A class of implicit upwind schemes for euler simulations with unstructured meshes

Loula Fatima Fezoui, Bruno Stoufflet

CENTRE
SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 517

A CLASS OF IMPLICIT UPWIND SCHEMES FOR EULER SIMULATIONS WITH UNSTRUCTURED MESHES

Fatma FEZOUI
Bruno STOUFFLET

Avril 1986

A CLASS OF IMPLICIT UPWIND SCHEMES FOR EULER SIMULATIONS WITH UNSTRUCTURED MESHES

BY F. FEZOUI(*) and B. STOUFFLET(**)

(*) INRIA Sophia-Antipolis, Route des Lucioles,
06560 VALBONNE (FRANCE)

(**) AMD-BA, B.P. 300, 92214 SAINT-CLOUD (FRANCE)

ABSTRACT :

A class of implicit upwind schemes for solving Euler equation 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 explicit 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.

RESUME :

Nous présentons dans ce papier une classe de schémas de résolution des équations d'Euler pouvant simuler des écoulements stationnaires a grand nombre de Mach et à forte incidence dans des géométries complexes (utilisation de maillages non structurés).
L'approximation spatiale est réalisée au moyen de schémas décentrés.
L'accélération de la convergence est obtenue par un schéma implicite reposant sur une linéarisation des flux et une résolution incomplète des systèmes par la relaxation de Gauss-Seidel.
Le schéma résultant est plus efficace, plus robuste et plus fiable que celui obtenu avec une intégration explicite en temps.

1. - INTRODUCTION

This paper follows the works of Dervieux et al. [2], [6], dealing with Euler flow simulation in complex geometries, such as aircraft, in transonic and supersonic regimes. For this purpose it is interesting and important to build schemes not relying too strongly on the regularity of 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, thus also affecting the accuracy and stability of the scheme. With unstructured meshes, the splitting of matrices along the x, y directions may be irrelevant, and the fully multidimensional matrices must then be used.

Using such grids in combination with explicit schemes, leads to strict limitations on the time-step hence, a large computing time needed to get to the steady state. Implicit solvers which permit large time-steps and CFL numbers lead to a significant decrease in computing time.

Many authors have developped the implicit methods.

Beam and Warming [7], among the earliest, have given an important contribution with their implicit difference schemes.

Lerat [14] built a class of centered second-order accurate implicit difference schemes, containing that of Beam and Warming.

Two implicit conservative and non-conservative versions of Harten's [12] scheme are presented by Yee, Harten and Warming [30], and recently Yee [31] gave very 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 [22] leads to a *triangular* linear system.

The scheme proposed by Mac Cormack [15] is non-centered and the linear systems are *bidiagonal*. This scheme was also studied by Casier, Deconinck and Hirsh {8} in a one dimensional context.

Rai and Chakravarthy [20] have presented an implicit version of the second-order finite difference scheme of Osher [18] where the linear systems are solved by a *relaxation* method.

Mulder and Van Leer [17] present an implicit upwind difference scheme for the one-dimensional Euler equations. They use upwind spatial differencing and *linearization* in time. This method is extended to the two-dimensional case in generalized coordinates by Van Leer et al. {1}, {25}. Two implicit solvers are described : a *factorization* method and a *linear relaxation*.

The implicit method proposed by Stoufflet [23] applies to *unstructured* meshes ; it involves a *linearization* of the first-order upwind scheme of Vijayasundaram [29] and a *relaxation iteration*. This method can be used to get an implicit version of a scheme, whether it be centered or upwind.

Our purpose is to extend this method to a large class of *upwind* fluxes of first or second-order accuracy, especially adapted to *unstructured meshes*.

Two basic ingredients are used in the present work.

Firstly, 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 [28] . 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 {29} and Fezoui {10} and to fully two-dimensional finite difference meshes by Montagné {16}.

Secondly, 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 specially successful to compute flows with extreme limits : large Mach number regimes, large angles of attack and very irregular meshes . The usual explicit schemes often fail when applied to such cases.

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 two-dimensional unstructured meshes.

Numerical experiments with different meshes and flux functions are presented.

2. - THE ONE-DIMENSIONAL STUDY

2.1. - The first-order implicit scheme

We consider a one-dimensional system of conservation laws :

$$W_t + F(W)_x = 0 \tag{1}$$

where W and F are m-component column vectors.

The system (1) can be rewritten as a quasi-linear system :

$$W_t + A(W)\, W_x = 0 \tag{2}$$

where A is the Jacobian matrix F_w.

Let us assume that the system (2) is hyperbolic i.e. : for each (x,t,W) there exists a similarity transformation T such that :

$$T^{-1}\, A\, T = \Lambda$$

where Λ is a real diagonal matrix carrying the eigenvalues of A.

We can write :

$$A = A^+ - A^-$$

$$|A| = A^+ + A^-$$

$$A^{\pm} = T^{-1}\, \Lambda^{\pm}\, T$$

where : $\Lambda^{\pm} = \mathrm{diag}(\lambda^{\pm})$; $\lambda^+ = \max(\lambda,0)$

$\lambda^- = \min(\lambda,0)$.

Moreover, we assume the system (1) to be homogeneous in the sense :

$$F(W) = A(W).W, \quad \forall W \in \mathbb{R}^m$$

Applied to system (1), a family of 3-point explicit conservative schemes is given by :

$$(3)\qquad \begin{cases} W_i^{n+1} - W_i^n + \sigma(\Phi_{i+1/2}^n - \Phi_{i-1/2}^n) = 0, \\ \Phi_{i+1/2}^n = \Phi(W_i^n, W_{i+1}^n), \\ \Phi_{i-1/2}^n = \Phi(W_{i-1}^n, W_i^n), \\ \sigma = \Delta t/\ \Delta x\ ;\ W_i^n = W(n\Delta t,\ i\Delta x)\ ; \end{cases}$$

where Δt and Δx are respectively, the time-step and the space-step.

Given in the form (3), these schemes differ only by their numerical flux function Φ which we require to be consistent with the physical flux in the sense $\Phi(W,W) = F(W)$.

Let us recall some first-order upwind flux functions, which will be used in the sequel.

<u>Van Leer's Q-scheme</u> [28] :

$$(4)\qquad \Phi^{VL}(U,V) = \{F(U) + F(V) - |A((U+V)/2)|\ (V-U)\}\ /\ 2.$$

<u>Vijayasundaram's scheme</u> [29]

$$(5)\qquad \Phi^{VS}(U,V) = A^{+}((U+V)/2)U + A^{-}((U+V)/2)V.$$

<u>Steger and Warming's scheme</u> [22]

$$(6)\qquad \Phi^{SW}(U,V) = A^{+}(U)U + A^{-}(V)V.$$

<u>Osher's scheme</u> [18]

$$(7)\qquad \Phi^{OS}(U,V) = \{F(U) + F(V) - \int_U^V |A(w)|\ dw\}\ /\ 2.$$

In the cases (4), (5) and (6) the function Φ may be rewritten :

$$(8)\qquad \Phi(U,V) = H_1(U,V)U + H_2(U,V)V$$

where H_1 and H_2 are m × m matrices.

For all these schemes, except Osher's, the flux function does not satisfy the following assumption :

(9) $\Phi(U,V)$ is differentiable w.r.t. (U,V).

Let us assume that (9) is true, then we can construct the Newton-like linearized implicit version of the scheme (3) :

$$(10)\quad \begin{cases} W^{n+1} - W^{n} + \sigma(\Phi^{N}_{i+1/2} - \Phi^{N}_{i-1/2}) = 0 \\ \Phi^{N}_{i+1/2} = \Phi^{N}(W^{n}_{i}, W^{n}_{i+1}, W^{n+1}_{i}, W^{n+1}_{i+1}) \\ \Phi^{N}_{i-1/2} = \Phi^{N}(W^{n}_{i-1}, W^{n}_{i}, W^{n+1}_{i-1}, W^{n+1}_{i}) \\ \Phi^{N}(U,V,W,Z) = \Phi(U,V) + \Phi_{U}(U,V)(W-U) + \Phi_{V}(U,V)(Z-V) \end{cases}$$

For Δt tending to infinity, scheme (9) becomes Newton's method for finding the stationary solution of scheme (3).

The derivatives of Φ may be very expensive to compute, so we introduce the simplified linearized version of (10) :

$$(11)\quad \begin{cases} W^{n+1}_{i} - W^{n}_{i} + \sigma(\Phi^{S}_{i+1/2} - \Phi^{S}_{i-1/2}) = 0, \\ \Phi^{S}_{i+1/2} = \Phi^{S}(W^{n}_{i}, W^{n}_{i+1}, W^{n+1}_{i}, W^{n+1}_{i+1}), \\ \Phi^{S} = \Phi^{S}(W^{n}_{i-1}, W^{n}_{i}, W^{n+1}_{i-1}, W^{n+1}_{i}), \\ \Phi^{S}(U,V,W,Z) = H_{1}(U,V)W + H_{2}(U,V)Z. \end{cases}$$

The resulting scheme is in fact a modified Newton's method where the exact Jacobians arising in (10) are replaced by simpler expressions.

Proposition (1) : Under the assumption (8), schemes (10) and (11) have the same equivalent system up to the second-order.

Proposition (2) : In the scalar linear case, schemes (10) and (11) are identical and unconditionally stable.

The proof of the above propositions can be found in [23].

Remark (1) : We cannot ensure that scheme (11) 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 efficiencyfor the two schemes for large Δt if the unknowns do not vary too much (as it is the case at convergence to steady-state).

Although schemes (4), (5) and (6) do not satisfy the assumption of differentiable fluxes (9), we propose to apply the linearization method to get implicit versions of these schemes. The first-order implicit version in each case given below, except (7), is then obtained by using in (11) the corresponding numerical flux function Φ.

Such iterative methods considered as modified Newton's method have been analyzed by Jespersen and Pulliam [13]. They 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.

One step of scheme (11) leads to a linear system of the form :

$$M(W^n).(W^{n+1} - W^n) = B(W^n)$$

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, see [22].

2.2. - Second-order extension

2.2.1. - The second order fluxes

We derive the second-order fluxes from the first-order ones by following the procedure given by Van Leer [28], with any numerical flux function given in (3).

Let us recall briefly that the method consists in a linear interpolation of W over interval $]i-1/2;i+1/2[$ which will be called the cell C_i.

The approximation becomes :

$$W(x) = W_i + (x-x_i)\,P_i, \quad \text{for } x \in C_i$$

$$P_i = (W_{i+1} - W_{i-1}) \,/\, 2\Delta x \;;$$

we compute the values of W at the boundary of the cell by :

$$W^{+}_{i-1/2} = W_i - P_i \frac{\Delta x}{2},$$

$$W^{-}_{i+1/2} = W_i + P_i \frac{\Delta x}{2}.$$

Then the fluxes are computed with the values of W at the interfaces of the cells.

The full second-order accurate version of (3) is then given by :

$$(12) \qquad \left\{ \begin{aligned} & W^{n+1}_i - W^n_i + \sigma(\Phi^n_{i+1/2} - \Phi^n_{i-1/2^{+}}) = 0 \\ & \Phi^n_{i+1/2} = \Phi(W^n_{i+1/2^{-}}, W^n_{i+1/2^{+}}) \\ & \Phi^n_{i-1/2} = \Phi(W^n_{i-1/2^{-}}, W^n_{i+1/2^{+}}) \end{aligned} \right.$$

Remark (2) : Scheme (12) is stable for Courant numbers unduly small.

However. explicit versions which are linearly stable can be build by using a two-step scheme (Hancock-Van Leer (11)] or a Runge-Kutta method (Turkel-Van Leer (26)].

2.2.2. - Implicit step

An efficient way yielding second-order accurate steady-state solutions while keeping the interesting properties of the first-order upwind matrix is to replace the right-hand side of (11).
by a second-order accurate spatial approximation as defined in (12).

We can present the resulting algorithm as a two-phases scheme :

A) Physical/explicit/second-order accurate phase

(13)
$$\begin{cases} \delta W_i^n = -\sigma(\Phi_{i+1/2}^n - \Phi_{i-1/2}^n) \\ \Phi_{i\pm 1/2}^n \quad \text{given by (12).} \end{cases}$$

B) Mathematical/implicit/first-order accurate phase :

Since we want to get the steady-state, we are not interested in time accuracy, so we choose in the implicit phase of the scheme a first-order flux which leads to a simpler system to solve than the second-order one.

Thus this step is the same as the first-order scheme; it can be rewritten in a delta formulation :

(14)
$$\begin{cases} \delta W_i^n + \sigma(\Phi_{i+1/2}^S - \Phi_{i-1/2}^S) = \delta W_i^n \\ \Phi_{i+1/2}^S = \Phi^S(W_i^n, W_{i+1}^n, W_i^{n+1}, W_{i+1}^{n+1}) \\ \Phi_{i-1/2}^S = \Phi^S(W_{i-1}^n, W_i^n, W_{i-1}^{n+1}, W_i^{n+1}) \\ \Phi^S(U,V,W,Z) = H_1(U,V)(W-U) + H_2(U,V)(Z-V) \\ \delta W_i^n = W_i^{n+1} - W_i^n \end{cases}$$

We can choose in (14) any flux function that can be rewritten in a $H_1 - H_2$ formulation.

We can use a different numerical flux in each phase ; however, in the first-order case, it is more convenient from the point of vue of efficiency to use the same flux in both phases.

The following properties are easy to prove :

Proposition (3) : In the scalar case, the scheme defined by (13) and (14) is unconditionally linearly 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. [25], although they point out in this paper that the splitted fluxes used in the explicit phase should be continuously differentiable.

3. - THE 2-D EXTENSION

3.1. - The equations

We consider the two-dimensional Euler equations given in a conservative form :

$$(1)\qquad \begin{cases} W_t + F(W)_x + G(W)_y = 0 & \text{in a bounded domain } \Omega \\ W = {}^t(\rho,\rho u,\rho v,\rho e) \ ; & \\ F(W) = {}^t(\rho u,\ \rho u^2+p,\ \rho uv,\ (e+p)u) & \\ G(W) = {}^t(\rho v,\ \rho uv,\ \rho v^2+p,\ (e+p)v) \ ; & \\ p = (\gamma-1)(e - \rho(u^2+v^2)/2) \ ; \ \gamma = 1.4 & \end{cases}$$

where ρ, p and e are respectively the density, the pressure and the total energy per unit of mass, u and v are the components of the velocity.

We derive from (1) the quasi-linear system :

$$(2)\qquad \begin{cases} W_t + A(W)\ W_x + B(W)\ W_y = 0 \\ A(W) = F_W \ ; \ B(W) = G_W \end{cases}$$

The functions F and G are homogeneous i.e. :

$$F(W) = A(W)W$$
$$G(W) = B(W)W.$$

The system (2) is hyperbolic.

Boundary conditions

For all the test-problems we solve here, the boundary conditions are the following :

A) Far field boundary

At "infinity" the flow is assumed to be uniform.

B) Wall boundary

At this boundary a slip condition is considered i.e. :

$$\vec{V}.\vec{n} = 0$$

where $\vec{V} = (u,v)$ is the velocity and $\vec{n}$ is the outward normal vector.

3.2. - The space discretization

3.2.1. - The grid : finite element method (F.E.M)

The triangulation used is of a finite element type one.

More precisely, it consists of a subdivision T_h of the domain Ω by triangles such that :

$$\bar{\Omega} = \cup K \ ; \ K \in T_h \ K \text{ triangle.}$$

$$K \cap K' = \begin{cases} \emptyset \text{ or} \\ \text{a vertex or} \\ \text{a side} \end{cases} \qquad \forall K, K' \in T_h$$

The nodes of the mesh are the vertices of triangles, the degrees of freedom are the values of the unknowns at the nodes :

$$W_i = W(a_i, t)$$

$$a_i = 1 \text{ node} \ ; \ 1 = 1, \text{ ns}$$

ns = number of nodes.

3.2.2. - Variational formulation : finite volume method (F.V.M.)

For any vertex a_i of T_h, we divide every triangle having a_i as a vertex, into six sub-triangles delimited by the medians ; then a control volume C_i (cell) is defined as the union of those sub-triangles having a_i as a vertex.

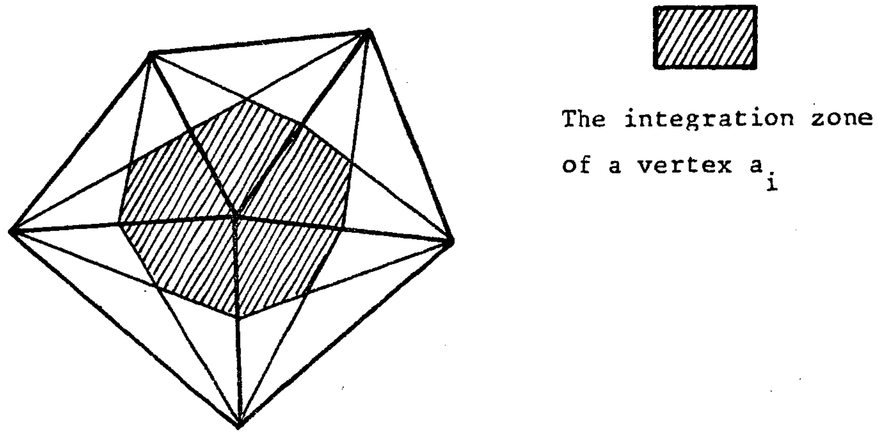


FIG.1 : Construction of an integration zone C_i around a vertex a_i

Integrating equations (1) with Green's formula, the F.V.M. method leads to :

$$(3) \qquad \text{area } (C_i)W_t + \int_{\partial C_i} (F(W)\nu_x + G(W)\nu_y)\, d\sigma = 0$$

where ∂C_i is the boundary of C_i and $\vec{\nu} = (\nu_x,\nu_y)$ is the unit outer normal vector of ∂C_i.

The numerical integration of (5) is the purpose of the next section.

For more details about the F.E.M. and F.V.M. methods, applied to this context we refer to [9].

3.3. - First-order fluxes

Let F^* and G^* be the approximate flux functions along the cell boundary ∂C_i ; we denote by Φ_{ij} the flux on the cell C_i in direction of j, neighbor of i ; then :

$$(4) \qquad \int_{\partial C_i} (F^* \nu_x + G^* \nu_y)\, d\sigma = \sum_{j\in N_i} \Phi_{ij}$$

where N_i is the set of neighbors of the node i.

The computation of Φ_{ij} is done as follows :

(i) The boundary ∂C_i of cell C_i is decomposed in bi-segments G_1IG_2 where I is the middle of the side joining vertex i to a neighboring j, G_1 and G_2 are the centroids of the triangles having ij as a common side, and $(F^* \nu_x + G^* \nu_y)$ is taken constant on the bi-segment ;

(ii) The value of Φ_{ij} is given by :

(5.1)
$$\begin{cases} \Phi_{ij} = H \dfrac{(U+V)}{2} - \dfrac{1}{2} \; P\left|\dfrac{(U+V)}{2}\right|(V-U) \\ H(U) = \eta_1 F(U) + \eta_2 G(U) \\ P(U) = \eta_1 A(U) + \eta_2 B(U) \\ \eta_1 = \displaystyle\int_{G_1IG_2} \nu_x \, d\sigma \quad ; \; \eta_2 = \int_{G_1IG_2} \nu_y \, d\sigma \end{cases}$$

(5.2)
$$U = W_i \; ; \; V = W_j$$

<u>Remark (3)</u> : The numerical flux written in (5.1) is the first-order flux of Vijayasundaram. Other fluxes can be used : it suffices to replace in (5.1) the functions H and P by the corresponding flux functions.

3.4. - The first-order scheme

The formulation is the same as in the 1-D case.

The algorithm is :

A) Explicit phase

(6)
$$\begin{cases} \delta W_i^n = \dfrac{\Delta t}{\text{area}(C_i)} \, \Phi_{ij}^n, \\ \Phi_{ij}^n \text{ is given by (5.1) with } U = W_i^n \; ; \; V = W_j^n. \end{cases}$$

B) Implicit phase

$$(7)\qquad \left\{\begin{array}{l} \frac{1}{\Delta t}\,\delta W_i^n = \frac{1}{\mathrm{area}(C_i)} \sum_j \Phi_{ij}^S \\ \Phi_{ij}^S = \Phi^S(W_i^n, W_j^n, W_i^{n+1}, W_j^{n+1}) \\ \Phi^S(U,V,W,Z) = H_1(U,V)(W-U) + H_2(U,V)(Z-V) \end{array}\right.$$

In all above, just the linearization of the interior upwind fluxes is considered;for boundary fluxes (inflow, outflow, wall boundaries) the same fluxes are applied to get a compatible system of boundary conditions : the same linearization holds.

For wall boundaries, the pressure integral is evaluated in the explicit phase.

We compute :

$$\int_{\partial\Omega} \Phi(p^n.V)\, d\sigma$$

where :

$$p^n = p(W^n) \quad \text{and} \quad V = (0,\nu_x,\nu_y,0).$$

In the implicit phase we compute :

$$\int_{\partial\Omega} \Phi p(W^n)\; \delta W^{n+1}\, d\sigma$$

where W^n and δW^{n+1} are evaluated in a such way that the slip conditions :

$$W_2^n\, \nu_x + W_3^n\, \nu_y = 0$$

$$\delta W_2^{n+1}\, \nu_x + \delta W_3^{n+1}\, \nu_y = 0$$

are satisfied.

W_2 and W_3 are the two components of the velocity.

The solution of the linear system arising in (7) is achieved by a Gauss-Seidel relaxation method.

3.5. - Second-order fluxes

The construction of the second-order fluxes is done in the same way as in the one-dimensional case.

The main idea is to introduce in a "compact" or "Hermitian" way, the gradients of the unknowns functions ; this is done by means of the Galerkin interpolation.

For the x-derivative it is written :

for each vertex i

$$W_x(i) = \frac{\int_{supp(i)} W_x \, dx \, dy}{\int_{supp(i)} dx \, dy}$$

where supp(i) is the support of the basis function related to vertex i i.e. : the union of the triangles having i as a vertex.

We get then the second-order fluxes by replacing in (5), (5.2) by :

$$U = W_i^n + \frac{\overrightarrow{\nabla W}}{2} \cdot \overrightarrow{ij}$$

$$V = W_j^n + \frac{\overrightarrow{\nabla W}}{2} \cdot \overrightarrow{ji}$$

$$\overrightarrow{\nabla W} = (W_x^n(i), W_y^n(i)).$$

3.6. - The second-order scheme

An efficient way to get the second-order accurate steady-state solutions is to use a second-order flux in the explicit phase of the scheme defined by (6) and (7) ; the implicit phase not being changed.

Thus in the discrete context, the implicit phase no longer includes the full linearization of the explicit flux, so that we cannot hope for as fast a convergence as in the case of the first-order scheme.

But our purpose is to compare this method with the explicit version of the scheme at the same order of accuracy and to show how much computing time is gained when we use the implicit method.

3.7. - Numerical experiments

A) Steady flow in a channel

We choose a test problem proposed at the GAMM workshop held in 1979 in Stockholm [21].

The bump is a 4.2 % thick circular arc with length 1, in a 2.073 high channel. Free-stream values correspond to a Mach number of .85, for which the flow is transonic.

For consistency with the GAMM test, we use a 72X21 triangular mesh.

The problem is solved with the 2-D extension of the first-order upwind scheme of Vijayasundaram.

We present in Fig. A and Table 1 the convergence histories and tables of efficiency.

For other numerical results (iso-values, ...) we refer to [23].

The results show a very good efficiency with an acceleration ratio of approximatively 8/1 between the implicit and the explicit version of the first-order scheme.

B) Steady flow around a NACA0012

This test problem was also proposed at the GAMM workshop [21].

We present four numerical experiments which differ by the Mach number M_∞, the incidence angle α and the quality of the mesh.

For the three first experiments, we used a 60x10 O-mesh (1680 triangles) with the following conditions :

B1) $M_\infty = .80$; $\alpha = 0°$ (GAMM test problem)
B2) $M_\infty = .30$; $\alpha = 20°$

B3) $M_\infty = .50$; $\alpha = 30°$

B4) $M\infty = .85$; $\alpha = 0°$

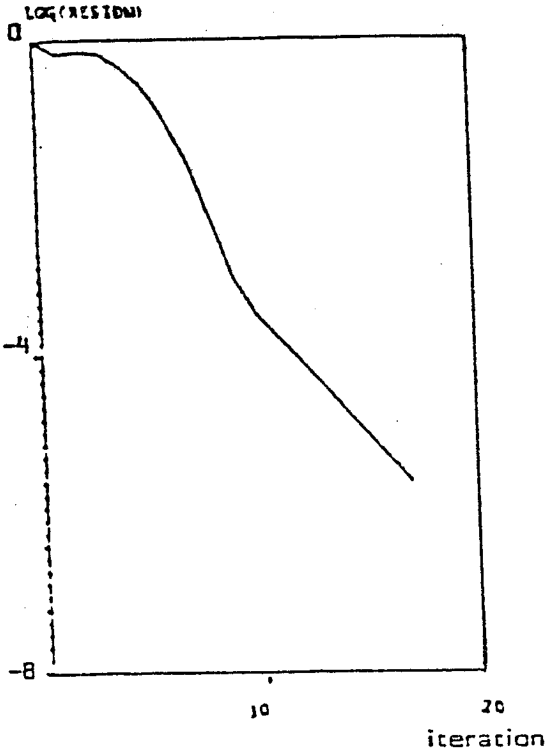
The mesh used in B4 is unstructured and irregular ; it has been obtained, starting from the above mesh by B. Palmerio's self adaptive program [19].

C) High speed flow past a blunt body

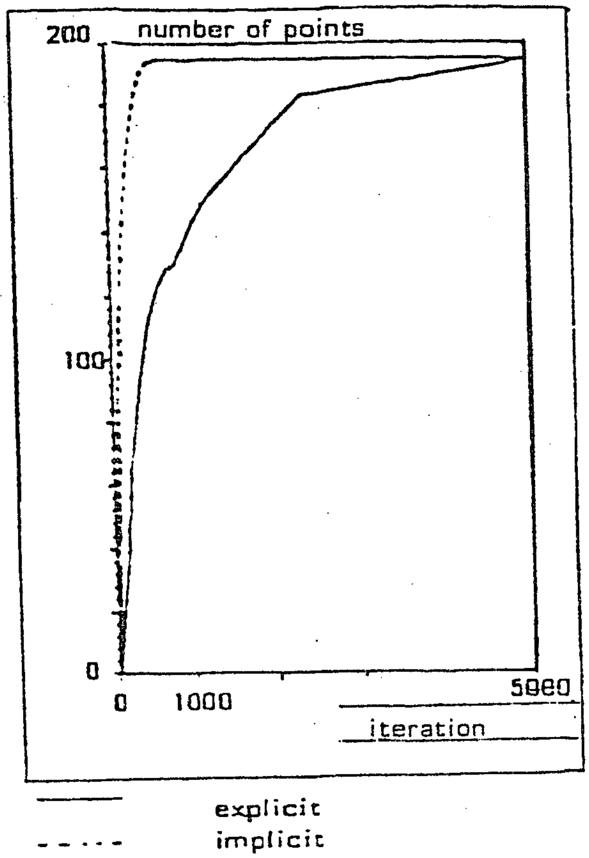
The Mach number is 8.

The calculation has been performed with the first-order scheme which uses in the implicit phase Steger's flux and in the explicit phase the Osher's flux.

A. Steady flow in a channel
M_∞ - .85 . α - 0° .



convergence history



build-up of the number of supersonic points.
time unity - cpu time of one explicit iteration

Method	average CPU cost	CFL	Number of iterations	ratio
Explicit	5.75	. 8	5 100	
Implicit	400	10^3	10	7.5

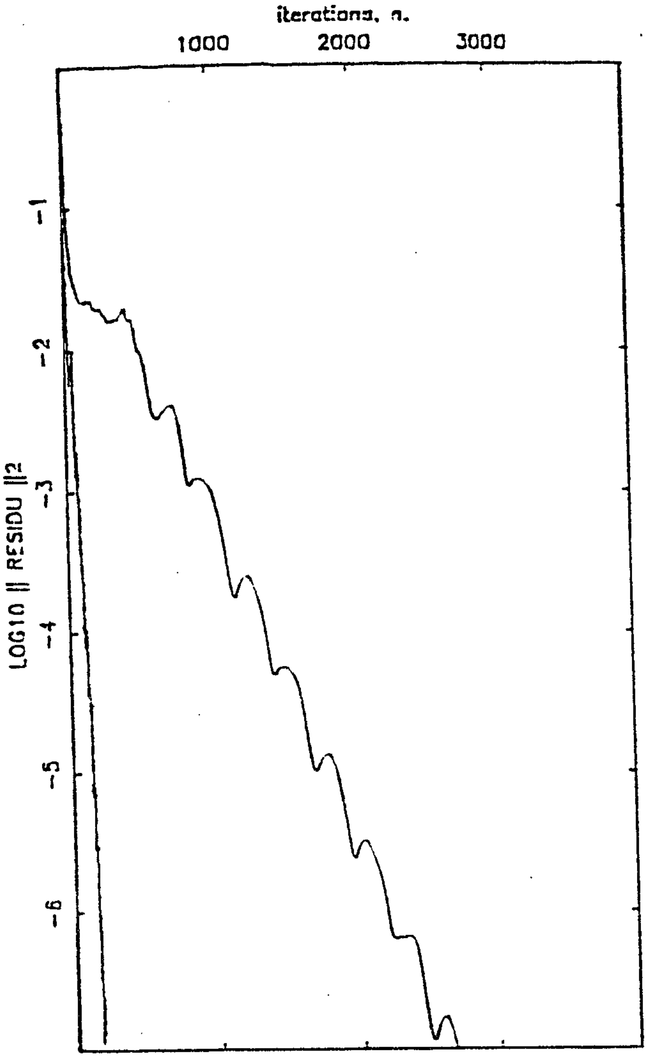
table 1

Method.average cpu cost of one iteration. CFL number.
number of iterations needed to reduce the residual by 10^3.

Scheme used : explicit and implicit versions of the first - order Q-scheme with about 50 relaxations sweeps on the linear system per time step.

B1.Steady flow around a NACA0012

- MACH= 0.80 - INCIDENCE= 0.00 -



Mechod	average CPU cosc	CFL	Number of iteracions	racio
Explicic	10 s	.7	1000	
Implicic	22 s	num. iter.	75	5.5

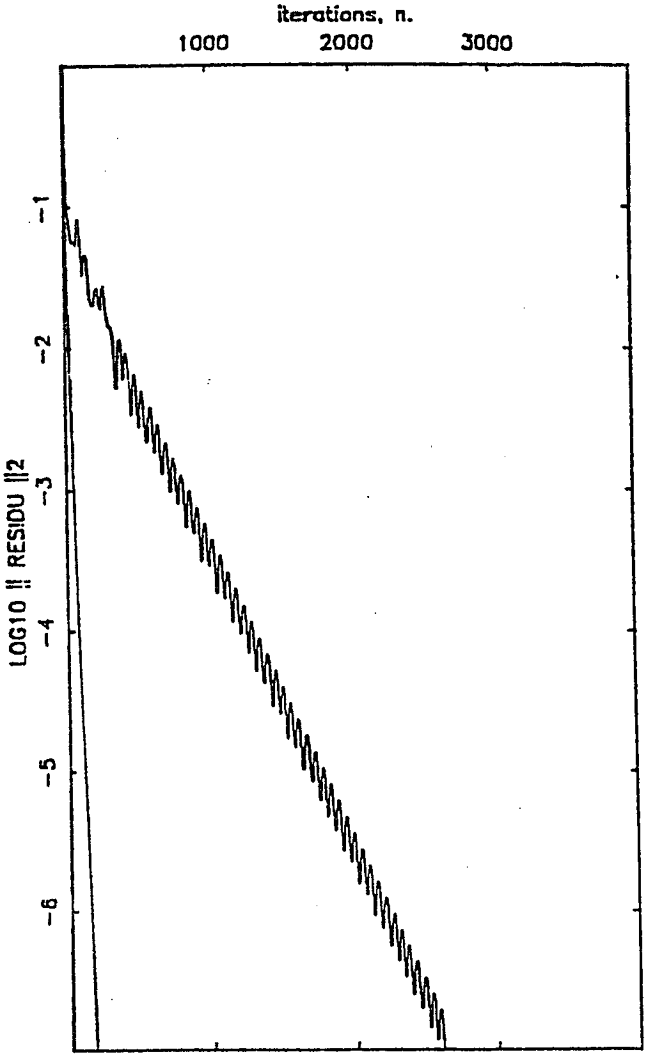
table 2

Schemes used :

- Explicit version of the second-order Q-scheme
- 2-phases schemes. Implicit first-order Vijayasundaram/
  Explicit second-order Q-scheme.

B2.Steady flow around a NACA0012

— MACH= 0.30 — INCIDENCE=20.00 —



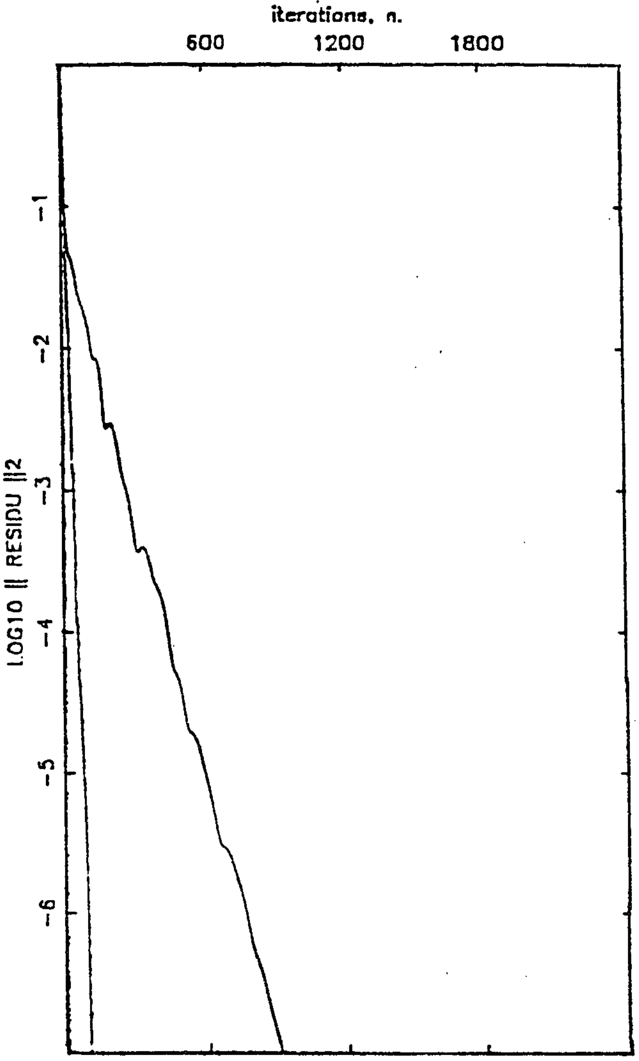
Method	average CPU cost	CFL	Number of iterations	ratio
Explicit second-order	10 s	.7	890	
Implicit second-order	22 s	num. iter.	75	4.9

table 3

Schemes used : . Explicit versions of the second-order accurateQ-scheme
. 2-phases scheme : Implicit first-order Vijayasundaram/
Explicit second-order Q-scheme
with about 2 relaxations sweeps
on the linear system per time step.

B3.Steady flow around a NACA0012

– MACH= 0.50 – INCIDENCE=30.00 –



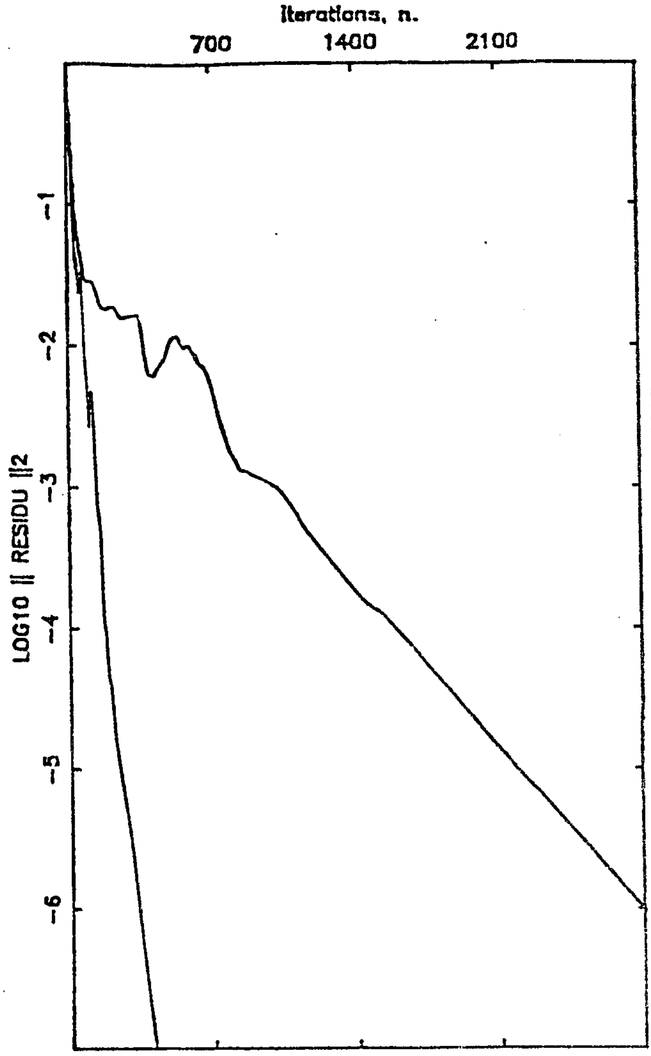
Method	average CPU cost	CFL	Number of iterations	ratio
Explicit	9 s	.7	480	
Implicit	18 s	num. iter.	60	4

table 4

Scheme used :

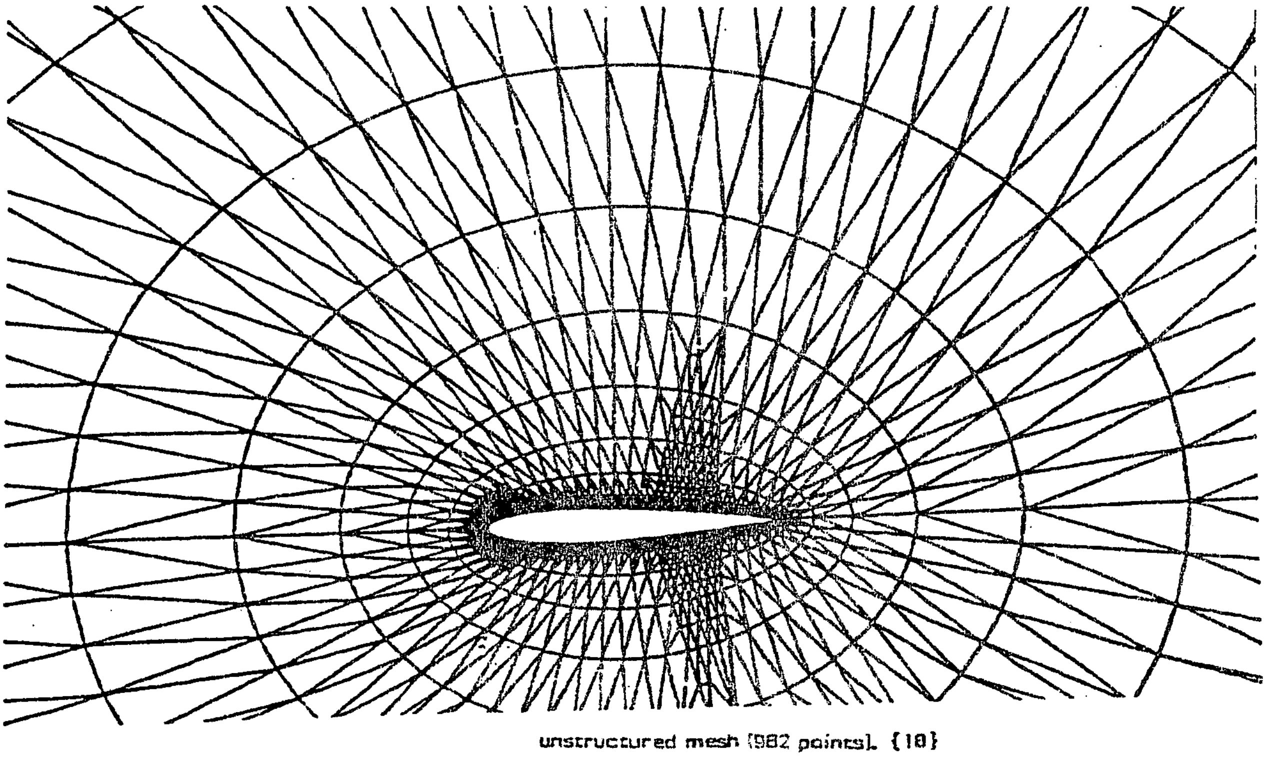
- Explicit version of the first-order Q-scheme
- 2 phases-scheme : Implicit first-order Vijayasundaram/ Explicit first-order Q-scheme with about 2 relaxations sweeps on the linear system per time step.

B4. Steady flow around a NACA0012
$M_\infty = .85$; $\alpha = 0°$



Method	average CPU cost	CFL	Number of iterations	ratio
Explicit	16 s	.7	1050	
Implicit	30 s	num. iter.	130	4.3

table 5

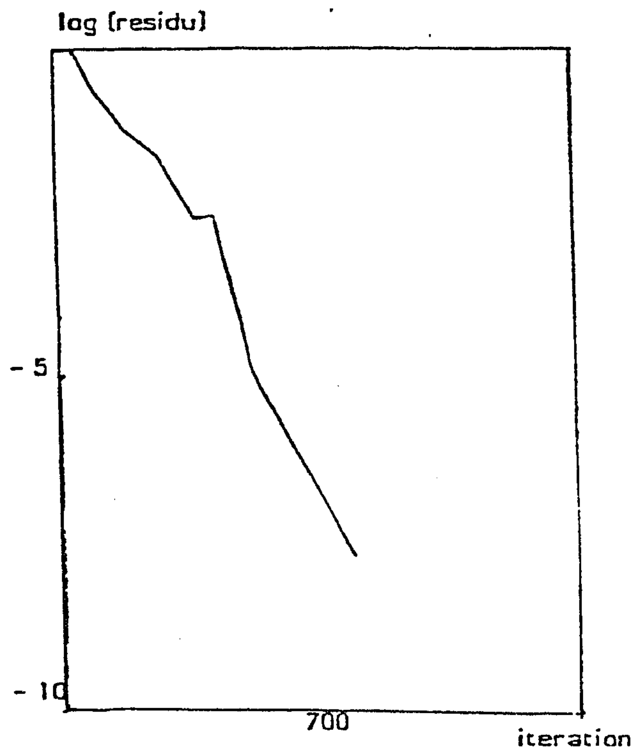


unstructured mesh (982 points). {18}

<u>Scheme used :</u> . Explicit version of the second-order Q-scheme
. 2-phases scheme : Implicit first-order Vijayasundaram/
Explicit second-order Q-scheme
with about 2 relaxation sweeps
on the linear system.

C. High speed flow past a cylinder

$M_\infty = 8. , \alpha = 0^\circ.$



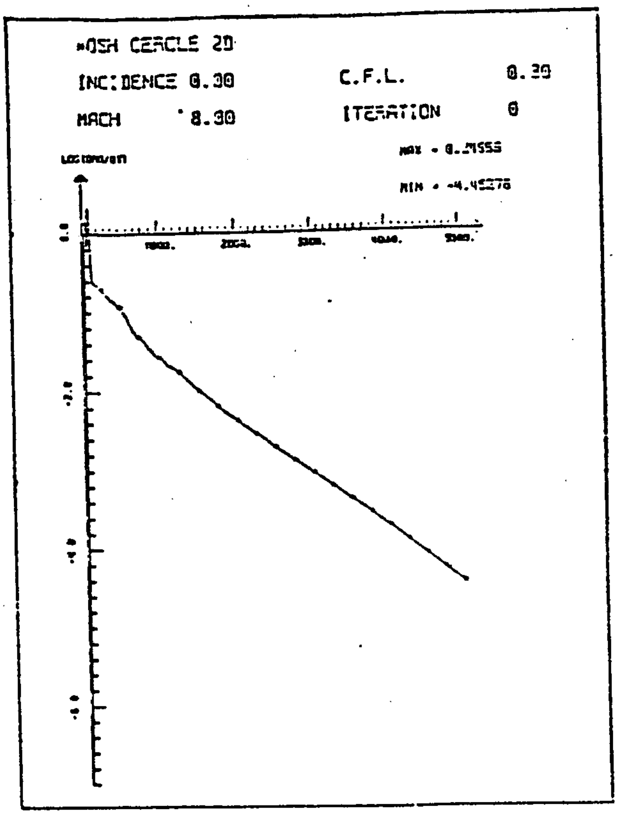
Convergence history
of the implicit version

Method	average CPU cost	CFL	Number of iterations	ratio
Explicit	9 s	.7	3000	
Implicit	18 s	15	400	3.75

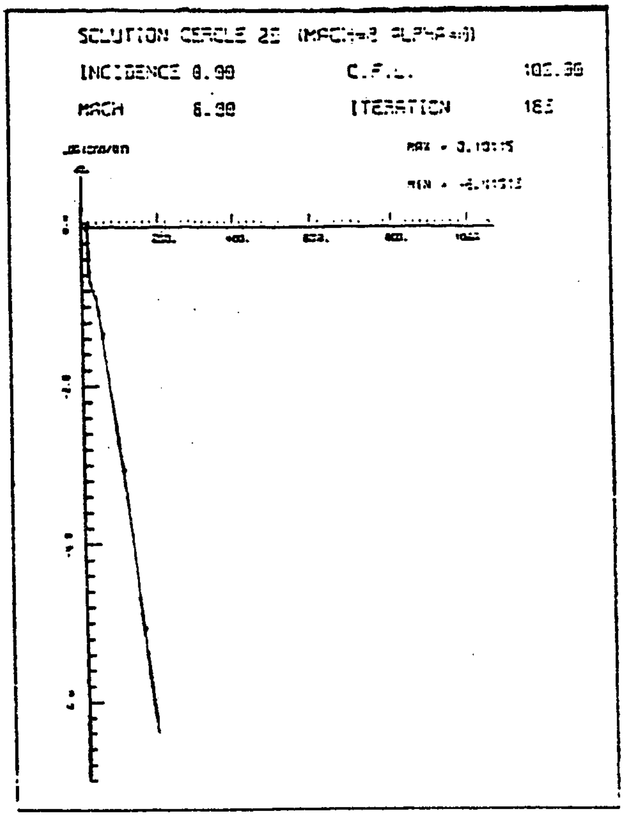
table 6

<u>Scheme used :</u> . Explicit version of the first-order scheme of Osher
. 2-phases scheme : Implicit first-order Steger and Warming/
Explicit first-order Osher
With about 2 relaxation sweeps on the linear
system per time step.

Flow past a blunt body
Mach = 8. ,α = 0°



explicit solution

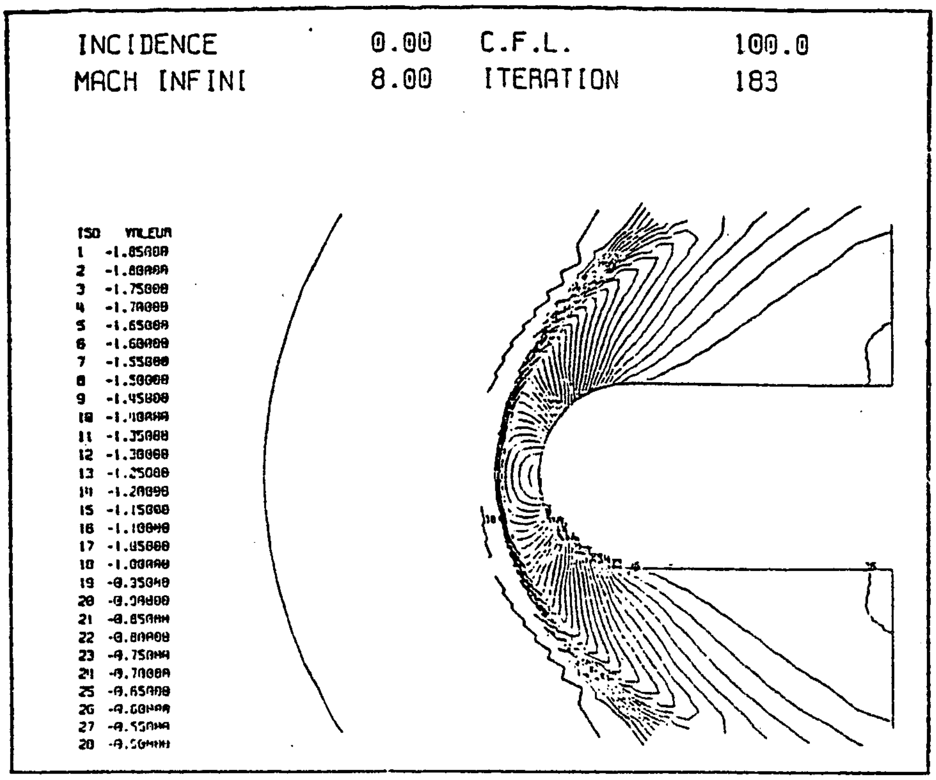


implicit solution

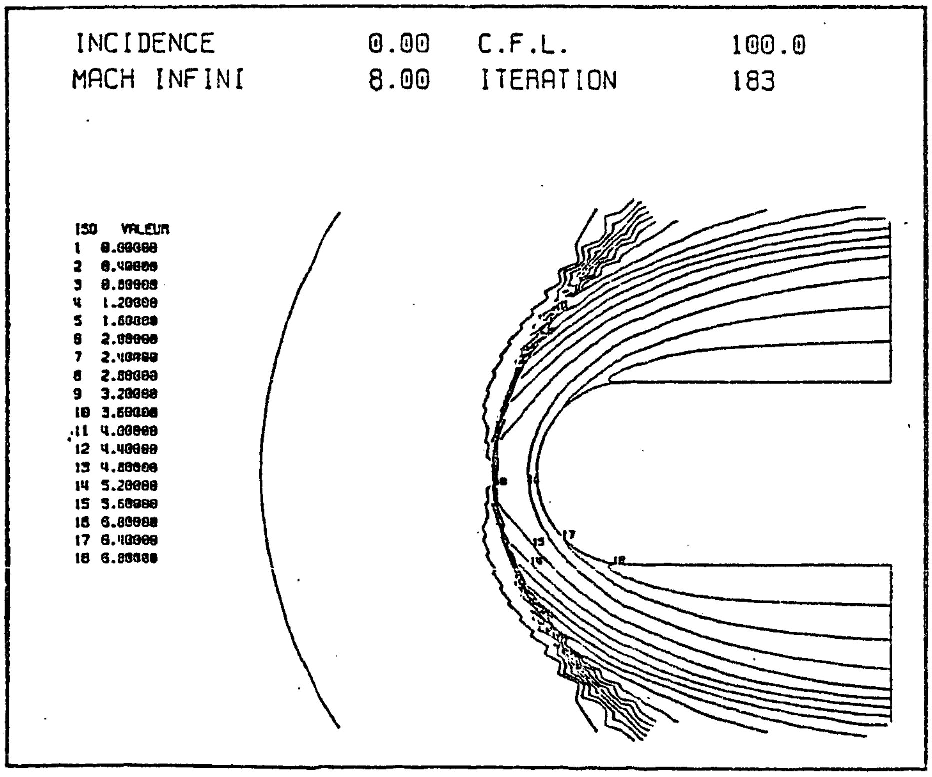
Method	average CPU cost	CFL	Number of iterations	ratio
Explicit	2 s	.80	3000	
Implicit	6 s	min (KT,100)	100	10

table 7

Flow past a blunt body
Mach = 8. ,α = 0°

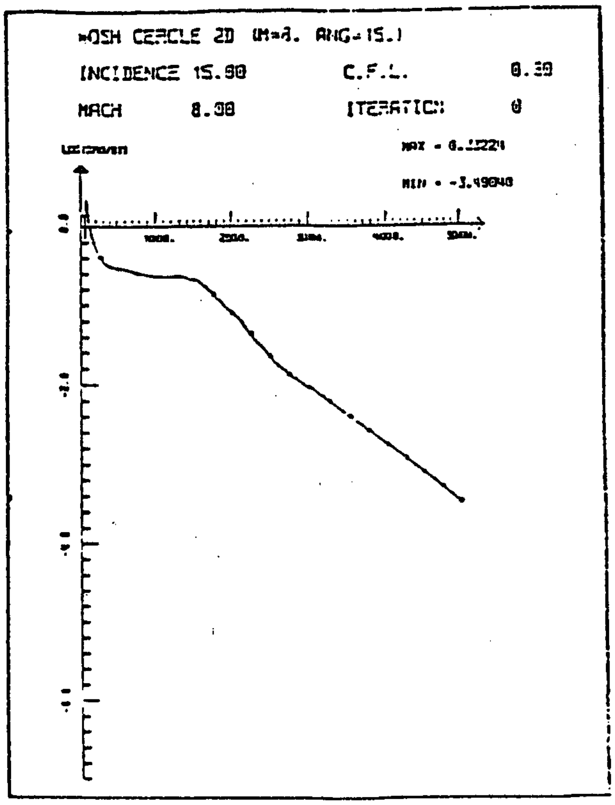


K_p contours

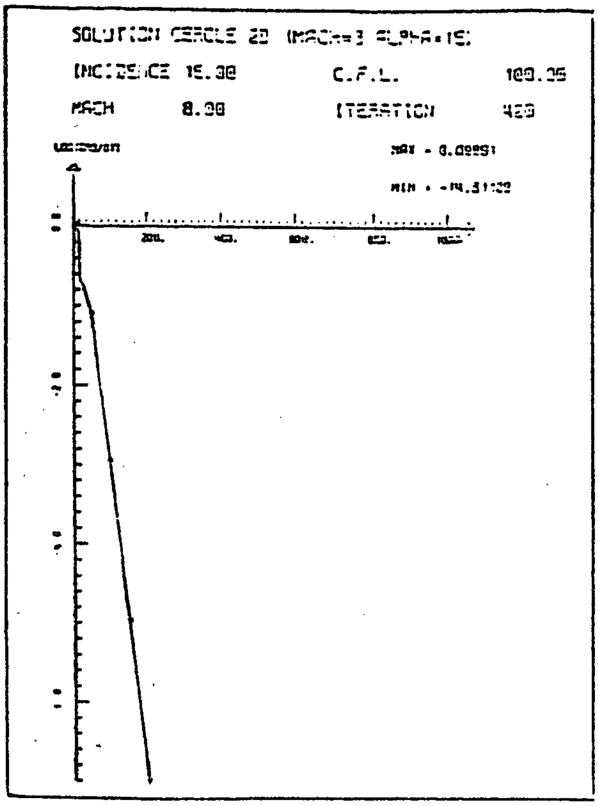


Entropy contours

flow past a blunt body
Mach = 8. ,α = 15°



Explicit time-stepping

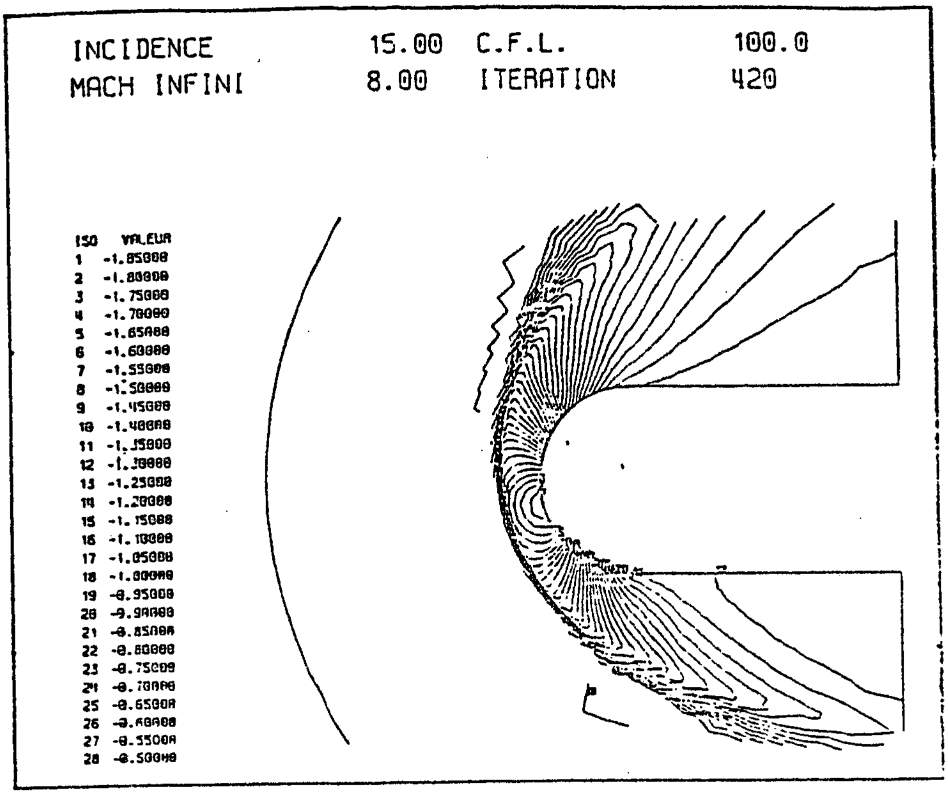


Implicit time-stepping

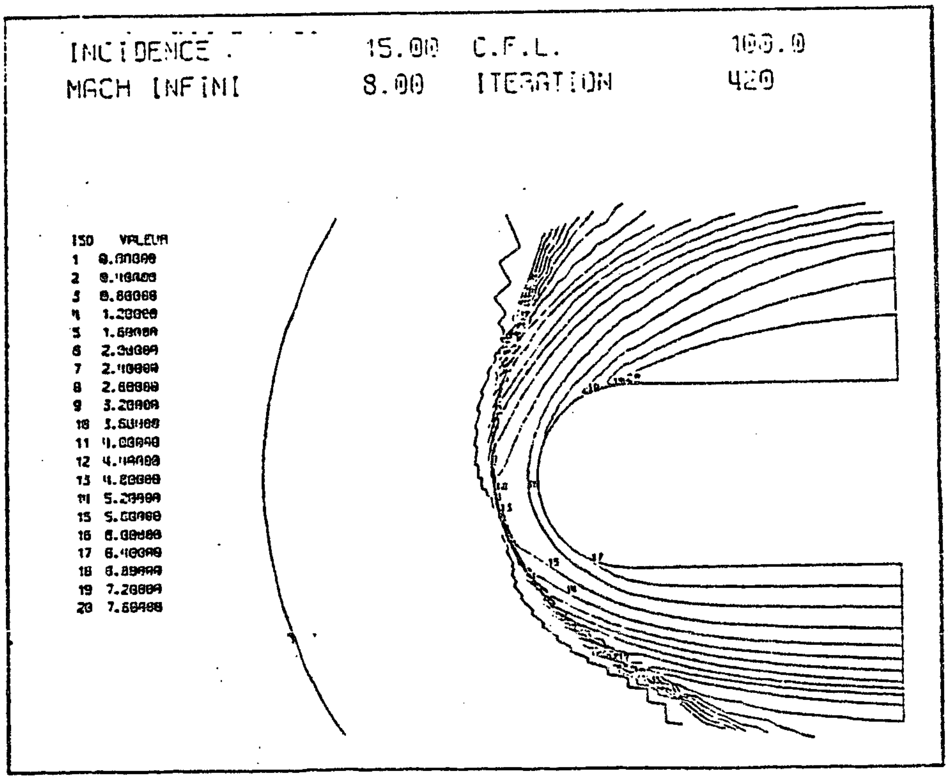
Method	average CPU cost	CFL	Number of iterations	ratio
Explicit	2 s	.80	3750	
Implicit	6 s	min (KT,100)	100	12,5

table 8

Flow past a blunt body
Mach = 8. ,α = 15°

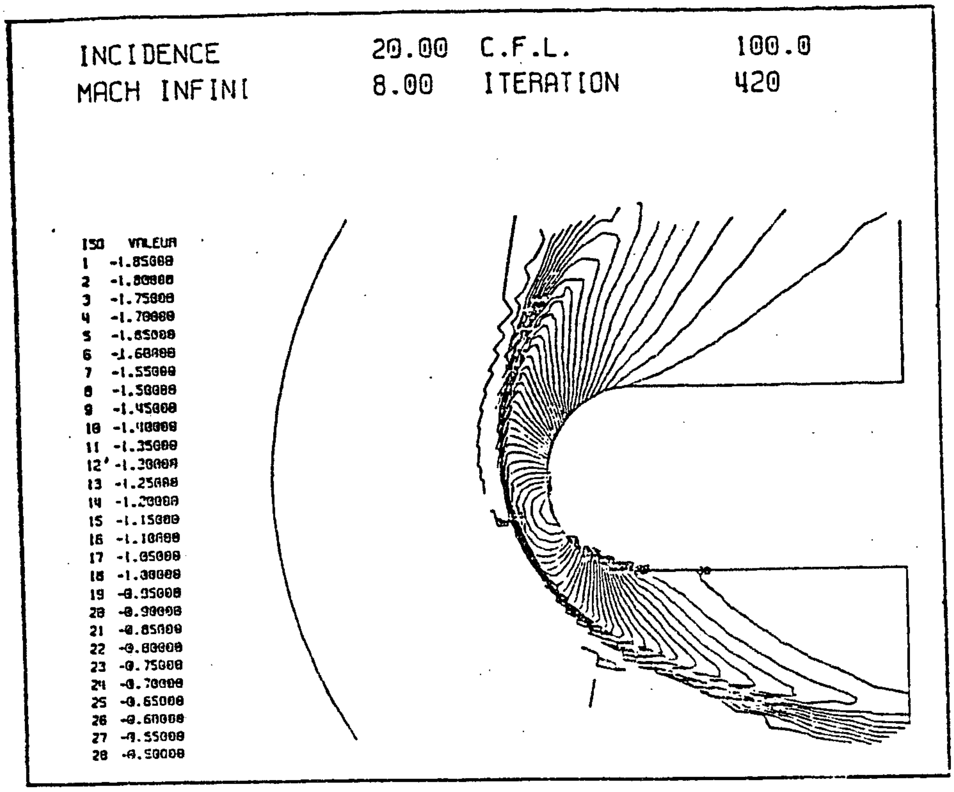


K_p contours

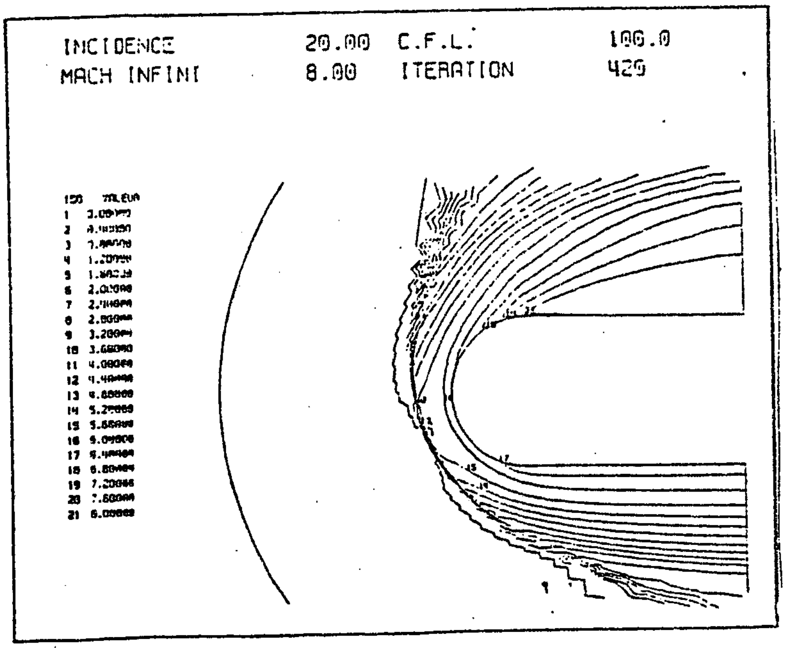


Entropy contours

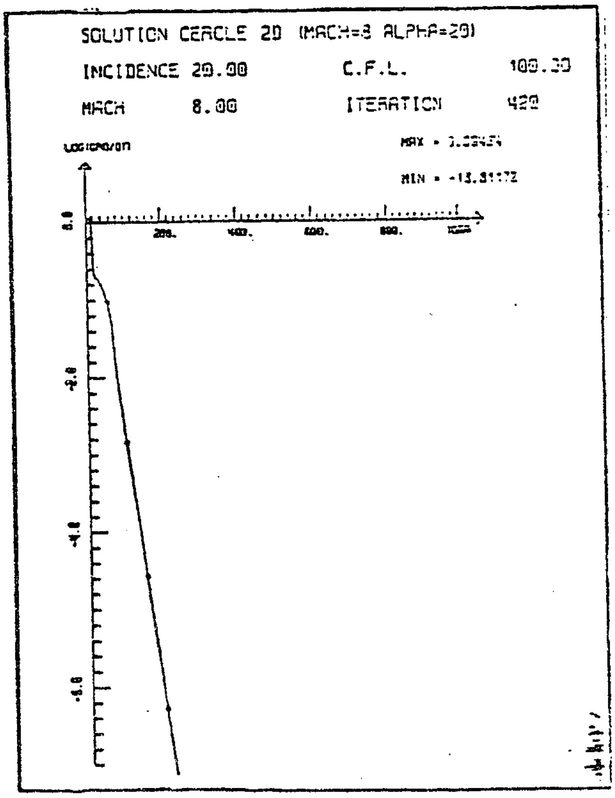
Flow past a blunt body
Mach = 8. α = 20°


INCIDENCE 20.00 C.F.L. 100.0
MACH INFINI 8.00 ITERATION 420

C_p contours


INCIDENCE 20.00 C.F.L. 100.0
MACH INFINI 8.00 ITERATION 420

Entropy contours


SOLUTION CERCLE 2D (MACH=8 ALPHA=20)
INCIDENCE 20.00 C.F.L. 100.00
MACH 8.00 ITERATION 420

convergence history

4. - CONCLUSION

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

With the first-order scheme we have a nearly quadratic convergence without the exact computation of the Jacobian.

The second-order version of the method is no longer quadratically converging but still very efficient.

The main quality of these implicit schemes is that they are more reliable than their explicit version. This is clearly shown in the case of unstructured meshes or in the case of high Mach number regimes and large incidences, revealing the greater robustness of the implicit version of the schemes.

We note however that the method is not yet fit for very heavy industrial use since for instance, we have to store 2-D matrices, but a study of this problem is in progress. For the numerical study of spatial approximations, it 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.

AKNOWLEDGMENTS

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] ANDERSON W.K. - THOMAS J.L. - VAN LEER B.
A comparison of Finite Volume Flux-Vector Splittings for the Euler Equations.
AIAA paper 85-0122.

[2] ANGRAND F. - DERVIEUX A.
Some explicit triangular Finite Element Schemes for the Euler Equations.
I.J. for Num. Meth. in Fluids, Vol 4, 749-764 (1984).

[3] ANGRAND F. - BOULARD V. - DERVIEUX A. - PERIAUX J. - VIJAYASUNDARAM G.
Transonic Euler Simulations by means of Finite Element explicit schemes.
AIAA paper 83-1924.

[4] ANGRAND F. - BOULARD V. - DERVIEUX A. - DESIDERI J.A. - PERIAUX J. - STOUFFLET B.
On the use of rational Runge-Kutta methods on Euler steady-state computations.
Lectures Notes in Physics 218 (1985).

[5] ANGRAND F. - BOULARD V. - DERVIEUX A. - PERIAUX J. - VIJAYASUNDARAM G.
Triangular Finite Elements methods for the Euler Equations.
Comp. Meth. in Appl. Sc. Eng., North Holland (1984) (to appear).

[6] ANGRAND F. - BILLEY V. - DERVIEUX A. - PERIAUX J. - STOUFFLET B.
2-D and 3-D Euler flow calculations with a second-order Galerkin finite Element Method. AIAA Paper 85-1706 (1985).

[7] BEAM R.M. - WARMING R.F.
An implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation Law form.
J. Comp. Phys.; 22 (1976).

[8] CASIER F. - DECONINCK H. - HIRSH C.
A class of Central Bidiagonal Schemes with implicit boundary conditions for the Euler Equations.
AIAA 83-0126, (1983).

REFERENCES (continued)

[9] DERVIEUX A.
Steady Euler Simulations using unstructured meshes.
Von Karman institute for Fluid Dynamics, lectures Series 1884-04.
March 26-29 (1985).

[10] FEZOUI F.
Résolution des équations d'Euler par un schéma de Van Leer en éléments finis.
INRIA Report 358 (1985).

[11] HANCOCK S.L.
Private communication, reported by VAN ALBADA et al. ref. [26].

[12] HARTEN A.
On a class of High Resolution Total-Variation-Stable Finite Difference schemes.
NYU Report (1982).

[13] JESPERSEN D.C. - PULLIAM T.H.
Flux Vector Splitting and Approximate Newton Methods.
AIAA Computational Fluid Dynamics Conference, Danvers, July 1983.

[14] LERAT A.
Implicit Methods of Second-order Accuracy for the Euler Equations.
AIAA Computational Fluid Dynamics Conference, Danvers, (1983).

[15] MAC CORMACK R.W.
A Numerical Method for solving the Equation of Compressible Vicous Flows.
AIAA Journal, 20 (1982).

[16] MONTAGNE J.L.
A Second-Order Accurate Flux Splitting Scheme in Two-Dimensional Gas Dynamics.
Lectures notes in Physics 218 (1985).

REFERENCES (continued)

[17] MULDER W.A. - VAN LEER B.
Implicit Upwind Methods for the Euler Equations.
AIAA paper 83-1930 (1983).

[18] OSHER S. - SALOMON F.
Upwind difference schemes for hyperbolics systems of conservation laws.
Journal Math. Comp., April 1982.

[19] PALMERIO B.
Self adaptative F.E.M. algorithms for the Euler equations.
INRIA Report 338 (1984).

[20] RAI M.M. - CHAKRAVARTY S.R.
An implicit Form for the Osher Upwind Scheme.
AIAA paper 84-0088 (1984).

[21] RIZZI A. - VIVIAND H. (Eds)
Numerical methods for the computation of inviscid transonic flow with shock waves.
Notes on Numerical Fluid Dynamics, Vol. 3, Wieweg and Sohn.
Braunschweig/Wiesbaden, 1981.

[22] STEGER J. - WARMING R.F.
Flux vector splitting for the inviscid gas dynamics with applications to finite difference methods.
Journal Comp. Physics, Vol. 40 n 2 (1981).

[23] STOUFFLET B.
Résolution numérique des équations d'Euler des fluides parfaits compressibles par des schémas implicites en éléments finis.
Thèse 3e cycle, PARIS VI, Mars 1984.

[24] STOUFFLET B.
Implicit Finite Element Methods for the Euler Equations.
Proceeding of the INRIA workshop on "Numerical Methods for Compressible Inviscid Fluids", 7-9 dec. 1983 Rocquencourt (France), Glowinski R., Ed., 1983, to appear SIAM.

REFERENCES (continued)

[25] THOMAS J.L. - VAN LEER B. - WALTERS R.W.
Implicit Flux Split for the Euler Equations.
Abstract for AIAA, 19th Fluid Dyn., Plasma & Phys. Lasers Conf., Cincinnati (1985).

[26] TURKEL E. - VAN LEER B.
Flux-Vector Splitting and Runge-Kutta methods for the Euler Equations.
ICASE Report No. 84-27 (1984).

[27] VAN ALBADA G.D. - VAN LEER B. - ROBERTS W.W.
A Comparative Study of Computational Methods in Cosmic Gas Dynamics.
Jr. (1982), Astron. Astrophys. 108, 76.

[28] VAN LEER B.
Towards the Ultimate Conservative Difference Scheme III.,
Journal of Comp. phys. 23, 263-275 (1977).

[29] VIJAYASUNDARAM G.
Résolution numérique des équations d'Euler pour des écoulements transsoniques avec un schéma de Godunov en Eléments Finis, Thèse 3e cycle Paris VI (1983).

[30] YEE H.C. - WARMING R.F. - HARTEN A.
Implicit Total Variation Diminishing (TVD) Schemes for Steady-State Calculations.
NASA Technical Memorandum SU342, 1983.

[31] YEE H.C.
Linearized Form of Implicit TVD Schemes for the Multidimensional Euler and Navier-Stokes Equations.
Advances in Hyperbolic partial Differential Equations, April 1985.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique