INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 34, 543–568 (1992)

IMPROVING THE CONVERGENCE RATE OF THE PETROV–GALERKIN TECHNIQUES 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

SUMMARY

This paper report progress on a technique to accelerate the convergence to steady solutions when the streamline-upwind/Petrov–Galerkin (SUPG) technique is used. Both the description of a SUPG formulation and the documentation of the development of a code for the finite element solution of transonic and supersonic flows are reported. 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, together with an appropriate convergence rate to the steady solution.

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.

It is shown that the velocities at which the error is absorbed in and ejected from the domain (that is damping and group velocities respectively) are strongly affected by the time step used, and that damping gives an $O(N^2)$ algorithm contrasting with the $O(N)$ one given by absorption at the boundaries. Nonetheless, the absorbing effect is very low when very different eigenvalues are present, such as in the transonic case, because the stability condition imposes a too slow group velocity for the smaller eigenvalues. To overcome this drawback we present a new *mass matrix* that provides us with a scheme having the highest group velocity attainable in all the components.

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.

1. SUPG FORMULATION

We begin by analysing 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:

$$\mathbf{U}_{,t} + \mathbf{F}_{,x} + \mathbf{G} = \mathbf{0}$$

*Graduate Research Assistant
†Professor and Scientific Researcher

0029–5981/92/060543–26$13.00
© 1992 by John Wiley & Sons, Ltd.

Received 30 August 1990
Revised 28 February 1991

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

$$\mathbf{U}_{,t} + \mathbf{A}\mathbf{U}_{,x} + \mathbf{G} = \mathbf{0} \tag{1}$$

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

Using Taylor's theorem we can write

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

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

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

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 Reference 1 for further details).

The Jacobian matrix $\mathbf{A}$ can be diagonalized (see Reference 2); therefore we can write

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

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

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \mathbf{\Lambda}\mathbf{V}^n_{,x}\Delta t + \mathbf{\Lambda}^2\mathbf{V}_{,xx}\frac{\Delta t^2}{2!} \tag{3}$$

which is a system of uncoupled scalar equations.

If equation (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 equation (3) with linear finite elements, which yield central differences in space, and investigate the behaviour of the resulting scheme. For an assemblage of elements of uniform length we get

$$\mathbf{V}^{n+1}_j = \mathbf{V}^n_j - \Delta t\frac{\mathbf{\Lambda}}{2h}(\mathbf{V}^n_{j+1} - \mathbf{V}^n_{j-1}) + \frac{\Delta t^2}{2}\frac{\mathbf{\Lambda}^2}{h^2}(\mathbf{V}^n_{j+1} - 2\mathbf{V}^n_j + \mathbf{V}^n_{j-1}) \tag{4}$$

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)

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

and that of the discretized case is

$$\mathbf{V}^n_j = \mathbf{H}^n(k_l)\,\mathrm{e}^{\mathrm{i}k_l jh} \tag{5}$$

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

$$\mathbf{V}^{n+1}_j = \left[1 - \frac{\mathbf{C}}{2}(\mathrm{e}^{\mathrm{i}k_l h} - \mathrm{e}^{-\mathrm{i}k_l h}) + \frac{\mathbf{C}^2}{2}(\mathrm{e}^{\mathrm{i}k_l h} - 2 + \mathrm{e}^{-\mathrm{i}k_l h})\right]\mathbf{H}^n(k_l)\,\mathrm{e}^{\mathrm{i}k_l jh}$$

$$\mathbf{V}^{n+1}_j = \mathbf{G}(\mathbf{C})\mathbf{V}^n_j$$

where $\mathbf{C}$ is a diagonal matrix in which the diagonal elements are the Courant-Friedrich-Lewy-numbers (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*

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

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

$$\rho(\mathbf{G}(\mathbf{C})) = \|\mathbf{G}(\mathbf{C})\|_2$$

and the *norm condition* is satisfied if

$$|\mu_i| \leqslant 1 \quad \forall i$$

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

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

As a result, we can conclude that

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

A foregone conclusion is that the only stable way of treating the system of equation (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

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

converges, but to reach convergence the term within the square brackets must vanish, i.e.

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

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

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

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

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

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:

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

Here, Γ_{u_i} refers to that part of the boundary on which a Dirichlet-type boundary condition (b.c.) is specified for the ith component of the primitive variables (i.e. ρ, u, p), and Γ_{f_i} to that part on which no b.c. is specified for the ith component. There exists another b.c. 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 followng 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 ith component of the trial solutions in conservation variables, and the function $\tilde{u}_i(x)$ the Dirichlet b.c. for the ith 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

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

where numnp denotes the total number of nodes in the discretization, $\mathbf{N}^j = \text{diag}(N_1^j, N_2^j, N_3^j)$ 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 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:

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

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

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

then

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

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

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

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:

$$a_i u_{,i} = 0 \tag{8}$$

where u is the unknown scalar field and a_i is the ith component of the flow velocity. Equation (8) together with the appropriate boundary conditions defined a well posed physical problem (see References 3 and 4 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:

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

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:

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

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 equation (9):

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

Integrating by parts, we obtain

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

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

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

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

In the unidimensional case this formulation reduces to

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

where ξ refers to the natural co-ordinate system $[-1, 1]$. Owing to the form of the resulting weighting functions when α takes the value 2 (see Figure 1), nodally exact solutions are obtained

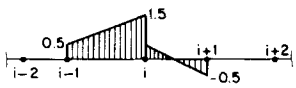

Figure 1. Weighting functions

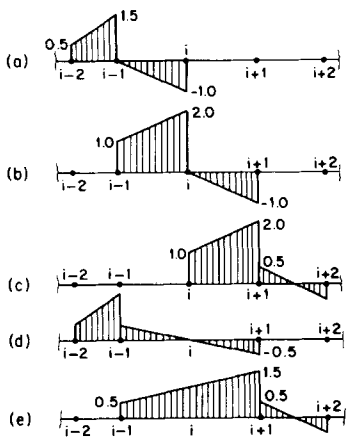

Figure 2. Sketches of the form functions: (a) node $i-1$; (b) node i; (c) node $i+1$; (d) compund weighting function for the node $i-1$; (e) compound weighting function for the node $i+1$

no matter how much different in length the elements may be. Therefore the optimal value of α to be used in equation (10) is 2.

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

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

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

in which ξ_i refers to the ith natural co-ordinate, 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 equation (10).

When adaptive refinement is used, neither equation (10) nor equation (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 neighbours (as is advocated in Reference 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:

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

In Figures 2(a), 2(b) and 2(c) we have the sketches of the form functions for the nodes $i-1$ 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 ith weighting function for each one of its neighbours.

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

1.3. Multidimensional compressible Euler equations

A Petrov–Galerkin formulation will be presented and it 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

$$\mathbf{U}_{,i} = \frac{\partial \xi_j}{\partial x_i} \mathbf{U}_{,\xi_j}$$

and

$$\mathbf{A}_i \mathbf{U}_{,i} = \mathbf{B}_j \mathbf{U}_{,\xi_j}$$

it follows that

$$\mathbf{B}_j = \mathbf{A}_i \frac{\partial \xi_j}{\partial x_i}$$

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 ith component of the trial solutions in conservation variables and the function $\tilde{u}_i(\mathbf{x})$ the Dirichlet b.c. for the ith 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 Reference 6):

$$\int_\Omega (\mathbf{N}^j + \mathbf{N}^j_{,\xi_j} \mathbf{B}^t_j (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1/2})(\mathbf{A}_i \mathbf{U}_{,i})\, \mathrm{d}\Omega = \mathbf{0} \quad \forall N^j_i \in V^i \tag{13}$$

where

$$\mathbf{N}^j = \mathrm{diag}(N^j_1, N^j_2, N^j_3, N^j_4)$$

and

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}^t_1 \\ \vdots \\ \mathbf{B}^t_{\text{nsd}} \end{pmatrix}$$

1.4. 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 References 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 levels.

From the condition of symmetry,

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

then

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

Using

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

we obtain the already known formulation, i.e.

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

2. Bidimensional scalar case.

Using

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

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

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

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.

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:

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

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:

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

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

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

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

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

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

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

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 References 4 and 10 respectively.

1.5.2. Multidimensional first-order systems of hyperbolic equations. As was done for the scalar case, the Jacobian matrices are split in the following way:

$$\mathbf{A}_{i}=\mathbf{A}_{i||}+\mathbf{A}_{i\perp}\quad(i=1,\ldots,\mathrm{nsd})$$

so that

$$\mathbf{A}_{i}\mathbf{U}_{,i}=\mathbf{A}_{i||}\mathbf{U}_{,i}$$

and

$$\mathbf{A}_{i||}\hat{\mathbf{U}}_{i}=\mathbf{0}\quad\forall\,\hat{\mathbf{U}}/\hat{\mathbf{U}}^{t}\nabla\mathbf{U}=0$$

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

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

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

and, following the development in natural co-ordinates, we define

$$\mathbf{B}_{i||}=\frac{\partial\xi_{i}}{\partial x_{j}}\mathbf{A}_{j||}$$

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

$$\int_{\Omega}\mathbf{N}_{,\xi_{j}}\mathbf{B}_{j||}^{t}(\mathbf{B}_{||}^{t}\mathbf{B}_{||})^{-1/2}\mathbf{B}_{i||}\mathbf{U}_{,\xi_{i}}\,\mathrm{d}\Omega$$

where

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

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

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

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

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

Substituting

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

in the eigenproblem, we get

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

The solution to this eigenproblem is

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

It follows that

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

and now, the shock capturing part of the formulation is

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

Again, as was done 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 equation (13), the contribution of the plain SUPG is

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

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:

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

2. DISCRETIZED EQUATIONS

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

2.1. Weighted residual formulation for the compressible Euler equations

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

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

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 Reference 2)

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

and

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

Let $n = (n_1, n_2)$ 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, $e = 1, \ldots, n_{el}$, where n_{el} is the number of elements. We assume

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

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

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

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

$$\Gamma = \overline{\Gamma_{u_i} \cup \Gamma_{f_i} \cup \Gamma_{slip}}, \quad \varnothing = \Gamma_{u_i} \cap \Gamma_{f_i}$$

$$\varnothing = (\Gamma_{u_i} \cup \Gamma_{f_i}) \cap \Gamma_{slip}, \quad (i = 1, 2, 3, 4)$$

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

The only natural b.c. is $\mathbf{F}_n^{(2)} = \mathbf{0}$ on Γ_{slip}, because on the inflow/outflow part of the boundary only Dirichlet b.c. 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:

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

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 ith component of the trial solutions in conservation variables, and the function $\tilde{u}_i(x)$ the Dirichlet b.c. for the ith 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. The finite approximation leads to

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

where numnp denotes the total number of nodes in the discretization, $\mathbf{N}^j = \mathrm{diag}(N_1^j, N_2^j, N_3^j, N_4^j)$ are the global piecewise bilinear basis functions and $\mathbf{U}^j$, $\mathbf{F}_i^j$ are the values of $\mathbf{U}$, $\mathbf{F}_i$ 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

$$\mathbf{0} = \sum_e \int_{\Omega^e} (\mathbf{N}^j + \tilde{\mathbf{P}}^j)(\mathbf{U}_{,t} + \mathbf{F}_{i,i})\, d\Omega$$

$$- \int_{\Gamma_{slip}} \mathbf{N}^j \mathbf{F}_n^{(2)} d\Gamma - \int_{\Gamma_{int}} \mathbf{N}^j [\mathbf{F}_n]\, d\Gamma \quad \forall N_i^j \in V^i$$

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

$$\begin{aligned} \mathbf{U}_{,t} + \mathbf{F}_{j,j} &= 0 \quad \text{on } \Omega && \text{governing equation} \\ \mathbf{F}_n^{(2)} &= 0 \quad \text{on } \Gamma_{\text{slip}} && \text{null flux condition} \\ [\mathbf{F}_n] &= 0 \quad \text{on } \Gamma_{\text{int}} && \text{continuity condition} \end{aligned}$$

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

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

Using the following splitting,

$$\begin{aligned} \sum_e \int_{\Gamma^e} \mathbf{N}^j\mathbf{F}_n\,\mathrm{d}\Gamma = &\int_{\Gamma_{\text{int}}} \mathbf{N}^j[\mathbf{F}_n]\,\mathrm{d}\Gamma \\ &+ \int_{\Gamma_{\text{int/outflow}}} \mathbf{N}^j\mathbf{F}_n\,\mathrm{d}\Gamma + \int_{\Gamma_{\text{slip}}} \mathbf{N}^j\mathbf{F}_n\,\mathrm{d}\Gamma \end{aligned}$$

we can write

$$\begin{aligned} \mathbf{0} = &\sum_e \int_{\Omega^e} (\mathbf{N}^j + \tilde{\mathbf{P}}^j)\mathbf{U}_{,t}\,\mathrm{d}\Omega + \sum_e \int_{\Omega^e} (\tilde{\mathbf{P}}^j\mathbf{F}_{i,i} - \mathbf{N}^j_{,i}\mathbf{F}_i)\,\mathrm{d}\Omega \\ &+ \int_{\Gamma_{\text{slip}}} \mathbf{N}^j\mathbf{F}_n^{(1)}\,\mathrm{d}\Gamma + \int_{\Gamma_{\text{in/outflow}}} \mathbf{N}^j\mathbf{F}_n\,\mathrm{d}\Gamma \quad \forall N_i^j \in V^i \end{aligned}$$

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 is more CPU-tme consuming; 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 well-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:

$$\mathbf{a}_{n+1}^j = \mathbf{a}_n^j + \mathbf{G}_j\tilde{\mathbf{D}}_j^{-1}\,\Delta b^j \quad (j = 1, \ldots, \text{Numnp})$$

where $\mathbf{G}_j$ is the identity matrix for internal nodes, and for boundary nodes it is a special rectifying matrix with absorbing characteristics (see Reference 11).

2.2. Stability, convergence and boundary conditions[17]

The SUPG method applied to the 1-D-advection equation in a uniform mesh gives

$$u_i^{n+1} = u_i^n + C(u_{i-\delta}^n - u_i^n), \quad C = \frac{\Delta t a}{\Delta x}, \quad \delta = \operatorname{sgn} a \tag{14}$$

Let the mesh be of uniform length, and the solution have the form

$$u_i^n = \lambda^n \xi^i, \quad \text{with} \quad |\xi| = 1 \tag{15}$$

then the amplification coefficient is obtained by substituting equation (15) in equation (14); this coefficient is

$$\lambda = 1 + C(\xi^{-\delta} - 1) \tag{16}$$

2.2.1. Stability condition. To have conditional stability we must find an upper bound for Δt such that

$$|\lambda| \leqslant 1, \quad \forall |\xi| = 1 \text{ and } C \leqslant C_{\max} \tag{17}$$

The geometrical interpretation of equation (17) is that λ (point B) lies on the line joining $\xi^{-\delta}$ and the point (0, 1), and that

$$\frac{\overline{\mathrm{BC}}}{\overline{\mathrm{AC}}} = C$$

As $\xi^{-\delta}$ lies in the unit circle, we conclude that *the scheme is stable if* $C \leqslant 1$ *and unstable otherwise*, that is

$$C \leqslant C_{\max} = 1 \tag{18}$$

The locus of $\lambda(\xi)$ is a circle of radius C and centre at $(1 - C, 0)$.

2.2.2. Convergence by damping. Supposing that the scheme (14) is utilized to capture the steady solution, each ξ component of the error is multiplied by λ in each iteration, and therefore, to reach as fast as possible the steady state, we must look for the C that minimizes $|\lambda|$. As for any ξ, $|\lambda|$ is the distance from B to the origin, it is evident that $|\lambda|$ is minimum when $C = 0{\cdot}5$.

To estimate the rate convergence of this algorithm, we use the fact that the maximum of $|\lambda|_{C=1/2}$ is given by the maximum wavelength admissible by the mesh, namely

$$\xi = \mathrm{e}^{\mathrm{i}\pi/N} = 1 + \frac{\mathrm{i}\pi}{N} - \frac{\pi^2}{2N^2} + O(N^{-3}) \tag{19}$$

then

$$\lambda = 1 - \frac{1}{2}\left(\frac{\mathrm{i}\pi}{N} + \frac{\pi^2}{2N^2}\right) + O(N^{-3}) \tag{20}$$

and

$$|\lambda|^2 = \left(1 - \frac{\pi^2}{4N^2}\right)^2 + \frac{\pi^2}{4N^2} + O(N^{-4}) \tag{21}$$

$$= 1 - \frac{\pi^2}{2N^2} + \frac{\pi^2}{4N^2} + O(N^{-4}) = 1 - \frac{\pi^2}{4N^2} + O(N^{-4}) \tag{22}$$

To reduce this component by a factor ε, we must iterate n times so that

$$|\lambda|^n = \varepsilon \tag{23}$$

Taking logarithms on both sides and using equation (22), we get

$$\frac{n}{2}\log\left(1 - \frac{\pi^2}{4N^2} + O(N^{-4})\right) = \log\varepsilon \tag{24}$$

$$n = \frac{8N^2}{\pi^2 \log(1/\varepsilon)} \tag{25}$$

2.2.3. Absorbent boundary condition. For the last node of the mesh (node N), the SUPG method gives the following equation:

$$u_N^{n+1} = u_N^n + 2C(u_{N-\delta}^n - u_N^n) \tag{26}$$

The limit of stability for this equation is $C = 0{\cdot}5$. If C were chosen equal to 1, and $u_i^0 = (-1)^i$, it can be shown that

$$u_N^n = (2n+1)(-1)^n \tag{27}$$

and here, there is a clear instability in the node N. This instability is not propagated into the domain because in this case *nothing* can be propagated upstream. However, if this scheme is applied to systems with any characteristic velocity directed inward at the boundary, instabilities generated only at boundary level couple with the components whose characteristic velocity introduces the boundary conditions into the domain, and the scheme is globally unstable. Fortunately, this problem can be solved by taking the mass of the boundary nodes twice as much as the real value. The resulting equations are

$$u_N^{n+1} = u_N^n + C(u_{N-\delta}^n - u_N^n) \tag{28}$$

The limit of stability for the boundary nodes is now $C = 1$, and we have the same limit for all the nodes.

2.2.4. Convergence by absorption at the downstream boundary. For $C = 1$ the scheme is not dissipative at all, there is no dispersion and the error is propagated downstream at a rate of 1 element per time step. As the perturbation is completely absorbed in the downstream boundary, it takes N iterations to eliminate the error. For $C < 1$, it can be shown that the scheme is $O(N/C)$ because the group velocity for large wavelengths is C elements per iteration.

Contrasting this rate of convergence with the rate of convergence by damping, we can see the best choice is *the maximum C compatible with stability.*

2.2.5. 1-D systems. When the forward Euler scheme is used in the temporal discretization, the SUPG method applied to 1-D linear systems gives (in the basis of eigenvectors of **A**):

$$\psi_{\mu i}^{n+1} = \psi_{\mu i}^n + \frac{\Delta t\,\lambda_m}{h}(v_{i-\delta}^n - v_i^n) \tag{29}$$

Here, $\delta = \operatorname{sgn}\lambda_m$ gives the correct upwind, and the stability limit for the time step is

$$\Delta t_{\max} = \frac{h}{\max\{|\lambda_\mu|\}} \tag{30}$$

The number of iterations needed to carry the component μ through N elements is

$$n = \frac{Nh}{\Delta t |v_{G\mu}|} = \frac{Nh}{\Delta t |\lambda_\mu|} \tag{31}$$

which is maximum for the lowest eigenvalue, that is

$$n = \frac{Nh}{\Delta t \min\{|\lambda_\mu|\}} \tag{32}$$

Using the maximum Δt, we have

$$n = \frac{N \max\{|\lambda_\mu|\}}{\min\{|\lambda_\mu|\}} \tag{33}$$

For the 1-D Euler equations, we have

$$\min\{|\lambda_\mu|\} = \min\{|u - c|, u\}, \quad \max\{|\lambda_\mu|\} = |u + c| \tag{34}$$

therefore

$$n = N \frac{M + 1}{\min\{M, |M - 1|\}} \tag{35}$$

The convergence is very low for:

(i) incompressible case: $M = \varepsilon \quad n = N/\varepsilon$;

(ii) transonic case: $M = 1 \pm \varepsilon \quad n = N\dfrac{2}{\varepsilon}$.

2.3. *The proposed scheme*

Because only the steady solution is of our concern, we can modify the temporal term to accelerate the convergence. For 1-D systems we choose the temporal term

$$a^{-1}|\mathbf{A}| \frac{\partial \mathbf{U}}{\partial t} = -\mathbf{A} \frac{\partial \mathbf{U}}{\partial x} \tag{36}$$

where a is an arbitrary velocity scale to be absorbed by the time step. Straightforward application of the SUPG method to the spatial part of this equation leads to the following equation for the ith node,

$$|\mathbf{A}| \frac{(\mathbf{U}_i^{n+1} - \mathbf{U}_i^n)}{C} = \frac{1}{2}\mathbf{A}(-\mathbf{U}_{i+1}^n + \mathbf{U}_{i-1}^n) + \frac{1}{2}|\mathbf{A}|(\mathbf{U}_{i+1} - 2\mathbf{U}_i + \mathbf{U}_{i-1})^n \tag{37}$$

and in the basis of eigenvectors of $\mathbf{A}$, equation (37) takes the form

$$C^{-1}(\psi_{\mu i}^{n+1} - \psi_{\mu i}^n) = \tfrac{1}{2}\{\mathrm{sgn}(\lambda_\mu)(\psi_{\mu, i-1} - \psi_{\mu, i+1}) + (\psi_{\mu, i+1} - 2\psi_{\mu i} + \psi_{\mu i-1})\}^n \tag{38}$$

$$\psi_{\mu i}^{n+1} - \psi_{\mu i}^n = C(\psi_{\mu i - \delta_\mu} - \psi_{\mu i}) \tag{39}$$

where

$$\delta_\mu = \mathrm{sgn}(\lambda_\mu)$$

From equation (39) we can see that all the components ψ_μ propagate with the same group velocity, that is, C elements per time step.

For the boundary nodes, this scheme gives a completely absorbent boundary condition in the eigenvectors basis, that is

$$\psi_{\mu N}^{n+1} - \psi_{\mu N}^{n} = \Pi_{\mu\mu}^{+}(\psi_{N-1}^{n} - \psi_{N}^{n})_{\mu} \tag{40}$$

where $\mathbf{\Pi}^{+}$ is the diagonal projection matrix

$$\mathbf{\Pi}^{+} = \mathrm{diag}\{\theta(\lambda_1), \theta(\lambda_2), \ldots, \theta(\lambda_m)\} \tag{41}$$

in which $\theta(x) = (x + |x|)/2$.

Transforming back to $\mathbf{U}$ variables, we get, from equation (40),

$$\mathbf{U}_N^{n+1} - \mathbf{U}_N^{n} = \theta(\mathbf{A})(\mathbf{U}_{N-1}^{n} - \mathbf{U}_N^{n}) \tag{42}$$

where

$$\theta(\mathbf{A}) = \mathbf{S}\mathbf{\Pi}^{+}\mathbf{S}^{-1} \tag{43}$$

In the case of $n_{\mathrm{d}} > 1$, we take $|\mathbf{A}|$ as

$$|\mathbf{A}| = \left(\sum_j \mathbf{A}_j^2\right)^{1/2} \tag{44}$$

In the diagonalizable case it can be shown that all the group velocities are optimal (i.e. $|\mathbf{v}_{\mathrm{G}\mu}| = O(h/\Delta t)$). The boundary condition previously specified is completely absorptive for waves with incidence normal to $\partial\Omega$, i.e. $\mathbf{k} \| \hat{\mathbf{n}}$ at $\partial\Omega$.

This b.c. was implemented in the following way: the absorbent boundary conditions for the decoupled system was

$$\psi_{N+1}^{n+1} = \psi_{N+1}^{n} + \mathbf{\Pi}^{+}(\psi_N^{n} - \psi_{N+1}^{n}) \tag{45}$$

$\mathbf{\Pi}^{+}$ is a projection operator onto the eigenvectors with positive eigenvalues. Coming back to the $\mathbf{U}$ variables,

$$\mathbf{U}_{N+1}^{n+1} = \mathbf{U}_{N+1}^{n} + (|\mathbf{A}|_{N+1}^{+})^{-1}\mathbf{R}_{N+1} \tag{46}$$

But, in practice one can fix only the primitive variables, so that the real boundary conditions are (in the ψ basis):

$$\psi_{N+1}^{n+1} = \psi_{N+1}^{n} + (\mathbf{S}^{-1}\mathbf{\Pi}'\mathbf{S})(\psi_N^{n} - \psi_{N+1}^{n}) \tag{47}$$

where $\mathbf{\Pi}'$ is the projection operator used in the primitive variables and $\mathbf{S}$ is the change of basis matrix. In general, the resulting projection matrix

$$\mathbf{\Pi}'' = \mathbf{S}^{-1}\mathbf{\Pi}'\mathbf{S} \tag{48}$$

will not be diagonal in the ψ basis, so that different components are mixed by the boundary condition. Furthermore, the resulting boundary condition could not be stable.

However, for supersonic flow, both matrices $\mathbf{\Pi}$ and $\mathbf{\Pi}'$ are the identity matrix in the downstream boundary and the null matrix at the upstream boundary so that (38) reduces to (35) and the resulting boundary condition will be stable and absorbent (at least for the linear range).

2.4. 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

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

where

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

Here, $\mathbf{A}_n = \partial \mathbf{F}_n / \partial \mathbf{U}$ is the Jacobian matrix (see Reference 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

$$\mathbf{\Lambda}^{\pm} = \tfrac{1}{2}(\mathbf{\Lambda} \pm |\mathbf{\Lambda}|)$$

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

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

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

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 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 shown in Table I.

2.5. Conclusions

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 forwad 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.

Table I

M_∞ number	Inflow	Outflow
< 1	u_1, u_2, ρ	p
> 1	$u_1, u_2\ \rho, p$	—

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 References 12–14).

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

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

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

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

   where $\mathbf{V}^j_{\substack{\text{after}\\ \text{iter.}}}$ is the velocity in the jth node obtained from $\mathbf{a}^j_{n+1}$ and $\mathbf{V}^j$ is the new value of this velocity to be assigned to $\mathbf{a}^j_{n+1}$.
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 = 0·9. This CFLN is reduced for those nodes that are on the boundary because of stability, the reduction being 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 one can choose between an overall refinement or a localized (adaptive) refinement. As a matter of fact, 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.

3. NUMERICAL RESULTS

3.1. Some preliminary results for the 1-D Euler equations

The proposed scheme was applied to the Euler equations (non-linearized) in the subsonic regime ($M = 0{\cdot}1, 0{\cdot}5$), transonic ($M = 1{\cdot}05$) and supersonic ($M = 2$). The mesh has 20 elements (21 nodes). The Δt was chosen always such that $C = 0{\cdot}9$ for the greatest eigenvalue.

3.1.1. Boundary conditions. The boundary conditions were:

(i) subsonic case ($M = 0{\cdot}5, 0{\cdot}1$): $M = u$, $\rho = 1{\cdot}4$ for the first node, $p = 1$ for the last node;
(ii) supersonic case ($M = 1{\cdot}05, 2$): $u = M$, $\rho = 1{\cdot}4$, $p = 1{\cdot}0$ at the first node, no conditions at the last (downstream) node.

3.1.2. Results.

1. In the supersonic case there is no 'mixing' in the components. We mean by this that a perturbation in pressure is not transferred to the velocity or to the density components. This is a consequence of the fact that all components move with the same group velocity and *in the same sense.*
2. Comparing the two methods for $M = 2$, we see that for the proposed scheme the perturbation travels with no dispersion and with a velocity of $0{\cdot}9h/\Delta t$.
3. For $M = 1{\cdot}05$, running the standard scheme (Figure 3), we have a strong perturbation which travels downstream with velocity $u - c = 0{\cdot}05$. The fact that for node 3

$$c^{-2}\frac{\delta p}{\delta \rho} = \frac{0{\cdot}0367}{0{\cdot}035} \approx 1$$

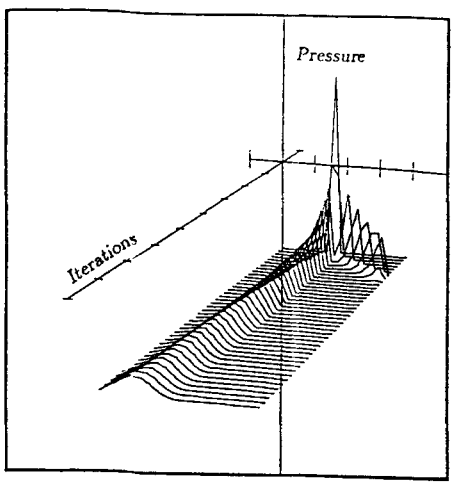

Figure 3. Pressure perturbation for the standard scheme

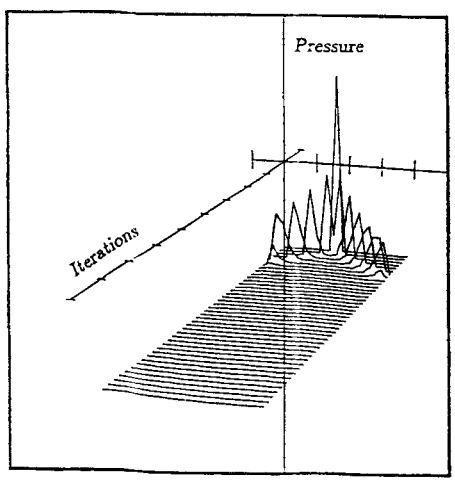

Figure 4. Pressure perturbation for the proposed scheme

and

$$\frac{\rho\delta v}{c\delta\rho} = \frac{1{\cdot}4 \times 0{\cdot}0245}{1 \times 0{\cdot}0355} \approx 1$$

confirms that it is a pressure wave.

With the proposed scheme all the components travels with the same velocity (Figure 4).

4. For all the subsonic cases, there is mixing of components, even with the proposed scheme. That is because the eigencomponents corresponding to both $u + c$ and u travel downstream at the same velocity and are partially reflected at the downstream boundary, which results in a perturbation of the $u - c$ eigencomponent which travels upstream, and again there is a reflection at the upstream boundary, and so on. If the overall gain of this process is greater than one the scheme is unstable. This effect can be observed in several cases.

3.2. Evaluation of an oblique shock wave

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 incident over a wedge (see Figure 5). This problem has already been used to test several schemes.[15] 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 shock.

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 d.o.f. ($u_2 = 0$), but for general curved surfaces this solution is not practical, and one necessarily has 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 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.

$$\text{inflow } (M = 2) \begin{cases} \rho &= 1{\cdot}0 \\ u_1 &= 0{\cdot}98481 \\ u_2 &= -0{\cdot}17365 \\ p &= 0{\cdot}178596 \end{cases}$$

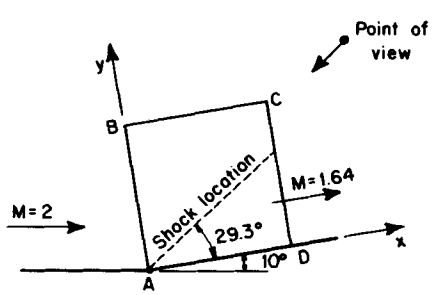

Figure 5. Oblique shock wave

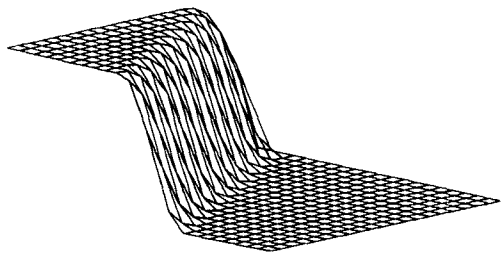

Figure 6. Density elevation on the oblique shock wave

The result is the following:

$$\text{outflow } (M = 1{\cdot}64) \begin{cases} \rho &= 1{\cdot}458 \\ u_1 &= 0{\cdot}887 \\ u_2 &= 0{\cdot}000 \\ p &= 0{\cdot}304 \end{cases}$$

Figure 6 shows this result in the form of density elevation (in Figure 5 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.

3.3. Calculations for the NACA0012 airfoil

Two test cases were chosen; both are lifting flows. The first of them is the ubiquitous case $M_\infty = 0{\cdot}80$ with an *angle of attack* $= 1{\cdot}25$ deg, and the second test has $M_\infty = 1{\cdot}20$ and an *angle of attack* $= 7{\cdot}00$ deg.

These are two of the cases considered in the AGARD Fluid Dynamics Panel Working Group 07 (see Reference 16).

3.3.1. Meshes. Figure 7 shows the final mesh for the first test and Figure 8 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. The mesh was overall refined when the rate of convergence decayed 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 adjacent nodes.

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

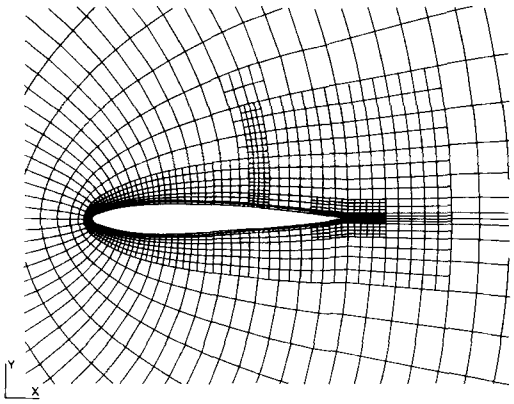

Figure 7. NACA0012 Airfoil: final mesh, first test

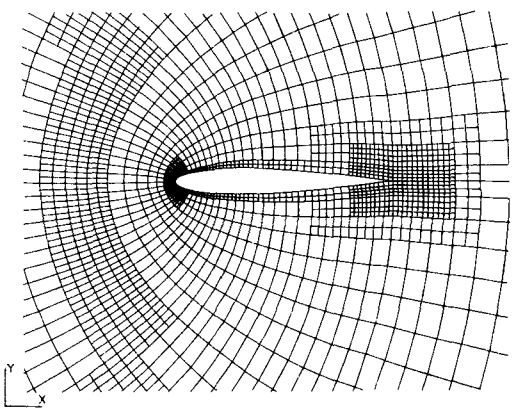

Figure 8. NACA0012 Airfoil: final mesh, second test

3.3.2. 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 $\rho = 1{\cdot}0$, $u_1 = \cos(1{\cdot}25)$, $u_2 = \sin(1{\cdot}25)$, and $p = 1{\cdot}11607$ everywhere.

The boundary conditions are 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 $\rho = 1{\cdot}0$, $u_1 = \cos(7{\cdot}0)$, $u_2 = \sin(7{\cdot}0)$, and $p = 0{\cdot}49603$ everywhere.

The boundary conditions are 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.

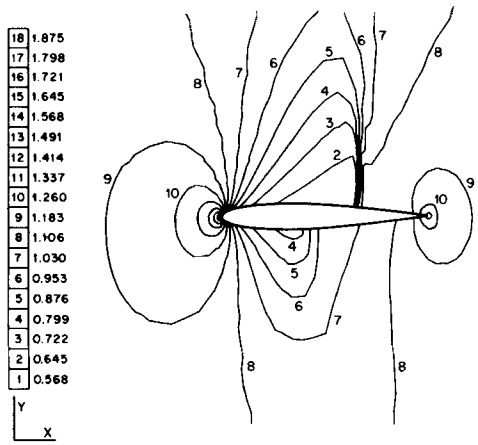

Figure 9. NACA0012 Airfoil: pressure contours, first test

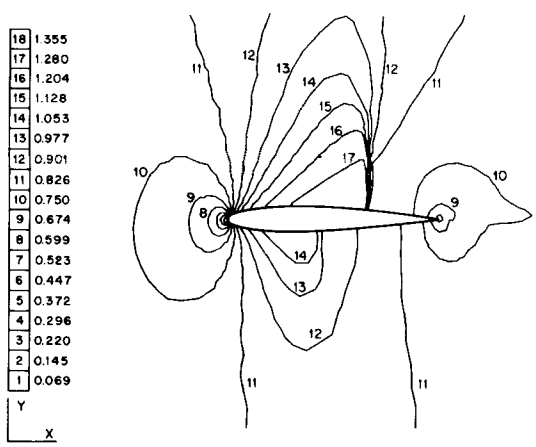

Figure 10. NACA0012 Airoil: Mach contours, first test

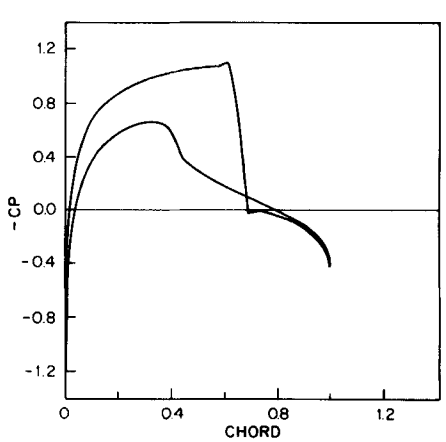

Figure 11. NACA0012 Airfoil: C_p distribution, first test

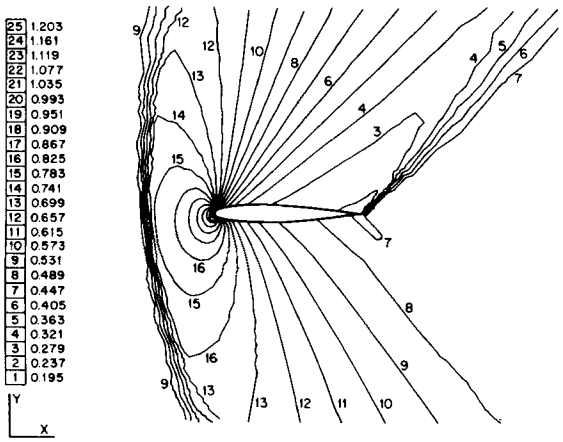

Figure 12. NACA0012 Airfoil: pressure contours, second test

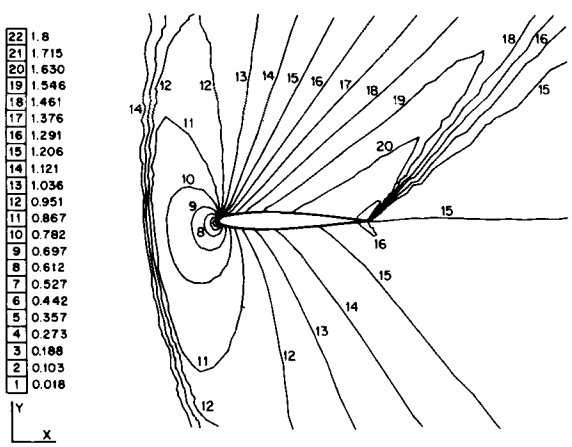

Figure 13. NACA0012 Airfoil: Mach contours, second test

3.3.3. Results. Figures 9–11 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 12 and 13 respectively.

The numerical results are in good agreement with those reported in Reference 16. 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 Reference 16.

3.4. 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 can not be fitted, this code still can solve the problem, no matter how complicated the domain or the boundary conditions may be.

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', *Int. j. numer. methods eng.*, **20**, 199–259 (1984).
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', *Comp. Methods Appl. Mech. Eng.*, **45**, 217–284 (1984).
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', *Comp. Methods Appl. Mech. Eng.*, **73**, 173–189 (1989).
4. T. J. R. Hughes, M. Mallet and M. Mizukami, 'A new finite element method for computational fluid dynamics: II. Beyond SUPG', *Comp. Methods Appl. Mech. Eng.*, **54**, 341–355 (1986).
5. P. Devloo, J. T. Oden and T. Strouboulis, 'Implementation of an adaptive refinement technique for the SUPG algorithm', *Comp. Methods Appl. Mech. Eng.*, **61**, 339–358 (1987).
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', *Comp. Methods Appl. Mech. Eng.*, **58**, 305–328 (1986).
7. A. Harten, 'On the symmetric form of systems of conservation laws with entropy', *J. Comp. Phys.*, **49**, 151–164 (1983).
8. E. Tadmor, 'Skew-selfadjoint forms for systems of conservation law', *J. Math. Anal. Appl.*, **103**, 428–442 (1984).
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', *Comp. Methods Appl. Mech. Eng.*, **54**, 223–234 (1986).
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', *Comp. Methods Appl. Mech. Eng.*, **54**, 223–234 (1986).
11. C. E. Baumann, M. A. Storti and S. R. Idelsohn, 'Absorbing boundary conditions for the solution of transonic flows', GTM internal communication.
12. 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', *Comp. Methods Appl. Mech. Eng.*, **56**, 327–362 (1986).
13. C. Koeck, 'Computation of three-dimensional flow using the Euler equations and a multiple-grid scheme', *Int. j. numer. methods fluids*, **5**, 483–500 (1985).
14. 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', *Int. j. numer. methods fluids*, **2**, 225–238 (1982).
15. S. F. Davis, 'A rotationally biased upwind difference scheme for the Euler equations', *J. Comp. Phys.*, **56**, 65–92 (1984).
16. T. H. Pullimam and J. T. Barton, 'Euler computations of AGARD Working Group 07 Airfoil Test Cases', *AIAA Paper 85-0018*, 1985.
17. M. A. Storti, C. E. Baumann and S. R. Idelsohn, 'Stability analysis for the calculation of transonic flows with SUPG-type schemes', GTM internal communication.