II CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA
II NATIONAL CONGRESS OF MECHANICAL ENGINEERING
12 a 16 de Agosto de 2002 - João Pessoa – PB

ADAPTIVE MESH REFINEMENT AND COARSENING FOR AERODYNAMIC FLOW SIMULATIONS

Leonardo Costa Scalabrin
Instituto Tecnológico de Aeronáutica
CTA/ITA/IEAA
São José dos Campos – SP – Brazil – 12228-900
scala@iae.cta.br

João Luiz F. Azevedo
Instituto de Aeronáutica e Espaço
CTA/IAE/ASE-N
São José dos Campos – SP – Brazil – 12228-904
azevedo@iae.cta.br

Abstract. *A new mesh refinement technique for unstructured grids is discussed. The new technique presents the great advantage of maintaining the original grid skewness, thanks to the capability of handling hanging nodes. The paper also presents an interpretation of MacCormack's method in an unstructured context. Results for a transonic convergent-divergent nozzle, for a convergent nozzle with a supersonic entrance and for a NACA 0012 airfoil are presented and discussed.*

Keywords: *Aerodynamics, Computational Fluid Dynamics, Numerical Methods, Finite Volume Technique, Adaptive Mesh Refinement*

1. INTRODUCTION

A major problem in computational fluid dynamics is the generation of an adequate mesh for the problem at hand. This task usually consumes a large amount of time and the quality of the generated mesh is often very dependent on the accumulated experience of who is generating it. One has to concentrate points in the regions where the aerodynamic flow presents significant variations. The regions in which these variations occur are dependent of the freestream conditions. For example, the position of shock waves in transonic and supersonic flows can varies substantially. Hence, one would have to generate one mesh for each freestream condition and, for each mesh, the user have to know where the relevant regions are. An approach that is particularly suited for unstructured grids is the use of adaptive mesh refinement/coarsening, since it allows the solution to dictate where points should be added to, or subtract from, the mesh.

A previous work in the group, Azevedo and Korzenowski (1998), has implemented a fairly efficient grid refinement procedure for unstructured triangular meshes. However, the procedure had a tendency of increasing the grid skewness and mesh coarsening was complicated, as discussed in Korzenowski et al (1997). In order to solve these problems, a new technique has been developed and the present paper will discuss it. Basically, the new technique divides the triangle into four new ones by the creation of nodes in the middle of the sides of the original triangle. This way of splitting triangles makes the new elements geometrically similar to the original triangle and, consequently, the mesh maintains its original quality. In addition, a hierarchical coarsening procedure was

implemented and the finite volume code was modified to handle the existence of hanging nodes in the mesh.

All simulations in the present case assume the flow to be inviscid and compressible, *i.e.*, the flow can be modeled by the compressible Euler equations. The code implements both a simple centered scheme with explicit time march, proposed by Jameson et al (1981), and an unstructured version of the method proposed by MacCormack (1969, 1985), which the authors have not seen in the literature yet. Both methods demanded the addition of artificial dissipation terms. This was true even for MacCormack's method which, despite being a Lax-Wendroff type scheme, would not remain numerically stable with only the numerical dissipation intrinsically provided by the 2nd-order Lax-Wendroff approach. In the present case, the formulation of the artificial dissipation terms follows the work of Mavriplis (1988, 1990) in an attempt to obtain steady state solutions which are independent of the time step, as discussed in Azevedo (1992)

The work initially describes the theoretical as well as the numerical formulation in the code. Particular attention is given to the numerical method and to the artificial dissipation terms. Results are presented for a transonic convergent-divergent nozzle, a convergent nozzle with a supersonic entrance and a NACA 0012 airfoil at transonic speeds. Finally, the paper draws some conclusions and it indicates the line of work that will be followed for the continuation of the present effort.

2. THEORETICAL FORMULATION

2.1. General Formulation

A first approach to solve an aerodynamic problem is to consider the flow to be compressible and inviscid. This kind of flow can be modeled by the Euler equations. The Euler equations in a dimensionless conservative form, for a two-dimensional flow, are

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=0, \tag{1}$$

where Q is the vector of dimensionless conserved variables, defined as

$$Q=\left[\rho \quad \rho u \quad \rho v \quad e\right]^{T}, \tag{2}$$

E and F are the dimensionless inviscid flux vectors, which can be written as

$$E=\begin{Bmatrix}\rho u\\ \rho u^{2}+p\\ \rho u v\\ (e+p)u\end{Bmatrix} \quad \text{and} \quad F=\begin{Bmatrix}\rho v\\ \rho u v\\ \rho v^{2}+p\\ (e+p)v\end{Bmatrix}. \tag{3}$$

In the previous expression, the dimensionless pressure p from the equation of state for a perfect gas as

$$p=(\gamma-1)\left[e-\frac{1}{2}\rho\left(u^{2}+v^{2}\right)\right], \tag{4}$$

where γ is the fluid ratio of specific heats. The adimensionalization process is detailed described in Pulliam and Steger (1980).

In this work, the finite volume technique was used in order to obtain the numerical solution of the previous set of equations. The formulation is obtained through an integration of the Euler equations in a finite volume. The use of the Gauss's theorem in each finite volume yields

$$\int_{V_i} \frac{\partial Q}{\partial t} dV + \int_{S_i} \left(E\,\vec{i} + F\,\vec{j} \right) \cdot d\vec{S} = 0 , \tag{5}$$

where $d\vec{S}$ is the outward oriented normal area vector of the surfaces that defines the i-th volume. Defining

$$Q_i = \frac{1}{V_i} \int_{V_i} Q\, dV_i , \tag{6}$$

as the mean value of the conserved properties in the volume, the final form of the finite volume formulation can be written for an elementary volume as

$$\frac{\partial Q_i}{\partial t} = -\frac{1}{V_i} \sum_{k=1}^{\text{number of faces}} \left(E_k\,\vec{i} + F_k\,\vec{j} \right) \cdot d\vec{S}_k , \tag{7}$$

which also shows that the integral was discretized assuming a constant value for the fluxes E and F on the faces. Many methods have been developed to solve Eq. (7). The paper will now discuss the two methods implemented.

2.2. Jameson's Method

The present paper used the Jameson et al (1981) method and a version of the method proposed by MacCormack (1969, 1985). For the first method, time integration of Eq. (7) can be written, using a 5-stage Runge-Kutta scheme, as

$$\begin{aligned} &Q_i^{(0)} = Q_i^n \\ &Q_i^{(l)} = Q_i^{(0)} - \frac{\alpha_l\, \Delta t_i}{V_i} \left[C\left(Q_i^{(l-1)} \right) - D\left(Q_i^{(l')} \right) \right] \qquad l = 1, \cdots, 5 \ , \\ &Q_i^{n+1} = Q_i^{(5)} . \end{aligned} \tag{8}$$

In the previous equations, $C(Q_i)$ is the convective operator calculated for the i-th control volume, which is the summation of the fluxes on the faces which constitutes the element, written as

$$C(Q_i) = \sum_{k=1}^{\text{number of faces}} \left(E(Q_k)\vec{i} + F(Q_k)\,\vec{j} \right) \cdot d\vec{S}_k , \qquad \text{where} \qquad Q_k = \frac{Q_m + Q_i}{2}, \tag{9}$$

Moreover, Q_m and Q_i are the conserved properties in each side of the k-th face and m indicates the neighbor of the i-th element. The α_l elements have the values $1/4$, $1/6$, $3/8$, $1/2$ and 1 for $l = 1, \cdots, 5$ respectively. The superscript l' indicates that the artificial dissipation terms are evaluated only in the two initial stages.

The artificial dissipation operator, $D(Q_i)$, is calculated according to the ideas developed by Mavriplis (1988, 1990). It is a blend of undivided harmonic and bi-harmonic operators. In regions where gradients are strong, the bi-harmonic operator is turned off to prevent oscillations whereas in

smooth regions the harmonic operator is turned off. A numerical pressure sensor does this switching between the operators. The operator can be written as

$$D(Q_i) = \sum_{m=1}^{\text{number of neighbors}} \left\{ \left(\frac{A_m + A_i}{2} \right) \left[\varepsilon_2 (Q_m - Q_i) + \varepsilon_4 \left(\nabla^2 Q_m - \nabla^2 Q_i \right) \right] \right\}, \tag{10}$$

where m represents a summation over the neighbors of the i-th element. The undivided harmonic operator ∇^2 and the A_i coefficients are written as

$$\nabla^2 Q_i = \sum_{m=1}^{\text{number of neighbors}} Q_m - Q_i \quad \text{and} \quad A_i = \sum_{k=1}^{\text{number of neighbors}} \left[\left| \left(u_k \, \vec{i} + v_k \, \vec{j} \right) \cdot \vec{S}_k \right| + a_k \left| \vec{S} \right| \right]. \tag{11}$$

As before, k indicates properties calculated on the faces, *i.e.*, using simple averages between the i-th element and its neighbor, the m-th element. The ε factors are based on the pressure sensor. The pressure sensor can be calculated as

$$\nu_s = \frac{\sum_{m=1}^{\text{number of neighbors}} \left| p_m - p_i \right|}{\sum_{m=1}^{\text{number of neighbors}} \left[p_m + p_i \right]}, \qquad \varepsilon_2 = \frac{1}{2} \max(\nu_{si}, \nu_{sm}) \quad \text{and} \quad \varepsilon_4 = \max\left(0, \frac{3}{256} - \varepsilon_2 \right). \tag{12}$$

2.3. MacCormack's Method

The original formulation for the MacCormack method, a two-stage, predictor-corrector method, can be found in MacCormack (1969, 1985). In Jameson's method, the conserved variables on the face, used do form the fluxes, were calculated as the arithmetic average between the values of the elements that contain the face. In the MacCormack method, the conserved variables on the faces are set equal to the conserved variables in one of the elements that contain the face, according to the stage of the method that is being calculated. In this work, an edge-based data structure was used in order to improve the computational efficiency. In this type of data structure, one stores the 2 nodes which constitute the edge and the two elements which contain the edge, i and m elements in the previous notation. In this work, in the first stage of the method, the conserved variables on the face are set equal to the conserved variables in the i-th element. On the second stage, the conserved variables on the face are set equal to the conserved variables in the m-th element.

The mathematical expression for MacCormack's method in an unstructured grid context can be written as

$$\begin{aligned}
&Q_i^{(0)} = Q_i^n, \\
&Q_i^{(1)} = Q_i^{(0)} - \frac{\Delta t_i}{V_i} \left[C\left(Q_i^{(0)}\right) - D\left(Q_i^{(0)}\right) \right], \\
&Q_i^{(2)} = \frac{1}{2} \left\{ Q_i^{(0)} + Q_i^{(1)} - \frac{\Delta t_i}{V_i} \left[C\left(Q_1^{(1)}\right) - D\left(Q_i^{(1)}\right) \right] \right\}, \\
&Q_i^{n+1} = Q_i^{(2)}.
\end{aligned} \tag{13}$$

The convective operators are calculated as in Eq. (9) with $Q_k = Q_i$ for the first stage and $Q_k = Q_m$ for the second stage. The artificial dissipation term is calculated using Eq. (10) dropping the bi-harmonic operator, *i.e.*, setting ε_4 to zero. The idea here was to add an artificial dissipation term

only in regions with strong property gradients, because the work of MacCormack (1985) states that the method does not need an artificial dissipation term. Nevertheless, the authors were never able to use this method without added artificial dissipation.

2.4. Initial and Boundary Conditions

In order to have a well-established mathematical problem, one has to set the boundary conditions. In this work, the necessary types of boundary conditions were wall, inlet, outlet and freestream conditions. In the finite volume context, the boundary conditions are set in virtual elements, called ghost elements, which are neighbors of the boundary elements. The idea here is set the conserved property values in the ghost volumes in order to have the correct boundary flux on the face. For example, in a wall boundary there is no flux through the wall. Hence, the expressions for the conserved variables in the ghost volume, using the properties of the internal element, are set in a way that, when using Eq. (9), the flux normal to the face is zero.

For the freestream surface, the conserved variables in the ghost volumes were set equal to those in the freestream. One should observe, however, that in this case it is necessary to place the far field boundary far away from the body in order to avoid reflection of perturbations at the boundary. For internal flows, there are the inlet and outlet boundary conditions. These types of boundary conditions can be better understood utilizing Riemman invariants, described in Long et al (1991), which determines how many variables one can fix in the boundary. For supersonic inlets, one can fix four variables. Therefore, for supersonic entrances, all the four variables in ghost volumes can be calculated using the stagnation properties and the Mach number. For supersonic exits, the four conserved variables are obtained by zero-th order extrapolation from inside the domain. For subsonic inlets 3 variables must be fixed. The variables normally chosen are the total pressure, P_0, the total temperature, T_0, and the angle of the flow entering the nozzle. The variable normally extrapolated from the domain is the u velocity component. For subsonic outlets, one variable must be fixed. In this work, the chosen variable was the static pressure. The extrapolated variables were the velocity components and the static temperature.

In this work, the initial conditions are very simple. For external flows, the initial conditions were taken as the freestream conditions. For internal flows, the initial conditions were taken as stagnation conditions.

3. ADAPTIVE MESH REFINEMENT

The quality of the numerical simulations is extremely dependent on the mesh. For a good quality solution, it is necessary to have points concentrated in the regions where the flow presents sudden variations. These regions may be determined by geometrical factors, such as corners or arcs, or by aerodynamic factors, such as a shock wave or a shear layer. Geometric factors are easy to handle, because they do not vary in time, at least for static configurations. The aerodynamic factors, however, vary according to the flow conditions. For example, a shock wave position over an airfoil varies according to the freestream Mach number. Hence, to obtain good numerical solutions, one should create a mesh for each aerodynamic condition, which requires a previous knowledge of the solution. Of course, sometimes aerodynamicists do not have this previous knowledge and the first numerical solution is not good.

The idea behind the adaptive mesh refinement is to attribute to the flow the responsibility of concentrating computational points by using automatic routines that alter the mesh. These routines identify the regions where more computational points are needed by using a numerical sensor. Then, the routine concentrates points in this region by many forms. One way is to move points from regions that do not need many points to the regions where they are needed. Another way is to create more points in the region. In this work, the option of adding points was chosen, because of its easiness of implementation in an unstructured grid context.

The sensor for regions that need refinement was developed in the work of Azevedo and Korzenowski (1998). It uses an undivided density gradient, normalized by the largest difference in density verified in the flow, *i.e.*,

$$\left(\text{sensor}\right)_i = \left[\frac{\left| \sum_{m=1}^{\text{number of neighbors}} \left(\rho_m - \rho_i\right) \right|}{\left|\rho_{\max} - \rho_{\min}\right|} \right]_i . \tag{15}$$

Note that, for cell centered finite volume calculations, discrete properties refer to control volumes. If this sensor is greater than a threshold value, the volume is refined.

After the sensor decided that the volume has to be refined, the volume is divided in 4 elements, as sketched in Fig. (1a). One can observe the presence of hanging nodes in the mesh. Another criterion for refinement is that the logic in the code does not allow elements which have been twice refined adjacent to elements that have not been refined. This is required in order to have a smooth decrease in element size throughout the mesh. This criterion is illustrated in Fig. (1b).

(a) (b)

Figure 1. Refinement of elements caused by sensor criterion and by size decrease criteria.

When the element is refined, the original element is destroyed. Its number is used to store the newly created center triangle. Three more elements are added to the end of the element array. The edge data structure changes too. The edges that constitute the original element are kept in memory, but they are marked with a flag that zeroes out the flux calculation for these edge. Besides the flag, the original edges store the number of the newly created edges, in order to make possible a hierarchical de-refinement. The edges that are created have two elements that contain them. One is a newly created element and the other is one of the neighbors of the original element. Of course, if an element besides a previously refined element is also refined, there is no creation of new edges. The code just changes the information concerning the neighbor elements of the previously created edges. In order to permit the hierarchical mesh de-refinement, the information concerning the new elements which are created from an existing element is also stored. After each refinement pass, a check for the size decrease criterion is performed and new refinement passes are done until such criterion is satisfied. These passes do not use the sensor criterion, just the size decrease criterion.

The treatment of hanging nodes is a very important aspect of the code, but it is a very simple implementation. The data structure of the code is edge-based. Therefore, it is irrelevant for the code if the node is a hanging node or not. The important aspect is that elements in this approach will no longer be treated just as triangles. In fact, each element may have the number of edges ranging from 3 to 6. Moreover, one is limited to 6 edges in the present case because of the size decrease criterion. Furthermore, this does not mean that the code wastes memory, because the data structure is edge-based and each edge stores the elements that contain it. All loops are edge-based and this makes the flux and dissipation factor calculations independent of the type of element.

The mesh de-refinement is basically the opposite of refinement. The code evaluates the refinement sensor for the elements that were refined in the last refinement pass. Despite the fact that the original element is no longer stored, it can be rebuilt using the array for creation of elements. If the sensor for the original element is smaller than a threshold value, it is marked for de-refinement.

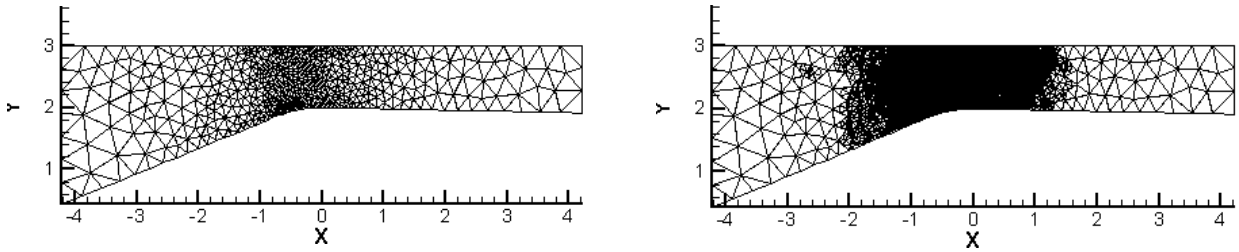
The code destroys the elements created by refinement and changes the connectivity array back in order to have no holes in it. This process requires changing all the arrays that have element information, such as the edge data structure. The code has to check if the refined edges can be destroyed because the neighbors of the original element can still be refined. After the de-refinement pass, a check for the size decrease criterion is performed and refinement passes are done until the size decrease criterion is satisfied. Again, these passes do not use the sensor criterion, just the size decrease criterion.

4. RESULTS AND DISCUSSION

The technique presented so far was used to simulate aerodynamic flow inside 2 different nozzles and transonic flow over a NACA 0012 airfoil. The first nozzle is a transonic convergent-divergent nozzle and the second one is a convergent nozzle with a supersonic entrance. The calculations for the nozzles do not have experimental results for comparison but they are very interesting to demonstrate the mesh refinement and de-refinement capability.

4.1. Transonic Convergent-Divergent Nozzle

The geometry and the initial mesh for this nozzle can be observed in Fig. (2a). The nozzle is symmetric, hence, only one half of the nozzle was included in the simulation. The pressure at the exit of the nozzle is set to $P_0/3$ in order to start the flow. Figure (2b) presents the final mesh after 2 refinement passes. The density sensor acts only where the flow variations are more important. Figure (3a) shows details of the interface between a refined and a non-refined region. The size decrease criterion enforcement is evident.



(a) (b)

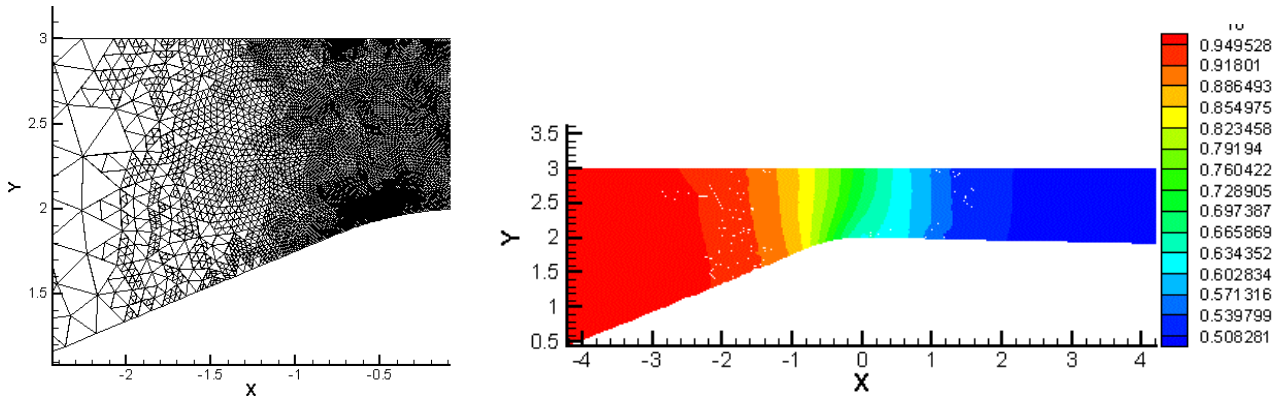
Figure 2. Transonic convergent-divergent nozzle geometry and initial mesh and final mesh after 2 mesh refinement passes.

Figure (3b) presents the dimensionless density contours for the converged solution using MacCormack's method. The results for Jameson's method are essentiality equal to those shown in Fig. (3b) and they will not be presented here. It is interesting to note that the classical theory for nozzle flow is a good approximation for the nozzle, except for the throat region, where the one-dimensional approximation does not predict that the flow accelerates faster near the wall.

4.2. Convergent Nozzle with a Supersonic Entrance

Figure (4a) presents the geometry of the nozzle. This nozzle has a supersonic flow with $M = 1.6$ on its entrance. Figure (4a) also shows the initial mesh for the problem and the evolution of the mesh. From the original mesh to the second, shown in Fig. (4b), there was one mesh refinement pass. From the second mesh to the third, shown in Fig. (4c), there were one pass of de-refinement followed by one pass of mesh refinement. Finally, from the third mesh to the fourth, shown in Fig. (4d), there was another pass of mesh refinement. One can see that the de-refinement routine does not reduce significantly the quantity of points in the mesh. The de-refinement routine is suited for

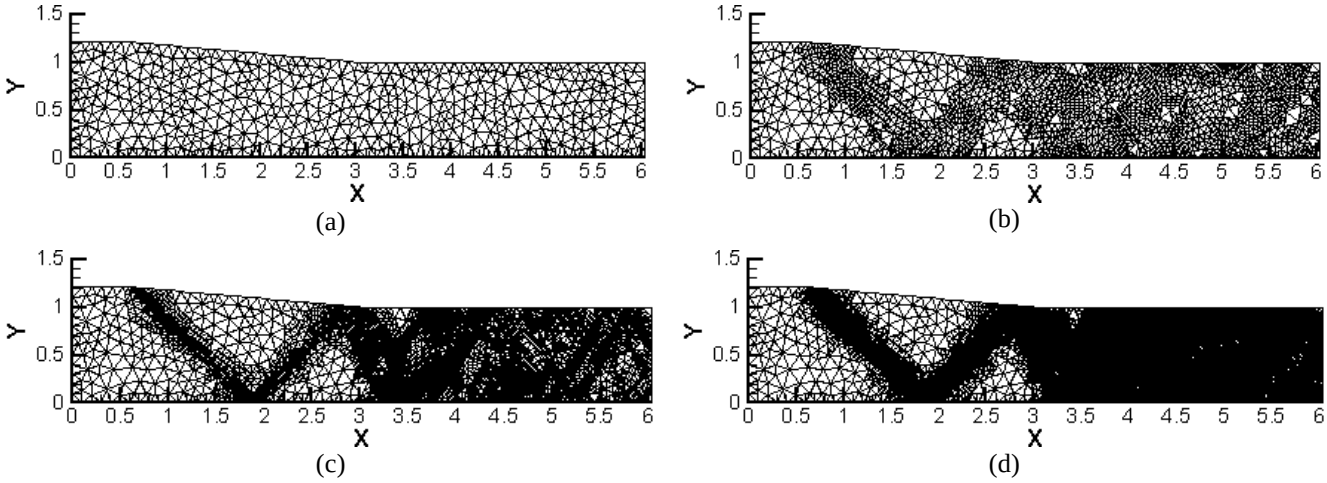
non-stationary problems in which the regions of the flow with strong gradients can vary substantially.



(a) (b)

Figure 3. Mesh detail of the final mesh after two mesh refinement passes and dimensionless density contours on the final mesh for the transonic convergent-divergent nozzle.

The converged solution for dimensionless pressure on the fourth mesh is presented in Fig. (5). The good shock capturing for the two first shock waves can be clearly seen in the figure. The reflection of the second shock wave is very near the corner of the convergent nozzle, where a supersonic expansion occurs. This fact reduces the intensity of the subsequent shocks reflections. In fact, these reflections could be solved only because of the massive mesh refinement in this region provided by the adaptive mesh routine.



(a) (b)

(c) (d)

Figure 4. Convergent nozzle geometry and mesh history.

4.3. Transonic Flow over a NACA 0012 Airfoil

In order to apply the mesh refinement routines in external flows, a NACA-0012 airfoil was simulated under transonic flow conditions. The initial mesh for this problem is presented in Fig. (6a). The final mesh after 2 refinement passes is shown in Fig. (6b). After the results for the convergent nozzle with supersonic entrance, the authors opted not to use the de-refinement routine because the gain in stationary problems is irrelevant. The entire region over the airfoil was refined and probably a larger threshold value for the sensor is needed.

Figure (7) presents the Mach contours for the problem in the final mesh. The shock wave is very well defined because of the extreme refinement created in its region. Despite the fact that the mesh is not symmetrical, the contour plot shows a symmetrical solution because of the refinement.

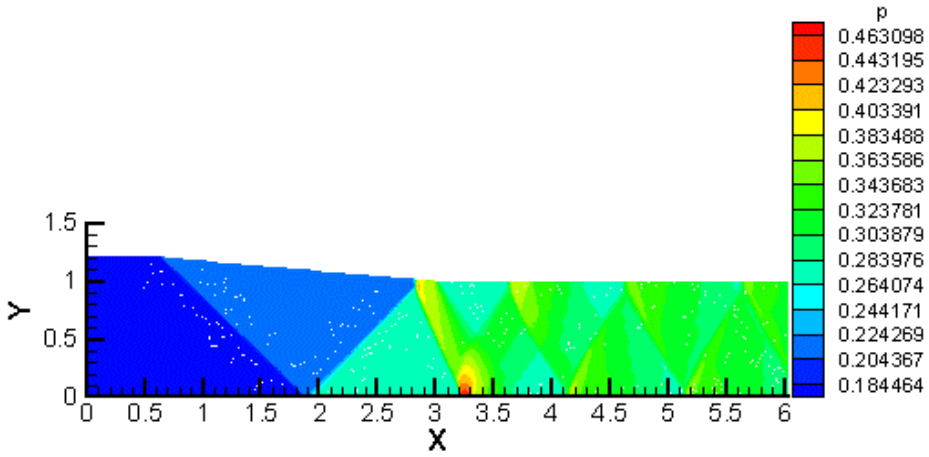


Figure 5. Dimensionless pressure contours for the convergent nozzle with a supersonic entrance in the final mesh.

5. CONCLUDING REMARKS

An interpretation for MacCormack's method in an unstructured mesh context was presented. It uses the elements which contains each edge in an alternate way to calculate the fluxes in the predictor and in the corrector stages. The results obtained are very close to those obtained with Jameson's method, which is largely used for unstructured calculations.

Furthermore, a new approach for mesh refinement was presented. The new technique does not increase the skewness of the original mesh and it allows hierarchical mesh de-refinement. The mesh de-refinement does not decrease significantly the number of points in a stationary problem, but it certainly can be extremely important in a non-stationary problem. Some results presented showed an excessive mesh refinement that can be solved with some adjustment in the sensor threshold value. The adaptive refinement technique increases the quality of the solution, as can be seen in all the results presented.

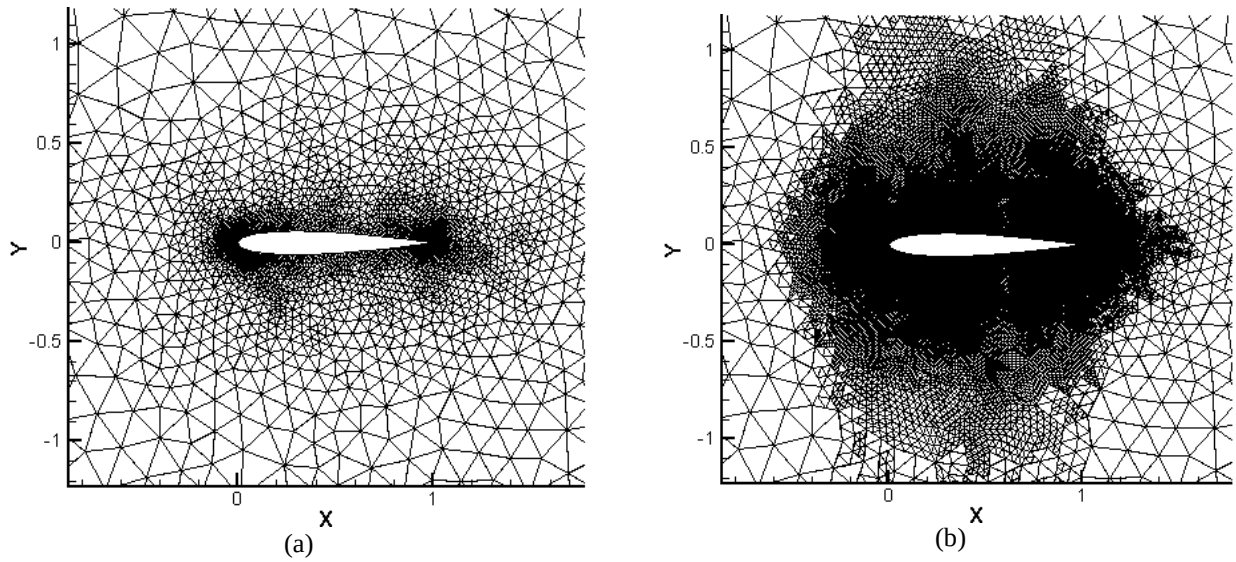


Figure 6. Initial mesh and final mesh after 2 mesh refinement passes for the NACA 0012 airfoil .

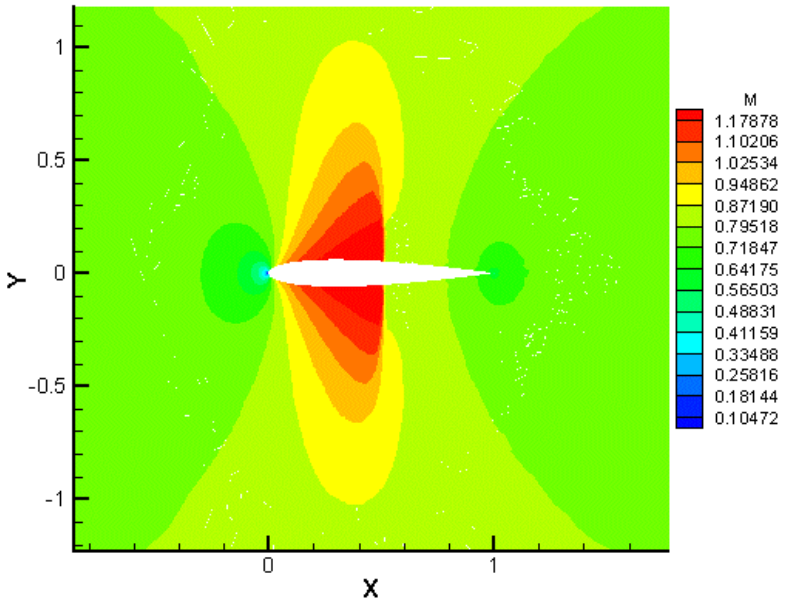


Figure 7. Mach number contours over the NACA 0012 airfoil for $M_\infty = 0.8$ and $\alpha = 0$ in the final mesh.

3. ACKNOWLEDGEMENTS

The authors would like to acknowledge Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, which provides a graduate scholarship for the first author and Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, which partially supported the project under the Integrated Project Research Grant No. 522413/96-0.

4. REFERENCES

Azevedo, J.L.F., 1992, "On the Development of Unstructured Grid Finite Volume Solvers for High Speed Flows", Report NT-075-ASE-N/92, Instituto de Aeronáutica e Espaço, S. J. Campos, SP.

Azevedo, J.L.F. and Korzenowski, H., 1998, "Comparison of Unstructured Grid Finite Volume Methods for Cold Gas Hypersonic Flow Simulations", AIAA Paper 98-2629, Albuquerque, NM.

Jameson, A., Schmidt, W. and Turkel, E., 1981, "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes", AIAA Paper 81-1259, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, CA.

Korzenowski, H., Figueira da Silva, L.F. and Azevedo, J.L.F, 1997, "Unstructured Adaptive Grid Flow Simulations of Inert and Reactive Gas Mixtures", Proceedings of the 14th Brazilian Congress of Mechanical Engineering, Bauru, SP, publication in CD-ROM format.

Long, L.N., Khan, M. and Sharp, H.T., 1991, "A Massively Parallel Euler/Navier-Stokes Method", AIAA Journal, Vol. 29, No. 4.

MacCormack, R.W., 1969, "The effect of Viscosity in Hypervelocity Impact Cratering", AIAA Paper 69-354, Cincinnati, OH.

MacCormack, R.W., 1985, "Current Status of Numerical Solutions of the Navier-Stokes Equations", AIAA Paper 85-0032, AIAA 23rd Aerospace Sciences Meeting, Reno, NV.

Mavriplis, D.J., 1988, "Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes", AIAA Journal, Vol. 26, No. 7, pp. 824-831.

Mavriplis, D.J., 1990, "Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes", AIAA Journal, Vol. 28, No. 2, pp. 213-221

Pulliam, T.H. and Steger, J.L., 1980, "Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow", AIAA Journal, Vol. 18, No. 2, pp. 159-167.