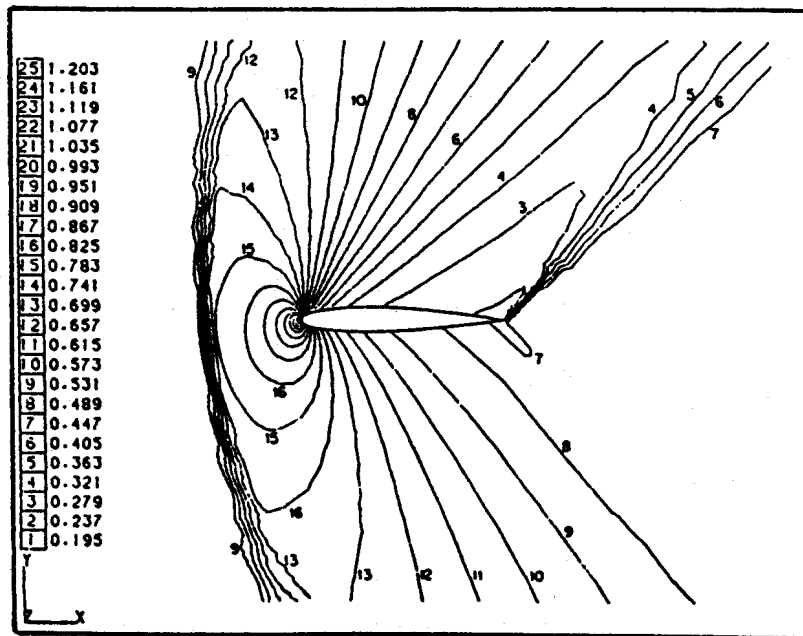


Figure 10

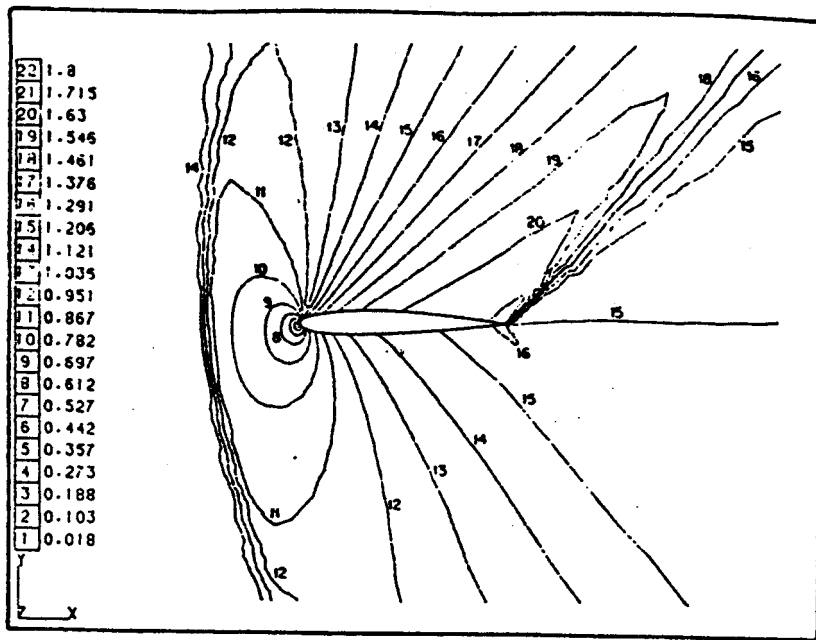


Figure 11

