An Edge-Based Solver for Compressible Flow

E. Ortega
R. Flores
E. Oñate

Publication CIMNE Nº-275, October 2005

An Edge-Based Solver for Compressible Flow

E. Ortega
R. Flores
E. Oñate

Publication CIMNE Nº-275, October 2005

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, España

AN EDGE-BASED SOLVER FOR COMPRESSIBLE FLOW

Enrique Ortega, Roberto Flores and Eugenio Oñate

International Centre for Numerical Methods in Engineering (CIMNE)
Universidad Politécnica de Cataluña
Building C1, Campus North, UPC
Gran Capitán, s/n, 08034 Barcelona, España
eortega@cimne.upc.edu, rflores@cimne.upc.edu, onate@cimne.upc.edu

Abstract. *An edge-based high-resolution scheme for the solution of the compressible Euler equations on unstructured finite elements grids is presented. The flow solver adopted in the present work is the approximate Riemann solver developed by Roe. A high-order spatial approximation is achieved by means of a piecewise linear reconstruction of the interface variables according to the MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) formulation. Finally, non-linear limiters are introduced in order to prevent the generation of oscillations. The proposed approach corresponds with the methodology presented by Löhner in [1].*

1. INTRODUCTION

When element-based data structures are used in finite elements calculations, certain redundancies of information occurs. For linear solvers based on triangular and tetrahedral elements, an alternatively and more efficient data structure using only edge information can be considered. Edge-based data structures for finite elements calculations have been introduced by Morgan *et al.* drawing from earlier finite volume schemes [2]. This mesh representation allows both to take advantage of unstructured meshes and reduce CPU time and memory required by the calculations [1,3,4] which is of vital importance when considering three-dimensional flow calculations. Additionally, when compressible or incompressible flow solvers are dealt with, edge-based representations make a straightforward implementation of upwind schemes in the finite element method context possible. Several comparisons between element and edge-based data structures performance can be found in the literature, see for instance [1]. An excellent review of the most common upwind schemes and its application on unstructured grids is set forth in [4].
Next the Euler equation system and its edge-based representation for two-dimensional flow is presented. Then the flow solver scheme is shown and subsonic, transonic and supersonic test cases are presented to demonstrate the algorithm performance.

2. EULER EQUATIONS

The Euler equations model a non-viscous, compressible and non-conductive fluid flow. This set of conservation laws is represented by a coupled non-linear system of first order partial differential equations, which can be written in different ways. The conservative form of the equations must be used when it is necessary to take into account discontinuities in the fluid

$$\widehat{U} = \sum_{j=1}^{nnode} U_j N_j \tag{6}$$

where *nnode* is the total number of nodes in the element, N_j is the standard finite elements shape function associated with node j and U_j is the value of $\widehat{U}$ at the same node. Due to the fact that the flux vectors F^K are generally a non-linear function of U, the formers are also approximated by the element shape functions as follows (group representation)

$$F^K(\widehat{U}) = \sum_{j=1}^{nnode} F_j^K N_j = \sum_{j=1}^{nnode} F^K(\widehat{U}_j) N_j \tag{7}$$

Assuming the Galerkin finite element formulation, $W_j = N_j$ and the approximate weak formulation can be written for a generic node i as

$$\sum_{e \in i} \int_{\Omega^e} N_i \frac{\partial \widehat{U}}{\partial t} d\Omega = \sum_{e \in i} \sum_{K} \int_{\Omega^e} \frac{\partial N_i}{\partial x_K} F^K(\widehat{U})\, d\Omega - \sum_{b \in i} \int_{\Gamma} N_i \bar{F}^n\, d\Gamma \tag{8}$$

where the summation extends over all the elements e and boundaries b which contains the generic node i.

Taking into consideration triangular elements with standard C^0 shape functions and the approximate forms for the conservative variables vector and the flux vectors given in eqs. (6) and (7) respectively, the integral terms in eq. (8) can be evaluated as

$$\sum_{e \in i} \int_{\Omega^e} N_i \frac{\partial \widehat{U}}{\partial t} d\Omega = \sum_{e \in i} \left[\int_{\Omega^e} N_i N_j\, d\Omega \right] \frac{dU_j}{dt} = \left[M \frac{dU}{dt} \right]_i \tag{9}$$

$$\sum_{e \in i} \sum_{K} \int_{\Omega^e} \frac{\partial N_i}{\partial x_K} F^K(\widehat{U})\, d\Omega = \sum_{e \in i} \sum_{K} \left[\frac{\Omega^e}{3} \frac{\partial N_i}{\partial x_K} \right] \left(F_i^K + F_j^K + F_k^K \right) \tag{10}$$

$$\sum_{b \in i} \int_{\Gamma} N_i \bar{F}^n\, d\Gamma = \sum_{b \in i} \left[\frac{\Gamma_b}{6} \left(2\bar{F}_i^n + \bar{F}_j^n \right) \right] \tag{11}$$

where M is the consistent mass matrix and Ω^e is the area of element e with nodes i, j and k. In the last expression Γ_b is the length of the boundary edge b defined by nodes i and j.

3.1 Edge-Based formulation

Using and edge-based data structure, nodal values of the diverse quantities are obtained adding edge contributions. The typical edge-data consists of the nodal coordinates and a list of all the edges in the mesh and its connectivities, i.e. the nodes to define each edge. A list of boundary edges where physical boundary conditions are imposed is also necessary. Several routines to obtain an edge-data structure for finite elements calculations can be found in [1]. Once the edge-data is obtained for a particular problem, the right hand side of eq. (8) can be evaluated for triangular linear elements in the following manner [4]

$$S_{ij}^{\kappa} = \frac{C_{ij}^{\kappa}}{\left|C_{ij}\right|} \quad \text{and} \quad \left|C_{ij}\right| = \sqrt{C_{ij}^{\kappa} C_{ij}^{\kappa}} \tag{17}$$

and taken into account the above expressions, eq. (12) becomes

$$\left[M\frac{dU}{dt}\right]_i = \sum_{ed=1}^{ne_i} \left|C_{ij}\right| \underbrace{\left(f_i + f_j\right)}_{\mathcal{F}_{ij}} + \left[\sum_{f=1}^{2} D_f \left(4\bar{F}_i^n + 2\bar{F}_{j_f}^n + F_i^n - F_{j_f}^n\right)\right]_i \tag{18}$$

Due to certain properties of the edge weights it is observed that the discretization scheme is conservative in the sense that the sum of the contributions made for any interior edge is zero. It is demonstrable too that the discretization scheme is a central difference type discretization for the spatial derivatives and some dissipation terms must be introduced in order to provide the necessary stabilization for the scheme. This fact brings about the replacement of the flux function $\mathcal{F}_{ij}$ defined in (18) by new consistent numerical fluxes. Adopting different forms for the latter, it is possible to obtain a wide variety of algorithms in which the numerical dissipation is introduced in a explicit manner or the flux function is modified according to the physics of the problem. An excellent recompilation and a comparative study of the most common algorithms is presented in [4]. Here, following Löhner's work [1], a first order numerical flux according to the Roe's approximate Riemman solver is employed in conjunction with a limiting stage with the aim of reducing the amount of dissipation and increasing the order of the scheme in regions where the flow is smooth.

4. ROE'S APPROXIMATE RIEMMAN SOLVER

The Roe's solver for the Euler equations based on flux difference splitting is one of the most popular and less dissipative approximate Riemman solvers and was developed by Roe in 1981. The idea behind this method is to solve the Riemman problem (Godunov) at the interface of two piecewise constant states U_L and U_R in a approximate manner reducing the computational cost and obtaining equally good results. The first order flux for this solver is defined for each edge by

$$\mathcal{F}_{ij} = f_i + f_j - \left|A(U_i, U_j)\right|\left(U_j - U_i\right) \tag{19}$$

where U_i and U_j is the vector of conservative variables at the edge nodes i and j and $|A(U_i, U_j)|$ is the absolute value of the Roe matrix calculated in the direction of the edge e_{ij}. The last is obtained projecting the conservative Jacobian matrices A^{κ} in the direction of the edge e_{ij} and replacing the variables in the resulting Jacobian matrix by the density-average Roe variables. Then the absolute value of the Roe matrix is achieved decomposing the last matrix by means of its eigenvalues and eigenvectors matrices. General expressions for the Jacobian, eigenvalues and eigenvectors matrices can be found in [5].
The density-average Roe variables are obtained by

$$A\left(U_i,U_j\right)=R_{ij}^{-1}\Lambda_{ij}R_{ij} \tag{24}$$

and

$$\left|A\left(U_i,U_j\right)\right|=R_{ij}^{-1}\left|\Lambda_{ij}\right|R_{ij} \tag{25}$$

where $\left|\Lambda_{ij}\right|=diag\left\{\left|\lambda_1\right|,\left|\lambda_1\right|,\left|\lambda_1\right|,\left|\lambda_2\right|,\left|\lambda_3\right|\right\}$ and

$$\begin{aligned}\lambda_1&=\hat{u}_{ij}\\ \lambda_2&=\hat{u}_{ij}+\tilde{c}_{ij}\\ \lambda_3&=\hat{u}_{ij}-\tilde{c}_{ij}\end{aligned} \tag{26}$$

It must be remembered here that all the variables correspond with the average-density Roe variables calculated for the edge e_{ij}. In eqs. (26) $\hat{u}_{ij}$ is the interface velocity projected on the edge, i.e. $\tilde{u}_{ij}^{\kappa}\cdot n^{k}$, and $n=(n^{(1)},n^{(2)},n^{(3)})$ is a unitary vector in the direction of the edge e_{ij}.

Introducing the factorization (25), the dissipative term of the Roe first order numerical flux can be written as

$$D=\left|A(U_i,U_j)\right|\left(U_j-U_i\right)=\left(R_{ij}^{-1}\left|\Lambda_{ij}\right|R_{ij}\right)\Delta U_{ji} \tag{27}$$

where $\Delta U_{ji}=(U_j-U_i)$ is a five-component vector which entries are the differences between the conservative variables vector components at edge nodes j and i. After some algebraic manipulations [7] the dissipative term in the first order Roe flux can be evaluated as follows

$$D=\left|\lambda_1\right|\Delta U_{ji}+\frac{\left|\lambda_2\right|-\left|\lambda_1\right|}{2}\Psi_1\begin{bmatrix}1\\ \tilde{u}_{ij}^{(1)}/\tilde{c}_{ij}+n^{(1)}\\ \tilde{u}_{ij}^{(2)}/\tilde{c}_{ij}+n^{(2)}\\ \tilde{u}_{ij}^{(3)}/\tilde{c}_{ij}+n^{(3)}\\ \tilde{H}_{ij}/\tilde{c}_{ij}+\hat{u}_{ij}\end{bmatrix}+\frac{\left|\lambda_3\right|-\left|\lambda_1\right|}{2}\Psi_2\begin{bmatrix}1\\ \tilde{u}_{ij}^{(1)}/\tilde{c}_{ij}-n^{(1)}\\ \tilde{u}_{ij}^{(2)}/\tilde{c}_{ij}-n^{(2)}\\ \tilde{u}_{ij}^{(3)}/\tilde{c}_{ij}-n^{(3)}\\ \tilde{H}_{ij}/\tilde{c}_{ij}-\hat{u}_{ij}\end{bmatrix} \tag{28}$$

where

$$\begin{aligned}\Psi_1=&\left(\frac{\gamma-1}{\tilde{c}_{ij}}q-\hat{u}_{ij}\right)\Delta U_{ji}^{(1)}+\left(-\frac{\gamma-1}{\tilde{c}_{ij}}\tilde{u}_{ij}^{(1)}+n^{(1)}\right)\Delta U_{ji}^{(2)}+\left(-\frac{\gamma-1}{\tilde{c}_{ij}}\tilde{u}_{ij}^{(2)}+n^{(2)}\right)\Delta U_{ji}^{(3)}\\ &+\left(-\frac{\gamma-1}{\tilde{c}_{ij}}\tilde{u}_{ij}^{(3)}+n^{(3)}\right)\Delta U_{ji}^{(4)}+\frac{\gamma-1}{\tilde{c}_{ij}}\Delta U_{ji}^{(5)}\end{aligned} \tag{29}$$

$$U_i^+ = U_i + \frac{1}{4}\Big[(1-k)\Delta_i^- + (1+k)\big(U_j - U_i\big)\Big]$$
$$U_j^- = U_j - \frac{1}{4}\Big[(1-k)\Delta_j^+ + (1+k)\big(U_j - U_i\big)\Big] \tag{32}$$

where Δ_i^-, Δ_j^+ are difference operators given by

$$\Delta_i^- = U_i - U_{i-1} = 2l_{ji}\cdot\nabla U_i - \big(U_j - U_i\big)$$
$$\Delta_j^+ = U_{j+1} - U_j = 2l_{ji}\cdot\nabla U_j - \big(U_j - U_i\big) \tag{33}$$

and l_{ji} is a vector from edge node i to node j, k is a parameter that allows to obtain different spatial order approximations and the gradients of the variables at edge nodes, ∇U_i and ∇U_j, are obtained via a recovery of the first derivatives at nodes procedure. It is possible to note that the approximations of the difference operators in (33) belong to an approximation to the gradient of the variables at nodes i and j using a central difference formulae. As was mentioned earlier, the parameter k in eqs. (32) allows to obtain different order or approximation, for example

$k = -1 \rightarrow$ second-order fully upwind scheme
$k = 0 \rightarrow$ From's scheme (see [9])
$k = 1/3 \rightarrow$ third-order upwind scheme
$k = 1 \rightarrow$ three-point central difference scheme

A graphical representation of the high-order approximation is presented in Figure 1.

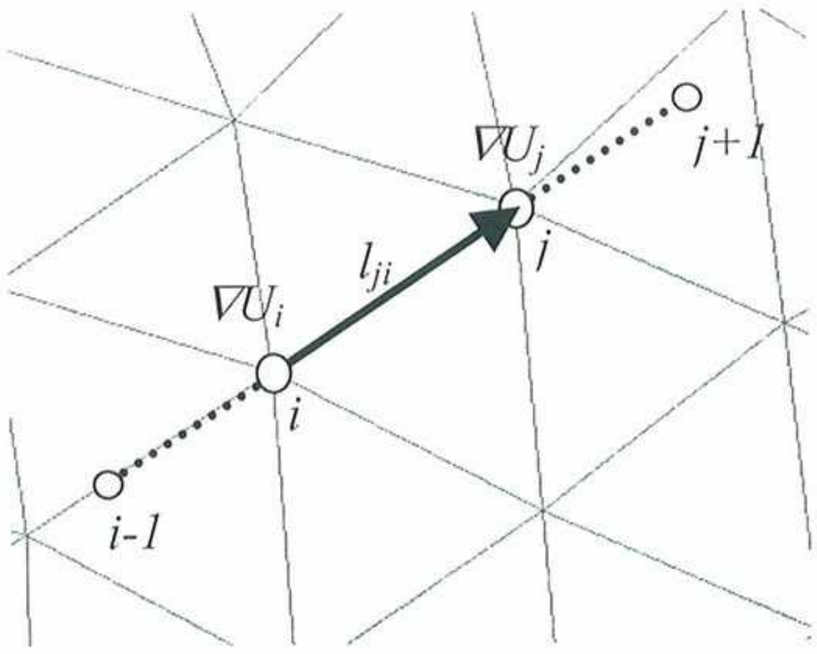

Figure 1: High-order approximation

The proposed extrapolation for the interface values (32) does not guarantee monotonicity properties, then in order to obtain an oscillation-free solution, non-linear limiters are introduced in the extrapolation stage through

$$
\begin{aligned}
&U_i^{(0)} = U_i^n \\
&\vdots \\
&U_i^{(s)} = U_i^n + \alpha_s \Delta t \left[M_L^{-1} \right]_i RHS_i^{s-1} \\
&\vdots \\
&U_i^{n+1} = U_i^{(s)}
\end{aligned}
\tag{37}
$$

where $\left[M_L^{-1} \right]_i$ is the lumped mass matrix at node $\underline{i}$ and α_s are coefficients that depend on the number of stages s employed. For three and four stages these parameters are set according to

3 stages $\rightarrow \alpha_1 = 3/5$, $\alpha_2 = 3/5$ and $\alpha_3 = 1.0$
4 stages $\rightarrow \alpha_1 = 1/4$, $\alpha_2 = 1/3$, $\alpha_3 = 1/2$ and $\alpha_4 = 1.0$

With the aim of reducing the computational cost in the residual evaluation, the dissipative term (28) is calculated at time $t=t^n$ and remains frozen for the next s stages of the scheme. Other possibilities proposed by Jameson, Schmidt and Turkel can be found in the literature.

5.1 Time step calculation

In the last paragraph the value of the time increment Δt must be bounded by stability criteria. In the present work, the time increment is determined at each node i as follows

$$\Delta t_i = \min\left\{ C \, \Delta t_{ij} \right\} \tag{38}$$

where C is the Courant number and Δt_{ij} is calculated, for each edge in the mesh that contains node i, according to

$$\Delta t_{ij} = \frac{l_{ji}}{\left| V_i^l \right| + c_i} \tag{39}$$

and

$$V_i^l = V_i \cdot l_{ji} \tag{40}$$

In the equations above, l_{ji} is a vector from node i to node j and V_i and c_i are the velocity vector and the speed of sound at node i respectively. If a global time step is chosen to advance in time, Δt in eqs. (37) corresponds to the minimum Δt_i calculated over the mesh.

6. NUMERICAL EXAMPLES

6.1 Subsonic flow around a NACA 0012 airfoil (M=0.3)

In this example the inviscid flow field around a NACA 0012, set at an incidence angle $\alpha=4^o$ is obtained. The unperturbed mach number is $M_\infty = 0.3$ and the flow is subsonic in the whole domain. A non-structured 6932 triangular-element mesh is employed in the numerical simulation. Figure 2 shows coefficient of pressure (Cp) contours for the flow field around the airfoil.

Figure 3 above presents the comparison between the pressure distribution over the airfoil calculated with the present methodology and the solution obtained by means of a potential flow calculation. A good agreement between both methods can be observed although some discrepancy between the numerical solutions appears in the leading edge suction peak.

6.2 Transonic flow around a NACA 0012 airfoil (M=0.8)

The numerical solution for the M=0.8 inviscid flow field around a NACA 0012 airfoil set at zero incidence angle is presented. A 7348 triangular-element non-structured mesh is employed in the numerical simulation and certain zones where discontinuities in the solution field are expected are refined in order to improve the accuracy. In Figure 4 Mach number contours around the airfoil are shown.

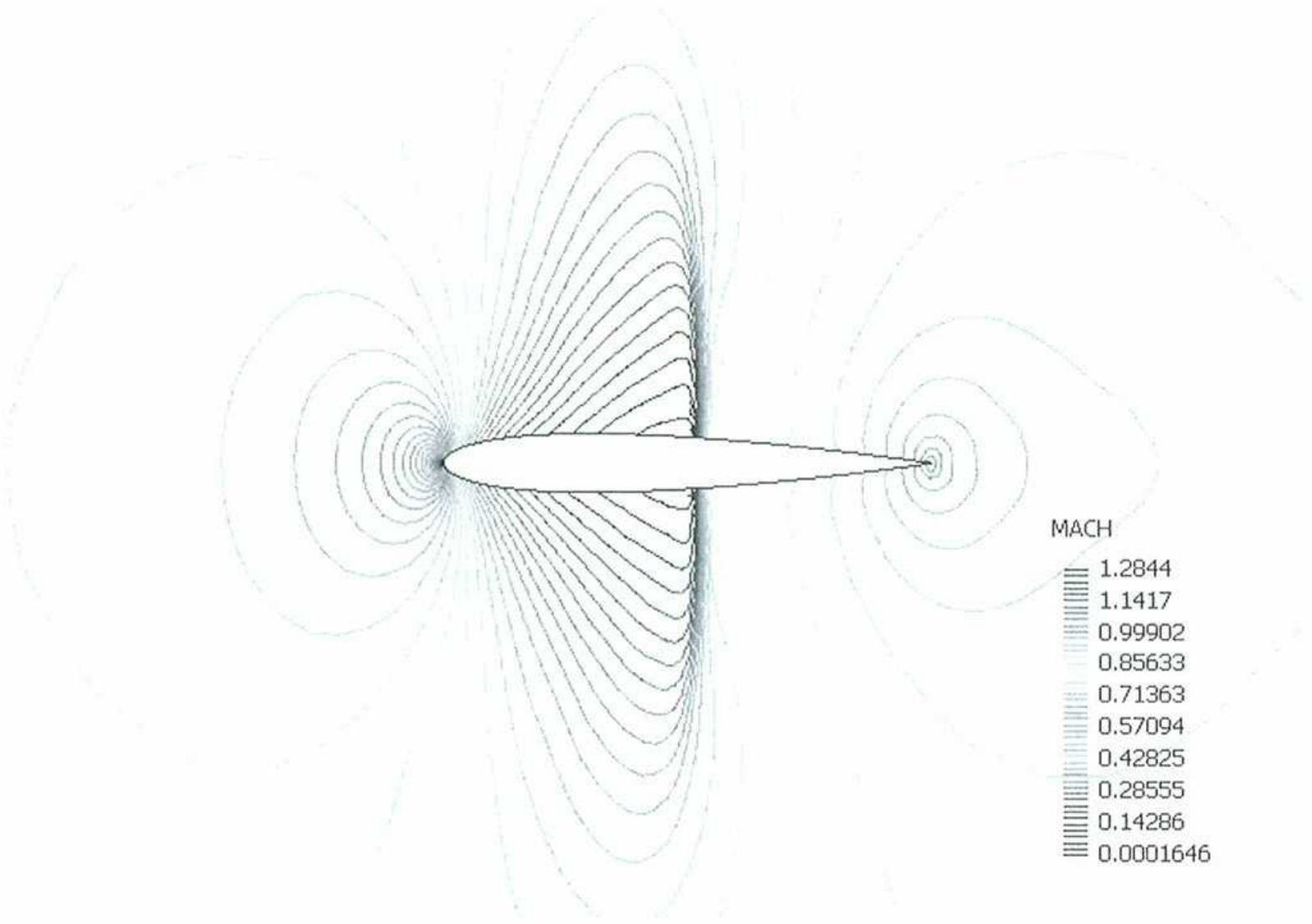

Figure 4: Mach number contours: NACA 0012, M = 0.8 and $\alpha = 0^{\circ}$

Next, in Figure 5, Cp distribution over the airfoil obtained by the present methodology is compared to the potential flow results due to Hafez *et al.* (1984) reproduced in [5]. In the figure below a good agreement between numerical and experimental results can be noticed.

The stagnation density value obtained in the numerical simulation is ρ_0=1.3288 and it differs less than 2% with respect to the analytical value given by ρ_0=1.3513.

6.3 Supersonic flow around a NACA 0012 airfoil (M=1.2)

In this example, a NACA 0012 airfoil set to zero incidence angle and immersed in a M_∞=1.2 supersonic flow is studied. The mesh is composed of 10076 triangular elements and, as in the previous example, zones where discontinuities are expected in the solution field are refined in order to improve the sharpness of the solution. Figure 7 shows coefficient of pressure contours for the flow field around the airfoil

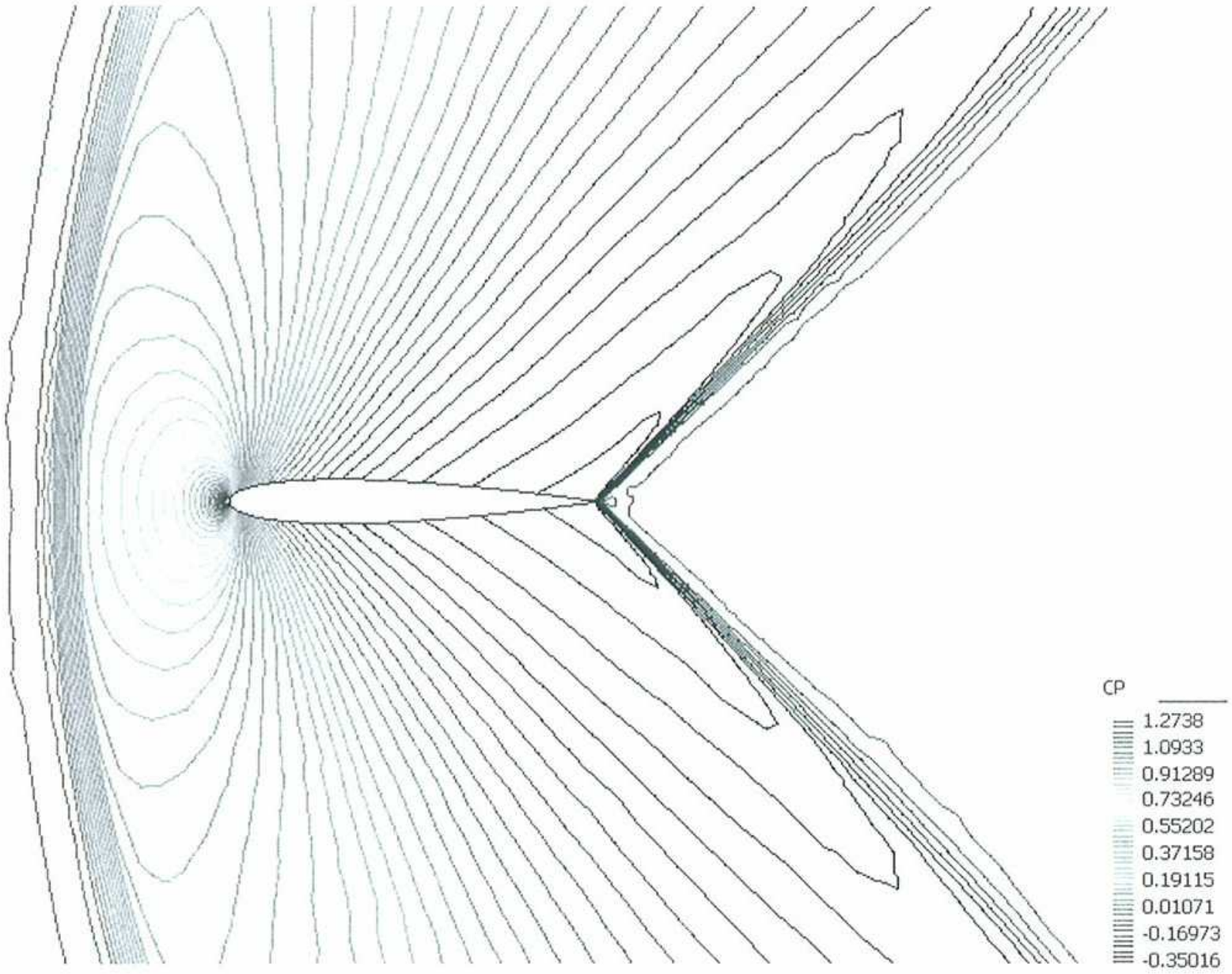

Figure 7: Cp contours around the NACA 0012 airfoil, M = 1.2 and $\alpha = 0^o$

A comparison between numerical and experimental results for Cp distribution over the airfoil is shown in Figure 8. There a good agreement can be observed.

9. CONCLUSIONS

The performance of the edge-based finite element methodology here exposed was tested on several academic simulations involving subsonic, transonic and supersonic flows. All of them were solved with satisfactory accuracy. However, numerical solutions for particular applications such as strong bow shocks present a negligible dependency of the α_1 and α_2 parameters that limit the eigenvalues of the Roe matrix. Also small oscillations could appear. The performance of other sets of limiters must be studied in order to fix these problems.
The extension of the present algorithm to solve three-dimensional applications is straightforward and in that case the advantages of the edge-based structure over the element-based structure are really noticeable.

ACKNOWLEDGEMENTS

The first author would like to acknowledge the support of Alβan Program, the European Union Programme of High Level Scholarships for Latin America, scholarship N° E04D027284AR.

REFERENCES

[1] Löhner R., '*Applied CFD Techniques*', John Wiley & Sons Ltd., 2001
[2] Morgan K., Peraire J., Peiró J., *'Unstructured Grid Methods for Compressible Flows'*, In Report 787 – Special Course on Unstructured Grid Methods for Advection Dominated Flows. AGARD, 1992.
[3] Morgan K., Peraire J., '*Unstructured Grid Finite-Element Methods for Fluid Mechanics*', Rep. Prog. Phys. 61: 569-638 (1998)
[4] Lyra P. R. M., Morgan K., '*A Review and Comparative Study of Upwind Biased Schemes for Compressible Flow Computation. Part III: Multidimensional Extension on Unstructured Grids*', Arch. Comput. Meth. Engrg Vol 9: 207:256 (2002)
[5] Hirsch C., '*Numerical Computation of Internal and External Flows*', Volume 2, John Wiley & Sons. (1990)
[6] Turkel E., '*Improving the Accuracy of Central Difference Schemes*', ICASE Report 88-53, September 1988
[7] Hu G., '*The Development and Applications of a Numerical Method for Compressible Vorticity Confinement in Vortex Dominant Flows'*, PhD Thesis, Virginia Polytechnic, 2001
[8] Swanson R.C., Turkel E., '*Multistage Schemes with Multigrid for Euler and Navier-Stokes Equations. Components and Analysis'*, NASA Technical Paper 3631, August 1997
[9] Hirsch C., '*Numerical Computation of Internal and External Flows*', Volume 1, John Wiley & Sons. (1990)