INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **45**:1107–1122 (DOI: 10.1002/fld.731)

Adaptive mesh refinement and coarsening for aerodynamic flow simulations

Leonardo Costa Scalabrin[1,‡] and João Luiz F. Azevedo[2,*,†]

[1]*Instituto Tecnológico de Aeronáutica, CTA/ITA/IEAA, São José dos Campos, SP 12228-900, Brazil*
[2]*Instituto de Aeronáutica e Espaço, CTA/IAE/ASE-N, São José dos Campos, SP 12228-904, Brazil*

SUMMARY

A new mesh refinement technique for unstructured grids is discussed. The new technique presents the important 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 grid context. Results for a transonic convergent–divergent nozzle, for a convergent nozzle with a supersonic entrance and for transonic flow over a NACA 0012 airfoil are presented and discussed. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: 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 on the freestream conditions. For example, the position of shock waves in transonic and supersonic flows can vary substantially. Hence, one would have to generate one mesh for each freestream condition and, for each mesh, the user has to know where the relevant regions are. An approach that is particularly suited for unstructured grids is the use of adaptive mesh refinement/coarsening,

*Correspondence to: João Luiz F. Azevedo, Instituto de Aeronáutica e Espaço, CTA/IAE/ASE-N, São José dos Campos, SP 12228-904, Brazil.
‡E-mail: scala@iae.cta.br
†E-mail: azevedo@iae.cta.br

Contract/grant sponsor: Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)
Contract/grant sponsor: Conselho Nacional de Desenvolvimento Cientifico e Tecnológico (CNPq); contract/grant numbers: 522413/96-0, 501200/2003-7

Published online 14 May 2004

Received 14 August 2002
Revised 7 March 2004

since it allows the solution to dictate where points should be added to, or subtracted from, the mesh [1].

A previous work in the group [2] 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 [3]. In order to solve these problems, a new technique has been developed and the present paper discusses 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 and allows the creation of hanging nodes in these positions. The use of hanging nodes 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 modelled by the compressible Euler equations. The code implements both a simple centred scheme with explicit time march [4] and an unstructured version of the method proposed by MacCormack [5, 6], 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 second-order Lax–Wendroff approach. In the present case, the formulation of the artificial dissipation terms follows the work of Mavriplis [7, 8] in an attempt to obtain steady state solutions which are independent of the time step, as discussed in Azevedo [9].

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.

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 type of flow can be modelled by the Euler equations. The Euler equations in 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

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

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

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

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

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

where γ is the fluid ratio of specific heats. The non-dimensionalization process is detailed described in Reference [10].

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

$$\int_{V_i} \frac{\partial Q}{\partial t}\,\mathrm{d}V + \int_{S_i} (E\mathbf{i} + F\mathbf{j})\,\mathrm{d}\mathbf{S} = 0 \tag{5}$$

where $\mathrm{d}\mathbf{S}$ is the outward oriented normal area vector for the surfaces that define the ith volume. If one defines

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

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

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

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

2.2. Jameson's method

The present paper used the Jameson method [4] and an unstructured version of the method proposed by MacCormack [5, 6]. For the first method, time integration of Equation (7) can be written, using a 5-stage Runge–Kutta scheme, as

$$\begin{gathered} Q_i^{(0)} = Q_i^n \\ Q_i^{(l)} = Q_i^{(0)} - \frac{\alpha_l \Delta t_i}{V_i}[C(Q_i^{(l-1)}) - D(Q_i^{(l')})], \quad l = 1, \ldots, 5 \\ Q_i^{n+1} = Q_i^{(5)} \end{gathered} \tag{8}$$

In the previous equations, $C(Q_i)$ is the convective operator calculated for the ith control volume. The convective operator comprises the summation of the convective fluxes on the faces that constitute the element, and it can be written as

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

Moreover, Q_m and Q_i are the conserved properties in each side of the kth face and m indicates the neighbour of the ith element. The α_l coefficients have the values 1/4, 1/6, 3/8, 1/2 and 1, for $l = 1, \ldots, 5$, respectively. The l' superscript 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 [7, 8]. 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 of the flow, the harmonic operator is turned off. A numerical pressure sensor does the switching between the operators. The operator can be written as

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

where m represents a summation over the neighbours of the ith element. The undivided harmonic operator, $\nabla^2()$, is written as

$$\nabla^2 Q_i = \sum_{m=1}^{\text{number of neighbours}} Q_m - Q_i \tag{11}$$

The A_i coefficients can be expressed as

$$A_i = \sum_{k=1}^{\text{number of neighbours}} [|(u_k\mathbf{i} + v_k\mathbf{j}) \cdot \mathbf{S}_k| + a_k|\mathbf{S}|] \tag{12}$$

As before, k indicates properties calculated on the faces, i.e. using simple averages between the ith element and its neighbour, the mth element. The ε factors are based on the pressure sensor. The pressure sensor can be calculated as

$$v_i = \frac{\sum_{m=1}^{\text{number of neighbours}} |p_m - p_i|}{\sum_{m=1}^{\text{number of neighbours}} [p_m + p_i]} \tag{13}$$

where

$$\varepsilon_2 = \tfrac{1}{2} \max(v_i, v_m) \quad \text{and} \quad \varepsilon_4 = \max\left(0, \frac{3}{256} - \varepsilon_2\right) \tag{14}$$

2.3. MacCormack's method

The original formulation for the MacCormack method, a two-stage, predictor-corrector method, can be found in References [5, 6]. In Jameson's method, the conserved variables on the face, used to form the fluxes, are calculated as the arithmetic average between the values of the elements that contain the face. In MacCormack's 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 define the edge and the two elements which share the edge. The edge is oriented from the first to the second node. An observer looking in the direction appointed by the edge orientation sees two elements sharing the edge. The element on the left side is called the ith element and the one on the right side is the mth element. In the present work, in the first stage of the method, the conserved variables on the edge are set equal to the conserved variables in the ith element. In the second stage, the conserved variables on the edge are set equal to the conserved variables in the mth element.

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

$$\begin{aligned} Q_i^{(0)} &= Q_i^n \\ Q_i^{(1)} &= Q_i^{(0)} - \frac{\Delta t_i}{V_i}[C(Q_i^{(0)}) - D(Q_i^{(0)})] \\ Q_i^{(2)} &= \frac{1}{2}\left\{Q_i^{(0)} + Q_i^{(1)} - \frac{\Delta t_i}{V_i}[C(Q_i^{(1)}) - D(Q_i^{(1)})]\right\} \\ Q_i^{n+1} &= Q_i^{(2)} \end{aligned} \tag{15}$$

The convective operators are calculated as in Equation (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 Equation (10), but dropping the bi-harmonic operator, i.e. setting ε_4 to zero. The idea here is to add an artificial dissipation term only in regions with strong property gradients because the artificial dissipation intrinsically provided by the numerical method is not enough to overcome the instabilities generated in these regions. Reference [6] states that the numerical method does not need an artificial dissipation term, but 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 appropriate 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 volumes, which are neighbours of the control volumes adjacent to the boundary. The idea here is to define 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 convective 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 Equation (9), the convective flux normal to the face is zero.

For the freestream surface, in the present case, the conserved variables in the ghost volumes are set equal to those in the freestream. However, one should observe 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. The use of Riemann invariants [11] could avoid this waste of points, but the authors opted not to use this approach because the applications here are simply 2-D and, therefore, fairly inexpensive. For internal flows, there are the inlet and outlet

boundary conditions. These types of boundary conditions can be better understood using one-dimensional characteristic relations. The one-dimensional characteristic relations describe how the information propagates in the flow. Hence, using these relations, one can determine how many conditions should be fixed at a given boundary and how many conditions are dependent on the flow inside the computational domain. The former case is usually associated with Dirichlet type boundary conditions, whereas the latter conditions typically lead to the need to extrapolate information from the interior to the boundary of the computational domain. A detailed description of the one-dimensional characteristic relations is presented in Reference [12].

For supersonic inlets, one can fix four variables in the ghost volume. Therefore, for supersonic entrances, the ρ, u, v and e values in the ghost cells are set equal to those values of the flow which enters the nozzle. For supersonic exits, the ρ, u, v and e values in the ghost volumes are obtained by zero-th order extrapolation from inside the domain. For subsonic inlets, only three variables can be fixed in the ghost volume. The variables normally chosen are the total pressure, P_t, the total temperature, T_t, and the angle of the flow entering the nozzle [6]. The variable normally extrapolated from inside the domain is the u velocity component, for nozzles which have their axes essentially aligned with the x-axis, as in the present case. Then, one can obtain ρ, v and e values in the ghost volume using thermodynamic relations. For subsonic outlets, only one variable must be fixed in the ghost volume. In this work, the chosen variable was the static pressure, p. The extrapolated variables are the u and v velocity components and the static temperature, T. Again, the ρ and e values in the ghost volume are calculated using thermodynamic relations.

In this work, the initial conditions are very simple. For external flows and for internal flows with supersonic inlet, the initial conditions are taken as the freestream conditions. For internal flows with subsonic inlet, the initial conditions are taken as the stagnation properties.

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. Two types of costs are involved in the generation of new meshes: the cost of mesh generation itself and the computational cost to calculate the wasted solutions.

The idea behind adaptive mesh refinement is to attribute to the flow the responsibility of concentrating computational points where needed by using automatic routines that alter the mesh [1]. These routines identify the regions where more computational points are needed by the use of numerical sensors based on gradients of indicator variables [13] or based on interpolation theory [1]. The refinement of the chosen regions can be done by three basic

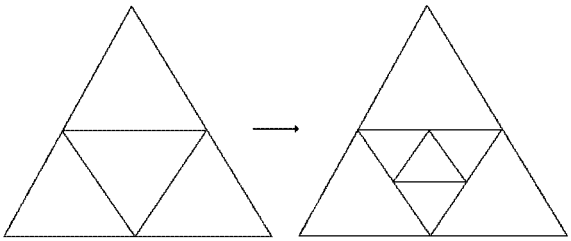

Figure 1. Refinement of elements caused by sensor criterion.

forms: mesh movement or repositioning, mesh enrichment and adaptive remeshing. In the mesh repositioning methods (R-Methods), points are moved from regions where they are not needed to regions where the refinement criteria indicates the need for more points. This approach is normally used with structured grids and its application on a unstructured mesh is limited [14]. A similar method to mesh repositioning is the adaptive remeshing technique (M-Methods), which can be understood as the repositioning of all the nodes of the mesh or of a particular region. The main disadvantage of this method is that it requires a very automatic grid generator. The mesh enrichment approach (H/P Methods) is characterized by the addition of points to the mesh (H-Methods) or by the use of higher-order shape-functions [15] (P-Methods) in a finite element context. In this work, the option of adding points was chosen, because of its ease of implementation in an unstructured grid, finite volume context.

The sensor for regions that need refinement was developed in Reference [2]. It uses an undivided density gradient, normalized by the largest difference in density verified in the flow, i.e.

$$(\text{sensor})_i = \left[\frac{|\nabla \rho|}{\rho_{\max} - \rho_{\min}} \right]_i \tag{16}$$

Note that, for cell centred 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 Figure 1. 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 Figure 2.

When the element is refined, the original element is destroyed. Its number is used to store the newly created centre 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 edges. Besides the flag, the array which contains the original edges also stores 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 neighbours of the original element. Of course, if a control volume adjacent to a previously refined element is also refined, there is no creation of new edges. The code only changes the information concerning the neighbour elements of the previously created

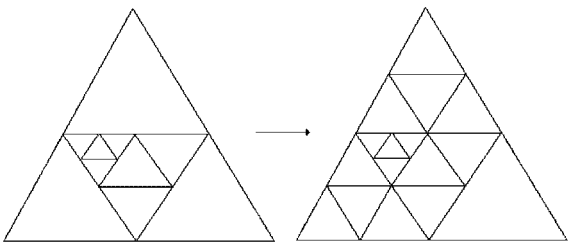

Figure 2. Refinement of elements caused by size decrease criterion.

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 requires 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 control volumes in this approach will no longer be treated simply as triangles. In fact, each element may have the number of edges ranging from 3 to 6. Moreover, one is limited to six edges in the present case due to the size decrease criterion and because only triangular grids are considered here. 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 term 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 original elements, i.e. the elements before 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. In a finite volume context, the property gradient components can be computed through a transformation of the derivative calculation into a line integral around the edges which form the volume. However, a line integral around the original volume is equal to the sum of the same line integral around the volumes generated by refinement of the original element. Hence, the property gradient components in the original element are equal to the sum of the respective gradient components in the volumes created by refinement. Therefore, it is fairly inexpensive to reconstruct the sensor in the original volume.

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 neighbours 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 was used to simulate aerodynamic flows inside two different nozzles and transonic flow over a NACA 0012 airfoil. The first test case considers a transonic convergent–divergent nozzle and the second one is a convergent nozzle with a supersonic entrance. The calculations for the supersonic entrance nozzle do not have experimental results for comparison but this test case is very interesting to demonstrate the mesh refinement and de-refinement capability.

4.1. Transonic convergent–divergent nozzle

The initial mesh for this nozzle can be observed in Figure 3. The nozzle is symmetric. Hence, only one half of the nozzle is included in the simulation. The static pressure at the exit of the nozzle is set to $P_t/3$ in order to start the flow, where P_t is the stagnation pressure at the nozzle entrance. Plates 1 and 2 present the dimensionless pressure contours in the nozzle calculated using MacCormack's and Jameson's methods, respectively. Both solutions are qualitatively correct and there are no significant differences between them. It is interesting to note that the classical quasi one-dimensional 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. MacCormack's and Jameson's methods detect a small shock wave after the throat. This shock wave can also be observed in Figure 4, which presents the pressure ratio on the nozzle wall for both methods and experimental data from Reference [16]. The shock wave is formed by an overturning of the flow through the curved throat section [17]. As the overturned flow impinges on the straight wall divergent section, it is deflected. Since the flow is already supersonic in this region, the deflection is accomplished by an oblique shock wave. In Figure 4, one can also observe that MacCormack's method provides a slightly better approximation of the experimental data.

Figure 5 presents the final mesh after two refinement passes. The first mesh refinement occurs after 5000 iterations and the second after more 5000 iterations. The density sensor threshold value is set to 0.1 in this simulation. The density sensor acts only where the flow variations are more important. Figure 6 shows details of the interface between a refined and a non-refined region. The effects of the size decrease criterion enforcement are evident in this figure. Plate 3 presents the dimensionless density contours for the converged solution using Jameson's method. The results for MacCormack's method are essentially equal to those shown in Plate 3 and they are not included here. Figure 7 presents a comparison of the pressure

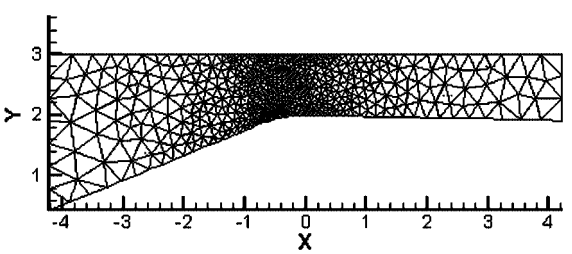

Figure 3. Transonic convergent–divergent nozzle initial mesh.

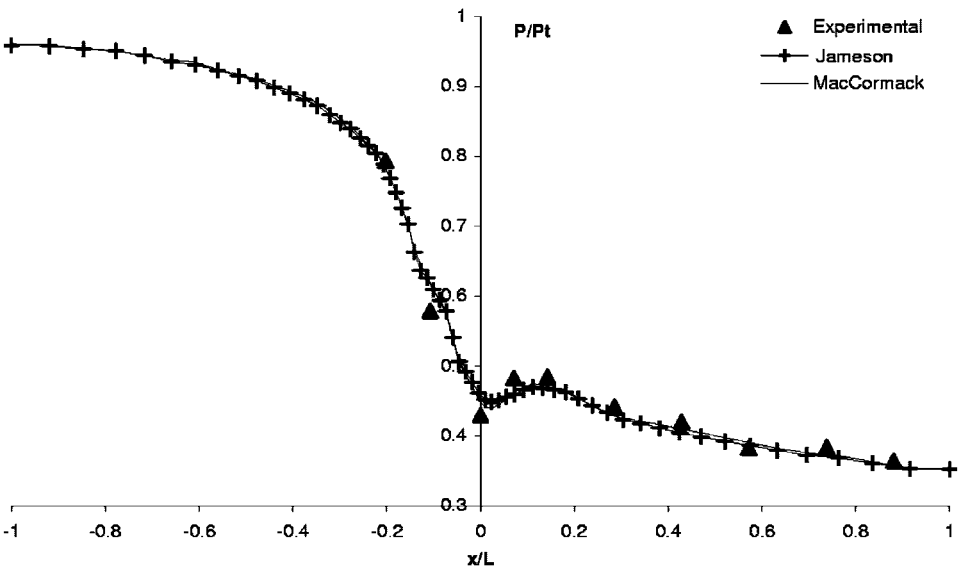

Figure 4. Pressure ratio on the transonic convergent–divergent nozzle wall.

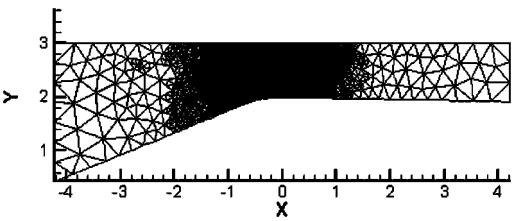

Figure 5. Transonic convergent–divergent nozzle final mesh after two grid refinement passes.

ratio distribution on the nozzle wall calculated using Jameson's method in the non-adapted and adapted meshes. In this figure, one can observe that the use of adaptive mesh refinement diminished the thickness of the shock, as expected. Furthermore, the solution obtained in the mesh after two adaptation passes has very good correlation with the experimental data.

4.2. Convergent nozzle with a supersonic entrance

Figure 8 presents the initial mesh for this nozzle. The test case considers a supersonic flow with $M = 1.6$ at the nozzle entrance. The evolution of the mesh can be observed from Figures 8–11. The threshold value for the sensor is set to 0.1 in this simulation both for refinement and de-refinement. The second mesh, shown in Figure 9, is obtained after 1500 iterations in the original mesh and one grid refinement pass. Figure 10 presents the mesh which results from a grid de-refinement pass, which occurs after 3000 iterations in the second mesh, followed by an immediate grid refinement of the resulting de-refined mesh. Finally, the final mesh is presented in Figure 11. This mesh is generated from the grid shown in Figure

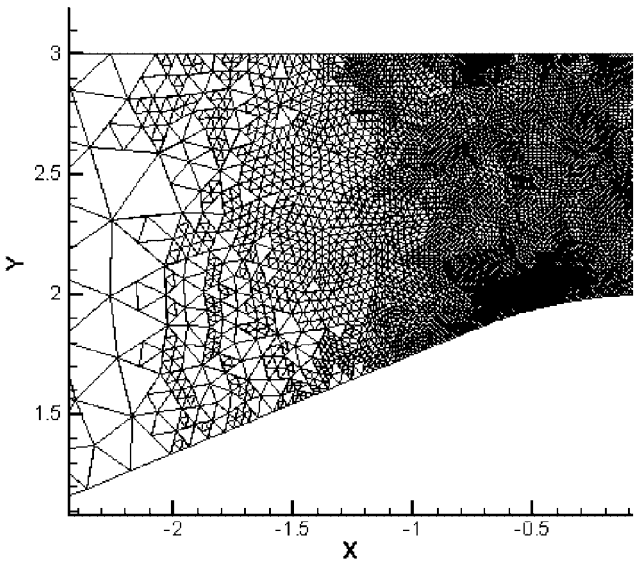

Figure 6. Mesh detail of the final mesh after two mesh refinement passes for the transonic convergent–divergent nozzle.

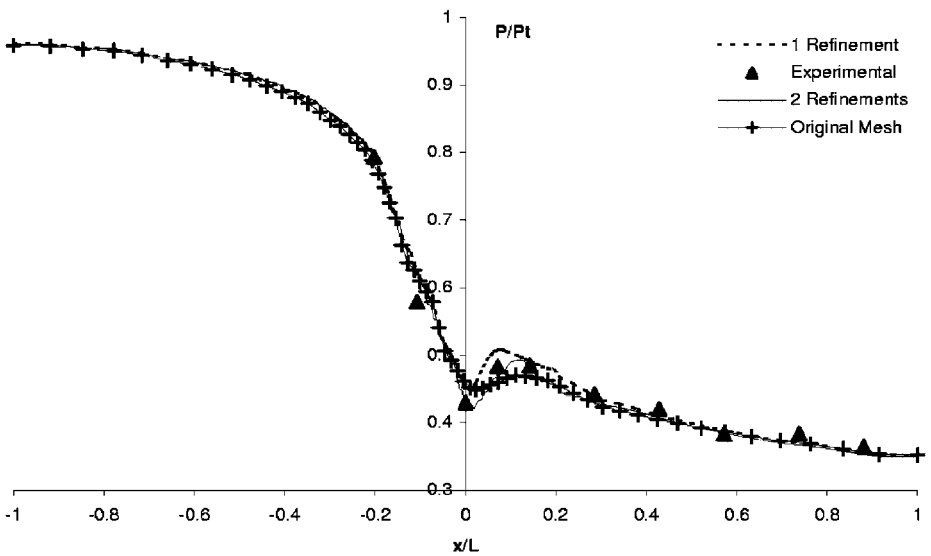

Figure 7. Pressure ratio on the transonic convergent–divergent nozzle wall.

10 by a refinement pass, which occurs after 3000 iterations. Experiences with this test case indicate that the de-refinement routine does not significantly reduce the quantity of points in the mesh. From these results, it appears that the de-refinement routine would be best suited

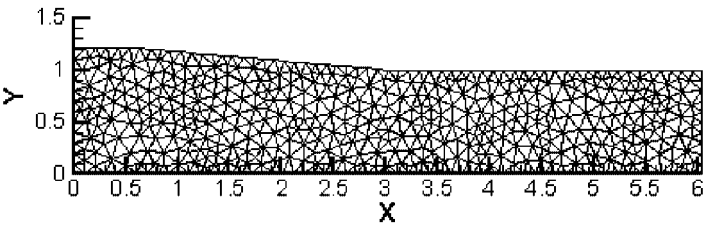

Figure 8. Convergent nozzle geometry and initial mesh.

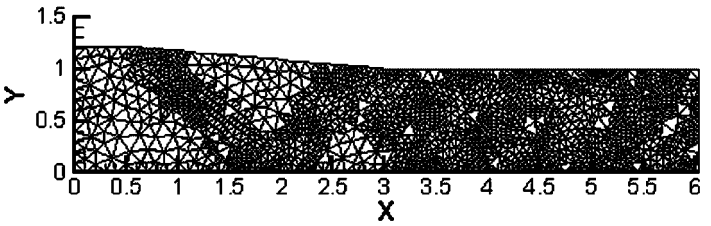

Figure 9. Convergent nozzle mesh after one grid refinement pass.

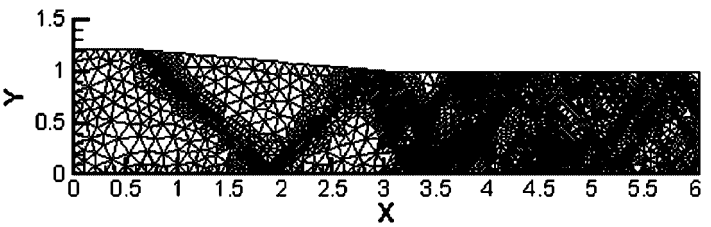

Figure 10. Convergent nozzle mesh after the original grid refinement pass followed by 1 mesh de-refinement pass and one mesh refinement pass.

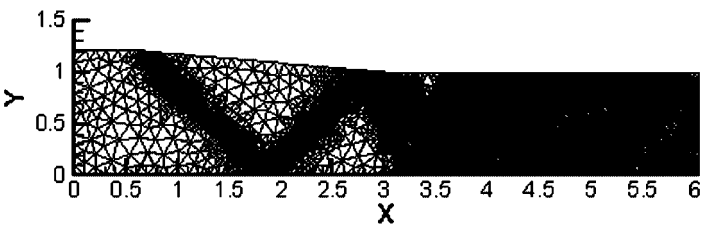

Figure 11. Convergent nozzle mesh after one grid refinement, one grid de-refinement and two grid refinement passes.

for non-stationary problems, in which the regions of the flow with strong gradients can vary substantially. In the present context, in which steady state flow simulations are of concern, the de-refinement procedure, although working correctly, does not appear to be very helpful since it does not significantly reduce the cost of the computations.

The converged solution for dimensionless pressure on the final mesh is presented in Plate 4. 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 shock reflections. In fact, these reflections could be resolved only due to the massive grid refinement in the downstream nozzle region provided by the adaptive mesh routine.

4.3. Transonic flow over a NACA 0012 airfoil

In order to apply the mesh refinement routines to external flows, a NACA 0012 airfoil was simulated under transonic flow conditions. The initial mesh for this problem is presented in Figure 12. The final mesh after two refinement passes is shown in Figure 13. The first mesh refinement occurs after 5000 iterations in the original mesh and the second refinement after additional 7000 iterations in the refined mesh. The threshold value for the density sensor is also set to 0.1 in this case. After the results for the convergent nozzle with supersonic entrance, described in the previous section, the authors opted not to use the de-refinement routine because the gain in computational cost for stationary problems is very small. Furthermore, one can observe that the entire region over the airfoil was refined in the present case and, probably, a larger threshold value for the sensor would be needed.

Plate 5 presents the Mach number contours for the problem in the final mesh for freestream Mach number equal to 0.8. The shock wave is very well defined because of the extreme refinement created in its region. Despite the fact that the mesh is not exactly symmetrical, the contour plots show a symmetrical solution due to refinement. The pressure coefficient distribution on the airfoil wall is presented in Figure 14, which also includes experimental

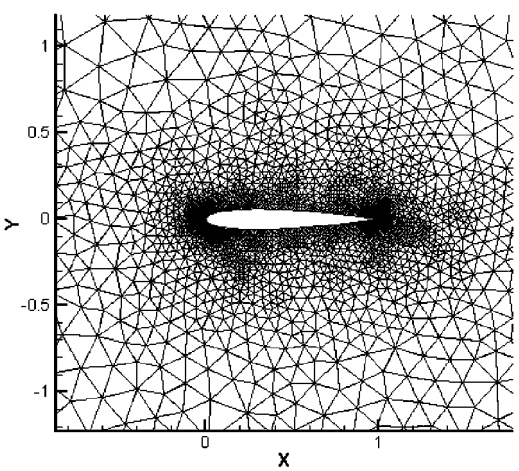

Figure 12. Initial mesh for the NACA 0012 airfoil.

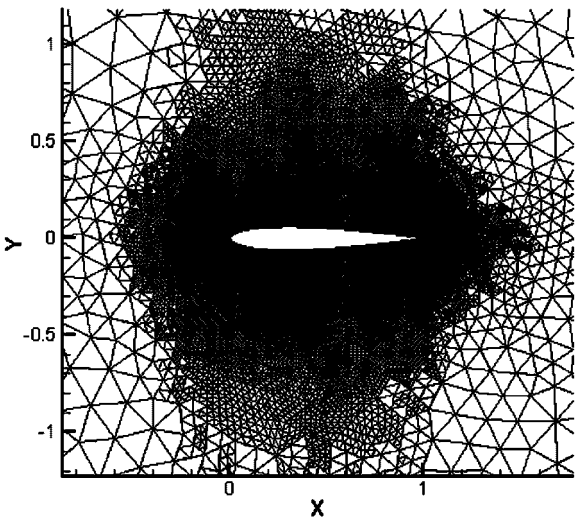

Figure 13. Final mesh after two mesh refinement passes for the NACA 0012 airfoil.

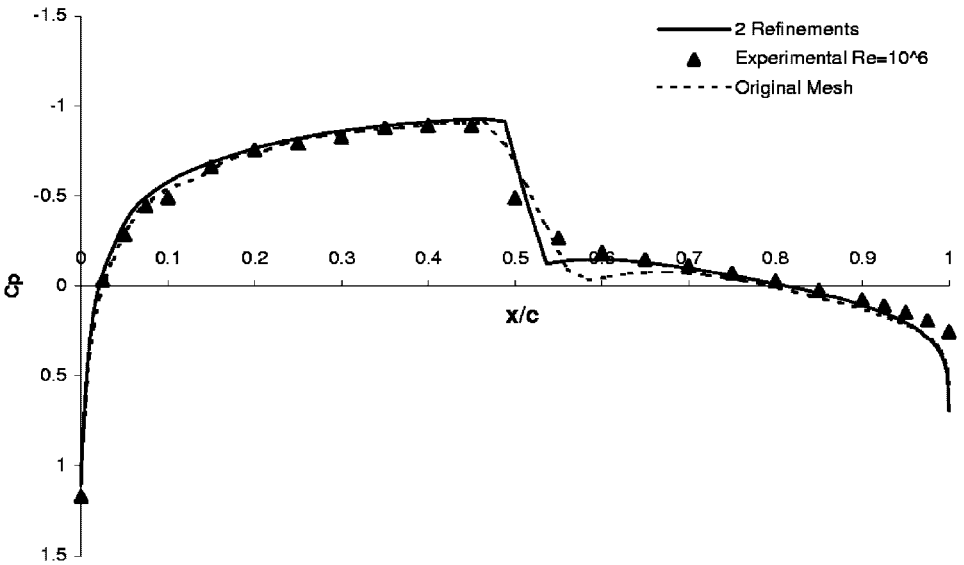

Figure 14. Pressure coefficient distribution on the NACA 0012 airfoil wall for $M_\infty = 0.8$ and $\alpha = 0$.

data from Reference [18]. The use of adaptive mesh refinement decreased the overshoot in the pressure coefficient after the shock and decreased the shock thickness. In this figure, one can observe that the experimental data presents a shock wave which seems to extend over a fairly large region of the airfoil. This is caused by interactions between the shock wave and the boundary layer which the present inviscid formulation cannot predict.

5. CONCLUDING REMARKS

An interpretation for MacCormack's method in an unstructured mesh context was presented. It uses the elements which contain 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 grid 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 significantly decrease the number of points in a stationary problem. At the present time, it seems that the de-refinement procedure would be more useful for non-stationary problems than for the steady state cases here considered. Some results presented showed an excessive mesh refinement that can be solved with some adjustment in the sensor threshold value. In addition, the treatment of hanging nodes in a finite volume context is introduced. The ability to treat hanging nodes in the mesh allowed the implementation of a robust mesh adaptation technique which does not need special routines, such as edge-swapping or edge-splitting, to guarantee mesh quality.

The adaptive refinement technique significantly increased the solution quality for all the cases here presented. The paper underlined the ability of the mesh adaptation to decrease shock thickness and to evidence aerodynamic phenomena not captured on coarse initial meshes.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Fundação de Amparo á 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 present work under the Integrated Project Research Grants No. 522413/96-0 and 501200/2003-7.

REFERENCES

1. Löhner R. *Applied CFD Techniques—An Introduction Based on Finite Element Methods*. Wiley: New York, 2001.
2. Azevedo JLF, Korzenowski H. Comparison of unstructured grid finite volume methods for cold gas hypersonic flow simulations. AIAA Paper 98-2629, *Proceedings of the 16th AIAA Applied Aerodynamics Conference*, Albuquerque, NM, 1998.
3. Korzenowski H, Figueira da Silva LF, Azevedo JLF. Unstructured adaptive grid flow simulations of inert and reactive gas mixtures. *Proceedings of the 14th Brazilian Congress of Mechanical Engineering*, Bauru, SP, Brazil, 1997, publication in CD-ROM format.
4. Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes. AIAA Paper 81-1259, *Proceedings of the 14th AIAA Fluid and Plasma Dynamics Conference*, Palo Alto, CA, 1981.
5. MacCormack RW. The effect of viscosity in hypervelocity impact cratering. AIAA Paper 69-354, *Proceedings of the AIAA Hypervelocity Impact Conference*, Cincinnati, OH, 1969.
6. MacCormack RW. Current status of numerical solutions of the Navier–Stokes equations. AIAA Paper 85-0032, *Proceedings of the 23rd AIAA Aerospace Sciences Meeting*, Reno, NV, 1985.
7. Mavriplis DJ. Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes. *AIAA Journal* 1988; **26**(7):824–831.
8. Mavriplis DJ. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA Journal* 1990; **28**(2):213–221.
9. Azevedo JLF. On the development of unstructured grid finite volume solvers for high speed flows. *Report NT-075*-ASE-N/92, Instituto de Aeronáutica e Espaço, São José dos Campos, SP, Brazil, 1992.
10. Pulliam TH, Steger JL. Implicit finite-difference simulations of three-dimensional compressible flow. *AIAA Journal* 1980; **18**(2):159–167.

11. Long LN, Khan M, Sharp HT. Massively parallel three-dimensional Euler/Navier–Stokes method. *AIAA Journal* 1991; **29**(5):657–666.
12. Azevedo JLF, Fico NGCR, Ortega MA. Two-dimensional and axisymmetric nozzle flow computations using the Euler equations. *Journal of the Brazilian Society of Mechanical Sciences* 1995; **17**(2):147–170.
13. Azevedo JLF, Figueira da Silva LF. The development of an unstructured grid solver for reactive compressible flow applications. AIAA Paper 97-3239, *33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Seattle, WA, 1997.
14. Palmerio B, Dervieux A. Application of a FEM moving node adaptive method to accurate shock capturing. *Proceedings of the First International Conference on Numerical Grid Generation in CFD*, Landshut, Germany, 1986.
15. Babuska I, Zienkiewicz OC, Gago J, de A Oliveira ERA. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*. Wiley: New York, 1986.
16. Mason ML, Putnam LE, Re RJ. The effect of throat contouring on two-dimensional converging-diverging nozzles at sonic conditions. *NASA TP 1704*, 1980.
17. Back LH, Cuffel RF. Detection of oblique shocks in a conical nozzle with a circular-arc throat. *AIAA Journal* 1966; **4**(12):2219–2221.
18. McDevitt J, Okuno AF. Static and dynamic pressure measurements on a NACA 0012 airfoil in the Ames high Reynolds number facility. *NASA TP 2485*, 1985.

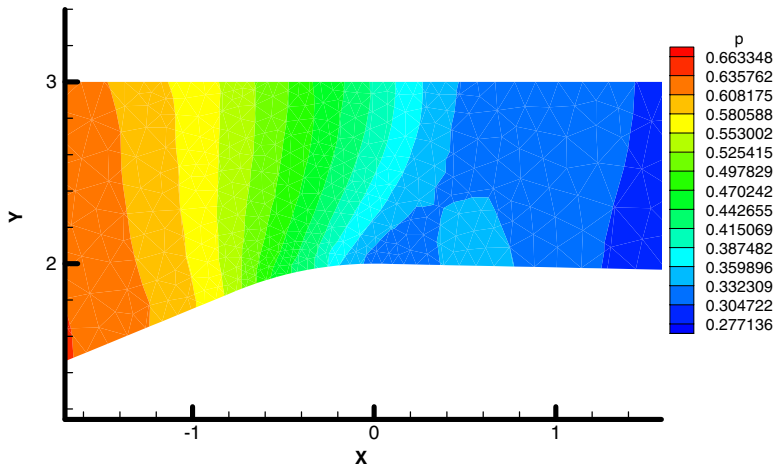

Plate 1. Dimensionless pressure contours on the initial mesh for the transonic convergent–divergent nozzle calculated using MacCormack's method.

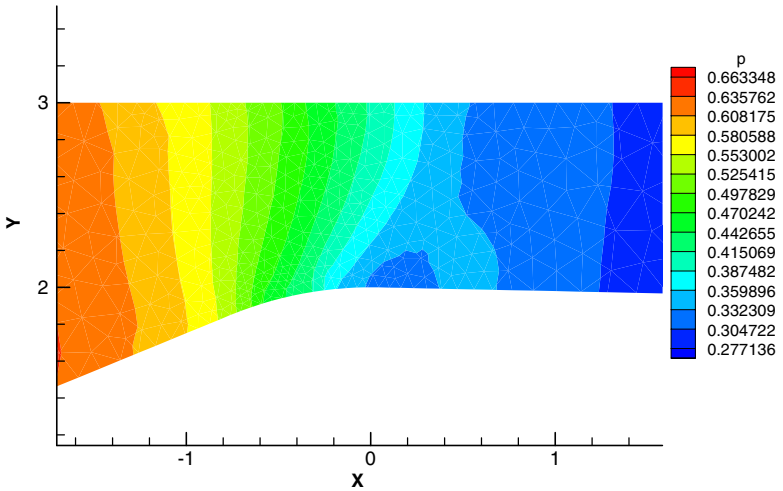

Plate 2. Dimensionless pressure contours on the initial mesh for the transonic convergent–divergent nozzle calculated using Jameson's method.

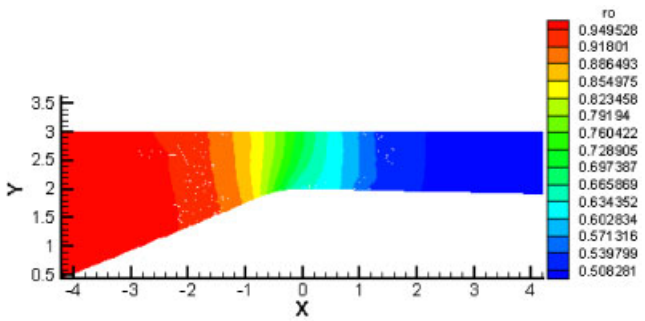

Plate 3. Dimensionless density contours on the final mesh for the transonic convergent–divergent nozzle.

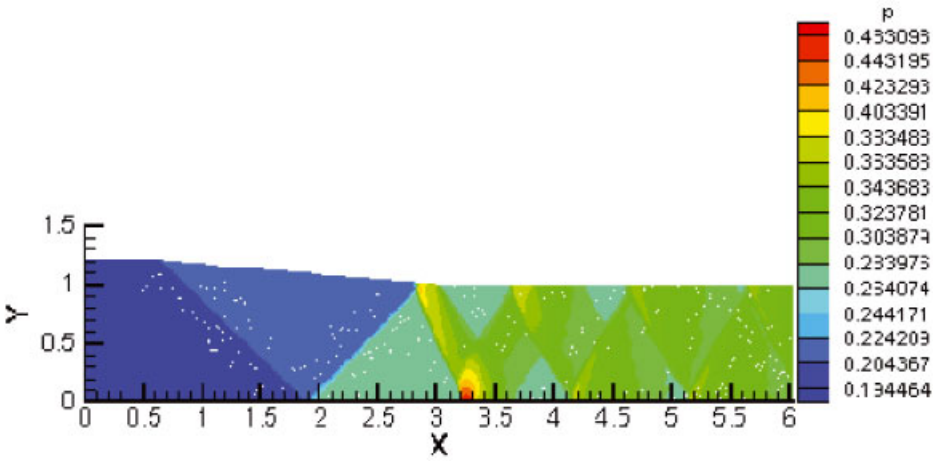

Plate 4. Dimensionless pressure contours for the convergent nozzle with a supersonic entrance in the final mesh.

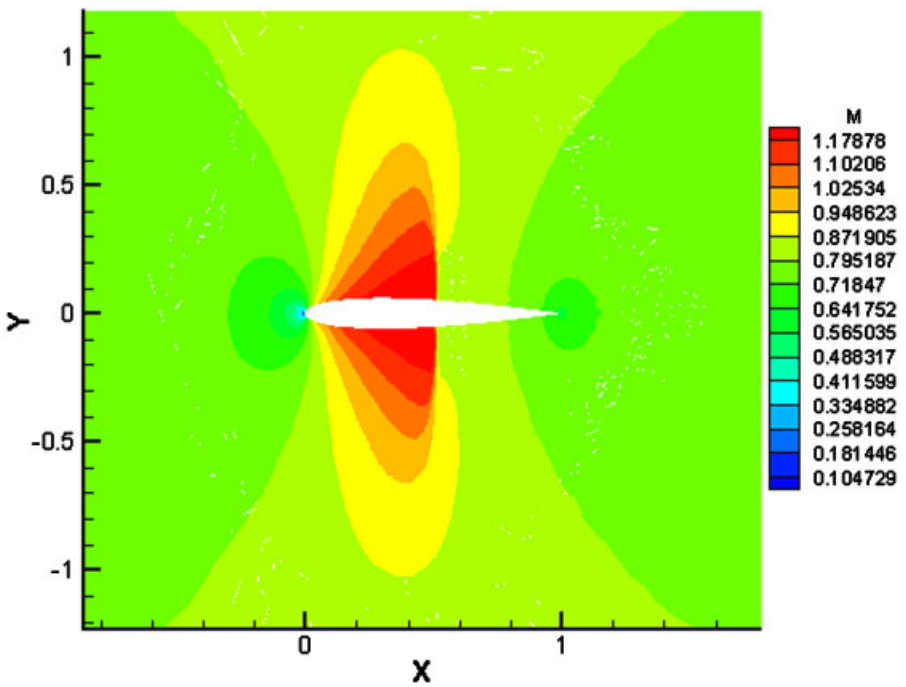

Plate 5. Mach number contours over the NACA 0012 airfoil for $M_\infty = 0.8$ and $\alpha = 0$ in the final mesh.