JOURNAL OF COMPUTATIONAL PHYSICS 84, 174–206 (1989)

A Class of Implicit Upwind Schemes for Euler Simulations with Unstructured Meshes

L. FEZOUI

INRIA Sophia-Antipolis, Avenue E. Hughes, 06565 Valbonne, France

AND

B. STOUFFLET

AMD-BA, 78 quai M. Dassault, 92214 Saint-Cloud, France

Received May 29, 1987; revised September 26, 1988

A class of implicit finite element upwind schemes for solving Euler equations is presented. Steady flows with extreme conditions such as high Mach numbers or/and large angles of attack on unstructured meshes can be simulated. Upwind methods are used for the spatial approximation. Higher rates of convergence are obtained by using an implicit scheme relying on a linearization of fluxes and a partial resolution of the systems by a Gauss–Seidel algorithm. The scheme that we get is more efficient and robust than explicit time integration.

I. INTRODUCTION

This paper follows the works of Dervieux *et al.* [2–6], dealing with Euler flow simulation in complex geometries, such as an aircraft, in transonic and supersonic regimes. For this purpose it is interesting and important to build schemes not relying strongly on the regularity of the mesh. Building an approximation scheme on an unstructured grid (of finite element type) or, possibly, on a distorted (maybe locally refined) grid is difficult since the strong variation in the spacing may disturb the internal viscosity of the scheme. With unstructured meshes, the splitting of matrices along the x, y directions may be irrelevant, and the fully multidimensional matrices must be used. Using such grids in combination with explicit schemes leads to strict limitations on the time-step; hence, a large computing time is needed to get to steady state.

Implicit solvers which permit large time steps and CFL numbers lead to a significant decrease in computing time.

0021-9991/89 $3.00

Two types of implicit methods can be found in the literature:

—schemes involving *elliptic operators* [14, 16].

—*linearization of the hyperbolic operator.*

Only the second approach is considered here.

Beam and Warming [7], among the earliest, have given an important contribution with their approximate-factorization finite-difference scheme. Two implicit conservative and non-conservative versions of Harten's scheme [13] are presented by Yee, Harten, and Warming [32], and recently Yee [33] gave accurate results with the TVD version. All these schemes require the solution of *block-tridiagonal* linear systems and are extended in two dimensions by an ADI technique.

The upwind flux-splitting scheme of Steger and Warming [24] leads to a *bidiagonal* linear system. The scheme proposed by MacCormack [17] is non-centered and the linear systems are also *bidiagonal.* An extension of this scheme was also studied by Casier, Deconinck, and Hirsch [8] in a 1-dimensional context.

Rai and Chakravarty [22] have presented an implicit version of the second-order finite difference scheme of Osher [20] where the linear systems are solved by a *relaxation* method. Mulder and Van Leer [19] present an implicit upwind difference scheme for the 1-dimensional Euler equations. They use upwind spatial differencing and *linearization* in time. This method is extended to the 2-dimensional case in generalized coordinates by Van Leer *et al.* [19, 27]; two implicit solvers are described: a *factorization* method and a *linear relaxation.*

The implicit method proposed by Stoufflet [25] applies to *unstructured* meshes and involves a *linearization* of the first-order upwind scheme of Vijayasundaram [31] solved by a *relaxation iteration.* This method can be used to get an implicit version of a given explicit scheme, whether it be centered or upwind. Our purpose is to extend this method to a large class of *upwind* approximations of first- or second-order accuracy, especially adapted to *unstructured* meshes. Two basic ingredients are used in the present work.

First for the spatial approximation, a second-order version of some first-order upwind schemes in conservative form is built following the method introduced by Van Leer [30]. This is done by local interpolation around nodes and some limitation on the slopes for the sake of monotonicity preservation. This method has been already extended to triangular finite elements grids by Vijayasundaram (unpublished) and Fezoui [10] and to fully 2-dimensional finite difference meshes by Montagné [18].

Second for advancing in time, an implicit integrating step is used. Since we are mainly interested by obtaining fast convergence to steady state solutions, the time integration is only an intermediate stage which should be performed as efficiently as possible.

The resulting scheme is, regarding the spatial approximation, a finite element-finite volume second-order scheme without any artificial viscosity parameter.

The time integrator is the first-order implicit linearized scheme of Stoufflet. This

method was especially successful to compute flows with extreme conditions such as, for instance, large Mach numer regimes, large angles of attack, and very irregular meshes in terms of aspect ratios of elements.

In the first part of the paper we recall the expressions for some numerical fluxes and study the time integration. The second part deals with the extension of the scheme to 2-dimensional unstructured meshes. Numerical experiments with different meshes and flux functions are presented.

II. Mathematical Modelling

In this section we recall the mathematical problem and set some definitions and notations which will be used in the sequel.

1. *Governing Equations*

Let $\Omega \subset \Re^2$ be the flow domain and Γ be its boundary. The conservative law form of the equations in two dimensions is

$$\frac{\partial}{\partial t} W + \frac{\partial}{\partial x} F_1(W) + \frac{\partial}{\partial y} F_2(W) = 0$$

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix} = (W^{(k)})_{k=1,4} \tag{2.1}$$

$$F_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{pmatrix}, \qquad F_2(W) = \begin{pmatrix} \rho \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{pmatrix}$$

where ρ is the density, $\mathbf{U} = (u, v)$ is the velocity vector, e is the total specific energy, and p is the pressure of the fluid, with the equation of state given for a perfect gas as

$$p = (\gamma - 1)(e - \tfrac{1}{2}\rho \, \|\mathbf{U}\|^2). \tag{2.2}$$

We introduce the vector

$$\mathbf{F}(W) = \begin{pmatrix} F_1(W) \\ F_2(W) \end{pmatrix} \tag{2.3}$$

The conservative system (2.1) can be rewritten

$$\frac{\partial}{\partial t} W + \nabla \cdot \mathbf{F}(W) = 0. \tag{2.4}$$

We recall that system (2.1) is a hyperbolic system, so that for every couple of real (α_1, α_2), the Jacobian matrix $P(\alpha_1, \alpha_2, W) = \alpha_1 F_1'(W) + \alpha_2 F_2'(W)$ is diagonalizable in the diagonal matrix $\Lambda(\alpha_1, \alpha_2, W) = \text{diag}[\lambda^{(k)}(\alpha_1, \alpha_2, W)]$ and its right eigenvector matrix is denoted by $T(\alpha_1, \alpha_2, W)$. For any real function f, we can define

$$f(P(\alpha_1, \alpha_2, W)) = T^{-1}(\alpha_1, \alpha_2, W)\, \text{diag}[f(\lambda^{(k)}(\alpha_1, \alpha_2, W))]\, T(\alpha_1, \alpha_2, W).$$

2. *Boundary Conditions*

We consider flows around airfoils of inlets, for instance; the domain of computation Ω is described by Fig. 1, where $\mathbf{n}$ is the unit vector of the normal to the boundary $\Gamma = \Gamma_b \cup \Gamma_\infty$.

We assume the flow to be uniform at infinitiy, and the variables to be non-dimensionalized by the free-stream vector W_∞ given by

$$\begin{aligned} \rho_\infty &= 1 \\ \mathbf{U}_\infty &= \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \\ p_\infty &= 1/\gamma M_\infty^2, \end{aligned} \tag{2.5}$$

where α is the angle of attack and M_∞ denotes the free-stream Mach number. On the wall boundary Γ_b, we assume the slip condition

$$\mathbf{U} \cdot \mathbf{n} = 0. \tag{2.6}$$

3. *Definitions*

We assume that Ω is a polygonal bounded domain of $\mathfrak{R}^2$. Let $\mathcal{T}_h$ be a standard triangulation of Ω and h the maximal length of the edges of the triangles of $\mathcal{T}_h$. We need to introduce the following notations.

For every vertex S_i $(i = 1, .., n_h)$ of $\mathcal{T}_h$, the cell $\hat{S}_i$ is the union of the subtriangles resulting from the subdivision by means of the medians of each triangle of $\mathcal{T}_h$ and having S_i as a vertex (Fig. 2). The boundary of $\hat{S}_i$ is denoted by $\partial \hat{S}_i$ and the unit

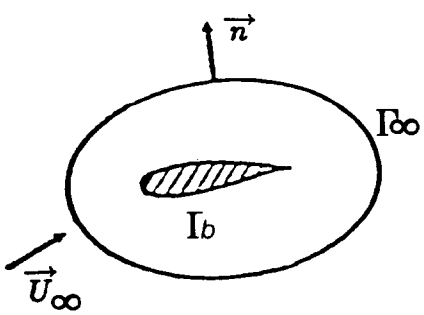

FIGURE 1

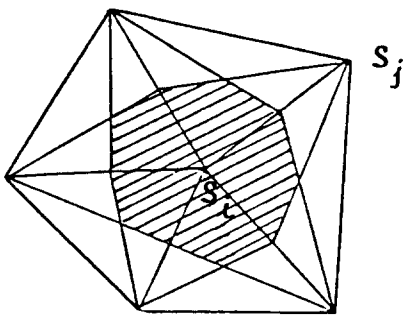

FIGURE 2

vector of the outward normal to $\partial\hat{S}_i$ by $\mathbf{v}_i = (v_{ix}, v_{iy})$. The union of all these cells constitutes a partition of domain Ω.

For every vertex S_i, $\kappa(i)$ is the set of neighboring nodes of S_i. For every domain $\theta \subset \Omega$ we recall that $\chi(\theta)$ is the characteristic function of θ.

We introduce the following discrete spaces (with P_k is the space of polynomials in two variables of degrees at most k):

$$\mathcal{V}_h = \{v_h \mid v_h \in C^0(\Omega),\ v_h|_T \in P_k,\ \forall T \in \mathcal{T}_h\}$$

$$\mathcal{W}_h = \{v_h \mid v_h \in L^2(\Omega),\ v_h|_{\hat{S}_i} = v_i = \text{constant};\ i = 1, ..., n_h\}.$$

Any function ϕ belonging to $\mathcal{V}_h$ is uniquely determined by its values $\phi(S_i)$ at each vertex S_i and if we note $(N_i)_{i=1}^{n_h}$ the basis set of $\mathcal{V}_h$, we have

$$\phi = \sum_{i=1,n_h} \phi(S_i) N_i.$$

There exists a natural bijection between spaces $\mathcal{V}_h$ and $\mathcal{W}_h$ defined by

$$\forall \phi \in \mathcal{V}_h,\ S(\phi) = \sum_{i=1,n_h} \phi(S_i)\, \chi(\hat{S}_i).$$

III. UPWIND APPROXIMATIONS

In this part we develop a systematic procedure to extend on standard triangulations any 3-points scheme defined by a numerical flux function. Then an extension to high-order approximations is proposed.

1. *First-Order Accurate Scheme*

We consider a 1-dimensional system of conservation law with a flux function given by $F: \mathfrak{R}^m \to \mathfrak{R}^m$, where m is the dimension of the system

$$\frac{\partial}{\partial t} W + \frac{\partial}{\partial x} F(W) = 0; \qquad W \in \mathfrak{R}^m. \tag{3.1}$$

The matrix $A(W) = dF/dW$ is the Jacobian matrix of the flux function. We know that a 3-points conservative finite difference scheme is characterized by its numerical flux function $\Phi_F(U, V)$ depending on two variables U and V and satisfying the consistency equation $\Phi_F(U, U) = F(U)$.

The expression of the semi-discretized approximation is given by

$$\begin{aligned} &\frac{\partial W_h}{\partial t} + \frac{1}{\Delta x}(\Phi_{i+1/2} - \Phi_{i-1/2}) = 0 \\ &\Phi_{i+1/2} = \Phi_F(W_i, W_{i+1}) \\ &\Phi_{i-1/2} = \Phi_F(W_{i-1}, W_i), \end{aligned} \tag{3.2}$$

where W_i is the value of the numerical approximation W_h of the solution of (3.1) at $x = i\,\Delta x$ and Δx is the space step.

We construct a 2-dimensional extension of this class of schemes as follows. Let us take the approximation W_h in the space $(\mathscr{V}_h)^m$. A variational approach of equation (2.1) is derived:

$$\begin{aligned} &\text{Find } W_h \in (\mathscr{V}_h)^m, \qquad \forall \phi_h \in \mathscr{V}_h \\ &\int_\Omega \frac{\partial W_h}{\partial t} \phi_h \, dx + \int_\Omega \nabla \cdot \mathbf{F}(W_h)\, S(\phi_h)\, dx = 0. \end{aligned} \tag{3.3}$$

The above formulation can be called a finite volume Galerkin (FVG) approximation. Taking in Eq. (3.3) the function ϕ_h as the shape function N_i associated to the node S_i and integrating by parts with Green's formula on each cell $\hat{S}_i$, the problem becomes

$$\begin{aligned} &\text{Find } W_h \in (\mathscr{V}_h)^m \\ &\int_\Omega \frac{\partial W_h}{\partial t} N_i \, dx = \int_{\partial \hat{S}_i} \mathbf{F}(W_h) \cdot \mathbf{v}_i \, d\sigma \end{aligned} \tag{3.4}$$

Equality (3.4) is just the formulation of the flux balance for the control volume $\hat{S}_i$.

The scheme will be completely defined if we define which approximation is used to compute the right-hand-side integral in (3.4). For this, the boundary $\partial \hat{S}_i$ of the cell $\hat{S}_i$ is split in bisegments $\partial \hat{S}_{ij}$ joining the middle point of the segment $S_i S_j$ to the centroids of the triangle having S_i and S_j as common vertices (Fig. 3).

Let us introduce the notations:

$$F_{ij}(U) = \mathbf{F}(U) \cdot \int_{\partial S_{ij}} \mathbf{v}_i \, d\sigma$$

$$P_{ij}(U) = F_1'(U) \int_{\partial \hat{S}_{ij}} v_{ix} \, d\sigma + F_2'(U) \int_{\partial \hat{S}_{ij}} v_{iy} \, d\sigma = \nabla \cdot \mathbf{F}(U) \cdot \int_{\partial \hat{S}_{ij}} \mathbf{v}_i \, d\sigma.$$

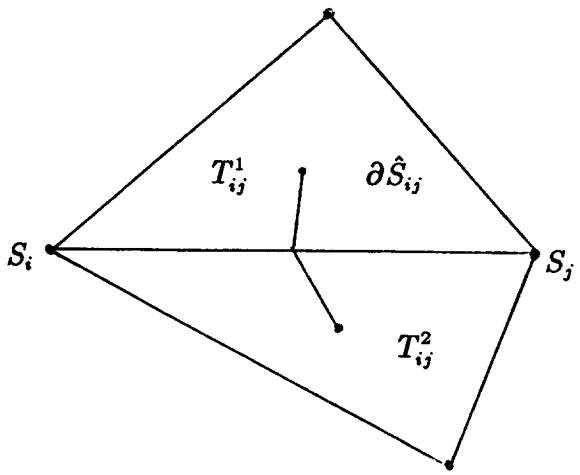

FIGURE 3

Problem (3.3) has the formulation:

$$\text{Find } W_h \in (\mathcal{V}_h)^m$$

$$\int_\Omega \frac{\partial W_h}{\partial t} N_i \, dx = -\sum_{j \in \kappa(i)} \int_{\partial \hat{S}_{ij}} \mathbf{F}(W_h) \cdot \mathbf{v}_i \, d\sigma \qquad \langle 1 \rangle$$

$$-\int_{\partial \hat{S}_i \cap \Gamma_b} \mathbf{F}(\bar{W}_h) \cdot \mathbf{n} \, d\sigma \qquad \langle 2 \rangle$$

$$-\int_{\partial \hat{S}_i \cap \Gamma_\infty} \mathbf{F}(\bar{\bar{W}}_h) \cdot \mathbf{n} \, d\sigma \qquad \langle 3 \rangle. \tag{3.5}$$

Let $H_{ij}^{(1)}$ denote the first term $\langle 1 \rangle$ of the right-hand side RHS of (3.4). The computation of this term will involve the numerical flux function Φ of a first-order accurate upwind scheme described in (3.2) by

$$H_{ij}^{(1)} = \Phi_{F_{ij}}(W_i, W_j),$$

where $W_i = W_h(S_i)$ and $W_j = W_h(S_j)$.

We recall the definition (by their flux numerical function) of the first-order-accurate schemes we have used in this study:

Steger and Warming's scheme [24]

$$\Phi_F^{\mathrm{SW}}(U, V) = A^+(U)U + A^-(V)V; \tag{3.6}$$

Vijayasundaram's scheme [31]

$$\Phi_F^{\mathrm{V}}(U, V) = A^+\left(\frac{U+V}{2}\right)U + A^-\left(\frac{U+V}{2}\right)V; \tag{3.7}$$

Osher's scheme [20]

$$\Phi_F^{\mathrm{OS}}(U, V) = \frac{1}{2}(F(U)+F(V)) - \frac{1}{2}\int_U^V |A(W)|\, dW. \tag{3.8}$$

Remark (1). On the other hand, if we compute now the term $\langle 1 \rangle$ by a centered numerical integration (for example, at the midpoints of the segments, we can easily check if we suppose that $\mathbf{F}$ varies linearly on each element that the resulting formulation is strictly equivalent to the finite element Galerkin approximation given by

$$\text{Find } W_h \in (\mathcal{V}_h)^m, \qquad \forall \phi_h \in \mathcal{V}_h$$

$$\int_\Omega \frac{\partial W_h}{\partial t}\phi_h\, dx = \int_\Omega \mathbf{F}(W_h)\cdot\nabla\phi_h\, dx.$$

The proof lies on the geometrical identity

$$\int_{\partial \hat{S}_{ij}} 3\mathbf{v}_i\, d\sigma = \mathrm{area}(T_{ij}^1)(\nabla N_i(T_{ij}^1) - \nabla N_j(T_{ij}^1)) + \mathrm{area}(T_{ij}^2)(\nabla N_i(T_{ij}^2) - \nabla N_j(T_{ij}^2)),$$

where T_{ij}^1 and T_{ij}^2 are the two triangles having S_iS_j as an edge (Fig. 3).

2. *Second-Order Extension*

The numerical integration with upwind scheme as described previously lead to approximations which are only first-order accurate ones. We present a modification of problem (3.5) in order to get a second-order accurate solution without changing the approximation space.

The ingredients to construct in such a way a second-order accurate approximation are:

(1) A dissipative first-order accurate (quasi-monotone) scheme.

(2) A second-order scheme derived from the previous one by using linear interpolations in the computations of the fluxes.

(3) A limiting procedure which reduces the oscillations of the solution.

A construction of this type introduced in [11, 26] is developed in the present context and is an extension of Van Leer method's [30] to the case of nonstructured meshes.

Let us recall briefly that in one space dimension the method consists in a linear

interpolation of W over each interval $I_i = |(i-\frac{1}{2})\,\Delta x;\,(i+\frac{1}{2})\,\Delta x|$ such that W is given by

$$W(x) = W_i + (x - x_i)P_i \qquad \text{for} \quad x \in I_i$$
$$P_i = (W_{i+1} - W_{i-1})/2\,\Delta x.$$

Then the fluxes are computed with the values $W^n_{i+(1/2)^-} = W_i + (\Delta x/2)P_i$ and $W^n_{i+(1/2)^+} = W_{i+1} - (\Delta x/2)P_{i+1}$ of W at each side of the interface $x_{i+1/2}$.

The spatially second-order accurate version of (3.2) is then given by:

$$\begin{aligned} &\frac{\partial W_h}{\partial t} + \frac{1}{\Delta x}(\Phi_{i+1/2} - \Phi_{i-1/2}) = 0 \\ &\Phi_{i+1/2} = \Phi_F(W_{i+(1/2)^-},\, W_{i+(1/2)^+}) \\ &\Phi_{i-1/2} = \Phi_F(W_{i-(1/2)^-},\, W_{i-(1/2)^+}) \end{aligned} \tag{3.9}$$

If we return to the 2-dimensional situation, to build a second-order-accurate approximation, we need to define the gradient of the solution at each vertex. Clearly, the gradient of a function v_h of $\mathcal{V}_h$ is constant in each element and discontinuous in the domain. We can use the following Galerkin projection to define a gradient $(\nabla v_h)_i$ at each vertex S_i:

$$(\nabla v_h)_i = \frac{1}{\text{area}(\hat{S}_i)} \int_{\hat{S}_i} \nabla v_h \, dx.$$

This interpolation is called "Hermitian interpolation" in [11].

The scheme obtained is a direct extension of the first-order accurate scheme (3.4) and its formulations is

$$\begin{aligned} &\text{Find } W_h \in (\mathcal{V}_h)^m \\ &\int_\Omega \frac{\partial W_h}{\partial t} N_i \, dx = -\sum_{j \in \kappa(i)} H^{(2)}_{ij} + \int_{\partial \hat{S}_i \cap \Gamma_h} \mathbf{F}(\bar{W}^n_h)\cdot \mathbf{n}\, d\sigma + \int_{\partial \hat{S}_i \cap \Gamma_\infty} \mathbf{F}(\bar{\bar{W}}^n_h)\cdot \mathbf{n}\, d\sigma, \end{aligned} \tag{3.10}$$

where

$$\begin{aligned} H^{(2)}_{ij} &= \Phi_{F_{ij}}(W^-_{ij},\, W^+_{ij}) \\ W^-_{ij} &= W_i + \tfrac{1}{2}(\nabla W)_i \cdot \mathbf{S}_i\mathbf{S}_j \\ W^+_{ij} &= W_j - \tfrac{1}{2}(\nabla W)_j \cdot \mathbf{S}_i\mathbf{S}_j. \end{aligned} \tag{3.11}$$

In some cases (transonic and supersonic flows, for example) we need to use a limiting procedure which acts to reduce numerical oscillations of the solution. This is done by a new definition of the values W^-_{ij} and W^+_{ij} at cell interfaces. We remark first:

$$\nabla W|_{T^1_{ij}} \cdot \mathbf{S}_i\mathbf{S}_j = \nabla W|_{T^2_{ij}} \cdot \mathbf{S}_i\mathbf{S}_j = W_j - W_i.$$

Then we define extrapolated gradients in such a manner:

$$(\nabla W_-)_i = 2(\nabla W)_i - \nabla W|_{T_{ij}^\alpha}$$
$$(\nabla W_+)_j = 2(\nabla W)_j - \nabla W|_{T_{ij}^\alpha}, \qquad \alpha = 1 \text{ or } 2,$$

where T_{ij}^1 and Tij^2 are the two triangles described in Fig. 3.

The values at interface needed to compute the flux $H_{ij}^{(2)}$ are now given by:

$$\tilde{W}_{ij}^- = W_i + L_{ij}^- \left(\frac{(1-\kappa)}{4} (\nabla W_-)_i + \frac{(1+\kappa)}{4} (\nabla W)|_{T_{ij}^\alpha} \right) \cdot \mathbf{S}_i \mathbf{S}_j$$
$$\tilde{W}_{ij}^+ = W_j - L_{ij}^+ \left(\frac{(1-\kappa)}{4} (\nabla W_+)_j + \frac{(1+\kappa)}{4} (\nabla W)|_{T_{ij}^\alpha} \right) \cdot \mathbf{S}_i \mathbf{S}_j,$$

where the parameter κ can be chosen to select the degree of upwinding in the interpolation and L_{ij}^- and L_{ij}^+ are the limiting matrices. In all applications we have taken $\kappa = 0$ which corresponds to the so-called Fromm scheme in the MUSCL approach [30]. Various slope limiters can be used to eliminate oscillations. The limiting function used in the applications is one proposed by Van Leer [30],

$$L_{ij}^- = \text{diag}\left[\frac{2(\Delta_-^{(k)})_i (W_j^{(k)} - W_i^{(k)}) + \varepsilon}{(\Delta_-^{(k)})_i^2 + (W_j^{(k)} - W_i^{(k)})^2 + \varepsilon} \right],$$
$$L_{ij}^+ = \text{diag}\left[\frac{2(\Delta_+^{(k)})_j (W_j^{(k)} - W_i^{(k)}) + \varepsilon}{(\Delta_+^{(k)})_j^2 + (W_j^{(k)} - W_i^{(k)})^2 + \varepsilon} \right] \tag{3.12}$$

where $(\Delta_-)_i = (\nabla W_-)_i \cdot \mathbf{S}_i \mathbf{S}_j$, $(\Delta_+)_j = (\nabla W_+)_j \cdot \mathbf{S}_i \mathbf{S}_j$, and ε is a small number that prevents zero division.

A better procedure in terms of accuracy is to use limiters on characteristic variables. For this, we compute these variables by the transformation taken at midpoint of the segment. If we denote by T_{ij} the transformation matrix corresponding to $P_{ij}(W((S_i + S_j)/2)$, the values at the interface are now given by

$$\tilde{W}_{ij}^- = W_i + T_{ij} Lc_{ij}^- T_{ij}^{-1} \left(\frac{(1-\kappa)}{4} (\nabla W_-)_i + \frac{(1+\kappa)}{4} (\nabla W)|_{T_{ij}^\alpha} \right) \cdot \mathbf{S}_i \mathbf{S}_j$$
$$\tilde{W}_{ij}^+ = W_j - T_{ij} Lc_{ij}^+ T_{ij}^{-1} \left(\frac{(1-\kappa)}{4} (\nabla W_+)_j + \frac{(1+\kappa)}{4} (\nabla W)|_{T_{ij}^\alpha} \right) \cdot \mathbf{S}_i \mathbf{S}_j$$

with

$$Lc_{ij}^- = \text{diag}\left[\frac{2(T_{ij}^{-1}(\Delta_-)_i)^{(k)} (T_{ij}^{-1}(W_j - W_i))^{(k)} + \varepsilon}{(T_{ij}^{-1}(\Delta_-)_i)^{(k)2} + (T_{ij}^{-1}(W_j - W_i))^{(k)2} + \varepsilon} \right]$$

and the same for Lc_{ij}^+.

3. *Boundary Conditions*

The second term $\langle 2 \rangle$ and the third term $\langle 3 \rangle$ of the RHS of (3.6) contain the physical boundary conditions. They are taken into account through the vector $\bar{W}_h^n$ which is computed from quantities depending on the interior value W_h^n and by quantities determined from physical boundary conditions.

Wall boundary. The vector $\bar{W}_h$ is assumed to satisfy the slip condition thus term $\langle 2 \rangle$ is computed as

$$\int_{\partial \hat{S}_i \cap \Gamma_b} \mathbf{F}(\bar{W}_h) \cdot \mathbf{n} \, d\sigma = \int_{\partial \hat{S}_i \cap \Gamma_b} \begin{pmatrix} 0 \\ \bar{p} \cdot n_x \\ \bar{p} \cdot n_y \\ 0 \end{pmatrix} d\sigma, \tag{3.13}$$

where $\bar{p}$ is taken as the interior pressure $p(W_h)$. (Note that in this procedure, the slip condition is applied in a weak variational way, as in cell-centered finite volume formulations.)

Inflow and outflow boundaries. At these boundaries we have to select a precise set of compatible exterior datas, depending on the flow regime and the direction of velocity. For this purpose a *plus–minus* flux splitting is applied between exterior data and interior values. More precisely, the boundary integral $\langle 3 \rangle$ is evaluated with the use of the flux-splitting,

$$\int_{\partial \hat{S}_i \cap \Gamma_\infty} \mathbf{F}(\bar{\bar{W}}_h) \cdot \mathbf{n} \, d\sigma = P_{i\infty}^+(W_i) \cdot W_i + P_{i\infty}^-(W_i) \cdot W_\infty, \tag{3.14}$$

where

$$P_{i\infty}(W) = \mathbf{F}'(W) \cdot \int_{\partial \hat{S}_i \cap \Gamma_\infty} \mathbf{n} \, d\sigma.$$

IV. Implicit Solvers

An explicit version in time of semi-discretized approximation (3.9) can be derived by using a diagonal mass-lumped matrix $\Sigma = \text{diag}(\text{area}(\hat{S}_i)/\Delta t)$ for the left-hand side and by replacing the time derivative by an explicit difference

$$\delta \hat{W}_h = \Sigma^{-1} E_h^2(W_h^n) \tag{4.1}$$

with $E_h^2(W_h^n) = \text{RHS}(3.9)$.

This scheme is only first-order accurate in time. However, explicit versions which are second-order accurate (at least) can be built by using a two-step scheme (Hancock–Van Leer [12]), Fezoui [10], or a Runge–Kutta method (Turkel and Van Leer [28]).

1. *First-Order Schemes*

First we show the procedure to get an implicit linearized version on the monodimensional case for simplicity. Suppose given the class of explicit upwind schemes,

$$\begin{aligned} &W_i^{n+1} - W_i^n + \frac{\Delta t}{\Delta x}(\Phi_{i+1/2}^n - \Phi_{i-1/2}^n) = 0 \\ &\Phi_{i+1/2}^n = \Phi(W_i^n, W_{i+1}^n) \\ &\Phi(U, V) = H_1(U, V)U + H_2(U, V)V \qquad \text{for} \quad U, V \in \mathfrak{R}^m \\ &H_1(U, U) + H_2(U, U) = A(U), \end{aligned} \tag{4.2}$$

where the third line is a tentative linearization and where the fourth line is a consistency property.

For instance, in the applications we have used the Steger–Warming scheme which belongs to this class of schemes with:

$$\begin{aligned} H_1(U, V) &= A^+(U) \\ H_2(U, V) &= A^-(V). \end{aligned} \tag{4.3}$$

This numerical flux function does not satisfy the assumption:

$$\Phi(U, V) \qquad \text{is differentiable w.r.t. } (U, V). \tag{4.4}$$

If (4.4) were true, then we could consider the Newton-like linearized implicit version of the scheme (4.2):

$$\begin{aligned} &W^{n+1} - W^n + \frac{\Delta t}{\Delta x}(\Phi_{i+1/2}^N - \Phi_{i-1/2}^N) = 0 \\ &\Phi_{i+1/2}^N = \Phi^N(W_i^n, W_{i+1}^n, W_i^{n+1}, W_{i+1}^{n+1}) \\ &\Phi_{i-1/2}^N = \Phi^N(W_{i-1}^n, W_i^n, W_{i-1}^{n+1}, W_i^{n+1}) \\ &\Phi^N(U, V, W, Z) = \Phi(U, V) + \Phi_{,U}(U, V)(W - U) + \Phi_{,V}(U, V)(Z - V). \end{aligned} \tag{4.5}$$

Now the derivatives of Φ may be very expensive to compute, so we introduce the simplified linearized version of (4.5)

$$\begin{aligned} &W_i^{n+1} - W_i^n + \frac{\Delta t}{\Delta x}(\Phi_{i+1/2}^S - \Phi_{i-1/2}^S) = 0 \\ &\Phi_{i+1/2}^S = \Phi^S(W_i^n, W_{i+1}^n, W_i^{n+1}, W_{i+1}^{n+1}) \\ &\Phi_{i-1/2}^S = \Phi^S(W_{i-1}^n, W_i^n, W_{i-1}^{n+1}, W_i^{n+1}) \\ &\Phi^S(U, V, W, Z) = H_1(U, V)W + H_2(U, V)Z. \end{aligned} \tag{4.6}$$

The resulting scheme is in fact a modified Newton's method where the exact Jacobians arising in (4.5) are replaced by simpler expressions.

We can rewrite (4.6) with a delta formulation,

$$\sigma H^n_{2,i+1/2}\,\delta W^{n+1}_{i+1} + \mathscr{H}^n_i\,\delta W^{n+1}_i - \sigma H^n_{1,i-1/2}\,\delta W^{n+1}_{i-1} = -\sigma(\Phi^n_{i+1/2} - \Phi^n_{i-1/2})$$
$$\mathscr{H}^n_i = I + \sigma H^n_{1,i+1/2} - \sigma H^n_{2,i-1/2} \tag{4.7}$$
$$\delta W^{n+1}_i = W^{n+1}_i - W^n_i,$$

where $\sigma = \Delta t/\Delta x$ and I is the identity matrix.

PROPOSITION (1). *Under assumption* (4.4) *schemes* (4.5) *and* (4.6) *have the same equivalent system up to the second-order, that is, the system approximated by the schemes with a second-order accuracy.*

PROPOSITION (2). *In the scalar linear case, schemes* (4.5) *and* (4.6) *are identical and unconditionally stable.*

The proof of the above propositions can be found in [25].

Remark (1). We cannot ensure that scheme (4.6) will become a quadratically converging method ("quasi-Newton") for Δt tending to infinity as Newton's method in the vicinity of the solution, but we may expect a similar efficiency for the two schemes for a large Δt if the unknowns do not vary too much (as in the case from convergence to steady state).

Although the scheme (3.5) does not satisfy a priori the assumption of differentiable fluxes (4.4), we propose nevertheless to apply the linearization method to get an implicit version of this formulation. The above linearization can be applied to the early formulation (3.5) by using the simplified scheme $\Phi^S_{F_{ij}}$ and can lead to a linear system of the form:

$$M(W^n_h)\cdot(W^{n+1}_h - W^n_h) = B(W^n_h). \tag{4.8}$$

The matrix M has the suitable properties (diagonaly dominant in the scalar case) allowing the use of a relaxation procedure to solve the linear system as shown in Ref. [25].

Such iterative methods considered as modified Newton's methods have been analyzed by Jespersen and Pulliam [15]. These authors showed by a rigorous analysis that the use of incorrect Jacobian matrices can lead to a conditional stability. Nevertheless, the numerical experiments presented below prove the efficiency of this simple approximate Newton's method.

2. *Second-Order Schemes*

If we consider Eq. (4.7) in the delta formulation, an efficient way to get second-order accurate stationary solutions while keeping the interesting properties of the first-order accurate upwind matrix is to replace the RHS of (4.7) by the second-order accurate spatial approximation introduced in Section III. It is convenient to present the resulting scheme as a two-phase algorithm. Starting from an arbitrary

second-order accurate approximation denoted by $E_h^2(W)$ the resulting scheme can be written as:

(i) Physical/explicit/second-order accurate phase

$$\delta \hat{W}_h = -\Sigma^{-1} E_h^2(W_h^n) \tag{4.9}$$

(ii) Mathematical/implicit/first-order accurate phase

$$\begin{aligned} M(W_h^n)\,\delta W_h^n &= \delta \hat{W}_h \\ W_h^{n+1} &= W_h^n + \delta W_h^n. \end{aligned} \tag{4.10}$$

The following properties are easy to prove:

PROPOSITION (3). *In the monodimensional scalar linear case, the scheme defined by* (4.8) *and* (4.9) *is unconditionally stable.*

PROPOSITION (4). *The steady state solutions are second-order accurate and do not depend on the time step used for their computation.*

The previous procedure is rather close to that of Van Leer *et al.* [27, 28], although they point out in this paper that the split fluxes used in the explicit phase should be continuously differentiable.

In most of the applications, we have built the mathematical part from Steger–Warming's flux splitting. In this case, the complete formulation (including linearization of boundary terms and use of local time-stepping) of the method in two space dimensions is given by

$$\text{Find } W_h^{n+1} \in (\mathscr{V}_h)^m$$

$$\delta \hat{W}_i = -\tilde{\sigma}_i (E_h^2(W_h^n))_i$$

$$\begin{aligned} &\delta W_i^n + \tilde{\sigma}_i \Big(\sum_{j \in \kappa(i)} P_{ij}^+ (W_i^n) \Big) \delta W_i^n + \tilde{\sigma}_i \Big(\sum_{j \in \kappa(i)} P_{ij}^- (W_j^n)\, \delta W_j^n \Big) \\ &+ \tilde{\sigma}_i P_{i\infty}^+ (W_i^n)\, \delta W_i^n + \tilde{\sigma}_i \int \sum_{\partial \hat{S}_i \cap \Gamma_b} Q_b(W^n)\, \delta W^n \, d\sigma = \delta \hat{W}_i \end{aligned} \tag{4.11}$$

$$W_h^{n+1} = W_h^n + \delta W_h^n, \tag{4.11}$$

where $\tilde{\sigma}_i = \widetilde{\Delta t}_i / \text{area}(\hat{S}_i)$, $\widetilde{\Delta t}_i$ is a local time step, and the matrix Q_b, derived from the linearization of the wall boundary conditions [28], is given by

$$Q_b(W) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ qn_x & -un_x & -vn_x & n_x \\ qn_y & -un_y & -vn_y & n_y \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.12}$$

with $q = (u^2 + v^2)/2$.

At the end we obtain a quasi-Newton method with three simplifications:

(1) We use a simplified Jacobian in the linearization.

(2) We use a different upwind scheme in the flux approximation and in the linearized part (in the applications, typically we compute the fluxes with Osher Riemann solver and the matrix with the simple Steger–Warming splitting).

(3) The fluxes are approximate with a second-order accurate scheme but the matrix is derived from the first-order one.

V. Numerical Experiments

Our purpose throughout these numerical experiments was not to test the accuracy of the spatial approximation but to emphase the efficiency of the implicit method and to compare with the explicit one to get a steady solution of the Euler equations in different situations. We have used either first-order accurate or second-order accurate approximations depending on the test case. Once the type of the spatial approximation and the iterative procedure have been chosen, the only remaining point is the resolution of the linear system. It appears that Gauss–Seidel relaxation resolution can be very efficient to get the steady solution. Concerning the matrix storage, no optimization in ordering the nodes of the mesh (which are a priori arbitrarily ordered) is needed because only the non-zero entries of the matrix are stored. For each test case, we compare in Table I the efficiency measured by the ratio of CPU time needed to reduce the residual by five orders of magnitude starting from free-stream conditions. The residual is defined in the L_2 norm of the flux evaluated for the continuity equation. In all computations, local time stepping based on a CFL number is used in both explicit and implicit versions of the scheme. The CPU times are given for an IBM3090 scalar computer. The average speed is about 5 Mflops. Efficiency for each test case is presented in Table I.

TABLE I

Efficiency for the Different Cases

Test case	Flow regime		Nodes number	Efficiency
A	0.85		1512	7.5
B1	0.63	2°	3200	11
B2	0.63	2°	4800	12
B3	0.85	1°	3200	7
B4	2.	10°	1360	7
B5	0.85	1°	2492	8.3
C1	8.	0°	1908	9
C2	8.	20°	1908	8

A. *Steady Flow in a Channel*

We choose first a test problem proposed at the GAMM workshop held in 1979 in Stockholm [23] to evaluate the efficiency of the implicit algorithm used as a quasi-Newton method.

Free-stream values correspond to a Mach number of 0.85, for which the flow is transonic. For consistency with the GAMM test, we use a 72×21 triangular mesh. The solver used for this problem relies on Vijayasundaram splitting in both explicit and implicit part. The linear system is solved by a complete nodewise collective Gauss–Seidel relaxation method (about 50 relaxation sweeps). If a high CFL number (10^3) is chosen, the steady solution is obtained with few iterations (about 10); the procedure behaves like a Newton–Raphson method when the initial solution is closed to the final one. An attractive strategy to determine the CFL number during the convergence is to take it as a function of the inverse of the residual. We present in Fig. 4 the convergence history and in Table I the efficiency of the implicit method with a CFL number equal to the inverse of the residual versus the explicit one (here a backward Euler time integrator). The implicit solver converges in about 10 iterations when the explicit one needs more than 5000 time steps.

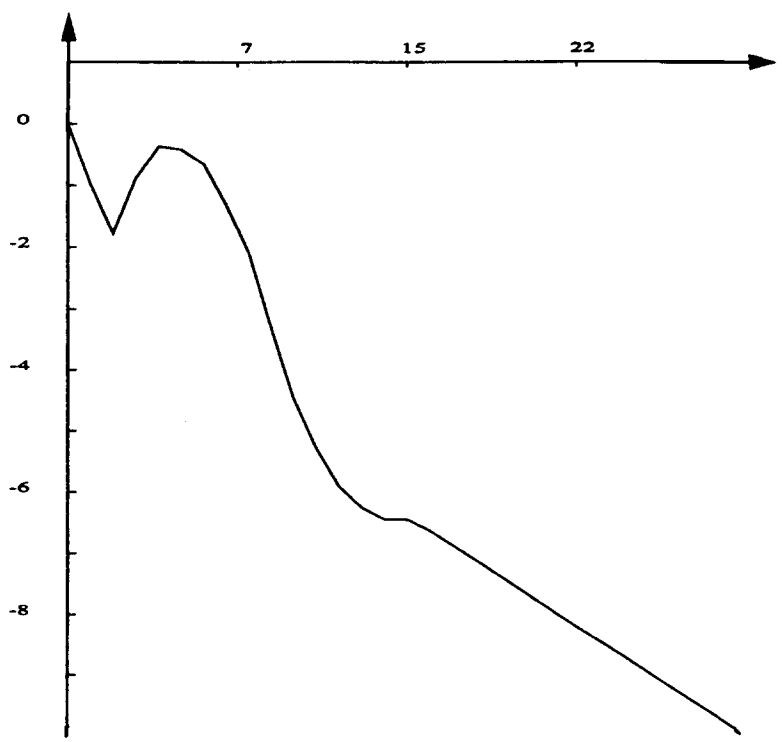

FIG. 4. Residual history for implicit first-order accurate method for the GAMM channel problem: angle of attack 0.00, CFL 1/residual; free-stream Mach 0.85; log residual.

We notice that the convergence is trully quadratic until the steady solution is practically reached (residual of the order of 10^{-4}) and beyond, the convergence becomes linear. This point confirms the efficiency of the chosen linearization. For other numerical results (iso-contours, distributions, ...) we refer to [25].

B. *Steady Flow around a NACA*0012 *Airfoil*

Once again, we are interesting in steady flows calculations around this profile proposed at the same GAMM workshop [23].

We present six numerical experiments which differ by the Mach number M_∞, the angle of attack α and the quality of the mesh in order to explore the capabilities of the described methodology. All calculations have been performed with second-order approximations relying on the Osher's Riemann solver and characteristic limiters. The linear system of the implicit method derived from the Steger–Warming splitting is solved by the nodewise collective Gauss–Seidel relaxation. A first statement can be deduced from these experiments: it is no longer possible in the present test cases to use so large CFL numbers as in (A) problem. This is probably due to the existence of a stagnation point where the gradients of the solution are important.

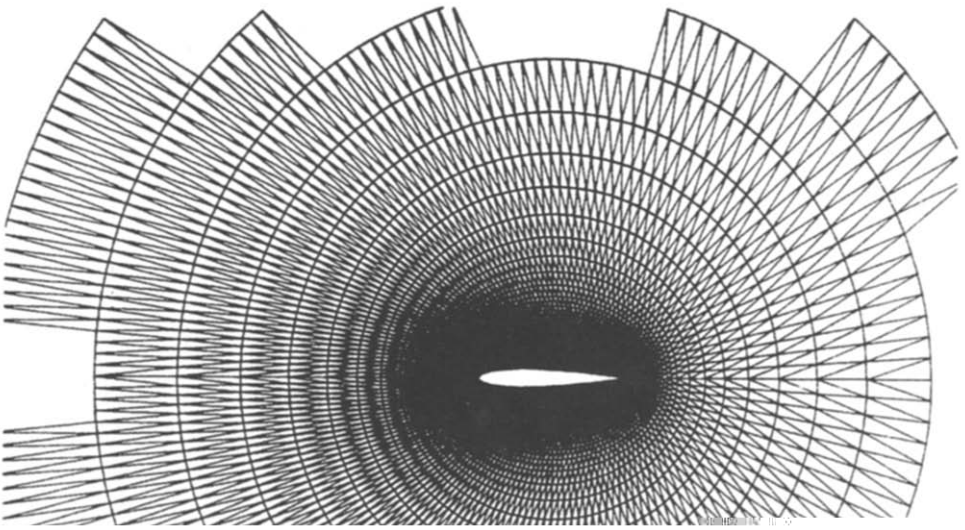

FIG. 5. Enlargement of the 4800 nodes mesh.

So, starting from a uniform flowfield as initial guess, very large transients occur which prevent from using a large CFL number during the initial period. A good strategy is then to choose the CFL number equal to the iteration number. The maximum value is fixed to 50. The method is no longer a Newton-like method and it becomes worthwhile to *solve exactly* the linear system involved in each pseudo-time iteration. In these conditions it seems sufficient to use only a small number of relaxation sweeps in order to obtain both stability and convergence to the steady state. In the applications, the resolution is stopped when the relaxation convergence is less than 10^{-3}. Moreover, relaxation sweeps are performed alternatively by taking points in the increased order then in the decreased one. No improvement in efficiency can be expected by the complete resolution of the linear system.

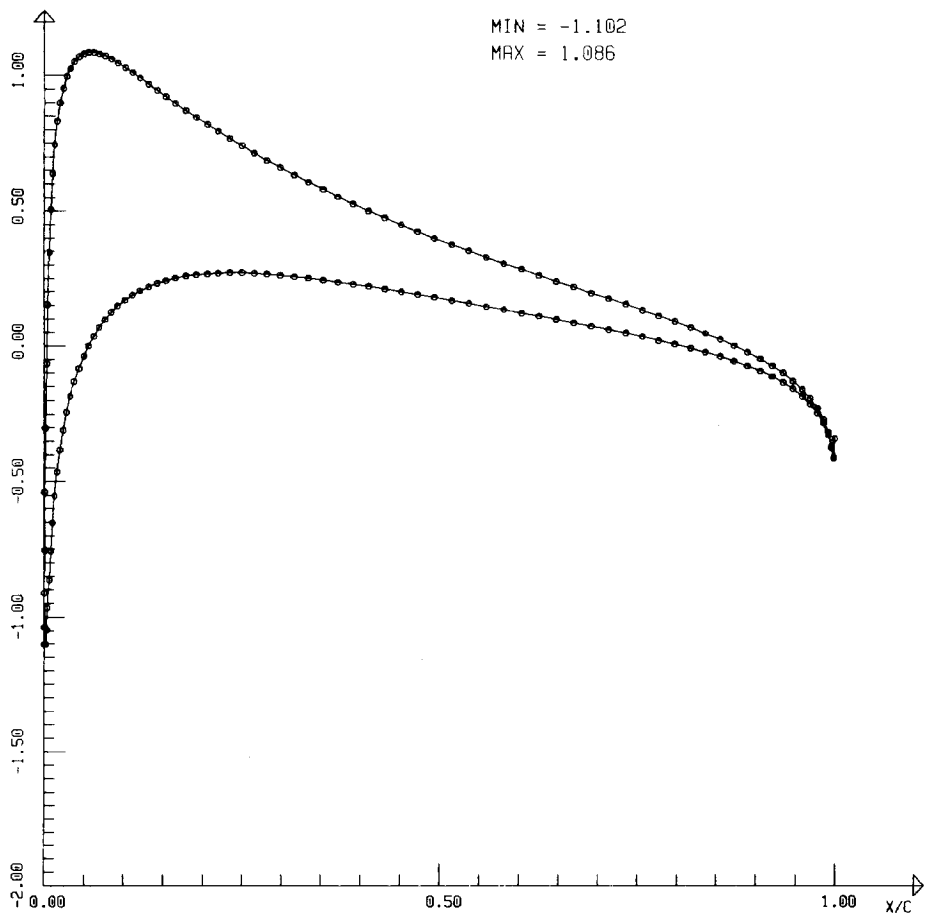

FIG. 6. Pressure coefficient distribution for NACA12 airfoil: $M_\infty = 0.63$; $\alpha = 2°$; CFL 50; 500 iterations.

(B_1) $M_\infty = 0.63$; $\alpha = 2°$; O-mesh 3200 nodes 6144 elements

(B_2) $M_\infty = 0.63$; $\alpha = 2°$; O-mesh 4800 nodes 10,240 elements

(B_3) $M_\infty = 0.85$; $\alpha = 1°$; O-mesh 3200 nodes 6144 elements

(B_4) $M_\infty = 2.00$; $\alpha = 10°$; O-mesh 1360 nodes 2560 elements

(B_5) $M_\infty = 0.85$; $\alpha = 1°$; 2492 nodes 4765 elements.

The unstructured and irregular mesh used in (B5) has been obtained from one of the above O-mesh (1360 nodes) by Palmerio's self-adaptive mesh refinement technique [21].

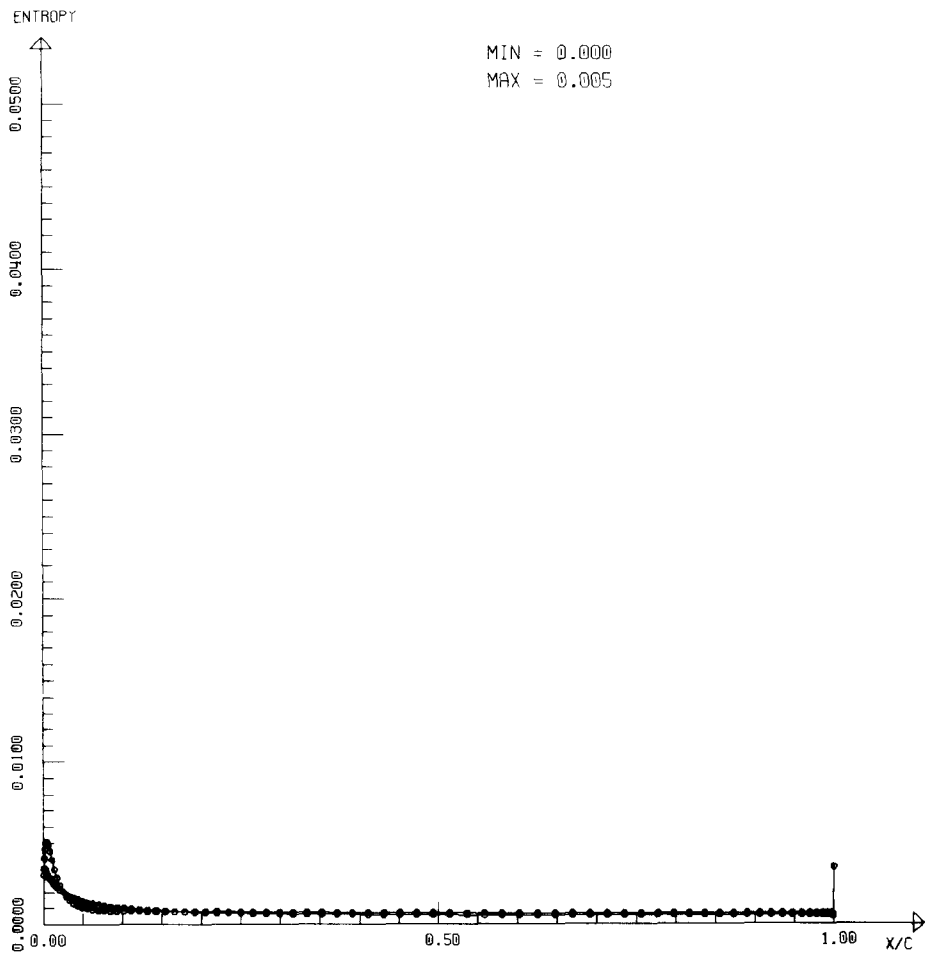

FIG. 7. Entropy distribution for NACA12 airfoil: $M_\infty = 0.63$; $\alpha = 2°$; CFL 50; 500 iterations.

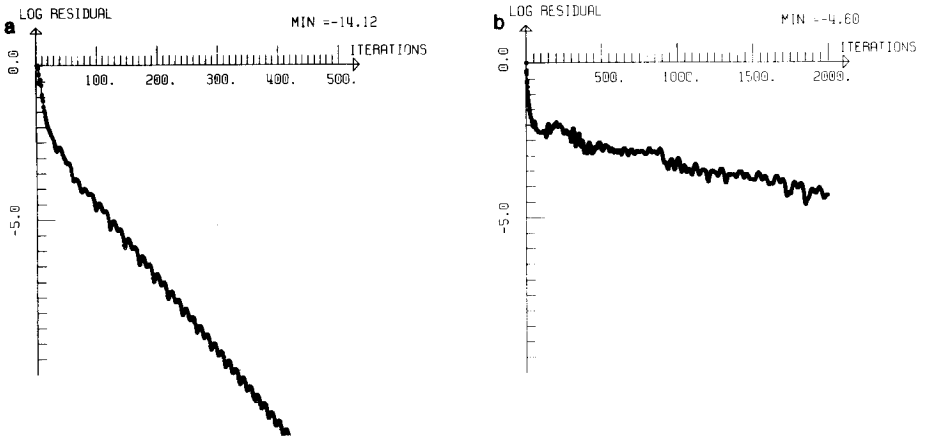

FIG. 8. Comparison of residual history for explicit and implicit methods for NACA12 airfoil at $M_\infty = 0.63$; $\alpha = 2°$ on the 3200 nodes mesh; CFL 50: CPU 98 min (a); 294 min (b).

In each test case, we compare explicit and implicit versions of the scheme with the same spatial approximation. For all computations, the reference explicit method is a four-step Runge–Kutta algorithm with coefficients $c_1 = 0.11$, $c_2 = 0.2766$, $c_3 = 0.5$, and $c_4 = 1.00$ used with a CFL number of 1.8.

The first two cases (B_1) and (B_2) considered are a subcritical flow around a NACA0012 airfoil at $M_\infty = 0.63$ and $\alpha = 2°$, computed with two O-meshes of 3200 nodes and 6144 elements for the first one and 4800 nodes and 10,240 elements for the second one which is presented on Fig. 5. The computed pressure coefficient and entropy distributions are shown on Figs. 6 and 7 for the second case. Convergence histories of residual for both methods are shown in Figs. 8 and 9. The average

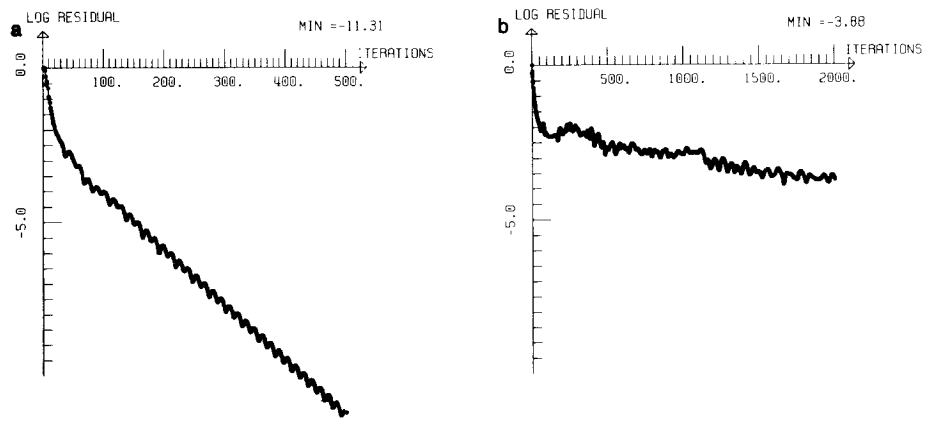

FIG. 9. Comparison of residual history for explicit and implicit methods for NACA12 airfoil at $M_\infty = 0.63$; $\alpha = 2°$ on the 4800 nodes mesh; CFL 50: CPU 176 min (a); 488 min (b).

number of relaxation sweeps at each iteration needed to reach the prescribed convergence level (10^{-3}) is about 40 for the first test case and 45 for the second one.

Lift coefficient C_l, drag coefficient C_d, and maximum of entropy level Σ_{max} obtained on each mesh are

(B_1) $C_l = 0.366$, $C_d = 0.00094$, $\Sigma_{max} = 0.0079$

(B_2) $C_l = 0.337$, $C_d = 0.00033$, $\Sigma_{max} = 0.0051$

which compare well with available results.

The efficiency ratio is more than 10 for both meshes; the efficiency of the implicit solver is preserved and even reinforced on the fine mesh.

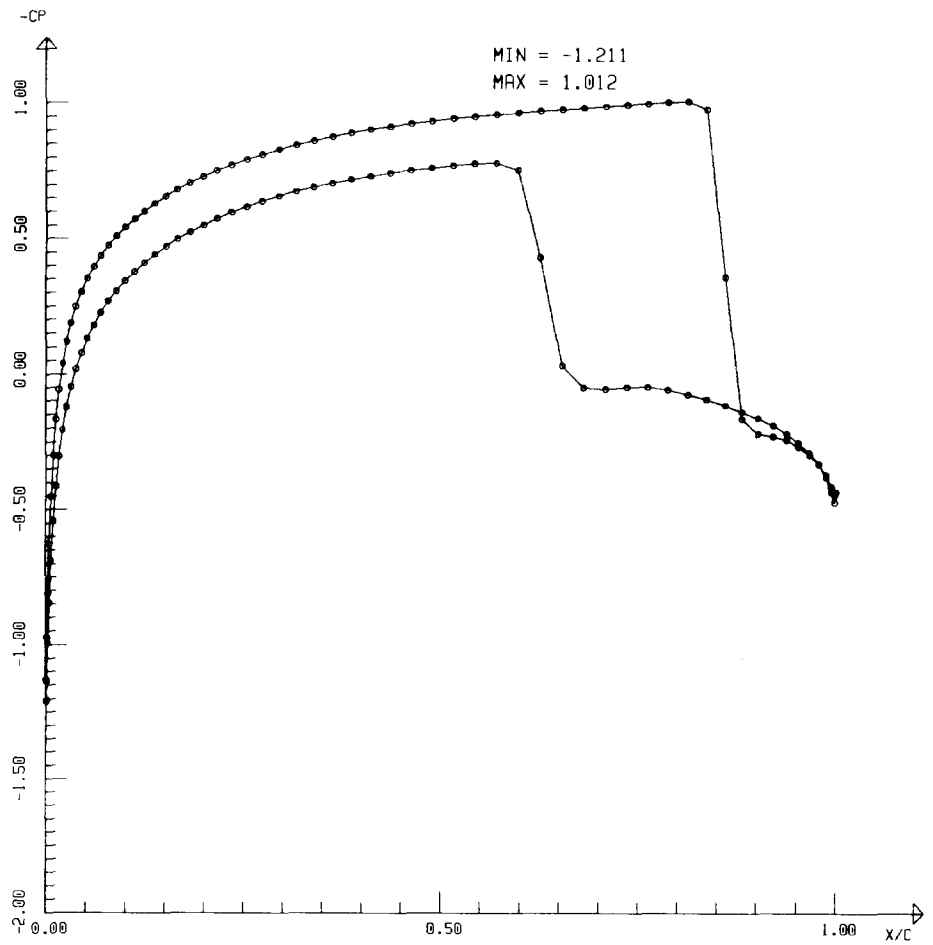

FIG. 10. Pressure coefficient distribution for NACA12 airfoil: $M_\infty = 0.85$; $\alpha = 1°$; CFL 50; 500 iterations.

The third test case (B_3) considered is a transonic flow around the same airfoil at $M_\infty = 0.85$ and $\alpha = 1°$, computed with the previous O-mesh of 3200 nodes and 6144 elements. The computed pressure coefficient and entropy distributions are shown on Figs. 10 and 11 and iso-Mach lines are plotted on Fig. 12. Convergence histories of residual for both methods are shown in Fig. 13.

Lift coefficient C_l, drag coefficient C_d, and pitching moment C_m are obtained without applying any correction at the farfield boundary:

(B_3) $C_l = 0.360$, $C_d = 0.0573$, $C_m = -0.1217$.

For this transonic test case, the residual is less than 10^{-10} after about 500 iterations which means about 5000 s on the IBM 3090 computer. The following test case (B_4)

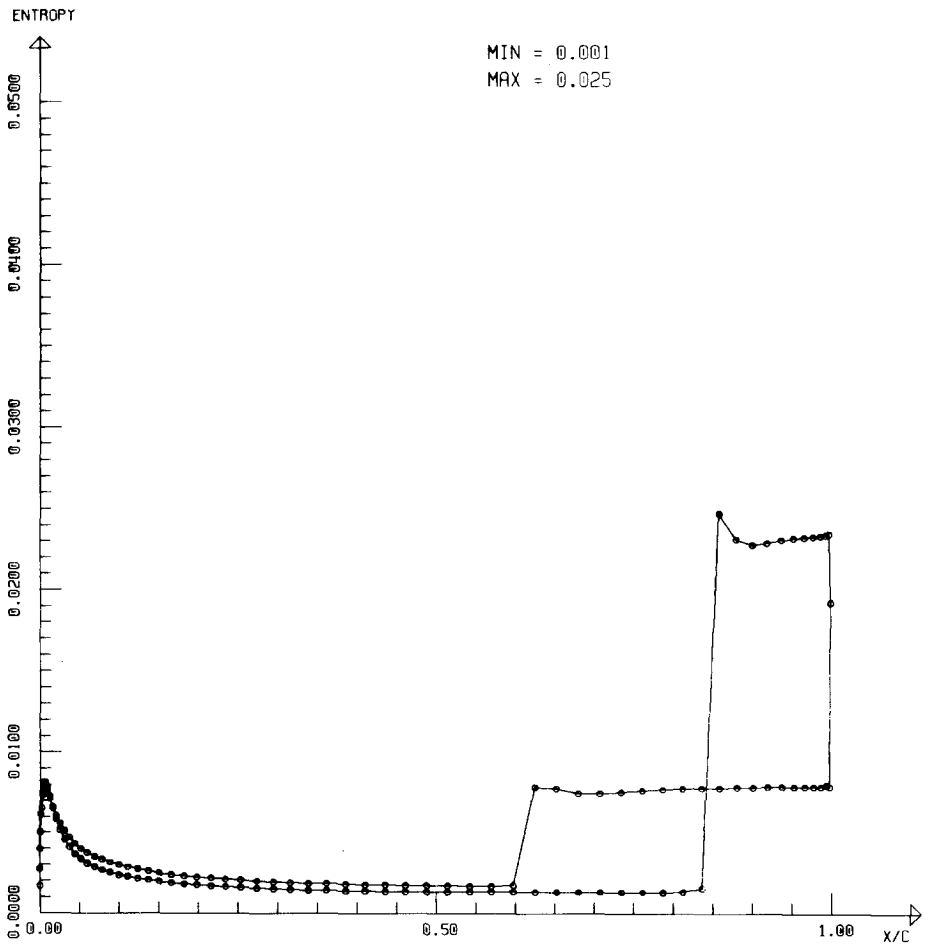

FIG. 11. Entropy distribution for NACA0012 airfoil: $M_\infty = 0.85$; $\alpha = 1°$; CFL 50; 500 iterations.

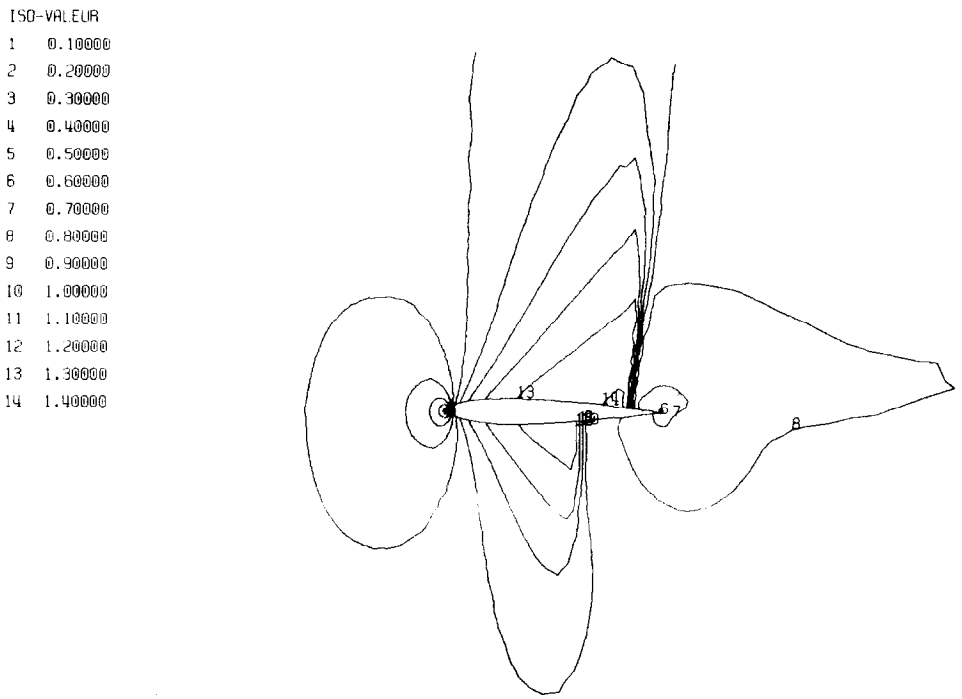

FIG. 12. Iso-Mach lines for NACA0012 airfoil: $M_\infty = 0.85$; $\alpha = 1°$; CFL 50; 500 iterations.

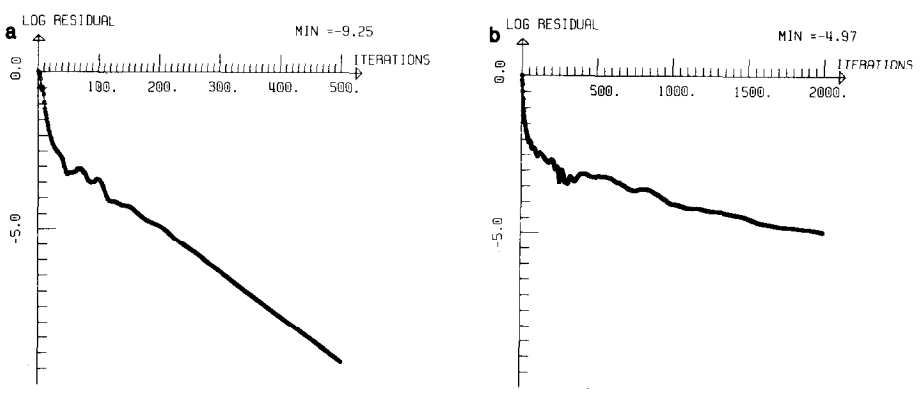

FIG. 13. Comparison of residual history for explicit and implicit methods for NACA12 airfoil at $M_\infty = 0.85$; $\alpha = 1°$ on the 3200 nodes mesh; CFL 50: CPU 84 min (a); 283 min (b).

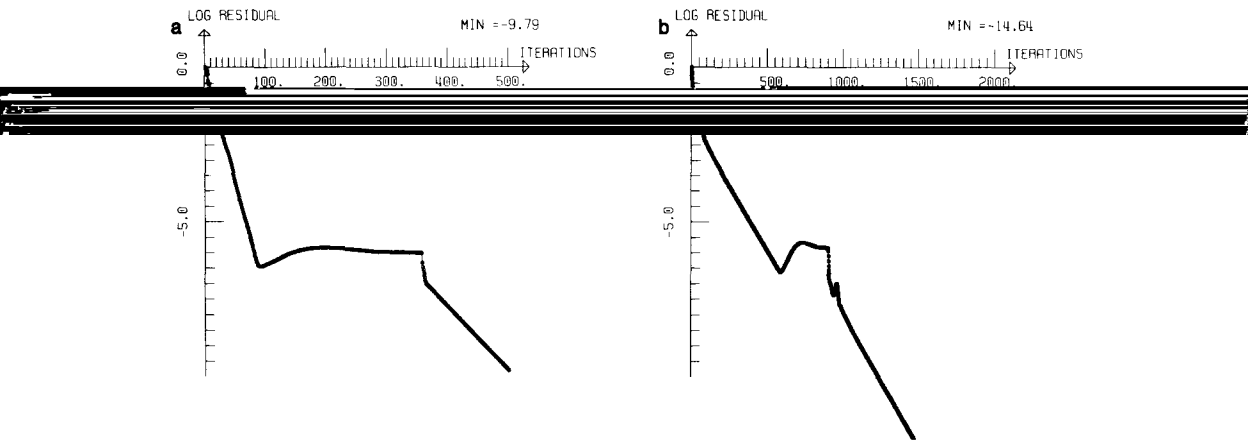

FIG. 14. Comparison of residual history for explicit and implicit methods for NACA12 airfoil at $M_\infty = 2.$; $\alpha = 10°$ on the 1360 nodes mesh; CFL 50: CPU 31 min (a); 150 min (b).

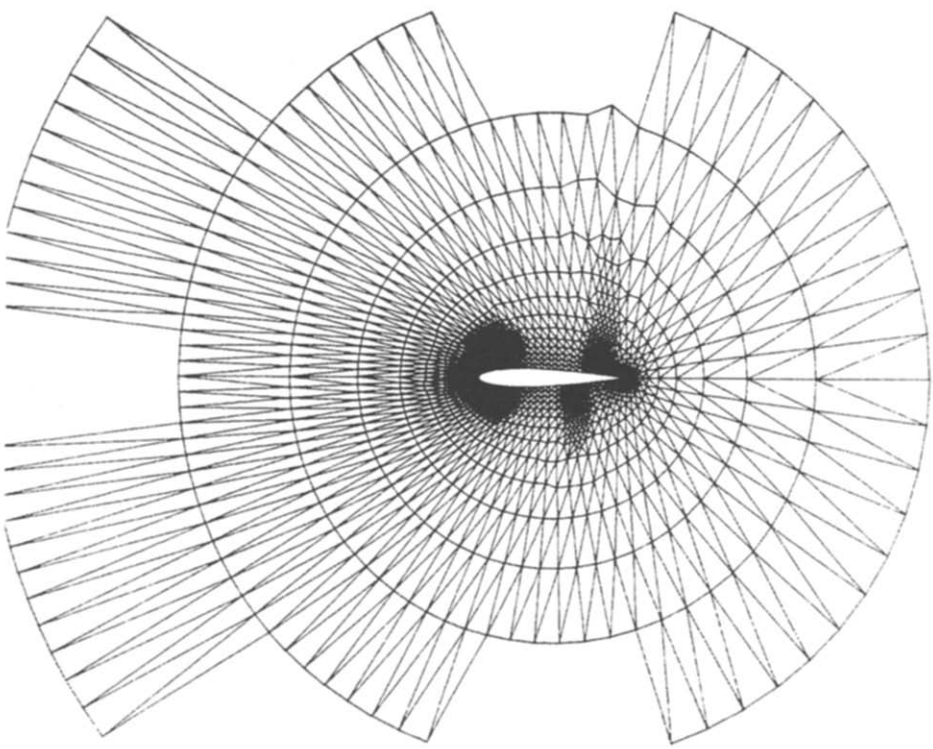

FIG. 15. Enlargement of the locally refined mesh.

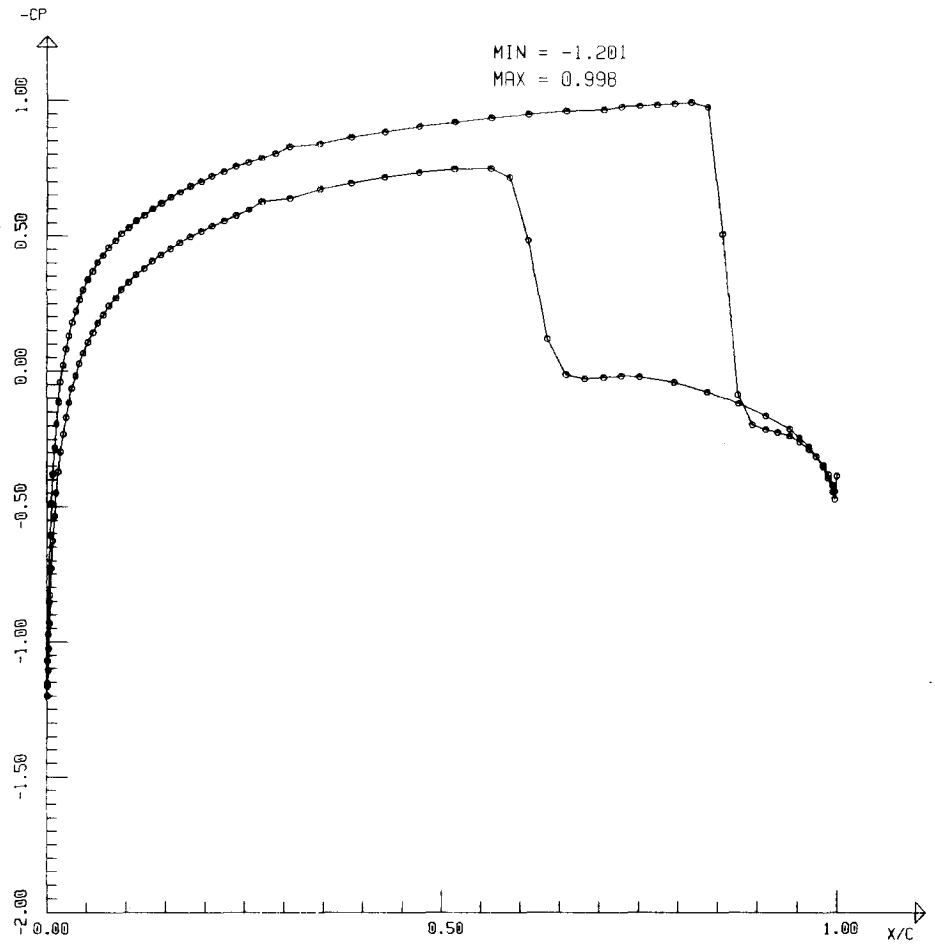

FIG. 16. Pressure coefficient distribution for NACA12 airfoil: $M_\infty = 0.85$; $\alpha = 1°$ on the refined mesh; CFL 50; 500 iterations.

is a supersonic flow around the same airfoil at $M_\infty = 2.$ and $\alpha = 10°$, computed on a quite coarse O-mesh of 1360 nodes and 2560 elements. We present only the history of convergence on Fig. 14 and we note that the efficiency rate of the implicit solver versus the explicit one for that problem is about 7.

The last problem (B_5) is the same as in (B_3) with a different mesh which is highly unstructured (Fig. 15). The pressure coefficient obtained on the body is shown on Fig. 16 and convergence histories for both methods on Fig. 17. Once again, the good efficiency of the implicit method is proved; its behaviour is not affected when using an unstructured grid. Aerodynamic coefficients obtained on this grid are:

(B_5) $C_l = 0.355$, $C_d = 0.0575$, $C_m = -0.1220$.

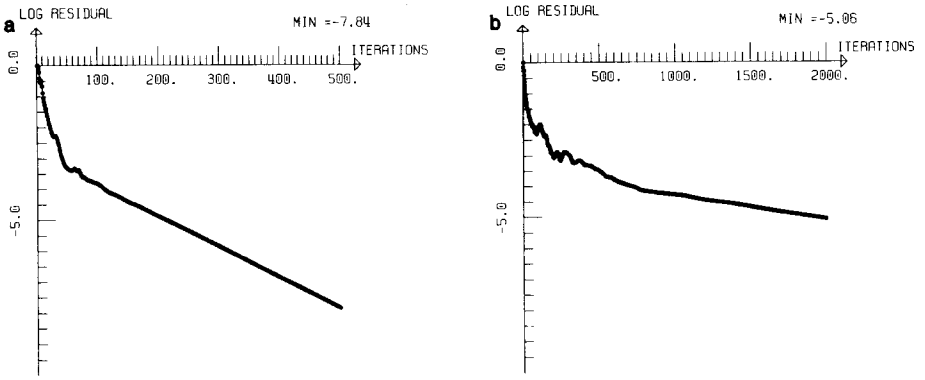

FIG. 17. Comparison of residual history for explicit and implicit methods for NACA12 airfoil at $M_\infty = 0.85$; $\alpha = 1°$ on the refined mesh; CFL 50: CPU 64 min (a); 290 min (b).

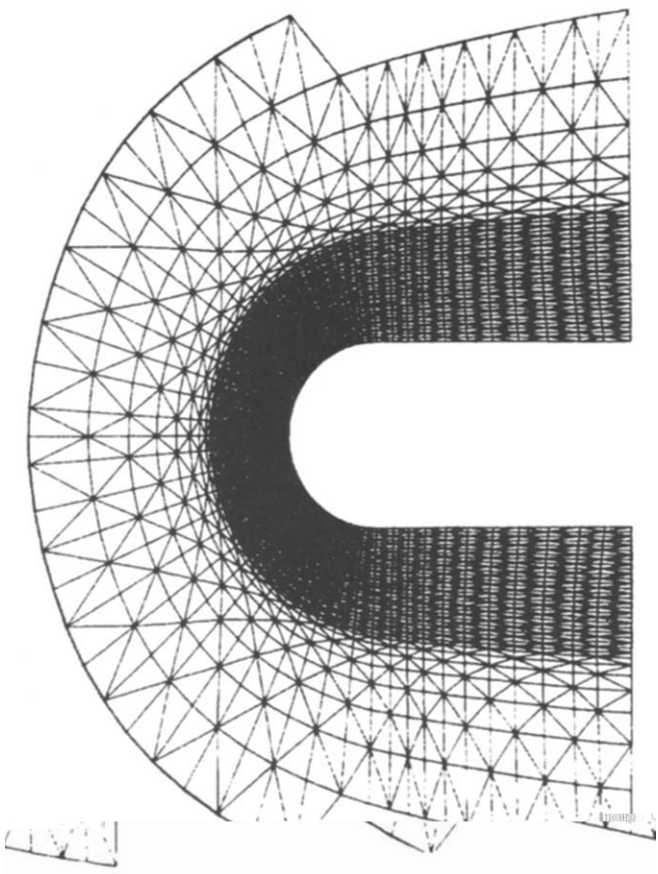

FIG. 18. Enlargement of the blunt body mesh.

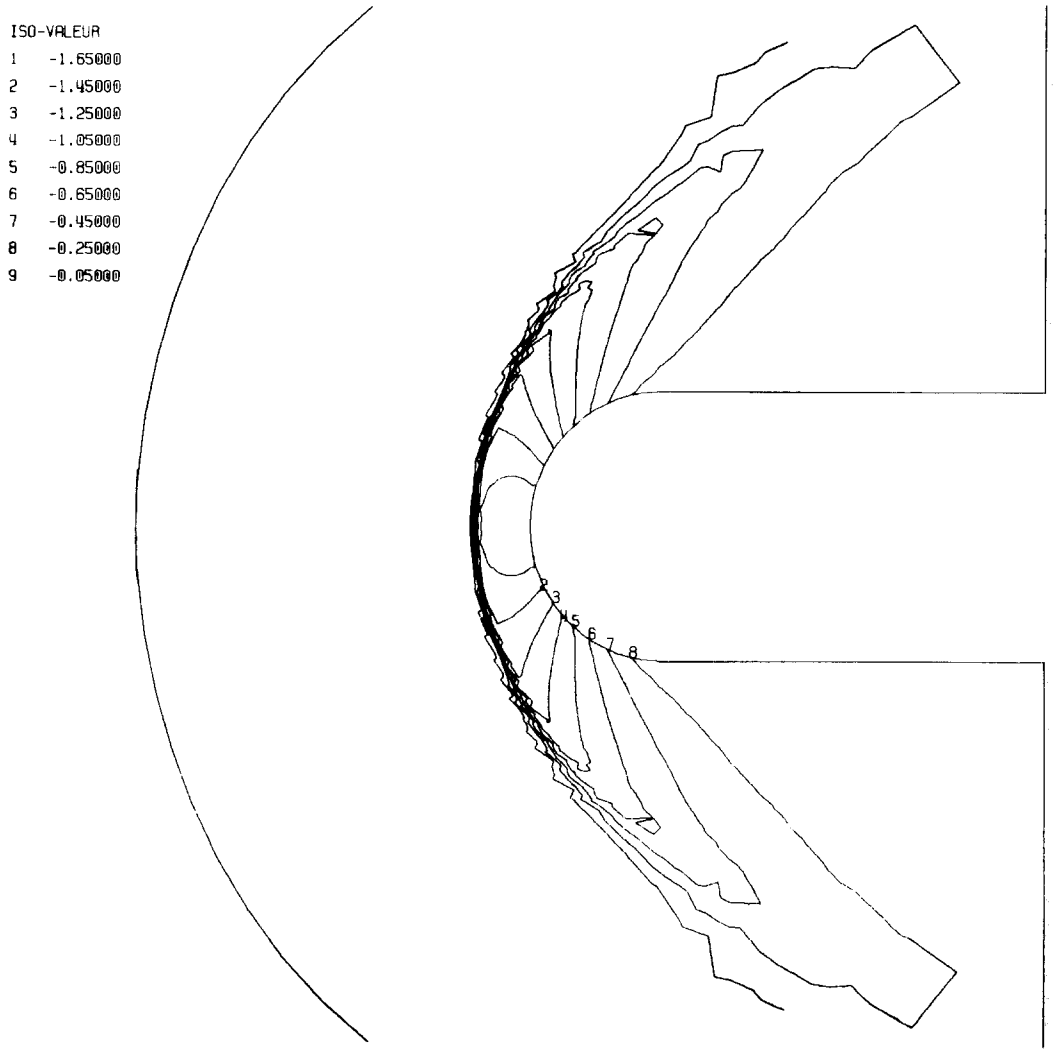

FIG. 19. Isopressure coefficient lines for blunt body: $M_\infty = 8.$; $\alpha = 0°$ ($\Delta C_p = 0.2$; the first line corresponds to the value -1.65); CFL 50; 285 iterations.

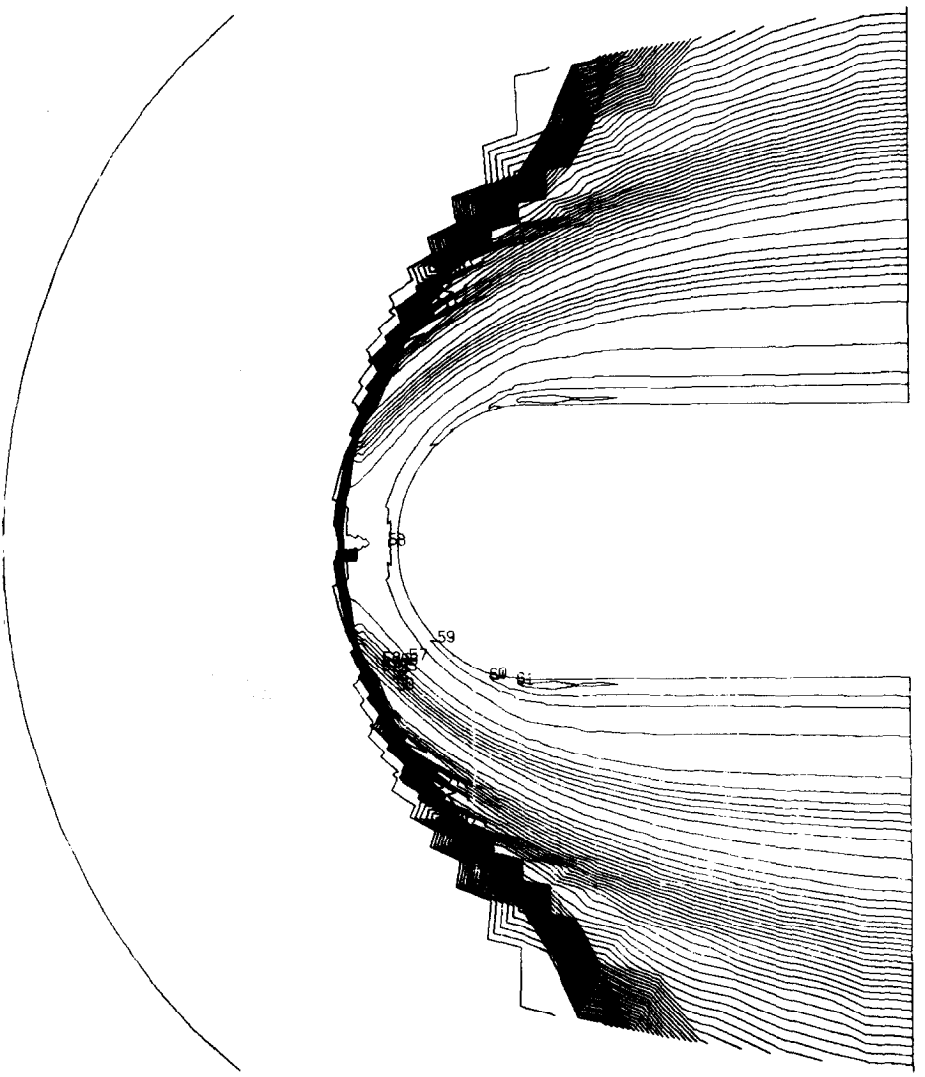

FIG. 20. Isoentropy lines for blunt body: $M_\infty = 8.$; $\alpha = 0°$ ($\Delta\Sigma = 0.1$; the first line corresponds to the value 0); CFL 50; 285 iterations.

In all computations, the efficiency ratio is between 7 and 12. We notice that the convergence is not altered when we use a second-order accurate approximation in the physical part of the scheme. This point argues for the use a first-order accurate preconditionner in the implicit phase of the scheme.

The results of this test case set off two other qualities of the method:

—First, the efficiency of the implicit solver is conserved when computing flows with a high angle of attack (cases (B_2) and (B_3)) which is important for future 3-dimensional computations on real configurations.

—Second, the use of unstructured grids obtained by mesh refinement does not alter either the efficiency of the implicit procedure. It was important to check that these designed algorithms combine well with all adaptive capabilities of finite element techniques.

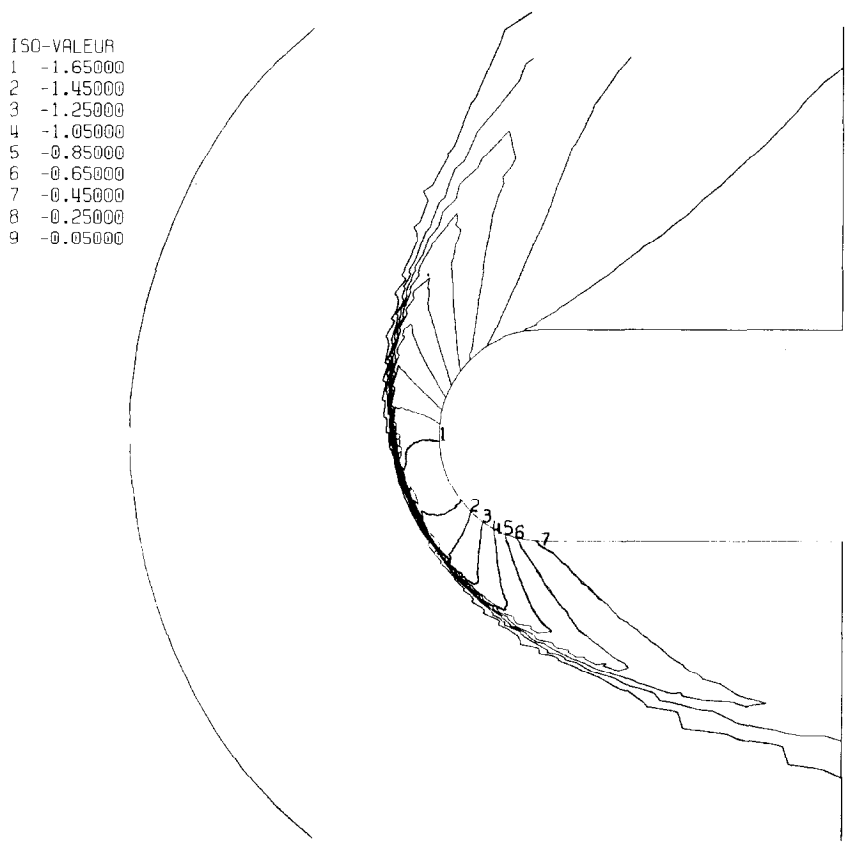

FIG. 21. Isopressure coefficient lines for blunt body: $M_\infty = 8.$; $\alpha = 20°$ ($\Delta C_p = 0.2$; the first one corresponds to the value -1.65); CFL 1.8; 2000 iterations.

C. *High Speed Flow Past a Blunt Body*

We are now concerned with steady hypersonic flow calculations around a blunt body with a circular cylinder as a nose. The difficulty in this problem remains in the robustness of the algorithms to "support" the displacement of the bow shock during the transient phase. All the computations were performed with a mesh of 1908 nodes and 3640 elements and is presented in Fig. 18.

Two test cases are presented for the same farfield Mach number of 8 and different angles of attack ($\alpha = 0°, 20°$). For the first test-case, a three-step explicit method was performed but did not converge for the second test-case and was replaced by the four-step one used for (B) problems. The linear system of the implicit method

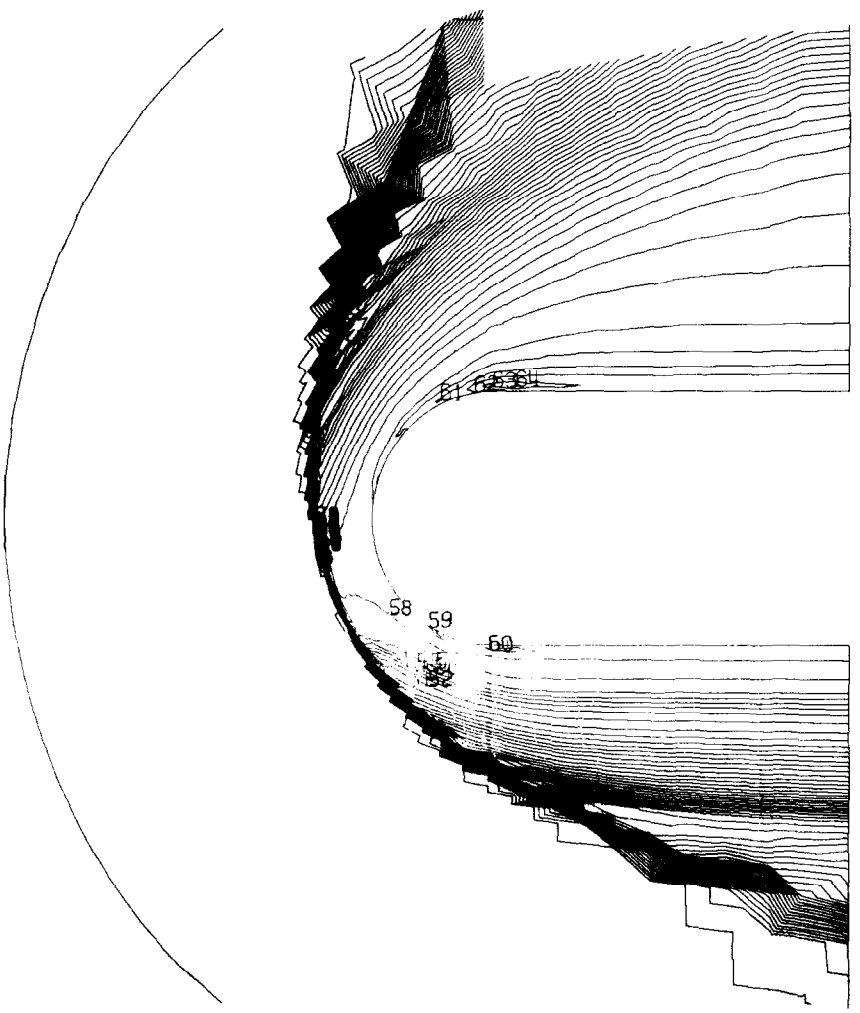

FIG. 22. Isoentropy lines for blunt body: $M_\infty = 8.$; $\alpha = 20°$ ($\Delta\Sigma = 0.1$; the first one corresponds to the value 0); CFL 1.8; 2000 iterations.

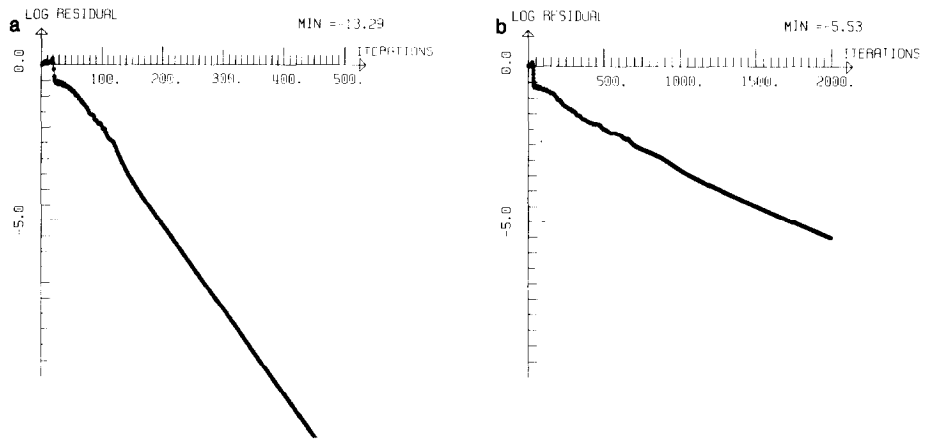

FIG. 23. Comparison of residual history for explicit and implicit methods for blunt body at $M_\infty = 8.$; $\alpha = 0°$: CFL 50, CPU 30 min (a); CFL 1.8, CPU 144 min (b).

derived from the Steger–Warming splitting is solved by the collective Gauss–Seidel relaxation described earlier. The resolution is stopped when the convergence rate is less than 10^{-3} which means an average number of sweeps of about 10 for these problems.

Isovalues for pressure coefficients and entropy distributions are presented in Figs. 19 and 20 for test case (C1) and Figs. 21 and 22 for test case (C2).

The results show a good behaviour of the implicit solver with efficiency ratios of about 8 for both test cases (Figs. 23 and 24 show the convergence histories of the implicit versus the explicit method for both cases). For larger values of α ($\alpha > 20°$), the explicit version does not converge anymore when starting from the uniform flowfield; however, the convergence can be assumed by incrementation of α. On the

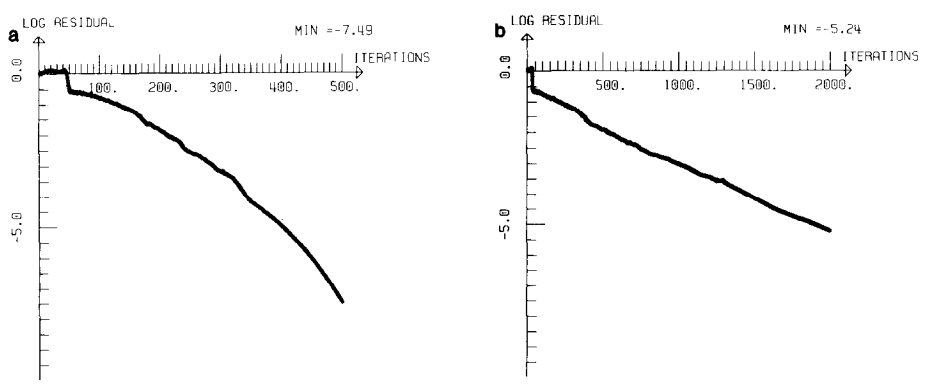

FIG. 24. Comparison of residual history for explicit and implicit methods for blunt body at $M_\infty = 8.$; $\alpha = 20°$: CFL 50, CPU 31 min (a); CFL 1.8, CPU 206 min (b).

other hand the implicit solver still converges for any value of α. The results show that implicit solvers seem very well adapted and very efficient to compute steady hypersonic flows.

CONCLUSION

The numerical results show a very clear gain in efficiency for the implicit scheme compared to its explicit version, for all flow regimes.

With the first-order scheme we have a nearly quadratic convergence (without the exact computation of the Jacobian) when the initial solution is closed to the steady solution (test case (A)).

When using a second-order accurate approximation the method is no longer quadratically converging but still very efficient. It appears that the incomplete solution of the linear system by relaxation is an efficient procedure and must be retained for 3D computations.

The main quality of these implicit schemes is their greater reliability with respect to the explicit version. This is clearly shown in the case of high Mach number regimes and large angles of attack, revealing the great robustness and efficiency of the implicit version of the schemes.

For the numerical study of spatial approximations, the methodology is particularly attractive because of its modular structure; we can change for instance the numerical flux in the explicit phase of the scheme without any further program changes. We note however that the method is not yet adapted to heavy industrial use since for instance, we have to store 2D matrices, but a study of this problem is in progress. A new version of the algorithm without any storage of matrices and which can be vectorized is under development.

ACKNOWLEDGMENTS

The authors wish to thank A. Dervieux for his many helpful suggestions and guidance of this paper. This work was supported in part under DRET Contract 84/014.

REFERENCES

1. W. K. ANDERSON, J. L. THOMAS, AND B. VAN LEER, AIAA Paper No. 85-0122, 1985 (unpublished).
2. F. ANGRAND AND A. DERVIEUX, *Int. J. Numer. Methods Fluids* **4**, 749 (1984).
3. F. ANGRAND, V. BOULARD, A. DERVIEUX, J. PERIAUX, AND G. VIJAYASUNDARAM, AIAA Paper No. 83-1924, 1983 (unpublished).
4. F. ANGRAND, V. BOULARD, A. DERVIEUX, J. A. DESIDERI, J. PERIAUX, AND B. STOUFFLET, in *Proceedings, Ninth International Conference on Numerical Methods in Fluid Dynamics, Saclay, France*, 1984, edited by Soubbaramayer and Boujot, Lecture Notes in Physics, Vol. 218 (Springer-Verlag, 1985).

5. F. ANGRAND, V. BOULARD, A. DERVIEUX, J. PERIAUX, AND G. VIJAYASUNDARAM, in *Computing Methods in Applied Sciences and Engineering*, edited by R. Glowinski *et al.* (North-Holland, Amsterdam, 1984).
6. F. ANGRAND, V. BILLEY, A. DERVIEUX, J. PERIAUX, AND B. STOUFFLET, AIAA Paper No. 85-1706, 1985 (unpublished).
7. R. M. BEAM AND R. F. WARMING, *J. Comput. Phys.* **22** (1976).
8. F. CASIER, H. DECONINCK, AND C. HIRSCH, AIAA Paper No. 83-0126, 1983 (unpublished).
9. A. DERVIEUX, Lectures Series 1884-04, Von Karman Institute for Fluid Dynamics, March 26–29, 1985 (unpublished).
10. F. FEZOUI, INRIA Report, 1986 (unpublished).
11. F. FEZOUI, B. STOUFFLET, J. PERIAUX, AND A. DERVIEUX, in *Proceedings, 4th International Symposium on Numerical Methods in Engineering, Atlanta, Georgia*, 1986.
12. S. L. HANCOCK, private communication, reported by Van Albada *et al.* in Ref. [29].
13. A. HARTEN, NYU Report, 1982 (unpublished).
14. A. JAMESON, MAE Report 1603, 1983 (unpublished).
15. D. C. JESPERSEN AND T. H. PULLIAM, "Flux Vector Splitting and Approximate Newton Methods," AIAA Computational Fluid Dynamics Conference, Danvers, July, 1983 (unpublished).
16. A. LERAT, J. SIDES, AND V. DARU, in *Proceedings, Eighth International Conference on Numerical Methods in Fluid Dynamics, Aachen*, 1982, edited by E. Krause, Lecture Notes in Physics, Vol. 170 (Springer-Verlag, New York/Berlin, 1982).
17. R. W. MACCORMACK, *AIAA J.* **20** (1985).
18. J. L. MONTAGNE, in *Proceedings, Ninth International Conference on Numerical Methods in Fluid Dynamics, Saclay, France*, 1984, edited by Soubbaramayer and Boujot, Lecture Notes in Physics, Vol. 218 (Springer-Verlag, New York/Berlin, 1985).
19. W. A. MULDER AND B. VAN LEER, AIAA Paper No. 83-1930, 1983 (unpublished).
20. S. OSHER AND F. SALOMON, *Math. Comput.* **22** (1976).
21. B. PALMERIO, INRIA Report 338, 1984 (unpublished).
22. M. M. RAI AND S. R. CHAKRAVARTHY, AIAA Paper No. 84-0088, 1984 (unpublished).
23. A. RIZZI AND H. VIVIAND, "Numerical methods for the computation of inviscid transonic flow with shock waves," in *Notes on Numerical Fluid Dynamics*, Vol. 3 (Wieweg, Braunschweig/Wiesbaden, 1981).
24. J. STEGER AND R. F. WARMING, *J. Comput. Phys.* **40**, 2 (1981).
25. B. STOUFFLET, thesis, University of Paris VI, 1984 (unpublished).
26. B. STOUFFLET, in *Proceedings, INRIA Workshop on Numerical Methods for Compressible Inviscid Fluids, Rocquencourt, France*, 1983, edited by F. Angrand *et al.* (SIAM, Philadelphia, 1984), p. 409.
27. J. L. THOMAS, B. VAN LEER, AND R. W. WALTERS, in *Proceedings, AIAA 19th Fluid Dynamics Plasma and Phys. Lasers Conf., Cincinnati, Ohio*, 1985.
28. E. TURKEL AND B. VAN LEER, ICASE Report No. 84–27, 1984 (unpublished).
29. G. D. VAN ALBADA, B. VAN LEER, AND W. W. ROBERTS, *J. Astron. Astrophys.* **108** (1982).
30. B. VAN LEER, *J. Comput. Phys.* **23**, 263 (1977).
31. G. VIJAYASUNDARAM, *J. Comput. Phys.* **63**, 416 (1986).
32. H. C. YEE, R. F. WARMING, AND A. HARTEN, NASA Technical Memorandum SU342, 1983 (unpublished).
33. H. C. YEE, *Comut. Math. Appl. A* **12**, 413 (1986).