

A Petrov–Galerkin technique for the solution of transonic and supersonic flows

C.E. Baumann, M.A. Storti and S.R. Idelsohn

*Computational Mechanics Laboratory of INTEC (Universidad Nacional del Litoral and CONICET),
Güemes 3450, 3000 Santa Fe, Argentina*

Received April 1990

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. The theoretical background of the SUPG formulation is described briefly. How the foregoing formulation was used in the finite element code and which are the appropriate boundary conditions to be used are also described. Finally some results obtained with this code are discussed.

1. Streamline-upwind Petrov–Galerkin formulations

We begin by 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.

1.1. 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

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

and the substitution of (1) in the above equation leads to

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \mathbf{A}\mathbf{U}_{,x}^n \Delta t + \mathbf{A}^2\mathbf{U}_{,xx}^n \frac{\Delta t^2}{2!} + o(\Delta t^2), \quad (2)$$

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 in the series (see [1] for further details).

The Jacobian matrix \mathbf{A} can be diagonalized (see [2]), therefore we can write

$$\mathbf{A} = \mathbf{S}\mathbf{A}\mathbf{S}^{-1},$$

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

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

which is a system of uncoupled scalar equations.

If (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 (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 obtain

$$\mathbf{V}_j^{n+1} = \mathbf{V}_j^n - \Delta t \frac{\mathbf{A}}{2h} (\mathbf{V}_{j+1}^n - \mathbf{V}_{j-1}^n) + \frac{\Delta t^2}{2} \frac{\mathbf{A}^2}{h^2} (\mathbf{V}_{j+1}^n - 2\mathbf{V}_j^n + \mathbf{V}_{j-1}^n). \quad (4)$$

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)

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

and that of the discretized case is

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

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

Substitution of (5) in (4) and the use of $\mathbf{C} = (\Delta t/h)\mathbf{A}$ gives

$$\begin{aligned} \mathbf{V}_j^{n+1} &= \left[1 - \frac{\mathbf{C}}{2} (e^{ik_l h} - e^{-ik_l h}) + \frac{\mathbf{C}^2}{2} (e^{ik_l h} - 2 + e^{-ik_l h}) \right] \mathbf{H}^n(k_l) e^{ik_l j h}, \\ \mathbf{V}_j^{n+1} &= \mathbf{G}(\mathbf{C})\mathbf{V}_j^n, \end{aligned}$$

where \mathbf{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*

$$\|\mathbf{G}(\mathbf{C})\| \leq 1$$

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

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

and the *norm condition* is satisfied if

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

in which μ_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, if $C_i = 1$ exact nodal values are obtained, and if $C_i < 1$ spurious oscillations develops, as will shortly become clear. A foregone conclusion is that the only stable way of treating the system of (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

$$\mathbf{V}_j^{n+1} = \mathbf{V}_j^n - \frac{C}{2} [(\mathbf{V}_{j+1}^n - \mathbf{V}_{j-1}^n) - C(\mathbf{V}_{j+1}^n - 2\mathbf{V}_j^n + \mathbf{V}_{j-1}^n)] \quad (6)$$

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

$$\mathbf{V}_{j+1}^n - \mathbf{V}_{j-1}^n = C(\mathbf{V}_{j+1}^n - 2\mathbf{V}_j^n + \mathbf{V}_{j-1}^n);$$

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 $\text{sgn}(\lambda_i)$ in every componential within the square brackets of (6). Introducing the modification in the original equation, we obtain the new difference equation

$$\mathbf{V}_j^{n+1} = \mathbf{V}_j^n - \frac{\Delta t}{2h} \mathbf{A}(\mathbf{V}_{j+1}^n - \mathbf{V}_{j-1}^n) + \frac{\Delta t}{2h} |\mathbf{A}|(\mathbf{V}_{j+1}^n - 2\mathbf{V}_j^n + \mathbf{V}_{j-1}^n).$$

Replacing \mathbf{V}_j by $\mathbf{S}^{-1}\mathbf{U}_j$ and premultiplying by \mathbf{S} , we obtain the final formulation

$$\mathbf{U}_j^{n+1} = \mathbf{U}_j^n - \frac{\Delta t}{2h} \mathbf{A}(\mathbf{U}_{j+1}^n - \mathbf{U}_{j-1}^n) + \frac{\Delta t}{2h} |\mathbf{A}|(\mathbf{U}_{j+1}^n - 2\mathbf{U}_j^n + \mathbf{U}_{j-1}^n). \quad (7)$$

The finite element discretization both in space and time of the previous formulation is the following. The boundary Γ of the domain Ω 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, Γ_{u_i} refers to that part of the boundary on which a Dirichlet-type boundary condition is specified for the i th component of the primitive variables (i.e., ρ , u , p), and Γ_{f_i} to that part on which no boundary condition is specified for the i th component. There exists another

boundary condition which is imposed on the slip part of Γ , 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:

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

and

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

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 the Dirichlet boundary condition 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 \mathbf{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

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

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 $\mathbf{U}^j, \mathbf{F}^j$ are the values of \mathbf{U}, \mathbf{F} at node j .

We now have 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:

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

It is important to recognize in this formulation a weighted residual method, that is, consistency is insured. Also this formulation is conservative, because in every point of Ω ,

$$\sum_{j=1}^{\text{NodElm}} N^j = 1,$$

and therefore

$$\sum_{j=1}^{\text{NodElm}} N^j_{,x} = 0,$$

then

$$\sum_{j=1}^{\text{NodElm}} \int_{\Omega^e} \left(N^j + N^j \left(\frac{h}{2} \right) \text{sgn} \mathbf{A} \right) \mathbf{A} \mathbf{U}_{,x} \, d\Omega = \sum_{j=1}^{\text{NodElm}} \int_{\Omega^e} N^j \mathbf{A} \mathbf{U}_{,x} \, d\Omega,$$

and integrating the right-hand side by parts we obtain

$$\sum_{j=1}^{\text{NodElm}} \left(\int_{\Gamma^e} N^j \mathbf{A} \mathbf{U} n_x \, d\Gamma - \int_{\Omega^e} N^j_{,x} \mathbf{A} \mathbf{U} \, d\Omega \right) = \sum_{j=1}^{\text{NodElm}} \int_{\Gamma^e} N^j \mathbf{A} \mathbf{U} n_x \, d\Gamma = \int_{\Gamma^e} \mathbf{F}_n \, d\Gamma,$$

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

1.2. 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. Equation (8) together with the appropriate boundary conditions define a well posed physical problem (see [3, 4] for 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:

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

and

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

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 τ 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 (9):

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

integrating by parts, we obtain

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

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 α can be chosen so that in the unidimensional case the solution will have nodal exactness.

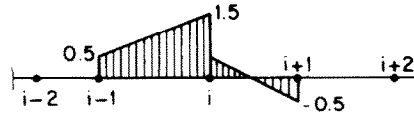


Fig. 1. Weighting functions.

In the unidimensional case this formulation reduces to

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

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

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

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

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

in which ξ_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 that of (10).

When adaptive refinement is used, neither (10) nor (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 [5]). 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

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

In Figs. 2(a)–(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 Figs. 2(d)–(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.

1.3. Multidimensional compressible Euler equations

A Petrov–Galerkin formulation is presented and it is 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

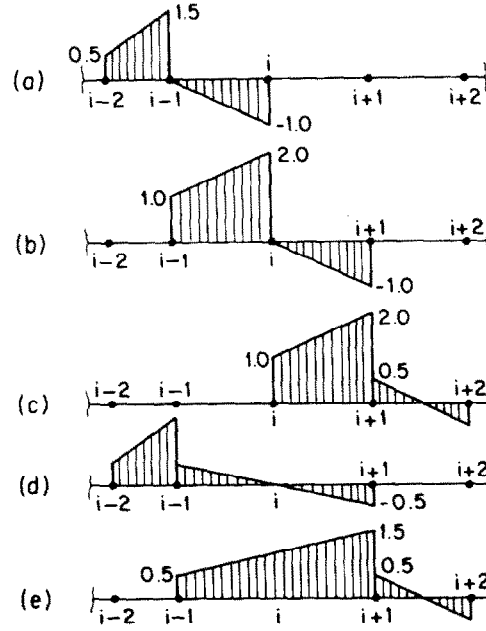


Fig. 2. Sketches of the form functions. (a) Node $i - 1$; (b) node i ; (c) node $i + 1$; (d) compound weighting function for node $i - 1$; (e) compound weighting function for node $i + 1$.

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

and

$$A_i U_{,i} = B_j U_{,\xi_j},$$

it 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:

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

and

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

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 $\tilde{u}_i(\mathbf{x})$ the Dirichlet boundary condition 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 [6]):

$$\int_{\Omega} (N^j + N_{,\xi_j}^j \mathbf{B}_j^t (\mathbf{B}^t \mathbf{B})^{-1/2}) (A_i U_{,i}) d\Omega = 0 \quad \forall N_i^j \in V^i, \quad (13)$$

where

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

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_1^t \\ \vdots \\ \mathbf{B}_{\text{nsd}}^t \end{pmatrix}.$$

1.4. Verifications

1.4.1. 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 [7–9].

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 subroutine level, whereas primitive variables were used at global level.

From the condition of symmetry,

$$\mathbf{A} = \mathbf{A}^t = \Phi \Lambda \Phi^{-1} = \Phi \Lambda \Phi^t.$$

Then

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

Using

$$N_{,\xi}^j = \frac{h}{2} N_{,x}^j,$$

we obtain the already known formulation, i.e.,

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

1.4.2. Bidimensional scalar case

Using

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

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

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

REMARK 1. 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 componential, and thus we have verified the comprehensiveness of this formulation once more.

1.5. Shock capturing concept

1.5.1. Bidimensional scalar advective equation

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

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

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

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

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

$$\mathbf{b}_{\parallel} \frac{1}{|\mathbf{b}_{\parallel}|} \mathbf{b}_{\parallel}^t.$$

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

$$\mathbf{b}_{\parallel} \mathbf{b}_{\parallel}^t \left(\frac{1}{|\mathbf{b}_{\parallel}|} - \frac{1}{|\mathbf{b}|} \right).$$

After the introduction of the above term, the final formulation is

$$\int_{\Omega} \left\{ \left[N^j + (\nabla_{\xi} N^j)^t \mathbf{b} \frac{1}{|\mathbf{b}|} \right] (\mathbf{b}^t \nabla_{\xi} u) + (\nabla_{\xi} N^j)^t \mathbf{b}_{\parallel} \left(\frac{1}{|\mathbf{b}_{\parallel}|} - \frac{1}{|\mathbf{b}|} \right) (\mathbf{b}^t \nabla_{\xi} u) \right\} d\Omega = 0 \quad \forall N^j \in V.$$

A comprehensive description of the shock capturing concept for the linear scalar advection-diffusion equation and the multidimensional advective-diffuse systems can be found in [4] and [10], respectively.

1.5.2. Multidimensional first-order systems of hyperbolic equations

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

$$\mathbf{A}_i = \mathbf{A}_{i_{\parallel}} + \mathbf{A}_{i_{\perp}}, \quad i = 1, \dots, \text{nsd},$$

so that

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

and

$$\mathbf{A}_{i_{\perp}} \hat{\mathbf{U}}_i = \mathbf{0} \quad \forall \hat{\mathbf{U}} / \hat{\mathbf{U}}^t \nabla \mathbf{U} = 0.$$

It 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:

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

and following the development in natural coordinates, we define

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

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

where

$$\int_{\Omega} N_{,\xi_j} \mathbf{B}_{j\parallel}^t (\mathbf{B}_{\parallel}^t \mathbf{B}_{\parallel})^{-1/2} \mathbf{B}_{i\parallel} U_{,\xi_i} d\Omega,$$

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

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

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

where λ and ϕ are the result of the following eigenproblem:

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

Substituting

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

in the eigenproblem, we obtain

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

The solution to this eigenproblem is

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

It 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

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

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 (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

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

2. Implementation and use of the method

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

2.1. Two-dimensional weighted residual formulation for the compressible Euler equations

The Euler equations can be written in conservation form as follows:

$$\mathbf{U}_{,j} + \mathbf{F}_{j,j} = \mathbf{0} \quad \text{on } \Omega \subset \mathbb{R}^2,$$

where

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

$$e = \varepsilon + \frac{u_i u_i}{2}.$$

Here, e is the total energy and ε is the internal energy per unit mass.

In order to complete the system we must specify an equation of state $p = p(\rho, \varepsilon)$. 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.,

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

where γ is the ratio of specific heats.

The flux vector $\mathbf{F}_j(\mathbf{U})$ is a homogeneous function of degree one in the conservative variables \mathbf{U} . It follows that (see [2])

$$\mathbf{F}_j(\mathbf{U}) = \mathbf{A}_j \mathbf{U}$$

and

$$\mathbf{F}_{j,j} = \mathbf{A}_j \mathbf{U}_{,j}.$$

Let $\mathbf{n} = (n_1, n_2)$ be the outward unit normal vector on the boundaries and let \mathbf{F}_j be split in the following way:

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

Later we will make reference to

$$\mathbf{F}_n^{(2)} = \mathbf{F}_j^{(2)} n_j = \begin{pmatrix} u_n \rho \\ u_n \rho u_1 \\ u_n \rho u_2 \\ u_n \rho e \end{pmatrix}.$$

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

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

Also let Γ^e be the whole boundary of element e , Γ the boundary of Ω , and Γ_{int} the following set:

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

The boundary Γ of the domain Ω is assumed to be decomposed as follows:

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

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

The only natural boundary condition is $\mathbf{F}_n^{(2)} = \mathbf{0}$ on Γ_{slip} , because on the inflow/outflow part of the boundary only Dirichlet boundary conditions are specified for a number of primitive variables (i.e., ρ , 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:

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

and

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

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 $\tilde{u}_i(\mathbf{x})$, the Dirichlet boundary condition 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 \mathbf{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

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

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

$$\begin{aligned} \mathbf{0} = & \sum_e \int_{\Omega^e} (N^j + \tilde{P}^j)(U_{,t} + F_{i,i}) \, d\Omega \\ & - \int_{\Gamma_{\text{slip}}} N^j F_n^{(2)} \, d\Gamma - \int_{\Gamma_{\text{int}}} N^j [F_n] \, d\Gamma \quad \forall N_i^j \in V^i, \end{aligned}$$

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:

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

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,

$$\begin{aligned} \mathbf{0} = & \sum_e \int_{\Omega^e} \tilde{P}^j (U_{,t} + F_{i,i}) \, d\Omega + \sum_e \int_{\Omega^e} (N^j U_{,t} - N^j F_i) \, d\Omega \\ & + \sum_e \int_{\Gamma^e} N^j F_n \, d\Gamma - \int_{\Gamma_{\text{slip}}} N^j F_n^{(2)} \, d\Gamma - \int_{\Gamma_{\text{int}}} N^j [F_n] \, d\Gamma \quad \forall N_i^j \in V^i. \end{aligned}$$

Using the following splitting:

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

we can write

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

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

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

where \mathbf{M} is the consistent mass matrix, \mathbf{R} the residue and $\Delta \mathbf{b}$ 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 uses more CPU time, therefore, a lumped mass matrix was used instead.

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

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

Now considering the nodal vector of primitive variables (\mathbf{a}), it is updated after each iteration as follows:

$$\mathbf{a}_{n+1}^j = \mathbf{a}_n^j + \tilde{\mathbf{D}}_j^{-1} \Delta \mathbf{b}^j, \quad j = 1, \dots, \text{numnp},$$

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

2.2. 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:

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

where α is equal to 1 for the interior nodes and to 2 for the boundary nodes.

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

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

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 ≤ 1 if $C \leq 0.90$ for the internal nodes and $C \leq 0.50$ 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 boundary nodes remain the same.

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

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

and

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

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

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

The two-dimensional Euler equations cannot be analyzed as a system of decoupled scalar equations, because there is not a similarity transformation that diagonalizes both Jacobians simultaneously. Nevertheless, we had no instability running the program with $\text{CFLN} \leq 0.90$.

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. 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 boundary condition 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 boundary condition of a heat transfer analysis. However, this proved to be a most unstable boundary condition, 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 [11, p. 335]).

The code avoids this shortcoming evaluating automatically, for each node of the declared slip boundaries, a unit vector $\tilde{\mathbf{n}}^j$, that takes 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.,

$$\tilde{\mathbf{n}}_i^j = \int_{\Gamma_{\text{slip}}} N^j \mathbf{n}_i \, d\Gamma \left(\sum_{i=1}^2 \left(\int_{\Gamma_{\text{slip}}} N^j \mathbf{n}_i \, d\Gamma \right)^2 \right)^{-1/2}.$$

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

$$\mathbf{V}^j = \mathbf{V}_{\text{after iter.}}^j - (\tilde{\mathbf{n}}^j \cdot \mathbf{V}_{\text{after iter.}}^j) \tilde{\mathbf{n}}^j,$$

where $\mathbf{V}_{\text{after iter.}}^j$ is the velocity in the j th node obtained from \mathbf{a}_{n+1}^j and \mathbf{V}^j is the new value of this velocity to be assigned to \mathbf{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 $\text{CFLN} = 0.9$. 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 it 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. The first ones are always 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 what 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.

2.3. 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 boundary condition for the inflow/outflow part is introduced in our formulation through the integral term

$$\text{where } \int_{\Gamma_{\text{in/outflow}}} \mathbf{N} \mathbf{F}_n \, d\Gamma ,$$

$$\mathbf{F}_n = \mathbf{F}_i n_i = (\mathbf{A}_i n_i) \mathbf{U} = \mathbf{A}_n \mathbf{U} .$$

Here, $\mathbf{A}_n = \partial \mathbf{F}_n / \partial \mathbf{U}$ is the Jacobian matrix (see [2]).

The matrix \mathbf{A}_n has a complete set of real eigenvalues for any flow condition. Therefore, \mathbf{A}_n can be written as follows:

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

where $\mathbf{\Lambda}$ is a diagonal matrix with entries

$$\lambda_1 = \lambda_2 = \lambda_3 = u_i n_i ,$$

$$\lambda_4 = \lambda_1 + a ,$$

$$\lambda_5 = \lambda_1 - a .$$

Here, a stands for the local sound speed.

Considering

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

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

$$\mathbf{A}_n = \mathbf{A}_n^+ + \mathbf{A}_n^- ,$$

$$\mathbf{F}_n = \mathbf{A}_n^+ \mathbf{U} + \mathbf{A}_n^- \mathbf{U} ,$$

here, the Jacobian \mathbf{A}_n^+ (\mathbf{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 boundary condition, 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 shown in Table 1.

Table 1

| M_x number | Inflow | Outflow |
|--------------|---------------------|---------|
| <1 | u_1, u_2, ρ | p |
| >1 | u_1, u_2, ρ, p | – |

3. Results

3.1. Evaluation of an oblique shock wave

We present a problem of which we know the analytical solution; it consists of the evaluation of an oblique shock wave originated when a flow incides over a wedge (see Fig. 3). This problem has already been used to test several schemes [12]. 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 20×20 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 degrees of freedom ($u_2 = 0$), 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 either on the outflow (C–D) or on the slip boundary [D–A]. The boundary condition to be imposed in node A is not unique, but 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 evaluation (the observation point is indicated in Fig. 3). We can see that a sharp shock without oscillations was obtained. With

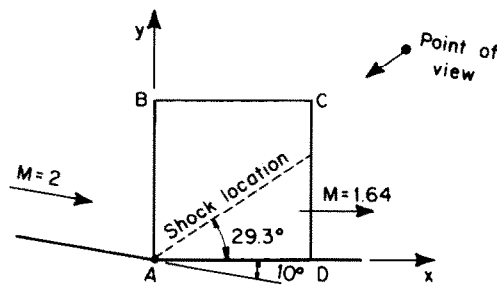


Fig. 3. Oblique shock wave.

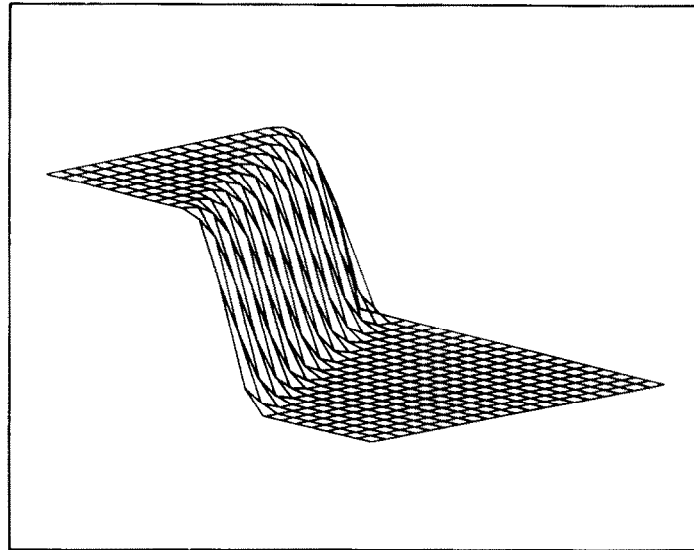


Fig. 4. Density evaluation on the oblique shock wave.

regard to the numerical values, we can say that there was complete agreement between the numerical and the analytical values.

3.2. Calculations for the NACA0012 airfoil

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

These are two of the cases considered in the AGARD Fluid Dynamics Panel Working Group 07 (see [3]).

3.3. Meshes

Figure 5 shows the final mesh for the first test and Fig. 9 that of the second test. Each one

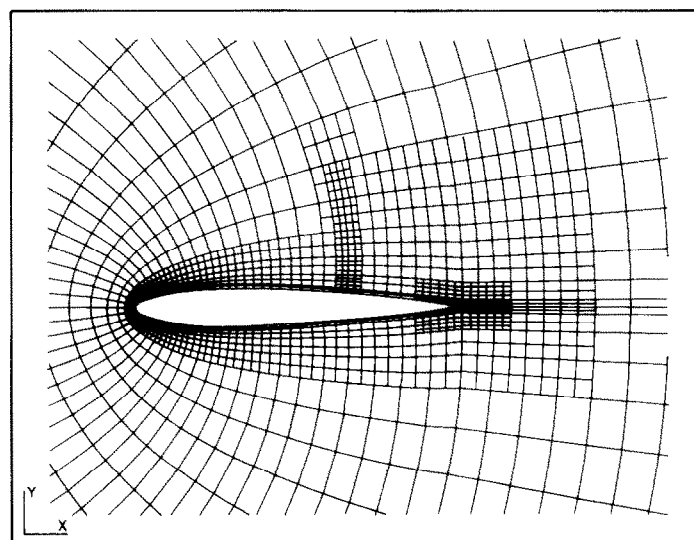


Fig. 5. NACA 0012 Airfoil: final mesh, first test.

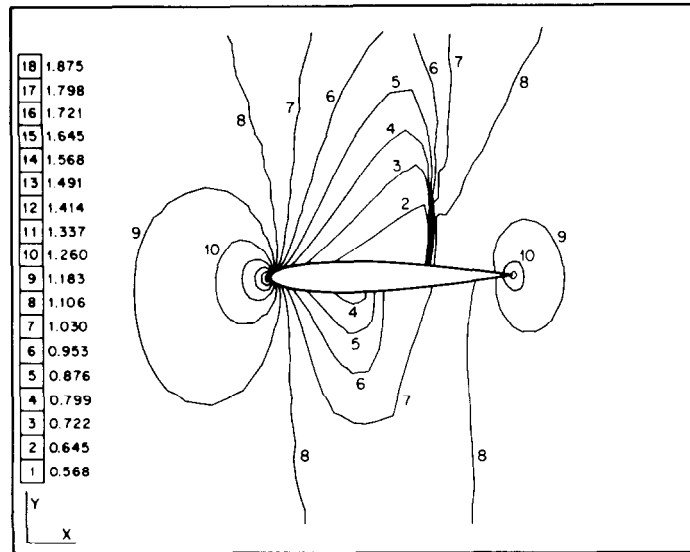


Fig. 6. NACA 0012 Airfoil: pressure contours, first test.

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. The mesh was refined overall when the rate of convergence decay became too much.

Only the final stages of refinement were of adaptive type. In this case, the gradients of the pressure were used as the criterion to switch the adaptive refinement. At the interfaces of different mesh sizes, the continuity is maintained by elimination of internal nodes, e.g. by sharing the forces between the adjacents nodes.

In fact, the refinement technique was introduced in the code in the simplest way. It is the object of current research and it is not considered in the scope of this paper.

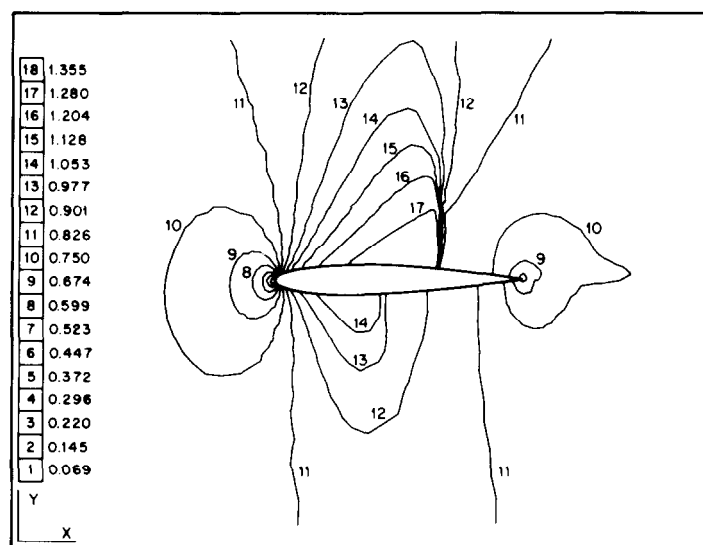


Fig. 7. NACA 0012 Airfoil: Mach contours, first test.

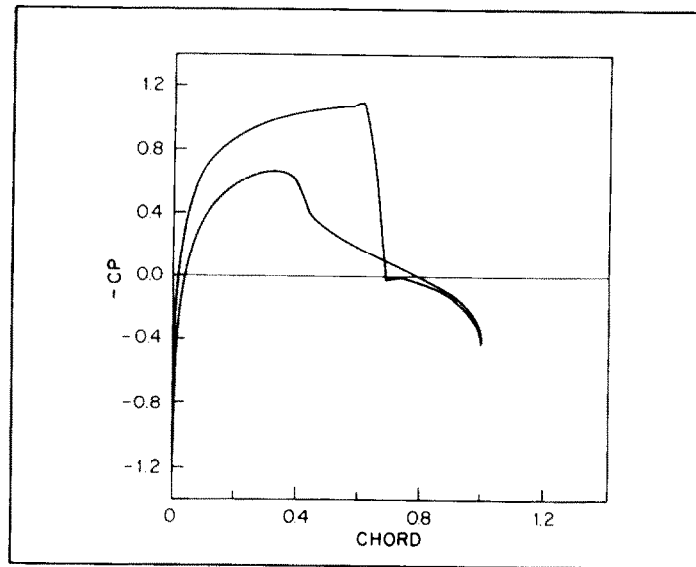


Fig. 8. NACA 0012 Airfoil: C_p distribution, first test.

3.4. 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 were $\rho = 1$, $u_1 = \cos(1.25)$, $u_2 = \sin(1.25)$ and $p = 1.11607$ everywhere. The boundary conditions were the following: (1) slipping boundary condition on the airfoil; (2) imposition of ρ , 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 $\rho = 1$, $u_1 = \cos(7.0)$, $u_2 = \sin(7.0)$ and $p = 0.49603$ everywhere. The boundary conditions were the following: (1) slipping boundary condition on the airfoil; (2) imposition of ρ , u_1 , u_2 and p on the inflow part of the domain.

3.5. Numerical results

Figures 6–8 show pressure contours, Mach contours and C_p distribution for the first case,

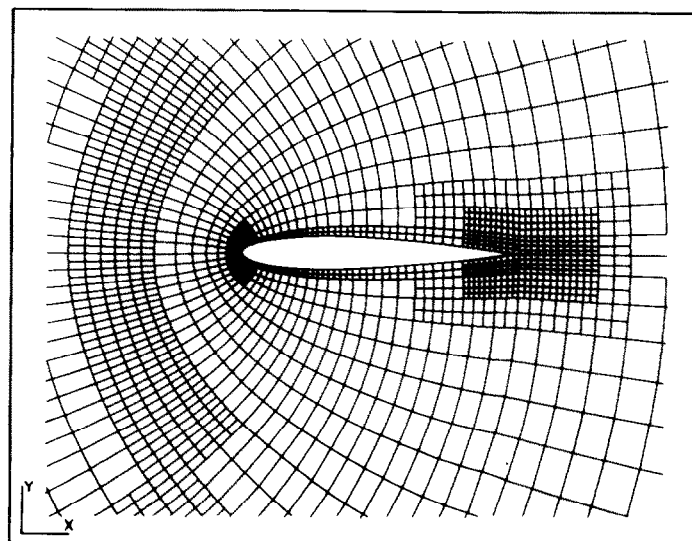


Fig. 9. NACA 0012 Airfoil: final mesh, second test.

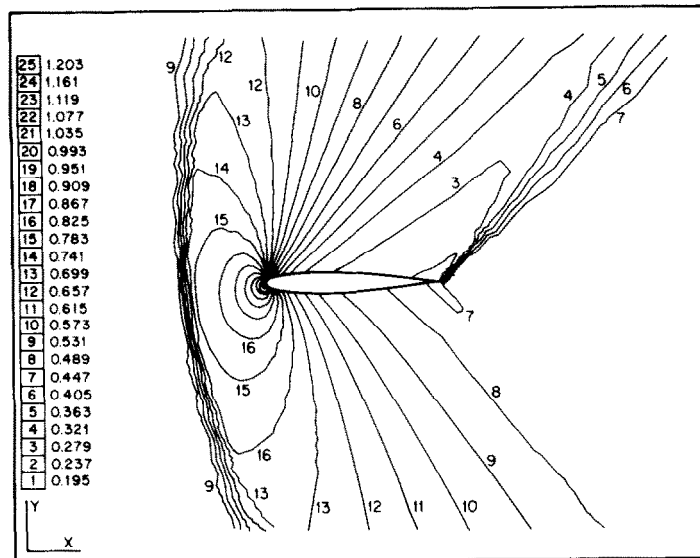


Fig. 10. NACA 0012 Airfoil: pressure contours, second test.

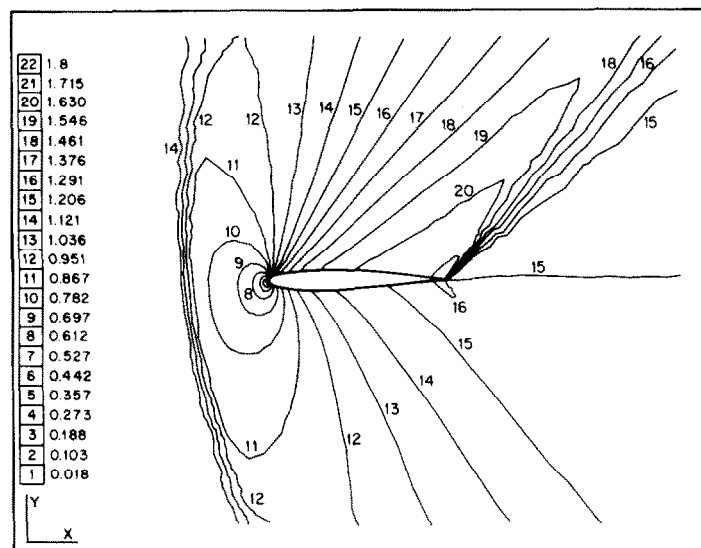


Fig. 11. NACA 0012 Airfoil: Mach contours, second test.

whereas for the second case, the pressure and Mach contours are shown in Figs. 10 and 11, respectively.

The numerical results are in good agreement with those reported in [13]. Only in the first test case is there 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 [13].

4. 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 cannot be fitted, the code can still solve the problem, no matter how complicated the domain or the boundary conditions.

Acknowledgment

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 Karl for his outstanding job in typing the manuscript.

References

- [1] J. Donéa, A Taylor–Galerkin method for convective transport problems, *Internat. J. Numer. Methods Engrg.* 20 (1984) 199–259.
- [2] 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.
- [3] 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, *Comput. Methods Appl. Mech. Engrg.* 73 (1989) 173–189.
- [4] T.J.R. Hughes, M. Mallet and M. Mizukami, A new finite element method for computational fluid dynamics: II. Beyond SUPG, *Comput. Methods Appl. Mech. Engrg.* 54 (1986) 341–355.
- [5] P. Devloo, J.T. Oden and T. Strouboulis, Implementation of an adaptive refinement technique for the SUPG algorithm, *Comput. Methods Appl. Mech. Engrg.* 61 (1987) 339–358.
- [6] 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, *Comput. Methods Appl. Mech. Engrg.* 58 (1986) 305–328.
- [7] A. Harten, On the symmetric form of systems of conservation laws with entropy, *J. Comput. Phys.* 49 (1983) 151–164.
- [8] E. Tadmor, Skew-selfadjoint forms for systems of conservation law, *J. Math. Anal. Appl.* 103 (1984) 428–442.
- [9] 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, *Comput. Methods Appl. Mech. Engrg.* 54 (1986) 223–234.
- [10] 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, *Comput. Methods Appl. Mech. Engrg.* 58 (1986) 329–336.
- [11] M.A. Storti, C.E. Baumann and S.R. Idelsohn, Stability analysis for the calculation of transonic flows with supg-type schemes, *GTM inter. comm.*
- [12] S.F. Davis, A rotationally biased upwind difference scheme for the Euler equations, *J. Comput. Phys.* 56 (1984) 65–92.
- [13] T.H. Pulliam and J.T. Barton, Euler computations of AGARD Working Group 07 Airfoil Test Cases, *AIAA Paper 85-0018*, 1985.
- [14] 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, *Comput. Methods Appl. Mech. Engrg.* 59 (1986) 327–362.
- [15] C. Koeck, Computation of three-dimensional flow using the Euler equations and a multiple-grid scheme, *Internat. J. Numer. Methods Fluids* 5 (1985) 483–500.
- [16] M.S. Engelman, R.L. Sani and P.M. Gresho, The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow, *Internat. J. Numer. Methods Fluids* 2 (1982) 225–238.
- [17] C.E. Baumann, M.A. Storti and S.R. Idelshon, Absorbing boundary conditions for the solution of transonic flows, *GTM inter. comm.*