Numerical simulation of 3-D flows with a finite element method

Vittorio Selmin

INRIA

UNITÉ DE RECHERCHE
INRIA-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° 706

NUMERICAL SIMULATION OF 3-D FLOWS WITH A FINITE ELEMENT METHOD

Vittorio SELMIN

JUILLET 1987

NUMERICAL SIMULATION OF 3-D FLOWS WITH A FINITE ELEMENT METHOD

SIMULATION NUMERIQUE D' ECOULEMENTS 3-D PAR UNE METHODE D'ELEMENTS FINIS

Vittorio SELMIN

INRIA Sophia-Antipolis
Avenue Emile Hughes
(Anc. Route des Lucioles)
Sophia Antipolis
06565 VALBONNE

Abstract

In this paper, we present 3-D Euler calculations by finite elements over unstructured meshes. The schemes which are used to perform the calculations are formulated within the framework of the Taylor-Galerkin finite element method. We have tested an artificial viscosity method and a TVD method to damp the oscillations which appear near the discontinuities. We also introduce a local mesh refinement technique. Finally, results of the numerical computation of transonic and supersonic flows are presented.

Résumé

Dans ce rapport, des calculs d' écoulements tridimensionnels régis par les équations d' Euler sont présentés dans le cadre d' une méthode d' éléments finis non-structurés. Les schémas utilisés pour effectuer les calculs sont construits un utilisant la méthode Taylor-Galerkin des éléments finis. On a testé des méthodes de viscosités artificielles et des méthodes TVD pour réduire l' amplitude des oscillations apparaissant près des discontinuités. On présente aussi un méthode de raffinement local du maillage. Finalement, des exemples numériques concernant la résolution d' écoulements transoniques et supersoniques sont présentés.

I. Introduction

The numerical solution of compressible flow problems has received much attention over the past few years due, to a large extent, to the interests of the aerospace industry. The solutions of such problems are characterized by the appearance of discontinuities, such as shock waves, in the flow field and a major topic of attention has been the development of numerical techniques which are able to adequately resolve such phenomena. Nevertheless, there is an ongoing interest in the application of unstructured grid finite element methods to the solution of this problems. In fact, this methods can take into account arbitrary geometries, including in particular the singular characteristics of an aircraft or a space shuttle.

In the following, we report on a finite element method for the solution of 3-D compressible flows. This method intends to combine the advantages of a first-order scheme, which produces no ripples near discontinuities, with those of a second-order one. These two schemes are formulated within the framework of the Taylor-Galerkin finite element method[1]. The difference between the two schemes is a discretization of a Laplacian operator which uses the dissipative effect of the well-known mass matrix operator of the finite element formulation. As shown previously in [2], using tetrahedral elements, the dissipative operator constructed on an element can be distributed over its sides which will be subsequently named segments. Thus, we can replace the element contributions to the discretization of the dissipative operator by segment contributions. The interest of the segments relies in the fact that being one-dimensional they can be use to generalize one dimensional methods (such as TVD methods for example).

An outline of the report follows. In Section II, we present the Euler equations and a very brief description of the finite element spatial discretization; then, in Section III, we discuss the first- and second-order schemes. In Section IV, we describe an artificial viscosity method and a flux limiter TVD method. In Section V, a local mesh refinement method is introduced. Finally, in Section VI, numerical results of the computation of transonic and supersonic flows are presented.

II. Mathematical modelling

1. Governing equations

The conservative law form of the Euler equations in the three-dimensional space is given by :

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + \frac{\partial G(U)}{\partial y} + \frac{\partial H(U)}{\partial z} = 0 \qquad (1)$$

in which

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix} , \; F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(e+p) \end{pmatrix} , \; G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(e+p) \end{pmatrix} \quad H = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(e+p) \end{pmatrix}$$

where ρ is the density, $\overrightarrow{V} = (u, v, w)$ is the velocity vector, e is the total energy per unit volume and p is the pressure. We assume that the fluid satisfies the perfect gas law :

$$p = (\gamma - 1)(e - \rho \frac{u^2 + v^2 + w^2}{2})$$

where γ is the ratio of specific heats ($\gamma = 1.4$ for air).
Let A, B and C denote the Jacobian matrices $\partial F(U)/\partial U$, $\partial G(U)/\partial U$ and $\partial H(U)/\partial U$, respectively, Eq.(1) can be written in the non conservative form :

$$\frac{\partial U}{\partial t} + A(U)\frac{\partial U}{\partial x} + B(U)\frac{\partial U}{\partial y} + C(U)\frac{\partial U}{\partial z} = 0$$

2. Finite element approximation

In order to discretize the continuous problem, the flow region is imbedded in a bounded polygonal domain Ω whose boundary is denoted by Γ. A tetrahedrization T with charateristic mesh spacing h is introduced on Ω. The vertices of the tetrahedra define a set of computational nodes, S , where the numerical solution of (1) is to be

computed. For each node S_i, a piecewise linear basis function $N_i(x,y,z)$ of support Ω_i is introduced where Ω_i is the union of all the elements T_i^l for which the node S_i is a common vertex. It is a first degree polynomial on each element T_i^l of T which takes the value 0 at all the nodes except S_i where $N_i(x_i,y_i,z_i)=1$.

Any scalar function $u(x,y,z)$ on Ω can be approximated by :

$$u^h(x,y,z) = \sum_i N_i(x,y,z)\ u(x_i,y_i,z_i)$$

where the sum is over all the nodes of S. In practice, for a given (x,y,z), the sum involves only four terms :

$$u^h(x,y) = \sum_{i=1}^{4} N_i(x,y,z)\ u(x_i,y_i,z_i) \quad ,$$

namely those corresponding to the basis functions N_i associated with the vertices of the tetrahedron containing (x,y,z).

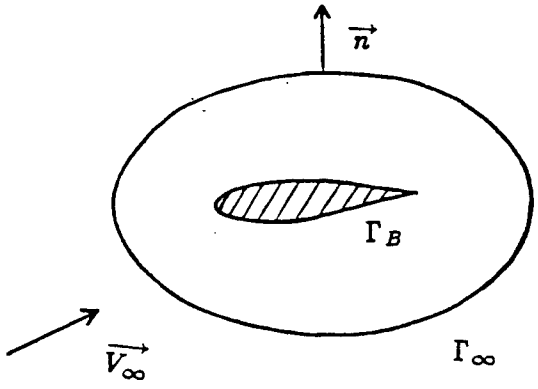


Fig.1: Boundary of the computational domain

3. Boundary conditions

In the sequel, we consider domains of computation related to external flows around bodies ; in Fig. 1 the body is represented by an airfoil which limits the domain of computation by its wall Γ_B and a second(artificial) boundary is introduced

to represent the farfield boundary denoted by Γ_∞.
We assume the flow to be uniform at infinity and we prescribe :

$$\rho_\infty = 1.0 \quad \vec{V}_\infty = \begin{pmatrix} cos\alpha\ cos\beta \\ sin\beta \\ sin\alpha\ cos\beta \end{pmatrix} \quad p_\infty = \frac{1}{\gamma M_\infty^2}$$

where α is the angle of attack, β is the yaw angle and M_∞ denotes the free-stream mach number.
Finally, on the wall Γ_B, we use the following slip conditions :

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

where $\vec{n}$ indicates the outward unit vector normal to the boundary $\Gamma = \Gamma_\infty \cup \Gamma_B$.

III. First- and second-order finite element schemes

1. Second-order scheme

Eq.(1) is discretized using a finite element version of the Richtmyer scheme[3]. For this scheme, we replace the following Taylor series expansion in the time increment Δt :

$$U^{n+1} = U^n + \Delta t \, \frac{\partial U^n}{\partial t} + \frac{1}{2}\Delta t^2 \, \frac{\partial^2 U^n}{\partial t^2} \quad , \tag{2}$$

by the two-step algorithm :

$$\begin{aligned} U^{n+\frac{1}{2}} &= U^n + \frac{1}{2}\Delta t \, \frac{\partial U^n}{\partial t} \\ U^{n+1} &= U^n + \Delta t \, \frac{\partial U^{n+\frac{1}{2}}}{\partial t} \end{aligned} \tag{3}$$

where superscript n is the time level so that $t^n = n\Delta t$ and $U^n = U(x, y, z, t^n)$. Note that the expansion (2) is basic to the classical Lax-Wendroff scheme.
Now, replacing the time derivative from Eq.(1), a Richtmyer two-step generalization of the Lax-Wendroff time differencing is obtained

$$\begin{aligned} U^{n+\frac{1}{2}} &= U^n + \frac{1}{2}\Delta t \, (\frac{\partial F^n}{\partial x} + \frac{\partial G^n}{\partial y} + \frac{\partial H^n}{\partial z}) \\ U^{n+1} &= U^n + \Delta t \, (\frac{\partial F^{n+\frac{1}{2}}}{\partial x} + \frac{\partial G^{n+\frac{1}{2}}}{\partial y} + \frac{\partial H^{n+\frac{1}{2}}}{\partial z}) \end{aligned} \tag{4}$$

in which $F^n = F(U^n)$, $G^n = G(U^n)$ and $H^n = H(U^n)$. Eqs.(4) are subsequently discretized in space as follows :

1.1. *The first step : predictor*

For each tetrahedron T of Ω, we compute

$$\tilde{U}^{n+\frac{1}{2}}(T) = \frac{1}{volume(T)} \, \{ \int_T U^n d\Omega - \int_{\partial T} (F^n \kappa_x + G^n \kappa_y + H^n \kappa_z) \, d\sigma \} \tag{5}$$

The result $\tilde{U}^{n+\frac{1}{2}}$ of the predictor phase is a constant function by tetrahedron (finite element P_0 or finite volume). The notation $(\kappa_x, \kappa_y, \kappa_z)$ indicates the outward unit vector normal to ∂T.

1.2. *The second step : corrector*

The second step of Eq.(4) is recast in a weak form according to the weighted residual formulation and the spatial approximation is performed using the standard Galerkin finite element method in which the weighting function is chosen to be the basis function $N_i(x,y,z)$.
The equation obtained in this way can be written in the form

$$\int_\Omega N_i\, U^{n+1} d\Omega = \int_\Omega N_i\, U^n d\Omega + RH_i^n \tag{6}$$

in which

$$\begin{aligned} RH_i^n = &+ \Delta t \int_\Omega \{\, F(\tilde{U}^{n+\frac{1}{2}}) \frac{\partial N_i}{\partial x} + G(\tilde{U}^{n+\frac{1}{2}}) \frac{\partial N_i}{\partial y} + H(\tilde{U}^{n+\frac{1}{2}}) \frac{\partial N_i}{\partial z} \,\} d\Omega \\ &- \Delta t \int_\Gamma \{\, F(\bar{U}) n_x + G(\bar{U}) n_y + H(\bar{U}) n_z \,\} d\Gamma \end{aligned} \tag{7}$$

where the boundary terms contain the physical boundary conditions (see section on the numerical examples). Since the left-hand side of Eq.(6) involves values of the variables for all the nodes lying in Ω_i, the resulting system is, by nature, implicit.

2. First-order scheme

If the consistent mass matrix is diagonalized(or lumped), i.e. if one introduces the approximation

$$\int_\Omega N_i\, U^{n+1} d\Omega \rightarrow \int_\Omega N_i\, U_i^{n+1} d\Omega$$

in the left-hand side of Eq.(4) (but not in the right-hand side), the following first-order accurate scheme is obtained

$$\int_\Omega N_i\, U_i^{n+1} d\Omega = \int_\Omega N_i\, U^n d\Omega + RH_i^n \tag{8}$$

where $U_i^n = U(x_i, y_i, z_i, n\Delta t)$.
This scheme can be interpreted as a second-order approximation of a parabolic equation of the form

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = \frac{h^2}{6\Delta t}\, \Delta U$$

where Δ is a laplacian operator.
The first-order scheme can also be implemented by means of a splitting-up procedure. In fact, Eq.(7) is equivalent to the following two-step scheme :

$$\begin{cases} \int_\Omega N_i \; U^* d\Omega = \int_\Omega N_i \; U^n d\Omega + RH_i^n \\ \int_\Omega N_i \; U_i^{n+1} d\Omega = \int_\Omega N_i \; U^* d\Omega \end{cases} \tag{9}$$

Notice that the diffusion step introduces only a real multiplicative coefficient into the amplification factor with respect to that associated to Eq.(4) and do not disturb the phases.

3. Structure of the dissipative effect of the first-order scheme

3.1. *Element contribution*

The dissipative operator of the first-order scheme at node S_i can be written as

$$D_i U_i = \int_{\Omega_i} N_i \; (U - U_i) d\Omega \tag{10}$$

This term is composed , in fact, of the contribution of all the elements T_i^l for which the node S_i is a common vertex :

$$D_i U_i = \sum_l D_i^l \; U_i \tag{11}$$

where $D_i^l \; U_i = \int_{T_i^l} N_i (U - U_i) d\Omega.$

3.2. *Segment contribution*

As shown previously in [2], the contribution of the element can be distributed over the segments joining two adjacent nodes. Denoting by S_{k1}, S_{k2} and S_{k3} the three other nodes of the element T_i^l, $D_i^l \; U_i$ can be written as :

$$D_i^l \; U_i = K_{ik1}^l (U_{k1} - U_i) + K_{ik2}^l (U_{k2} - U_i) + K_{ik3}^l (U_{k3} - U_i)$$

where $K_{ij}^l = \int_{T_i^l} N_i N_j d\Omega.$
Using this decomposition, Eq.(10) becomes :

$$D_i U_i = \sum_k K_{ik} (U_k - U_i) \tag{12}$$

where the sum is over all the nodes surrounding the node S_i and $K_{ik} = \sum_l K_{ik}^l$.

4. First-order scheme with modulated dissipation

The dissipation term (12) of the first-order scheme can be modulated by introducing a parameter d, $0 \leq d \leq 1$. Two different schemes are obtained depending on whether the one-step or two-step form of the monotone scheme is started from .

4.1. *One-step scheme with modulated dissipation*

$$\int_\Omega N_i \; U_i^{n+1} d\Omega = \int_\Omega N_i \; U_i^n d\Omega + RH_i^n + \sum_k d_{ik} K_{ik} (U_k^n - U_i^n) \tag{13}$$

In the case $d_{ik} = 0$, we obtain a Richtmyer type scheme[3].

4.2. *Two-step scheme with modulated dissipation*

$$\begin{cases} \int_\Omega N_i \; U^* d\Omega = \int_\Omega N_i \; U^n d\Omega + RH_i^n \\ \int_\Omega N_i \; U_i^{n+1} d\Omega = \int_\Omega N_i \; U_i^* d\Omega + \sum_k d_{ik} K_{ik} (U_k^* - U_i^*) \end{cases} \tag{14}$$

In the case $d_{ik} = 0$, scheme (6) is obtained.

For the latter two schemes, it can be useful to introduce the artificial viscosity operator

$$AD_i \; U_i = \sum_k D_{ik} \; U_i \tag{15}$$

where

$$D_{ik} \; U_i = d_{ik} K_{ik} (U_k - U_i) \tag{16}$$

IV. Numerical methods

1. Artificial viscosity method

The solution obtained with second-order schemes suffers from dispersive "ripples" particularly near discontinuities whereas the first-order scheme produces no ripples but suffer from excessive numerical diffusion. The concept of artificial viscosity consists in modulating the effect of this dissipation by means of the parameter d. This coefficient may be $O(1)$ near discontinuities and $O(h^2)$ in the regions where the flow is "regular" to preserve the accuracy of the second-order scheme. Hence, at the discontinuities, the scheme is essentially the non oscillatory first-order scheme. Of course, the parameter d must be a function of a sensor which recognizes discontinuities in the flow.

An effective sensor of the presence of the shock wave can be constructed by taking the second derivative of the pressure. Letting the nodes S_i and S_k be the endpoints of the segment $\{ik\}$, we introduce the quantity

$$d_i = |\frac{p_k - 2p_i + p_{i*}}{p_k + 2p_i + p_{i*}}| \tag{17}$$

where

$$p_{i*} = p_k - 2\overrightarrow{\nabla} p_i \cdot \overrightarrow{\eta} \tag{18}$$

is the pressure extrapolated in the direction $\overrightarrow{\eta} = (x_k - x_i; y_k - y_i; z_k - z_i)$ from the node S_i and $\overrightarrow{\nabla} p_i$ is the gradient of the pressure calculated at this node. Similarly, we define

$$d_k = |\frac{p_{k*} - 2p_k + p_i}{p_{k*} + 2p_k + p_i}| \tag{19}$$

where

$$p_{k*} = p_i + 2\overrightarrow{\nabla} p_k \cdot \overrightarrow{\eta} \tag{20}$$

Note that d_i and d_k appear to be evaluations, at nodes S_i and S_k , of a non dimensional second order derivative of the pressure in the direction $\overrightarrow{\eta}$.
Finally, the coefficient of artificial viscosity d_{ik} is expressed as

$$d_{ik} = \chi \ max(d_i, d_k) \tag{21}$$

where χ is an adjustable parameter. Here the quantity d_{ik} is maximum on both sides of a shock but vanishes inside.
The disadvantage of the artificial viscosity methods lies in that they introduce an adjustable parameter whose value is not always simple to determine. In the next section, a parameter-free method will be presented.

2. Flux limiter symmetric TVD method

We can obtain second-order TVD scheme (without any adjustable parameter) using the following strategy :

i) Locate the place where the second-order accurate scheme produces oscillations
ii) Insert there the maximum dissipation($\rightarrow$ monotone scheme)
iii) Reduce or (compensate) the dissipation in the other parts of the flow .

In the one-dimensional case, the coefficient of artificial viscosity d_e (related to an element) was evaluated using the variations of a sensor q over three contiguous elements. In the multidimensional case, we work with the segments and , using the extrapolated points $\{i*\}$ and $\{k*\}$, we construct the following variations :

$$\Delta_{i*}\, q \;=\; q_i - q_{i*}\;;\; \Delta_{ik}\, q \;=\; q_k - q_i\;;\;\; \Delta_{k*}\, q \;=\; q_{k*} - q_k$$

where q_{i*} and q_{k*} are determined by equations similar to Eq.(17) and Eq.(19) . Denoting by r^+ and r^- the ratios :

$$r^+ = \frac{\Delta_{k*}\, q}{\Delta_{ik}\, q} \;\;;\;\; r^- = \frac{\Delta_{ik}\, q}{\Delta_{i*}\, q}$$

we define

$$d_{ik} = \begin{cases} 0 & \text{if } r^+ < 0 \text{ or } r^- < 0 \;(extremum) \\ \text{1-B}(r_s^+, r_s^-) & \text{otherwise} \end{cases} \qquad (22)$$

where

$$r_s^+ = \frac{min(|\Delta_{ik}\, q|, |\Delta_{k*}\, q|)}{max(|\Delta_{ik}\, q|, |\Delta_{k*}\, q|)}$$

$$r_s^- = \frac{min(|\Delta_{i*}\, q|, |\Delta_{ik}\, q|)}{max(|\Delta_{i*}\, q|, |\Delta_{ik}\, q|)}$$

and $B(r_s^+, r_s^-) = min[B(r_s^+), B(r_s^-)]$.
The function $B(r)$ is a limiter function that takes values into the interval $[0, 1]$. Some examples are

$$B(r) = \; min(\frac{2r}{1+r}, 1) \quad (\; van\; Leer\;) \qquad (23)$$

$$B(r) = min(2r, 1) \quad (\ Roe's\ superbee\ ) \tag{24}$$

Our own numerical experience indicates, that if the conservative variables are chosen as sensor, the solution presents oscillations in the non-conservative variables such as the pressure. A better choice is to take an unique sensor for the whole system of equations, so that the dissipation coefficient is the same for all the equations. In the examples illustrated later, the sensor is the Mach number($M = \frac{\|\vec{w}\|}{c}$) which is a variable very sensitive to oscillations.

V. Local mesh refinement

The use of a local mesh refinement technique is interesting for the following reasons:
- At present, simulations utilizing the Euler equations model are still onerous in terms of CPU time per degree of freedom and in terms of memory storage. It can be of crucial importance to optimize the locations as well as the total number of mesh points.
- The truly interesting results are located on a relatively small part of the computational domain: basically on the airfoil of an airplane wing.
- The flow contains local singularities such as: shocks, stagnation points, slip surfaces or wakes. Local mesh refinement techniques for the solution of the two-dimensional Euler equations and using triangular elements have been proposed among others in [7]. We describe shortly in the following how to apply such a technique in a three-dimensional context with tetrahedra.

1. Local mesh refinement procedure

We intend to introduce in an initial tetrahedrization a preferably small number of additional mesh points. The introduction of the new mesh points is decided automatically by performing a test on every element that relies on one of the criteria defined further. We proceed in the following way.

1.1. *First sub-gridding*

A pointer indicates for every tetrahedron whether it needs to be subdivided into 8 sub-tetrahedra, according to the criterion employed. It remains to construct a finite element tetrahedrization containing:
1 - The sub-tetrahedra coming from the tetrahedra divided into 8.
2 - Triangles identical to those of the initial mesh.
3 - Triangles cut in 2, 4 or 8, permitting to patch tetrahedra of type 1 to tetrahedra of type 2.

1.2. *Projection of the new boundary points*

The points along the body are positioned using the equation of the body.

1.3. *Reorganization*

At the end of the preceding step, the flow variables at each new point are set equal to the mean of the corresponding flow variables at the end-points of the segments. Then the complete topological definition of the mesh is reorganized.

2. Criteria applied for mesh refinement

2.1. *Geometrical test*

We can add nodes in regions of the computational domain using geometrical criteria which do not depend of the numerical solution. For example, we can decide to introduce nodes near a body using the criteria that all the tetrahedra which possess at least one vertex on the skin of the body are systematically dividing into 8. In an other way, for supersonic flows, we can know a priori the regions where the initial flow is not perturbed and we can thus refine the other regions.

2.2. *Test on the numerical entropy production*

We now search for a criterion that would diminish the error in the numerical solution of the Euler equations and avoid to introduce unnecessary points. A measure of the discretization error can consist in evaluating how closely the equation:

$$u\frac{\partial s}{\partial x} + v\frac{\partial s}{\partial y} + w\frac{\partial s}{\partial z} = 0 \quad , \tag{25}$$

in which $s = \dfrac{p}{\rho^\gamma}$ is a function of the entropy, is satisfied. Note that (25) is only true when the flow is smooth, so that the quantity

$$u\frac{\partial s}{\partial x} + v\frac{\partial s}{\partial y} + w\frac{\partial s}{\partial z}$$

is expected not to be small either if the approximation is not accurate enough or near a singularity. In both cases, it is advantageous to locally refine the mesh. This leads us to define a test for dividing the tetrahedra in which in the mean:

$$|u\frac{\partial s}{\partial x} + v\frac{\partial s}{\partial y} + w\frac{\partial s}{\partial z}| \;\geq\; COEF\; |u\frac{\partial s}{\partial x} + v\frac{\partial s}{\partial y} + w\frac{\partial s}{\partial z}|_{max} \tag{26}$$

where

$$|u\frac{\partial s}{\partial x} + v\frac{\partial s}{\partial y} + w\frac{\partial s}{\partial z}|_{max} \;=\; max|u\frac{\partial s}{\partial x} + v\frac{\partial s}{\partial y} + w\frac{\partial s}{\partial z}|.$$

The value of the parameter $COEF$ is obviously problem-dependent.

VI. Numerical experiments

1. Boundary conditions

As described in section (II.1.2), the boundary terms of Eq.(6)

$$-\Delta t \int_\Gamma \{F(\bar{U})n_x + G(\bar{U})n_y + H(\bar{U})n_z)\}d\Gamma \tag{27}$$

contain the physical boundary conditions. They are taken into account through the vector $\bar{U}$ which is calculated partly from quantities depending of the "interior" value U^n and also partly from quantities specified by the physical boundary conditions. We precise now this procedure for the different families of boundary conditions present in the solution of the Euler equations.

- For the wall condition : The vector $\bar{U}$ must satisfy the slip condition :

$$\bar{\rho}\bar{u}\ n_x\ +\ \bar{\rho}\bar{v}\ n_y\ +\ \bar{\rho}\bar{w}\ n_z\ =\ 0\ . \tag{28}$$

It is easy to see that only the pressure $\bar{p}$ corresponding to $\bar{U}$ is neccessary to evaluate the flux vectors $F(\bar{U})$, $G(\bar{U})$ and $H(\bar{U})$. Using the condition (28), Eq.(27) reduces to the following expression :

$$-\Delta t \int_\Gamma \begin{pmatrix} 0 \\ \bar{p}\ n_x \\ \bar{p}\ n_y \\ \bar{p}\ n_z \\ 0 \end{pmatrix} d\Gamma \tag{29}$$

We choose the pressure $\bar{p}$ equal to the internal pressure :

$$\bar{p} = p(U^n)$$

- For inflow and outflow conditions, including the conditions at infinity : We use a flux splitting inspired by an upwind scheme[4]. This way, in Eq.(25) , the vector integrand :

$$K = F(\bar{U})n_x + G(\bar{U})n_y + H(\bar{U})n_z$$

is approximated in each node S_i of the boundary by a vector $K(i)$ which is constructed in terms of the value of U at node S_i and of conditions at infinity using the flux splitting. For example,

$$K(i) = E_\infty^+ U_i + E_\infty^- U_\infty$$

where

$$E_\infty = A(U_\infty)n_x + B(U_\infty)n_y + C(U_\infty)n_z$$

2. Numerical experiments

a) Shock tube problem : To illustrate the accuracy of the different schemes for transient problems, we present numerical results for the shock tube problem introduced by Sod[5]. In this problem, an initial discontinuity in the thermodynamical state of the gas breaks into a shock wave followed by a contact discontinuity and a rarefaction wave. A finite element mesh of 909 ($101 \times 3 \times 3$) nodes is used and the initial condition at $t = 0$ is specified by the data

$$\begin{cases} \rho = 1.000 \ ; \ u = 0.00 \ ; \ v = 0.00 \ ; \ w = 0.00 \ ; \ p = 1.00 & 0 \leq x \leq \frac{1}{2} \\ \rho = 0.125 \ ; \ u = 0.00 \ ; \ v = 0.00 \ ; \ w = 0.00 \ ; \ p = 0.10 & \frac{1}{2} < x \leq 1 \end{cases}$$

The profiles of density, velocity, pressure and temperature are compared with the analytical solution at $t = 0.16$.

In the numerical simulations illustrated in Fig. 2, the artificial viscosity method is the least dissipative. Notice that, for this method, the oscillations are rapidly damped using the two-step scheme (13) which generally gives a better overall accuracy than the one-step scheme (12). Nevertheless, we note that the amplitude of the oscillations is reduced, for the scheme (12), when the CFL number and consequently the time step increases (The CFL number is a non dimensional number defined by the following relation : $CFL = \lambda \Delta t / h$, where h is a characteristic mesh spacing of the elements and λ is an eigenvalue associated to the Jacobian matrices A, B and C). This situation can perhaps be explained by the fact that the dispersion error decreases when the CFL number increases in the case of linear problems. The TVD method gives monotonic results at the shock but is more dissipative in the regular regions of the flow.

b) Steady transonic and supersonic flows in a bumped channel : We now consider the extension in 3-D of a problem which has been proposed for the GAMM Workshop[6]. A channel of lengh 5, of depth 2 and of width 1 contains an obstacle of

chordlength 1, whith the shape of a 4.2% thick circular arc. We use a triangular finite element mesh from which a cut along the y-axis is similar of the 2-D mesh proposed by the GAMM. The tetrahedrisation contains 4599 nodes and 17280 tetrahedra (Fig. 3). The behaviours of the different schemes are compared for $M_\infty = 0.85$ (transonic flow) and for $M_\infty = 1.5$ (supersonic flow). All the calculations are carried out with a CFL number equal to 0.8 for the supersonic case and with a CFL number equal to 0.6 for the transonic case. A local timestepping technique is used to improve the convergence towards the steady-state solution. We present plots of the following non dimensional numbers :

the pressure coefficient : $C_p = 2\dfrac{p - p_\infty}{\rho_\infty \| \vec{w}_\infty \|^2}$ the Mach number : $M = \dfrac{\| \vec{w} \|}{c}$
and the entropy deviation : $\Sigma = (p/p_\infty)/(\rho/\rho_\infty)^\gamma - 1$.

In the transonic case, the initial subsonic flow is accelerated on the body. A supersonic region is shaped and a shock appears. There is a production of entropy corresponding to the shock, which propagates to the infinity. Note that the solutions illustrated in Fig. 4 have not completely reached the steady-state.
For the supersonic flow(see Fig. 5), two oblique shocks are formed at the edges of the obstacle. There is two productions of entropy corresponding to the oblique shocks. The sum of this productions propagates to the infinity.

In the case of the artificial viscosity method, the shocks spread out over two or three elements, approximatively. The TVD method is more dissipative, not necessarely near the discontinuities, but in the smooth regions of the flow.

c) Steady supersonic flow around a wing body : The last problem considered is the steady-state solution of a supersonic flow (M_∞=1.5) around a model of wing. Contrary to the previous problems, this one requires the solution of a true 3-D flow. The equation of the skin of the wing is defined as follows:

$$0 \leq x \leq 1 \ , \ 0 \leq y \leq 1 \ , \ x \leq y$$
$$z = (1 - x)[-a + \sqrt{r^2 - (f - 0.5)^2}]$$

where $a = 2.95519$, $r = 2.99719$ and $f = \dfrac{y}{x}$. The maximal height of the wing is 0.042.

This problem is chosen to illustrate the local mesh refinement technique. For example, we can refine the initial mesh(Fig. 6) near the wing using the criteria that all the tetrahedra which possess at least one vertex on the skin of the wing are systematically dividing into 8, as illustrated in Fig. 7. In another way, since the flow is supersonic, the initial conditions are not perturbed behind the wing body. We thus have divided into 8 all the tetrahedra from which the gravity center (x_g, y_g, z_g) satisfies the following inequalities:

$$y_g \geq -0.2 \ , \ x_g \leq 0.2 + y_g$$

This mesh is illustrated in Fig. 8 and is used to perform the computation of the problem. The solution calculated with the artificial viscosity method is illustrated in Fig. 9, after 500 timesteps. Two oblique shocks are formed at the edges of the wing. The two shocks interact at the tip of the body to produce a third shock and a slip surface. The numerical solution is thus used to refine again the mesh according to criteria (26) with $COEF = 0.1$. The new nodes are principally introduced over the skin of the wing and, particularly, at the shock's location(see Fig. 10). Using this refined mesh, we recalculate a new solution illustrated in Fig. 11 after 500 new timesteps. We note that the discontinuities are better located. Finally, we present a solution calculated with the TVD method on the last mesh and using the interpolated solution, obtained with the artificial viscosity method, as initial conditions(see Fig. 12).

VII. Conclusion

In this report, we have presented 3-D calculations using either an artificial viscosity method or a flux limiter TVD method. These calculations have been performed over unstructured meshes which have been constructed using a local mesh refinement technique. The artificial viscosity method appears to be the least diffusive but introduces a parameter which is problem dependent. On the other hand, the TVD method has not this disadvantage but it often adds dissipation in the smooth regions of the flow.

Numerous subsequent studies remain to be done. For example, we can study other type of sensors for the artificial viscosity method. We can also introduce, in the code, characteristic TVD methods already described for two-dimensional calculations in [2] and which appear less dissipative than the present TVD method. Finally, we could develop codes using upwind schemes instead of schemes based on a Lax-Wendroff type scheme with artificial diffusion added.

VIII. References

[1] J. DONEA, A Taylor-Galerkin Method for Convective Transport Problems; Internat. J. Numer. Methds. Engrg. 20(1984) 101-120 .

[2] V. SELMIN, Finite Element Solution of Hyperbolic Equations. II.Two Dimensional Case; INRIA Research Report (1987) .

[3] F. ANGRAND, A. DERVIEUX, V. BOULARD, J. PERIAUX and G. VIJAYASUNDARAM; Transonic Euler Simulations by Means of Finite Element Explicit Schemes, AIAA Paper 83-1924 .

[4] G. VIJAYASUNDARAM, Résolution numérique des équation d' Euler pour des écoulement transsoniques avec un schéma de Godunov en Eléments Finis; Thesis, University Paris VI (1983) .

[5] G.A. SOD, A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws; J. Comput. Phys. 27(1978) 1-31 .

[6] A.W. RIZZI and H. VIVIAND (eds.), Numerical Methods for the Computation of Inviscid Transsonic Flows with Shock waves; Vieweg and Sohn, Braunschweig-Wiesbaden (1982) .

[7] B. PALMERIO, Self-Adaptative F.E.M. Algorithms for the Euler Equations; INRIA Research Report 338(1984) .

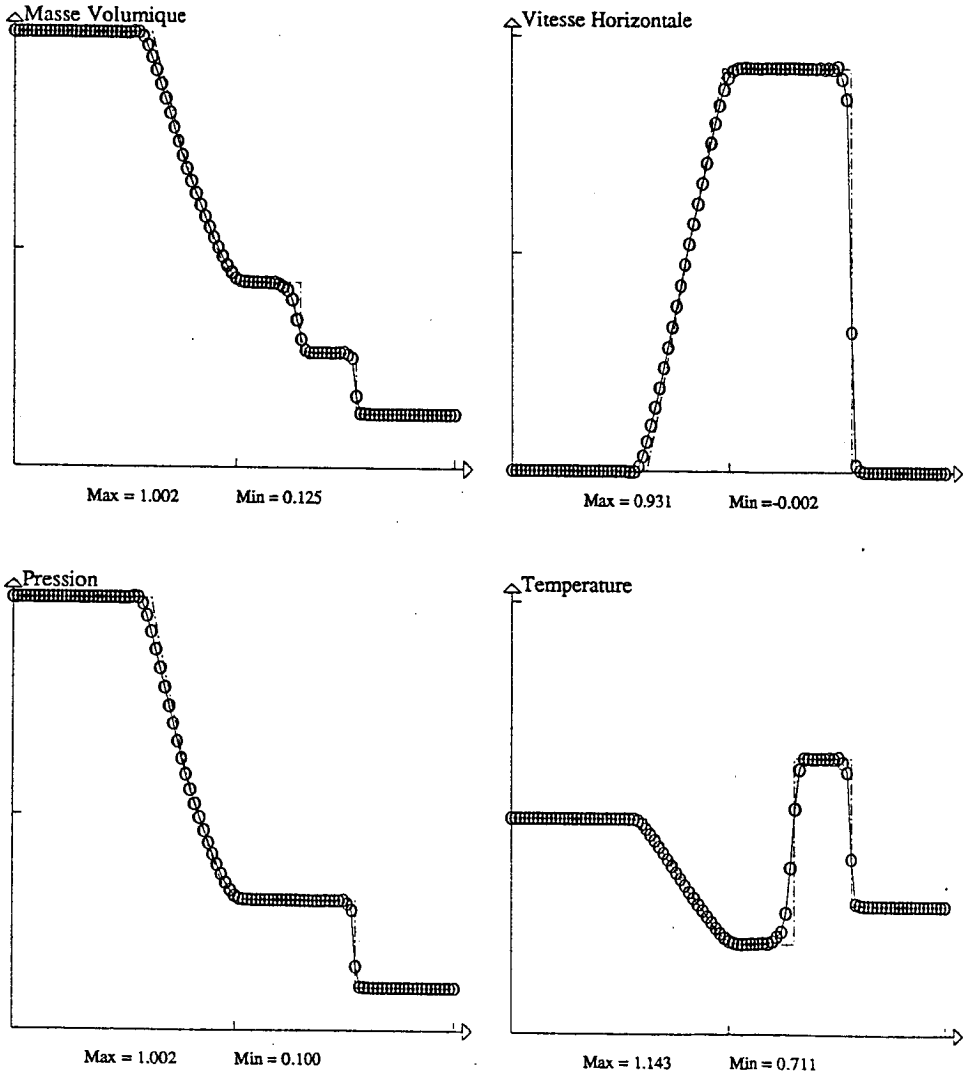


Fig. 2.a : Shock tube problem. Artificial viscosity method. Consistent mass matrix , $\chi = 3.0$, CFL=0.550.

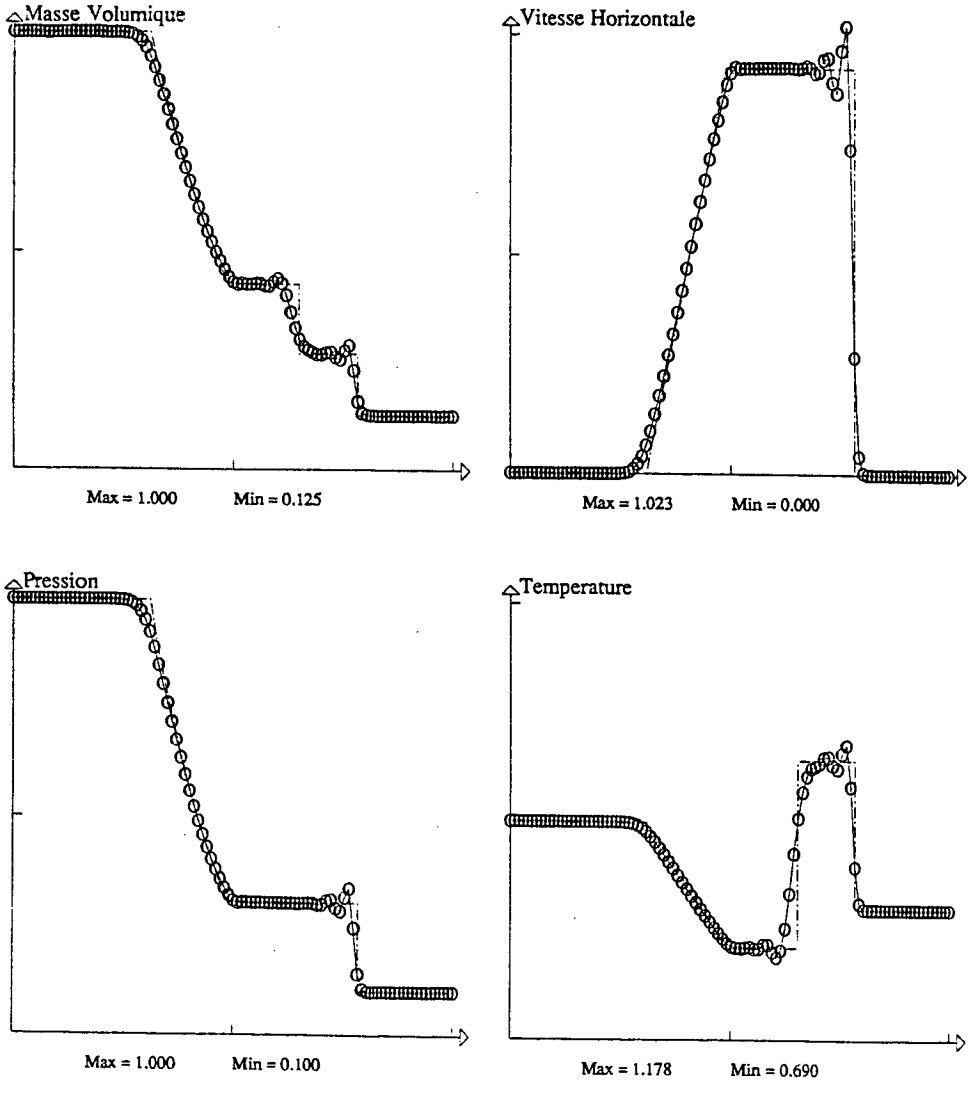


Fig. 2.b : Shock tube problem. Artificial viscosity method. Lumped mass matrix, $\chi = 3.0$, CFL=0.550.

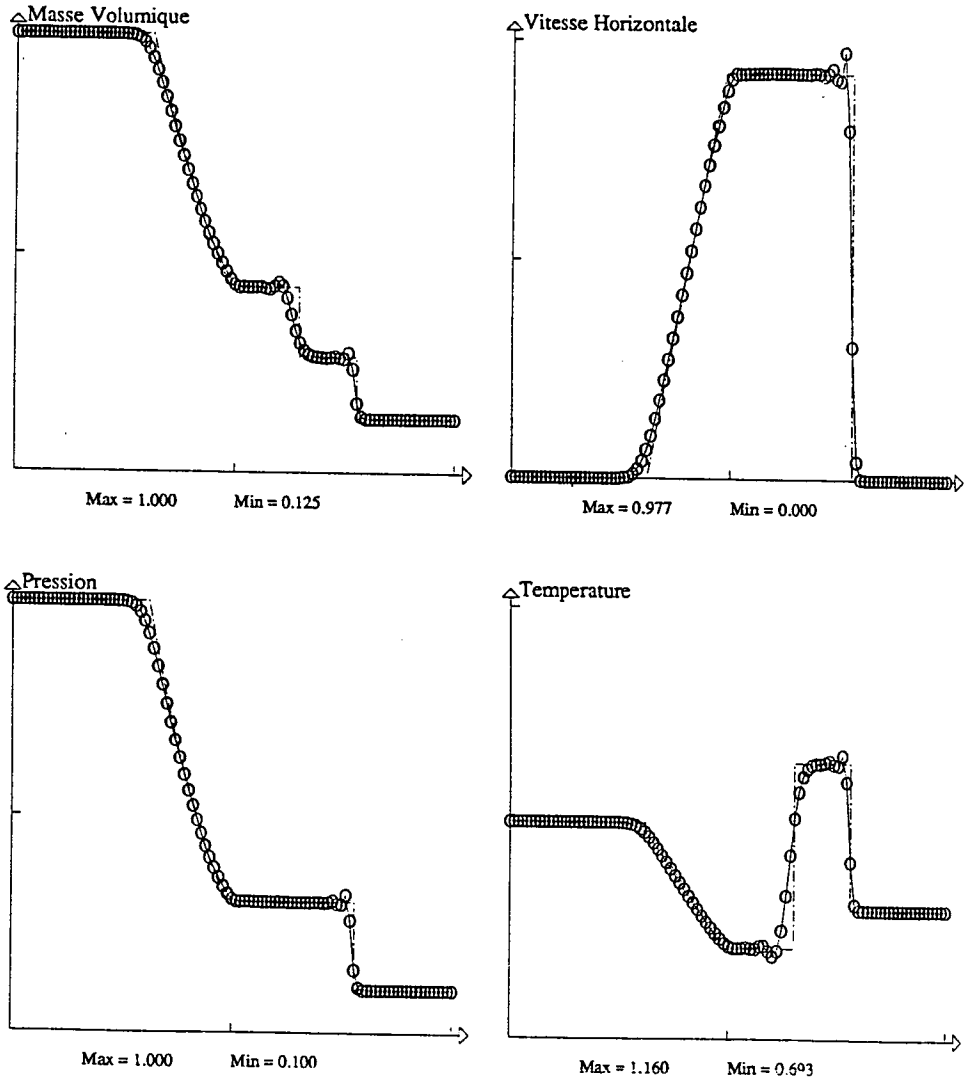


Fig. 2.c : Shock tube problem. Artificial viscosity method. Lumped mass matrix, $\chi = 3.0$, CFL=0.800.

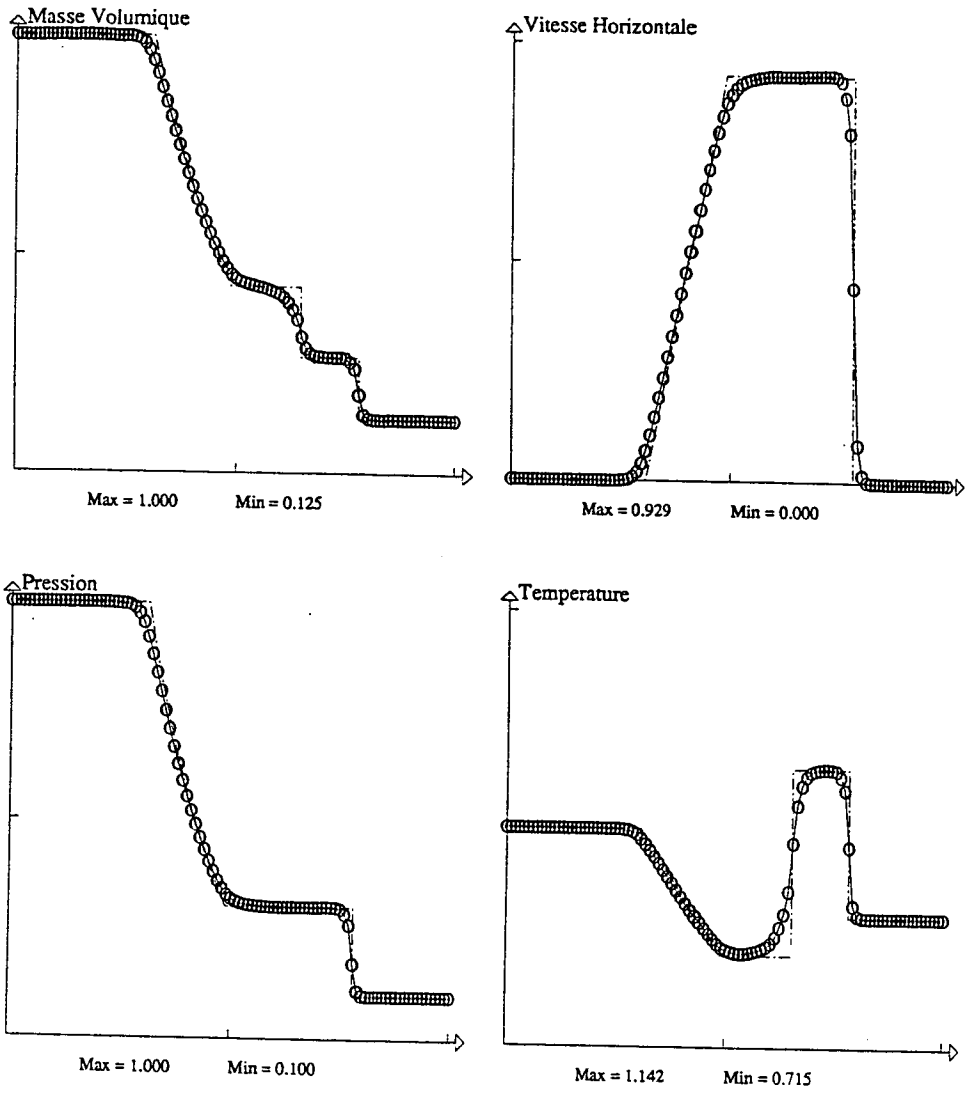


Fig. 2.d : Shock tube problem. TVD method. Consistent mass matrix , CFL=0.550.

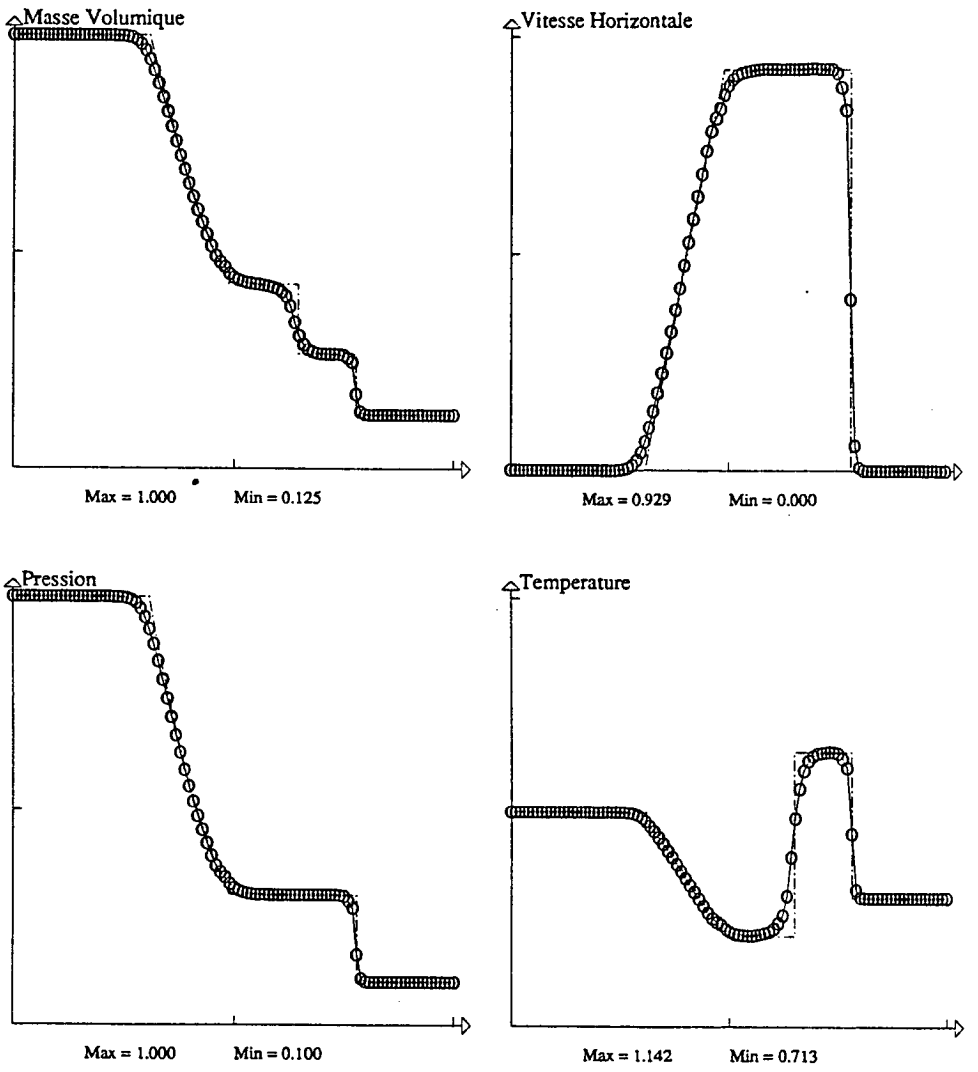


Fig. 2.e : Shock tube problem. TVD method. Lumped mass matrix , CFL=0.800.

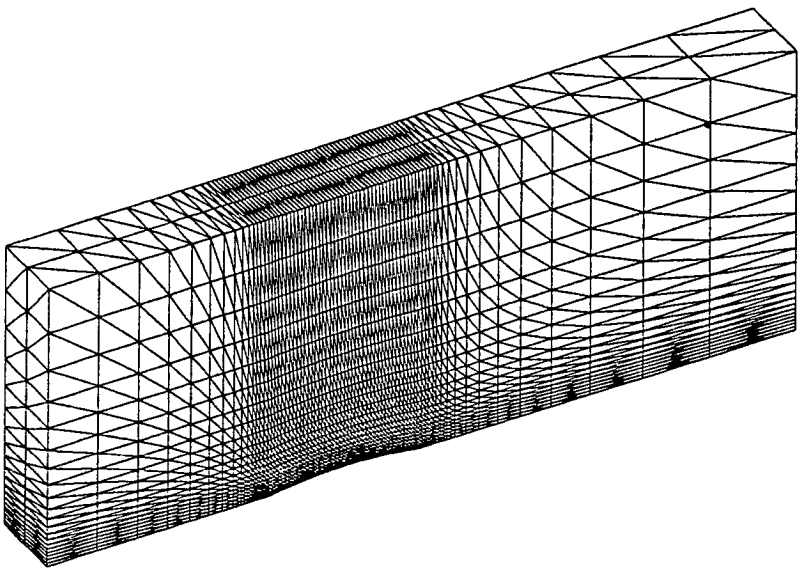
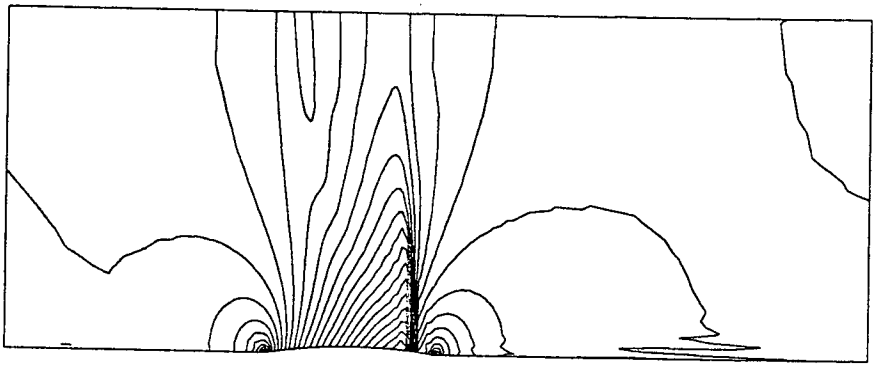


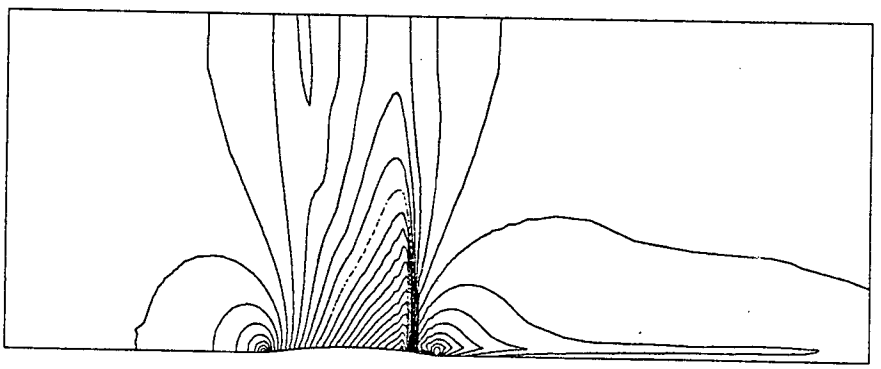
Fig. 3 : Bumped channel , mesh.

Artificial viscosity method
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.6 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting

CP-LINES : MIN=-0.450 MAX=0.800 DLTA=0.050



MACH-LINES : MIN=0.650 MAX=1.300 DLTA=0.025



ENTROPY-LINES : MIN=0.000 MAX=0.013 DLTA=0.001

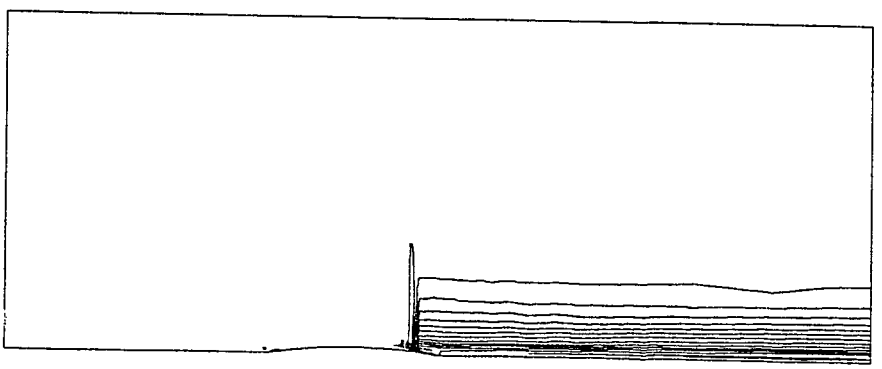
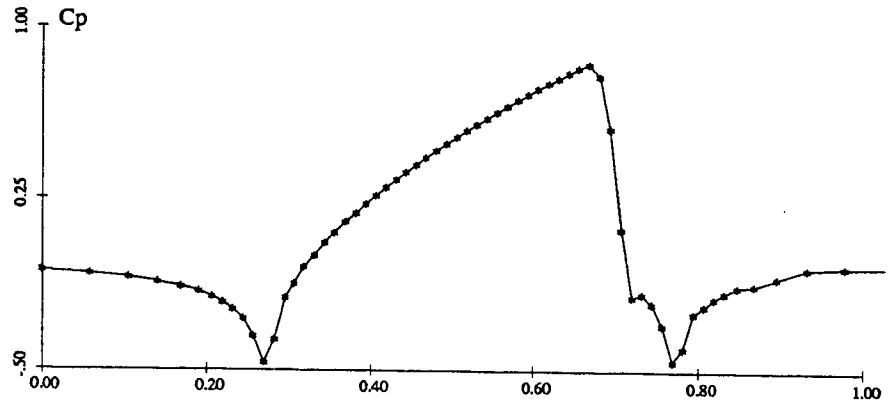
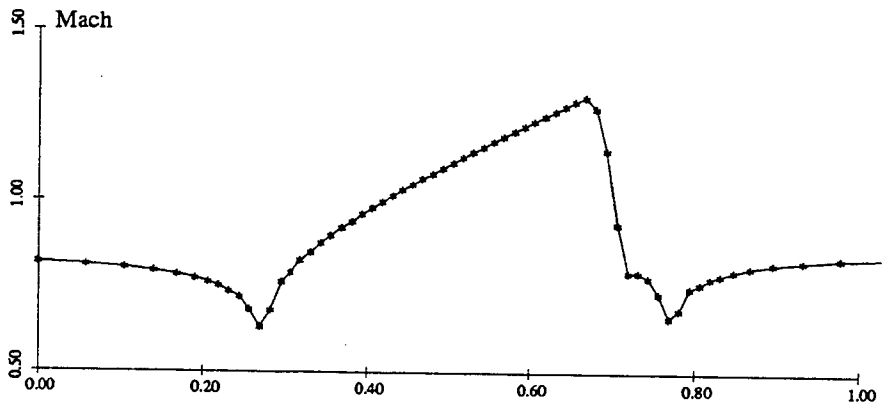


Fig. 4.a : Transonic flow in a bumped channel. Artificial viscosity method.

Artificial viscosity method
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.6 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting





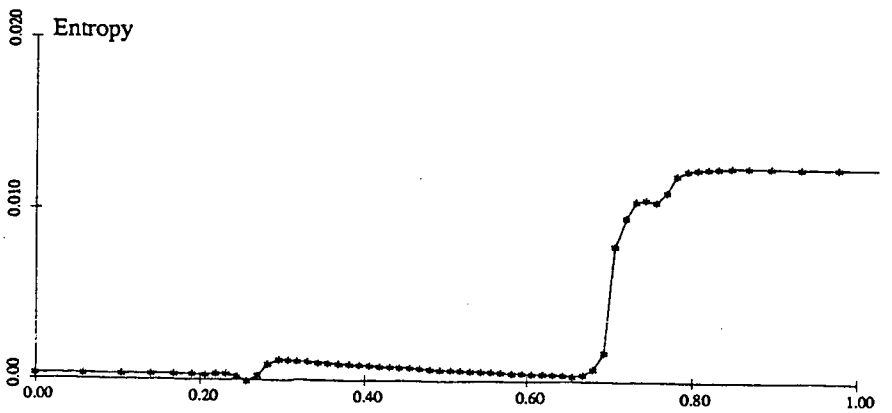
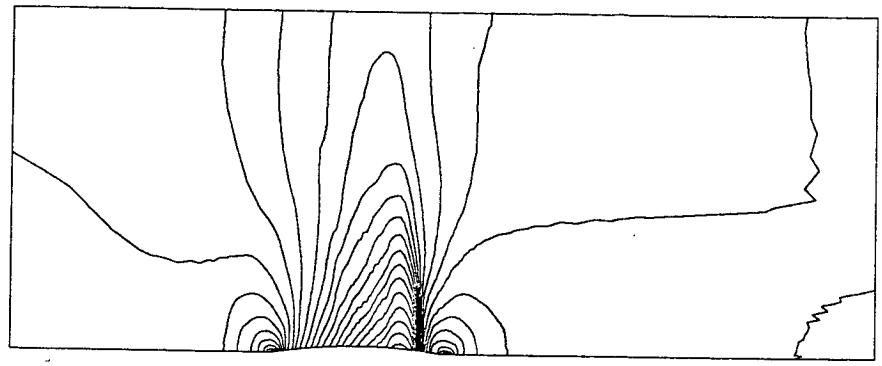


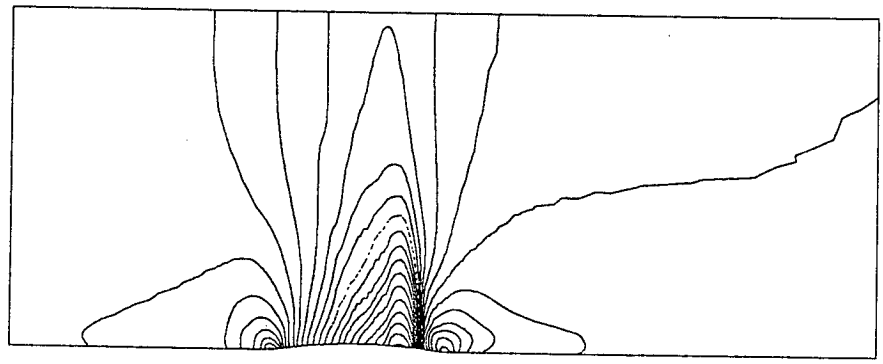
Fig. 4.a : Continued.

Flux limiter TVD method
Limiter : Roe 's superbee
Lumped mass matrix ; CFL=0.6
Local time stepping ; b.c. : Vijaya. flux splitting

CP-LINES : MIN=-0.400 MAX=0.750 DLTA=0.050



MACH-LINES : MIN=0.675 MAX=1.250 DLTA=0.025



ENTROPY-LINES : MIN=0.000 MAX=0.012 DLTA=0.001

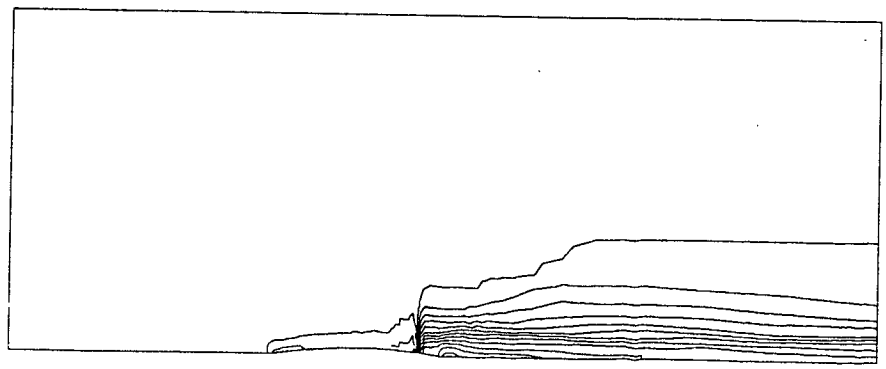


Fig. 4.b : Transonic flow in a bumped channel. TVD method.

Flux limiter TVD method
Limiter : Roe 's superbee
Lumped mass matrix ; CFL=0.6
Local time stepping ; b.c. : Vijaya. flux splitting

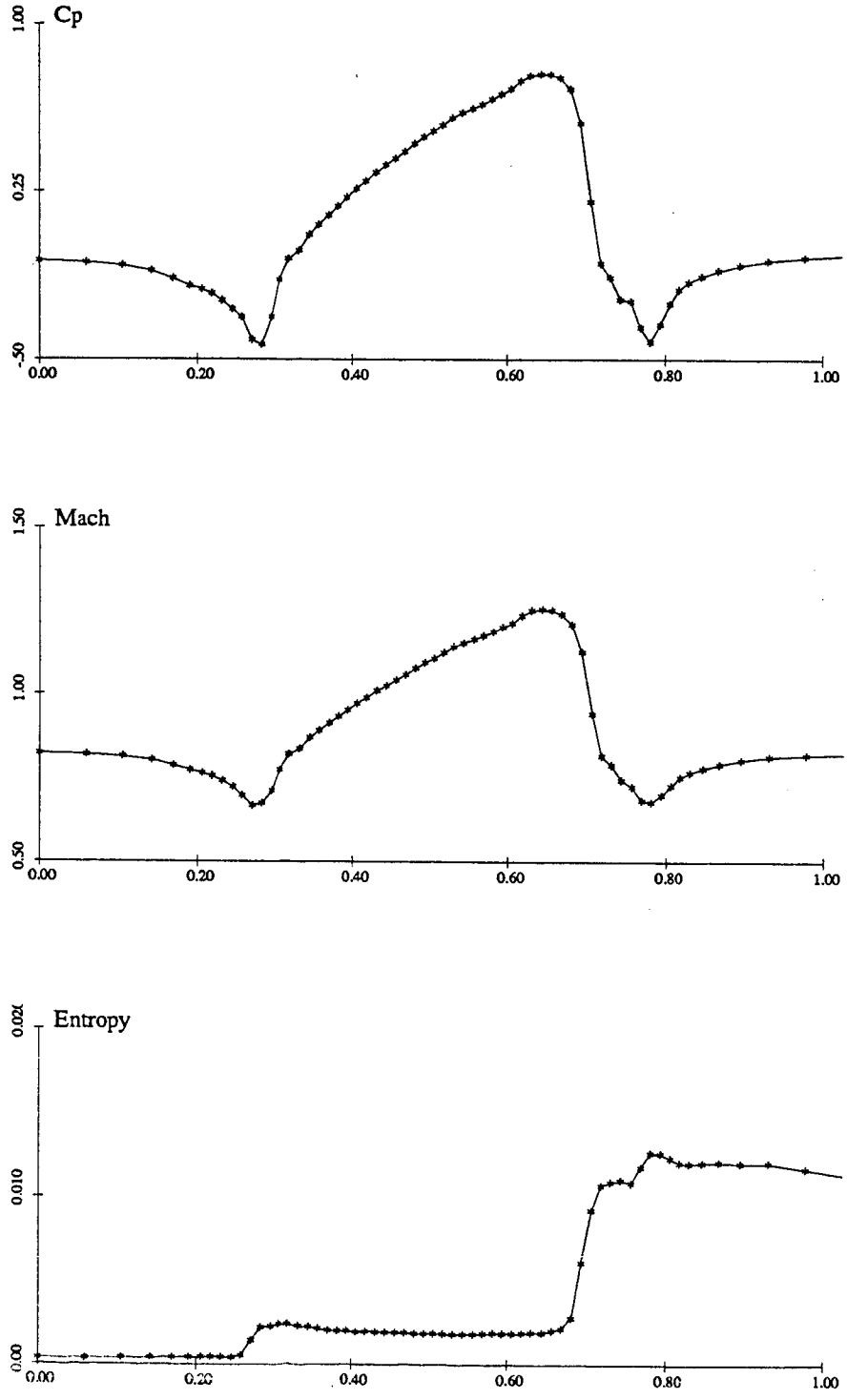


Fig. 4.b : Continued.

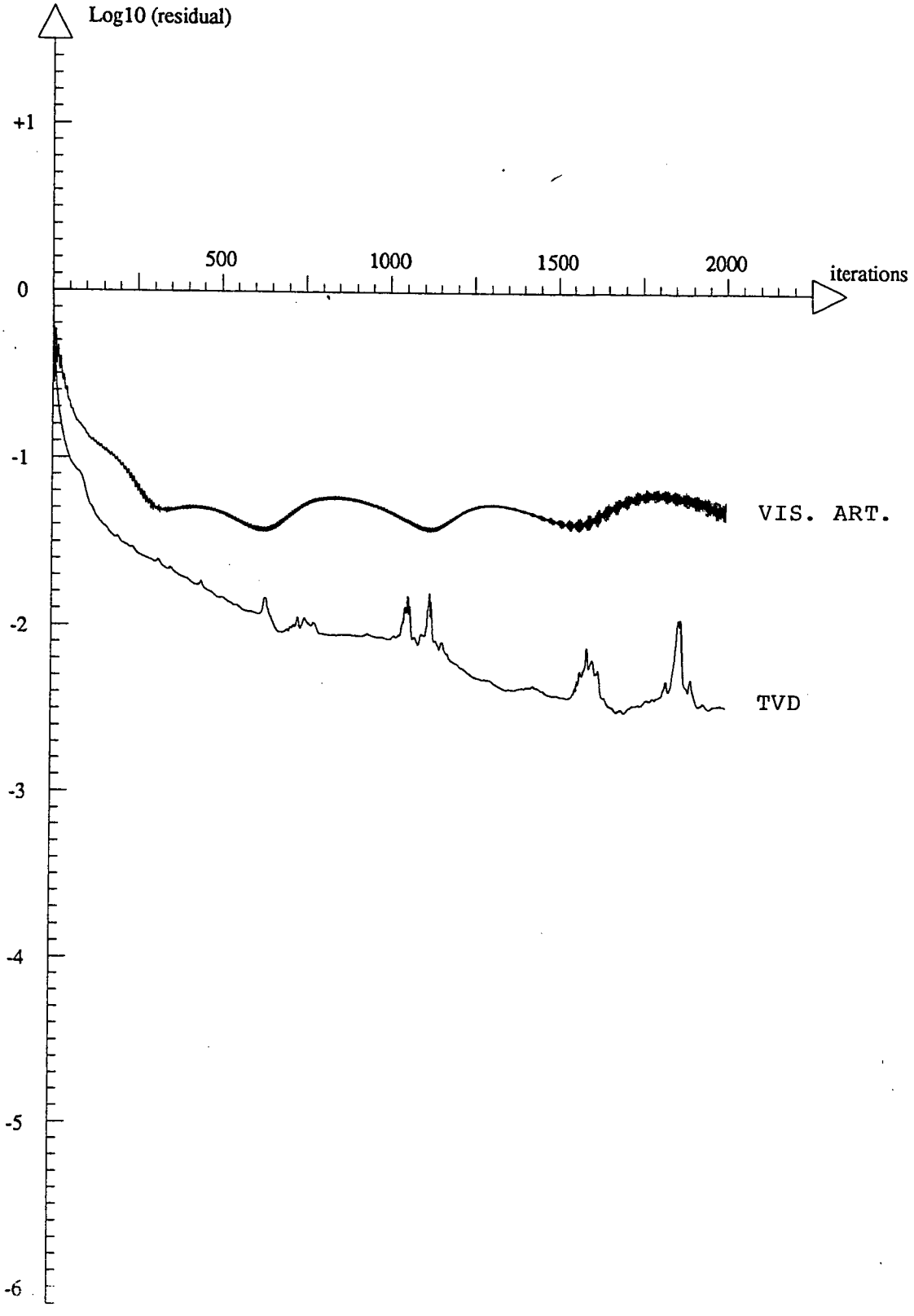
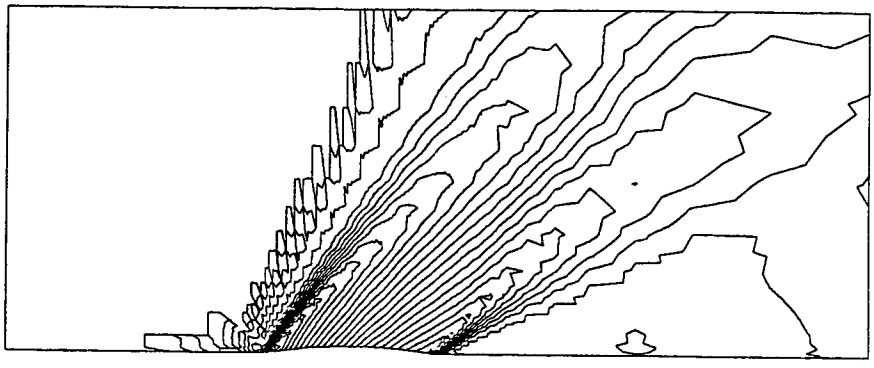


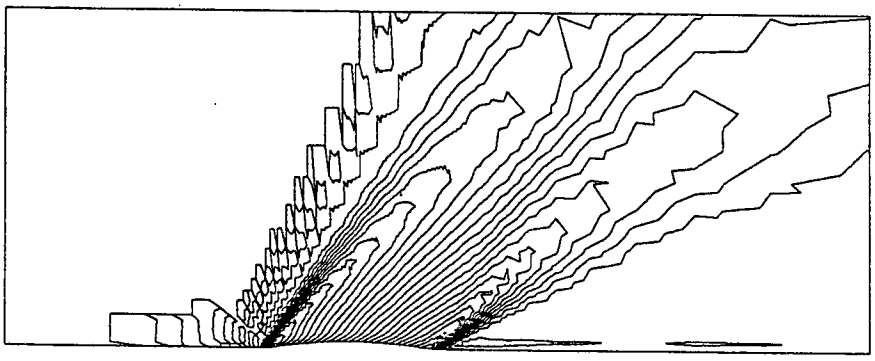
Fig. 4.c : Transonic flow in a bumped channel. Residual curves.

Artificial viscosity method
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.8 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting

CP-LINES : MIN=-0.350 MAX=0.225 DLTA=0.025



MACH-LINES : MIN=1.200 MAX=1.800 DLTA=0.025



ENTROPY-LINES : MIN=0.000 MAX=0.025 DLTA=0.001

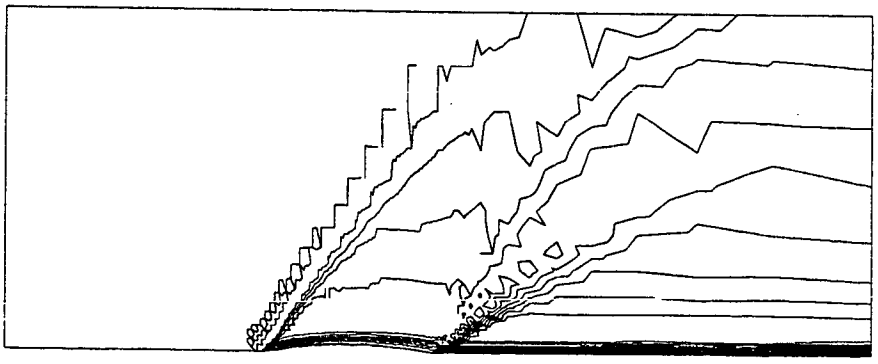


Fig. 5.a : Supersonic flow in a bumped channel. Artificial viscosity method.

Artificial viscosity method
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.8 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting

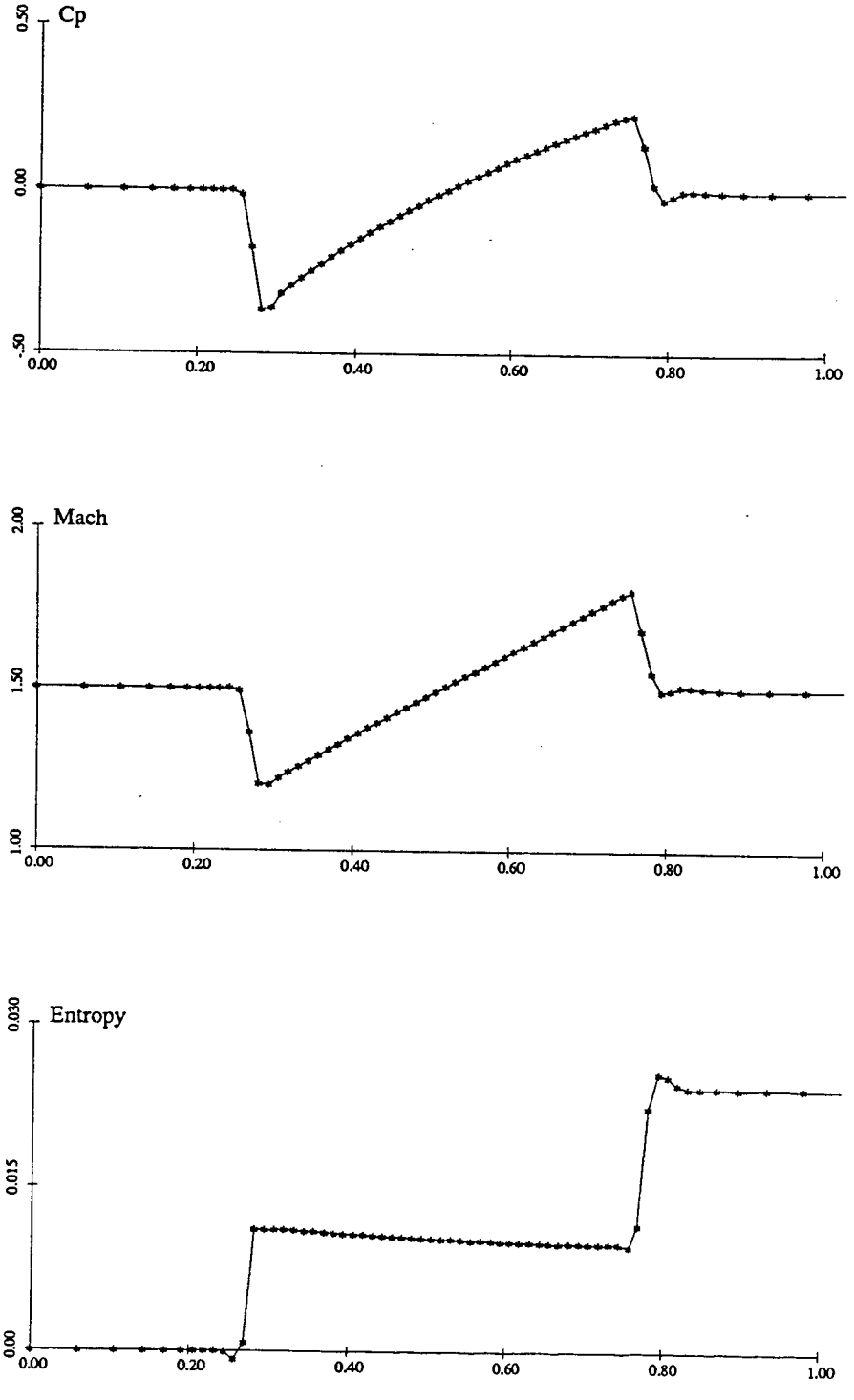
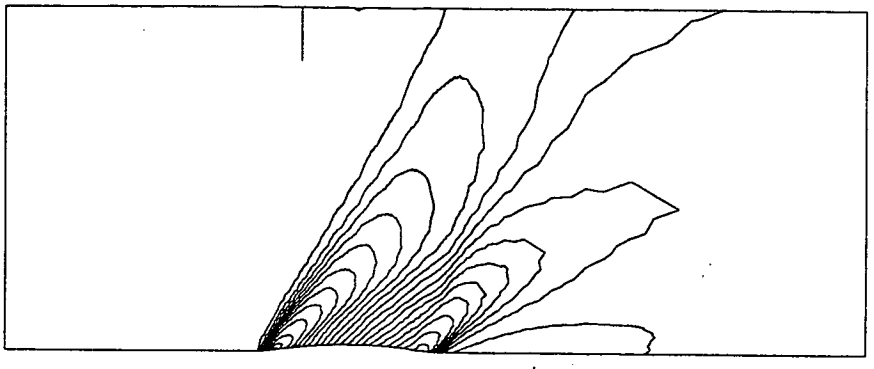


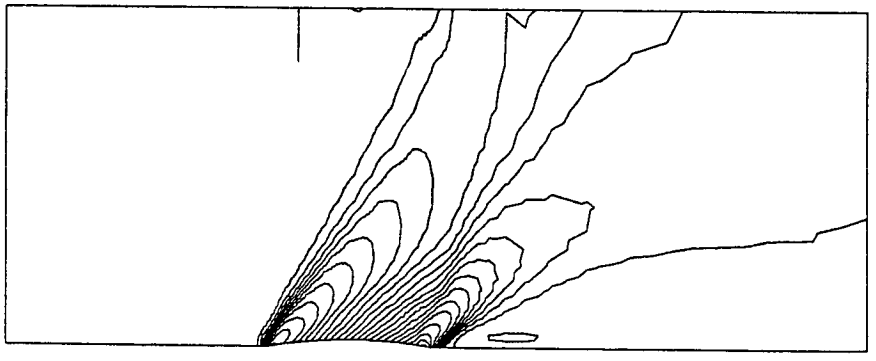
Fig. 5.a : Continued.

Flux limiter TVD method
Limiter : Roe 's superbeee
Lumped mass matrix ; CFL=0.8
Local time stepping ; b.c. : Vijaya. flux splitting

CP-LINES : MIN=-0.350 MAX=0.200 DLTA=0.025



MACH-LINES : MIN=1.200 MAX=1.775 DLTA=0.025



ENTROPY-LINES : MIN=0.000 MAX=0.026 DLTA=0.001

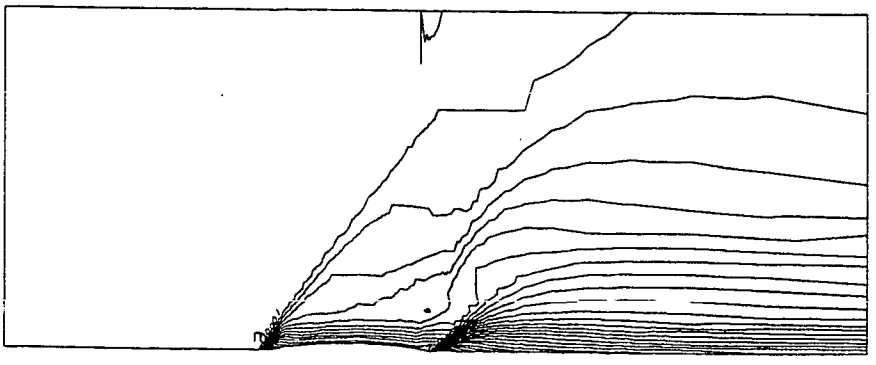


Fig. 5.b : Supersonic flow in a bumped channel. TVD method.

Flux limiter TVD method
Limiter : Roe 's superbee
Lumped mass matrix ; CFL=0.8
Local time stepping ; b.c. : Vijaya. flux splitting

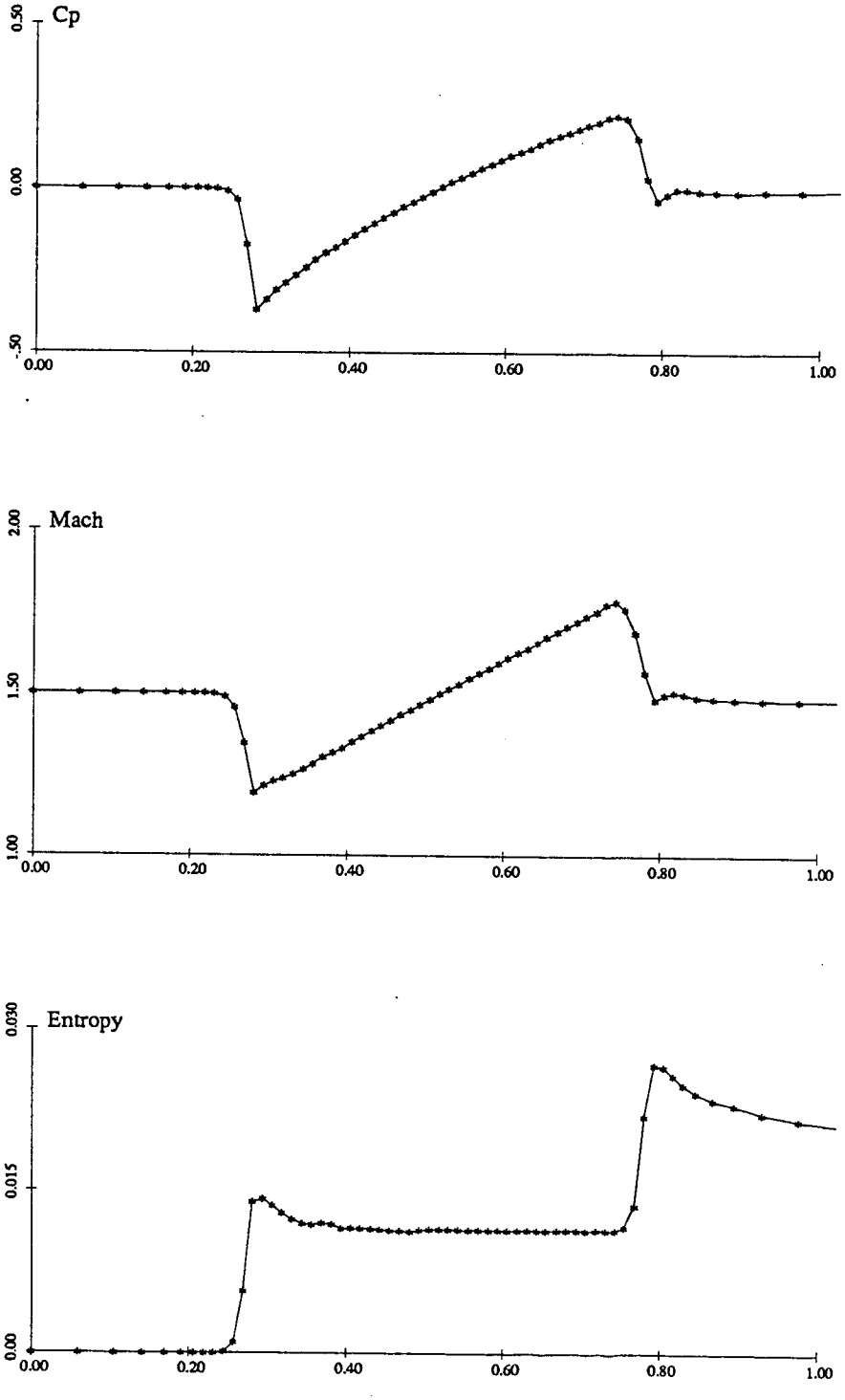


Fig. 5.b : Continued.

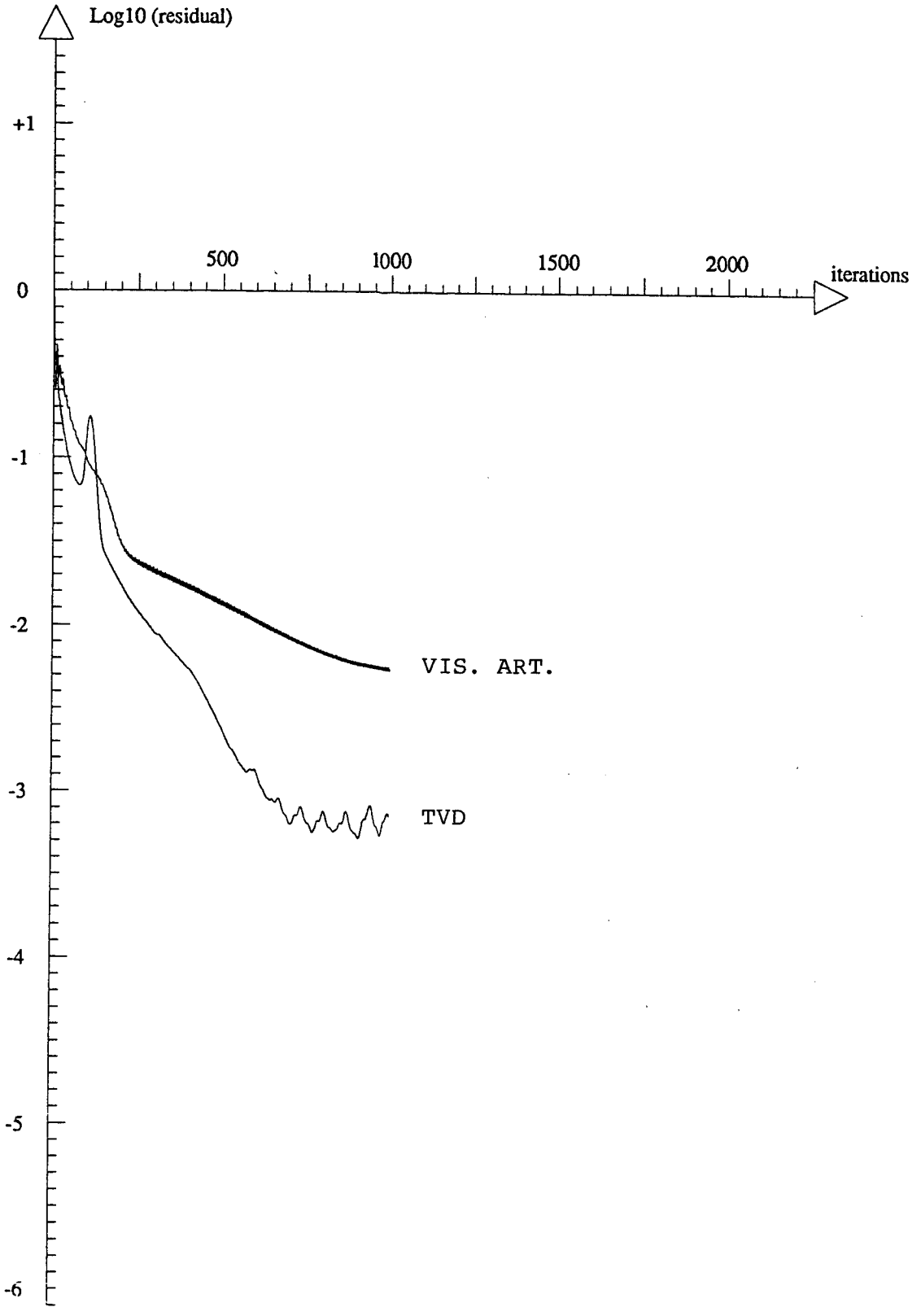
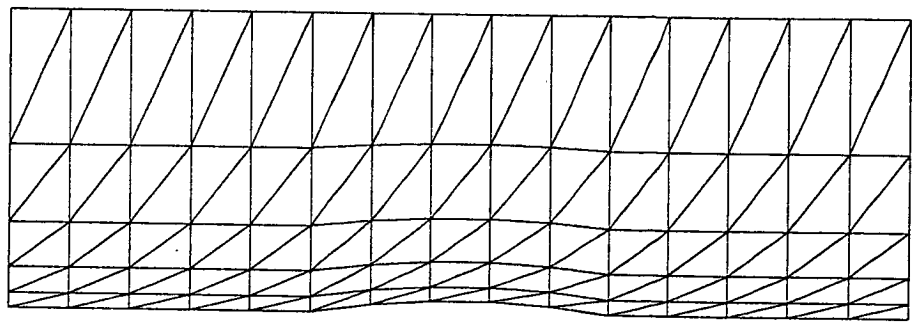


Fig. 5.c : Supersonic flow in a bumped channel. Residual curves.

INITIAL MESH
NO MESH-REFINEMENT
NS=1056 ; NT= 4500

MESH : CUT CORRESPONDING TO x=0.0



MESH : CUT CORRESPONDING TO z=zmin

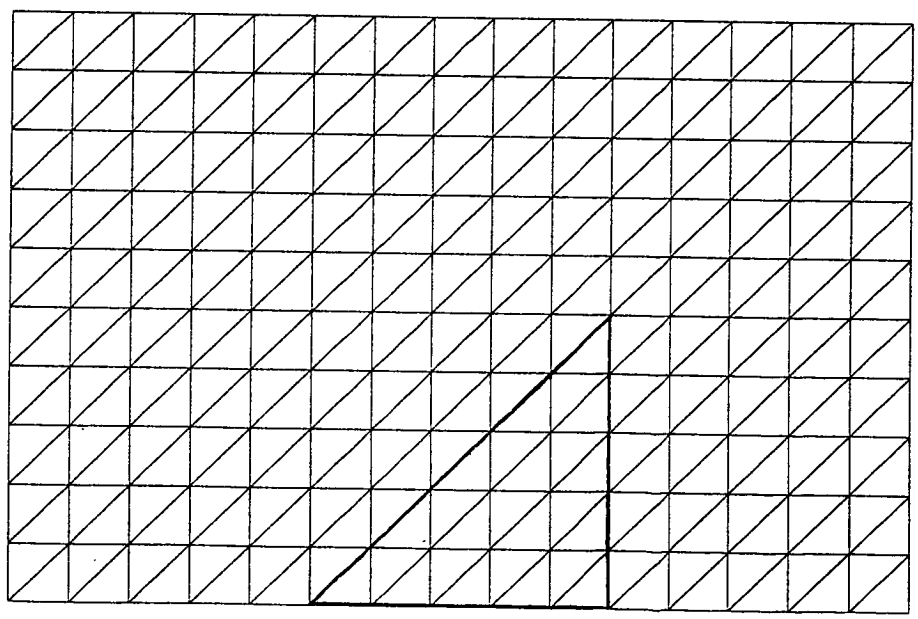
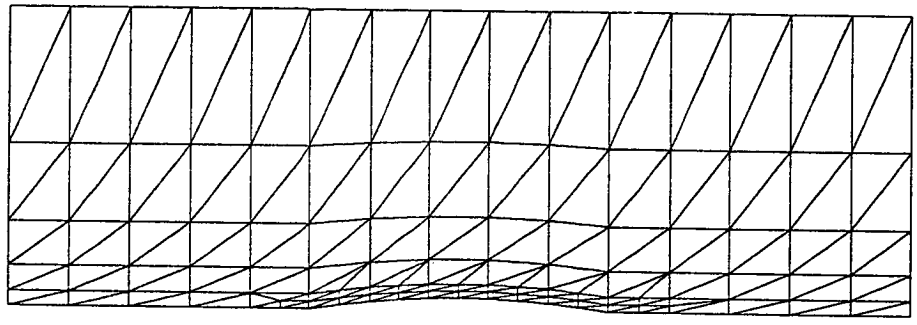


Fig. 6 : Supersonic flow past a wing. Initial mesh.

SECOND MESH
MESH-REFINEMENT ONLY NEAR THE BODY
NS=1305 ; NT= 5647

MESH : CUT CORRESPONDING TO x=0.0



MESH : CUT CORRESPONDING TO z=zmin

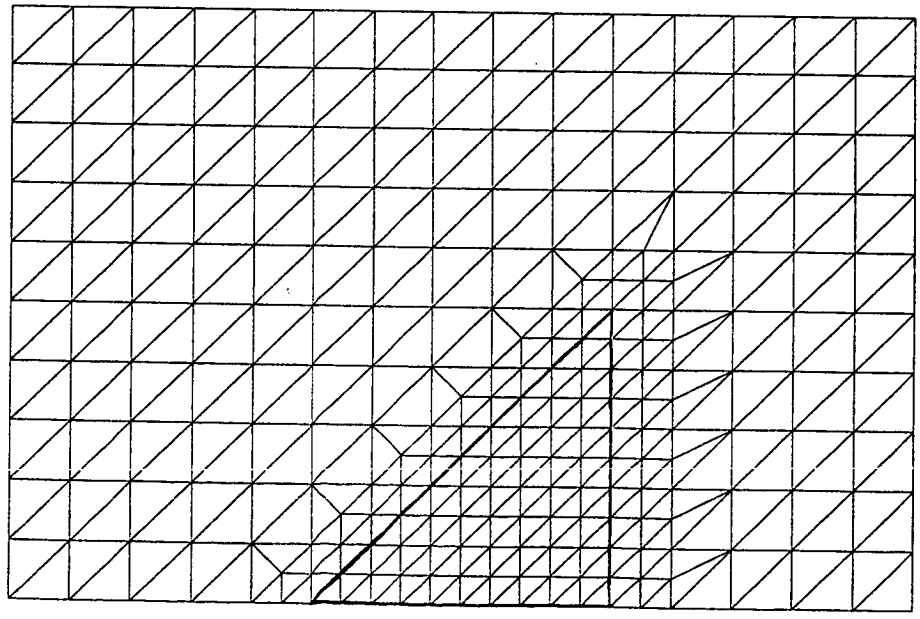
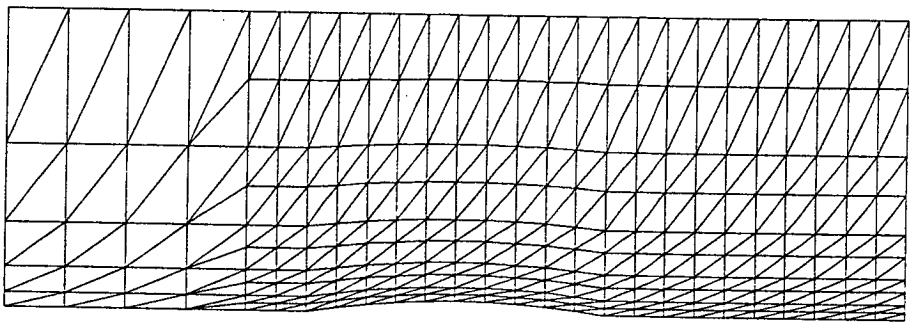


Fig. 7 : Supersonic flow past a wing. Refinement of the initial mesh using the first geometrical criteria.

THIRD MESH
MESH-REFINEMENT ONLY IN A PART OF THE INITIAL MESH
NS=3597 ; NT= 17500

MESH : CUT CORRESPONDING TO x=0.0



MESH : CUT CORRESPONDING TO z=zmin

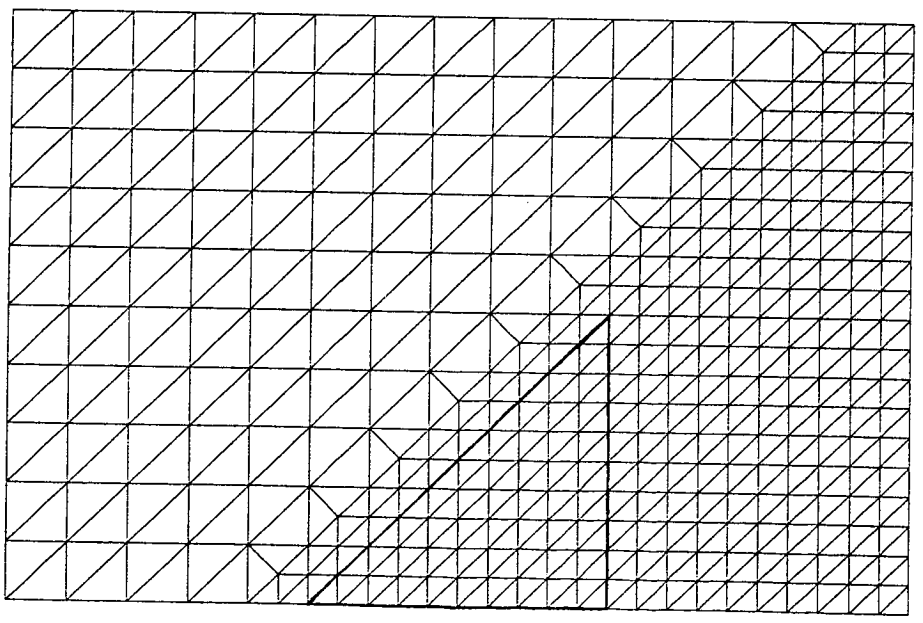
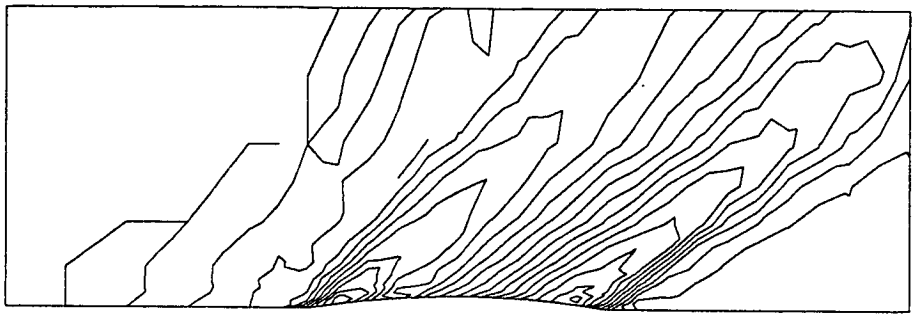


Fig. 8 : Supersonic flow past a wing. Refinement of the initial mesh using the second geometrical criteria.

Third mesh. Artificial viscosity method.
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.8 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting

CP-LINES : MIN=-0.160 MAX=0.180 DLTA=0.020



CP-LINES : MIN=-0.200 MAX=0.200 DLTA=0.020

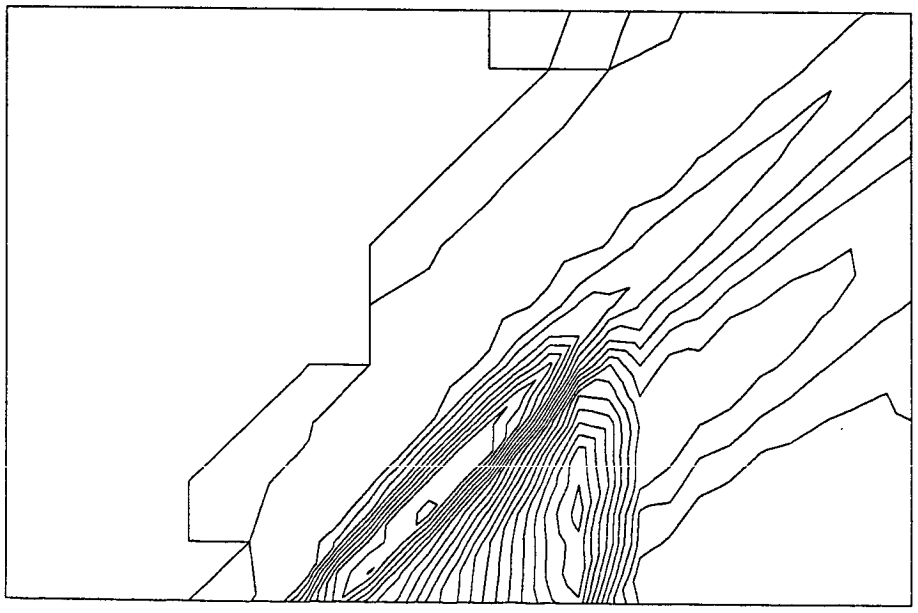
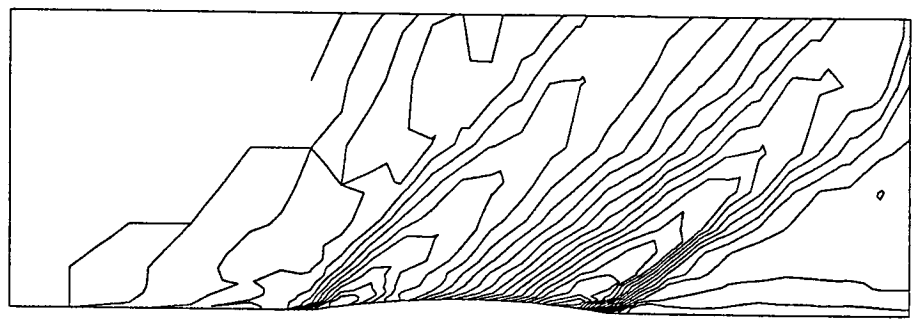


Fig. 9 : Supersonic flow past a wing. Artificial viscosity method. Solution obtained with the third mesh.

Third mesh. Artificial viscosity method.
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.8 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting

MACH-LINES : MIN=1.340 MAX=1.720 DLTA=0.020



MACH-LINES : MIN=1.300 MAX=1.740 DLTA=0.020

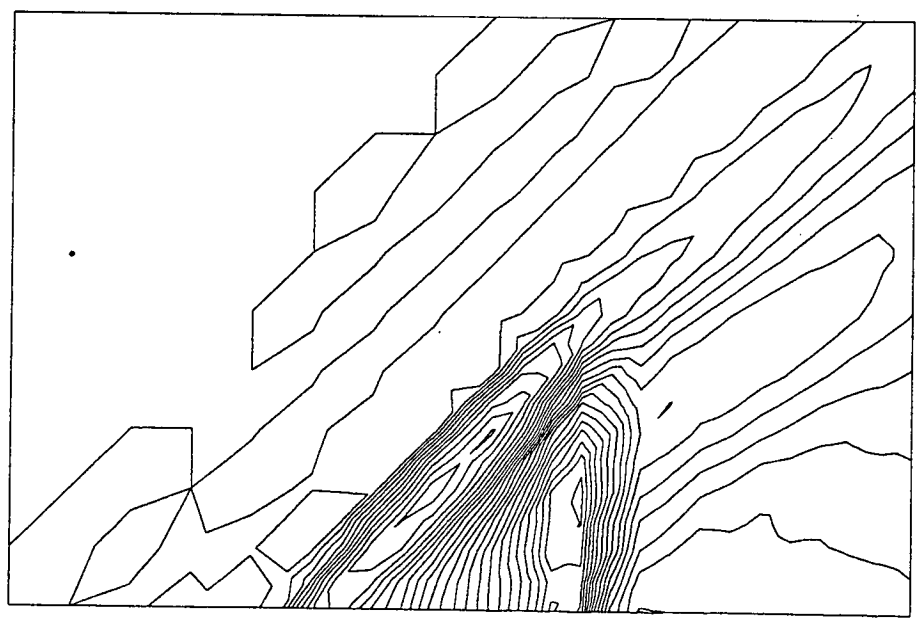
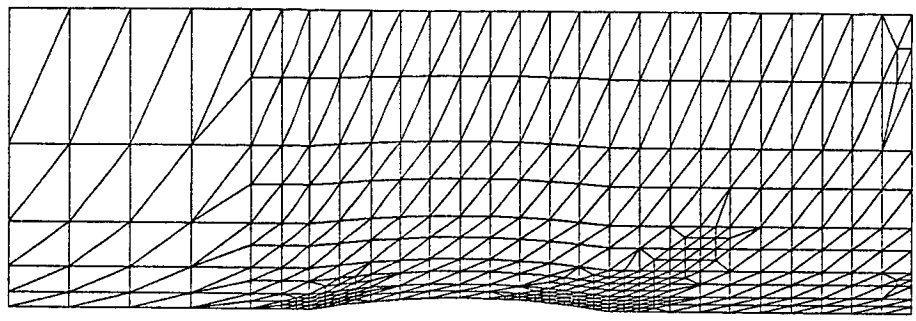


Fig. 9 : Continued.

MESH-REFINEMENT OF THE THIRD MESH
USE OF THE CRITERIA ; COEF=0.1
NS=5891 ; NT= 29857

MESH : CUT CORRESPONDING TO x=0.0



MESH : CUT CORRESPONDING TO z=zmin

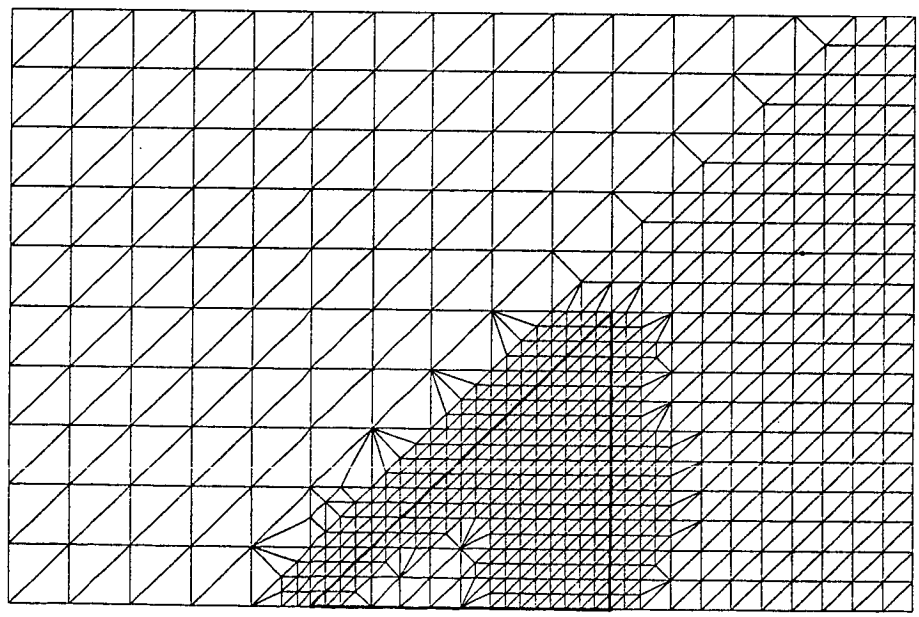
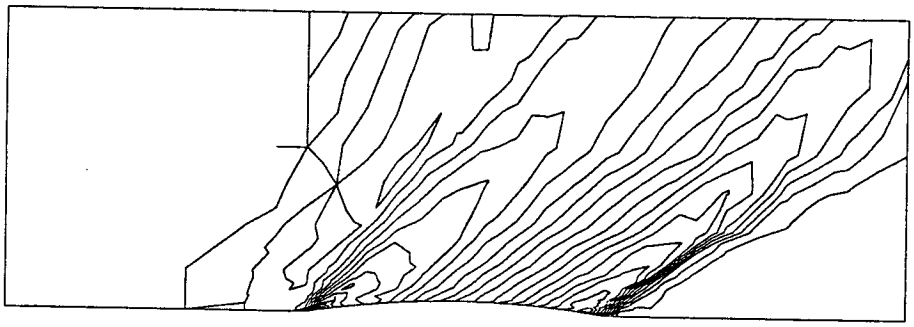


Fig. 10 : Supersonic flow past a wing. Refinement of the third mesh using the criteria (26).

Third mesh with local refinement. Artificial viscosity method.
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.8 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting

CP-LINES : MIN=-0.220 MAX=0.200 DLTA=0.020



CP-LINES : MIN=-0.280 MAX=0.240 DLTA=0.020

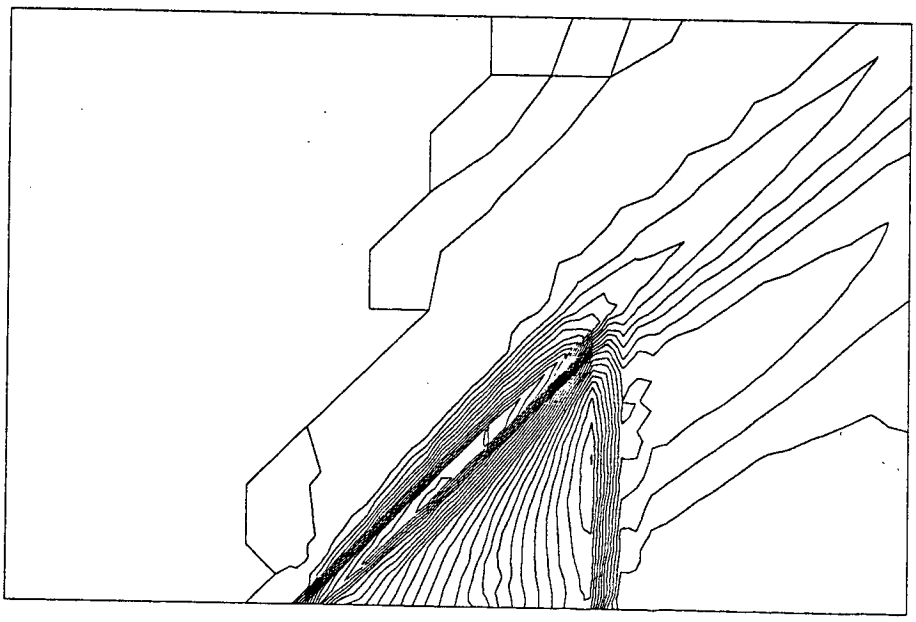
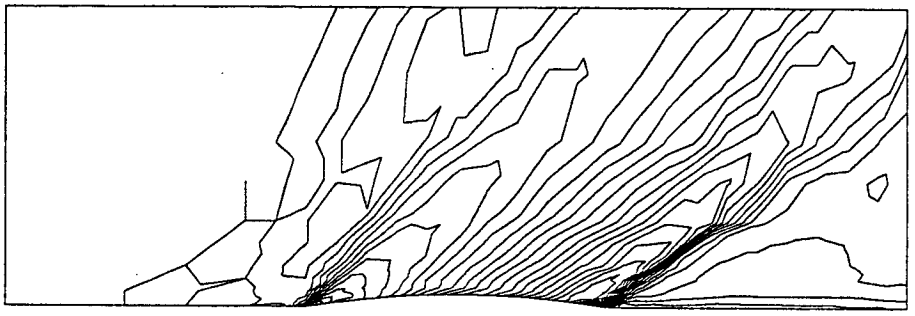


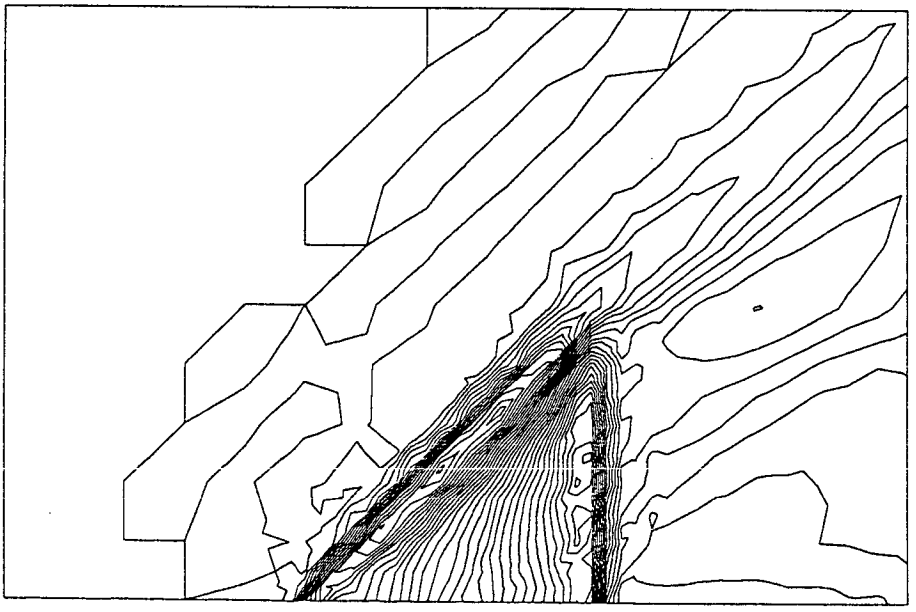
Fig. 11 : Supersonic flow past a wing. Artificial viscosity method. Solution obtained after mesh refinement.

Third mesh with local refinement. Artificial viscosity method.
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.8 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting

MACH-LINES : MIN=1.300 MAX=1.760 DLTA=0.020

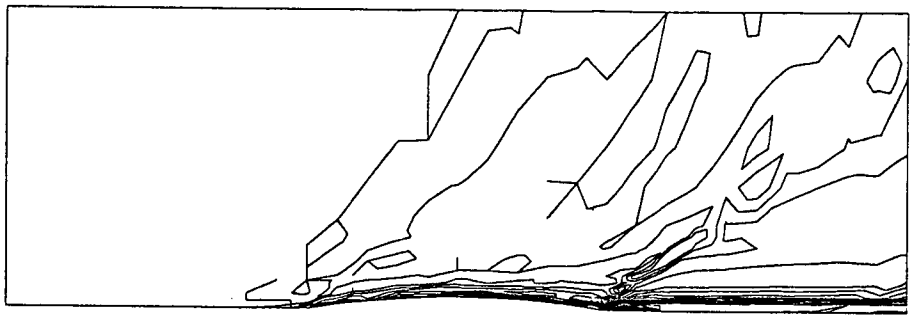


MACH-LINES : MIN=1.200 MAX=1.800 DLTA=0.020

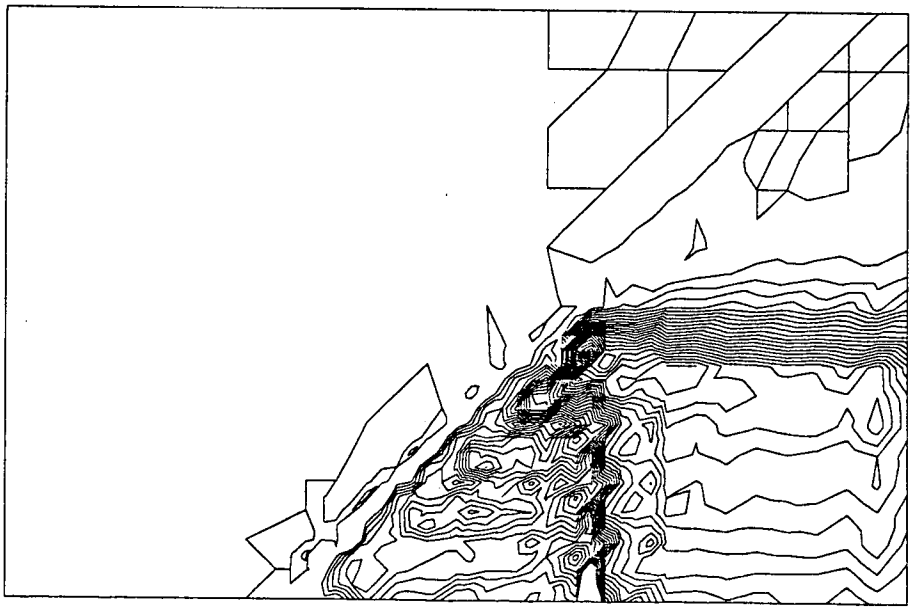


Third mesh with local refinement. Artificial viscosity method.
Sensor : second derivative of the pressure
Lumped mass matrix ; CFL=0.8 ; χ=4.0
Local time stepping ; b.c. : Vijaya. flux splitting

ENTROPY-LINES : MIN=0.000 MAX=0.015 DLTA=0.001

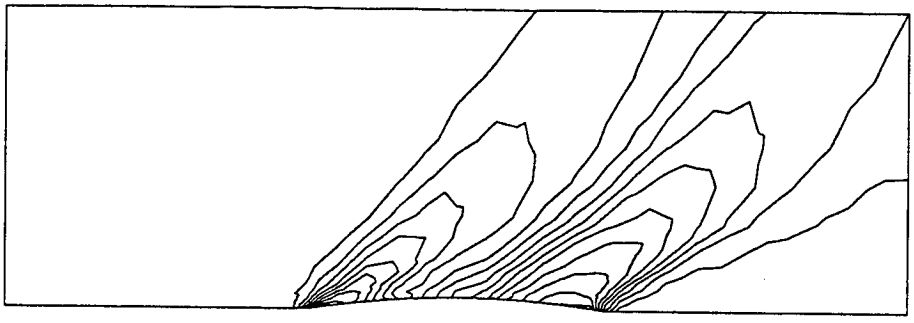


ENTROPY-LINES : MIN=0.000 MAX=0.023 DLTA=0.001



Third mesh with local refinement. TVD method.
Limiter : Roe's superbee
Lumped massmatrix ; CFL=0.8
Local time stepping ; b.c. : Vijaya. flux splitting

CP-LINES : MIN=-0.180 MAX=0.160 DLTA=0.020



CP-LINES : MIN=-0.220 MAX=0.200 DLTA=0.020

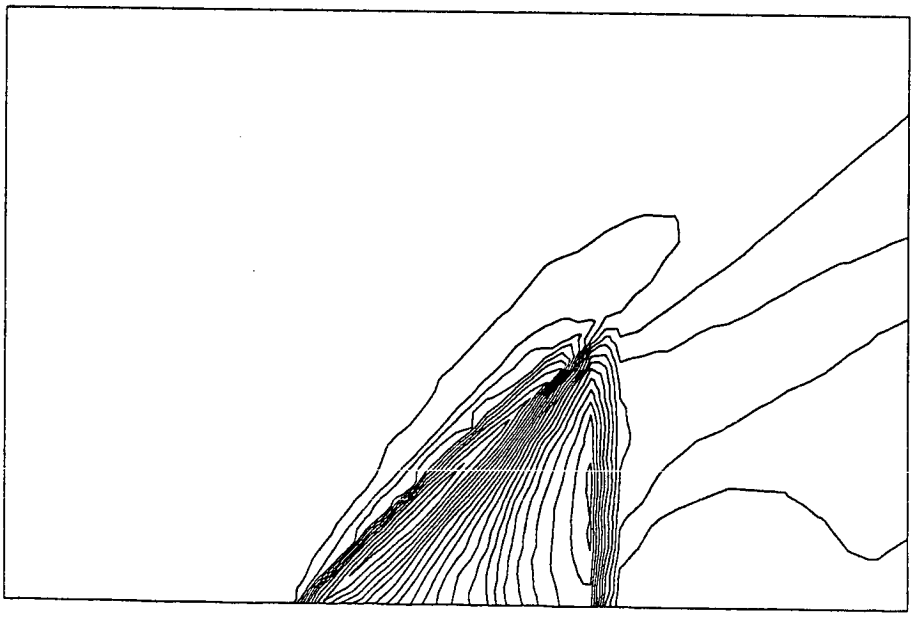
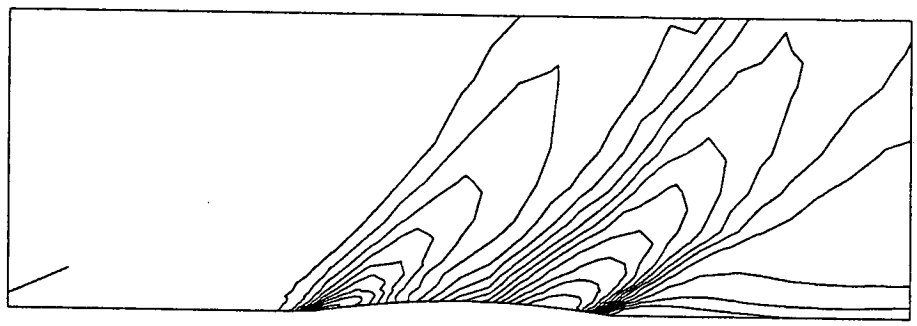


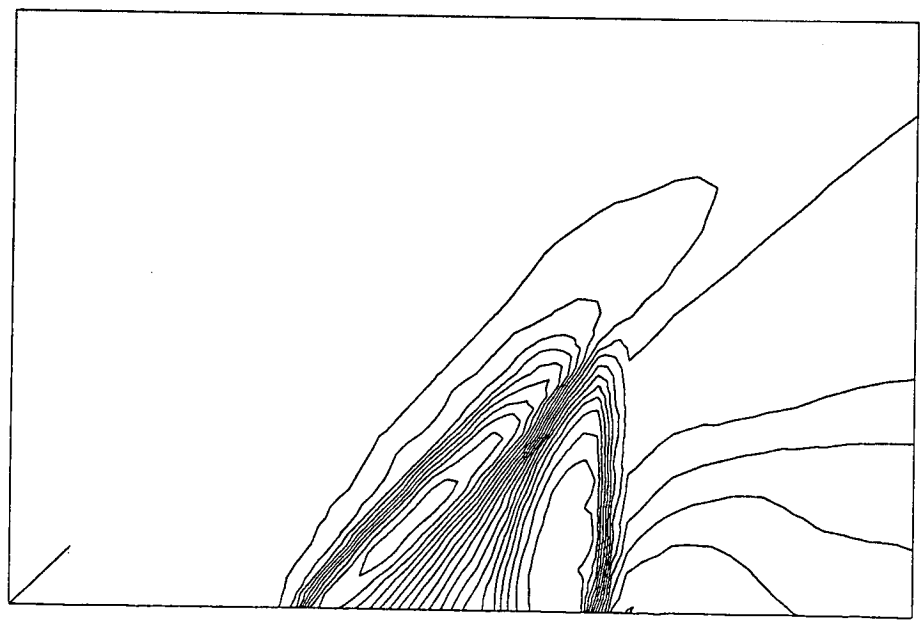
Fig. 12 : Supersonic flow past a wing. TVD method.

Third mesh with local refinement. TVD method.
Limiter : Roe's superbee
Lumped mass matrix ; CFL=0.8
Local time stepping ; b.c. : Vijaya. flux splitting

MACH-LINES : MIN=1.320 MAX=1.660 DLTA=0.020

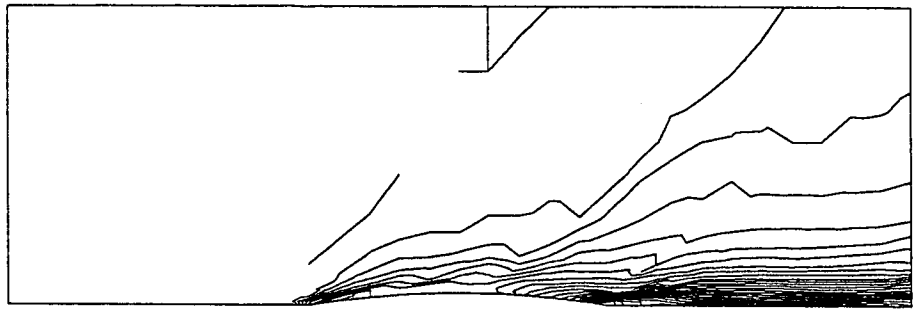


MACH-LINES : MIN=1.300 MAX=1.660 DLTA=0.020

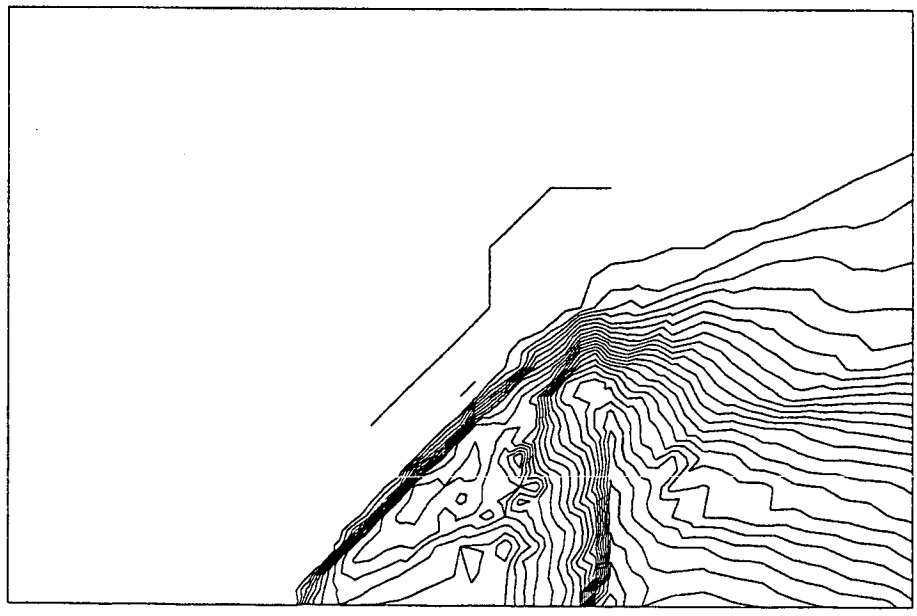


Third mesh with local refinement. TVD method
Limiter : Roe's superbee
Lumped mass matrix ; CFL=0.8
Local time stepping ; b.c. : Vijaya. flux splitting

ENTROPY-LINES : MIN=0.000 MAX=0.026 DLTA=0.001



ENTROPY-LINES : MIN=0.000 MAX=0.026 DLTA=0.001



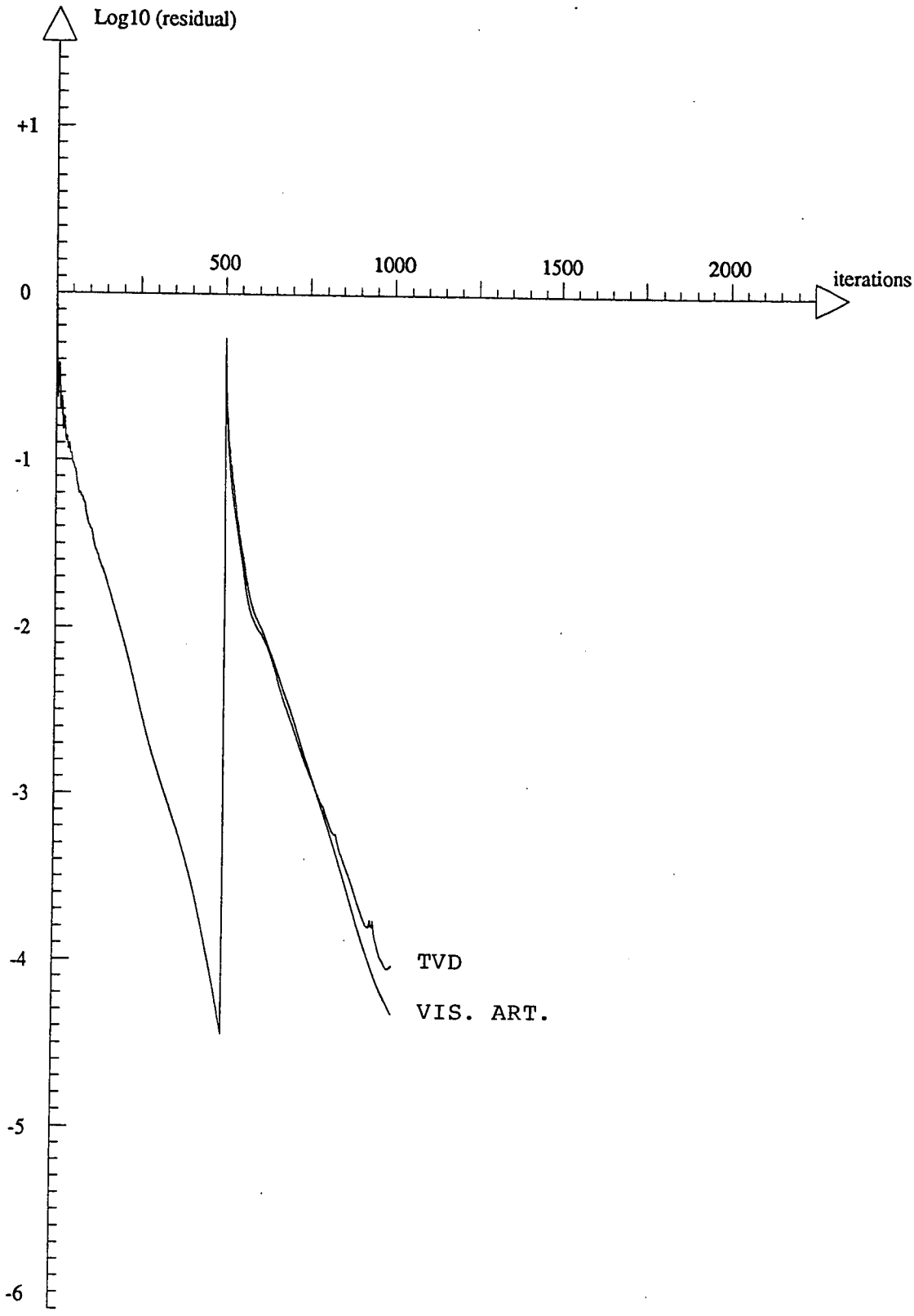


Fig. 13 : Supersonic flow past a wing. Residual curves.