

A MESH ADAPTATION METHOD BY NODE RE-ALLOCATION USING AN EDGE-BASED ERROR MEASURE

G. Bono^a,
and A. M. Awruch^b

^aFederal University of Rio Grande do Sul
Mechanical Engineering Postgraduate Program
Av. Sarmento Leite 425
CEP: 90050-170, Porto Alegre, RS, Brazil
gbono@mecanica.ufrgs.br

^bFederal University of Rio Grande do Sul
Applied and Computational Mechanical Center
(CEMACOM/PPGEC/UFRGS)
Av. Osvaldo Aranha 99, 3º andar
CEP: 90035-190, Porto Alegre, RS, Brazil
amawruch@ufrgs.br

ABSTRACT

A mesh adaptation technique implemented in an algorithm to simulate compressible flows characterized by the presence of strong shocks, using the finite element method (FEM), is presented in this work. The initial mesh is continuously adapted during the solution process using a node movement technique, keeping as much as possible mesh smoothness and local orthogonality with an unconstrained optimization method. The error is estimated as a function of the Hessian tensor, containing second derivatives of the specific mass, and a Riemann metric projected on the element edges is obtained in order to determine node movements. Time and spatial discretization of the governing equations are carried out using an explicit Taylor-Galerkin scheme and an isoparametric hexahedral element with eight nodes. An Arbitrary Lagrangean Eulerian (ALE) description is used to take into account mesh movement. Finally, some two-dimensional examples involving transonic and supersonic flows are presented to validate the algorithm.

Keywords: Adaptive meshes, compressible flows, finite element, error estimate

INTRODUCTION

Mesh adaptation methods are applied in many engineering fields such as solid mechanics, fluid dynamics, heat transfer, mass transport, combustion and material sciences. Physical phenomena characterizing these areas may develop singularities, localized in some small specific regions, such as stress concentrations, shock waves, detonation waves, boundary layers and recirculating flows. As a consequence of these phenomena, strong variations in some variables occur and the complexity of this kind of problem is based on the fact that regions, where high gradients of a specific variable may be found, are not known “a priori”.

Adaptive methods in computational mechanics are based on a single idea: when the estimated error is greater than a specified value, approximation (e.g. node location, elements size, order of the interpolation functions) must be changed in order to obtain an estimated error which is less than the previously specified value.

To estimate the error, it is necessary to generate a measure of the solution quality, and this measure may vary from an *ad hoc* examination of the solution gradient until accurate *a posteriori* error estimation. Once the error has been estimated, it is necessary to establish how to systematically reduce it to the required level.

The main strategies that may be adopted to obtain smaller values of the error estimate are: to increase the node density in an appropriate region by re-allocation (the *r*-method), to refine the mesh, creating new nodes and elements (the *h*-method), and to increase the order of the approximation function (the *p*-method). These techniques may be combined, as in, for example, the *h-p* method.

In this work the *r*-method is used, continuously and automatically re-allocating nodes in the space-time domain, with a greater concentration of nodes in regions where some variable gradients are higher, but preserving the mesh structure (number of elements, number of nodes and element connectivities), making the computational procedure simpler and cheaper.

The *r*-method was not very popular in the finite element community (Cao et al., 1999) due to difficulties in finding an accurate, efficient and general procedure to move the mesh. However, this method has some important features, such as simple data structure and meshes that may be continuously changed in unsteady flows leading, if a suitable strategy is implemented, to accurate results with low computational costs (Li et al., 2002). Examples of some adaptive node-movement technique in finite volume and finite element methods can be found in Gnoffo (1983), Nakahashi and Deiwert (1987), Palmeiro (1994), McRae (2000), Soni et

al. (2000), Cao et al. (2001), Li et al. (2002) and Azarenok and Tang (2005).

An edge-based interpolation error estimate drives nodal movement to satisfy an optimal mesh criterion. While the optimal mesh criterion is traditionally one in which the error is equidistributed over the elements, in this work the error is equidistributed over the edges. Mesh anisotropy is avoided employing a formulation based in variational principles.

GOVERNING EQUATIONS AND NUMERICAL SCHEME

The mass, momentum and energy conservation equations for compressible flows, neglecting viscous and heat diffusion terms, are given by:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x_i} - w_i \frac{\partial U}{\partial x_i} = 0 \quad (1)$$

with

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho e \end{pmatrix}; \quad \mathbf{F}_i = \begin{pmatrix} \rho v_i \\ \rho v_1 v_i + p \delta_{i1} \\ \rho v_2 v_i + p \delta_{i2} \\ \rho v_3 v_i + p \delta_{i3} \\ v_i (\rho e + p) \end{pmatrix} \quad (2)$$

where $i = 1, 2, 3$, and \mathbf{U} and \mathbf{F}_i are vectors containing field and flux variables, respectively. In these expressions, v_i and w_i are the fluid and mesh velocity components, respectively, ρ is the specific mass, p is the thermodynamic pressure and e is the total energy. Finally, x_i and t are the spatial and time coordinates, respectively.

The state equation is given by:

$$p = \rho(\gamma - 1) \left(e - \frac{v_i v_i}{2} \right) \quad (3)$$

with $\gamma = c_v/c_p$, where c_v and c_p are the specific heat coefficients at constant volume and pressure, respectively.

The system of partial differential equations is solved using the finite element method (Löhner

et al., 1985), employing a Taylor series and the classical Bubnov-Galerkin method for time and space discretization, respectively. The problem is completely defined when initial and boundary conditions are added to Eqs. (1), (2) and (3).

An isoparametric eight-node hexahedral element is used and the corresponding element matrices are analytically obtained employing reduced numerical integration. This code has been validated with respect to analytical and experimental results for several regimes of compressible gas flows (Kessler and Awruch, 2004; Bono, 2004).

ADAPTIVE METHOD

Accuracy of a numerical scheme may be improved if a fixed number of nodes is dynamically re-allocated, concentrating these nodes in regions where important variations of some variable values occur. This procedure may sometimes be more efficient in terms of processing time and computer memory than refinement techniques (where new nodes and elements are created).

Stability of most numerical schemes may depend on the mesh quality. For this reason, excessive mesh distortion, without any control, must be avoided using a smoothing process and preserving mesh regularity.

The adaptive method presented in this work is based on a variational principle, and its main objective is to build a mesh with an effective control of conflicting requirements such as mesh regularity (sudden variations of the size in neighbor elements must be avoided), local orthogonality (preserving the element shape without important distortions) and mesh adaptation (concentrating nodes where they are necessary).

The adaptive strategy presented in this work is based on the variational formulation presented by Brackbill and Saltzman (1982) and the optimization procedure follows those presented by Carcaillet et al. (1986) and Kennon and Dulikravich (1986). The edge-based error estimate is based on the work of Ait-Ali-Yahia et al. (1996). A node-moving scheme is then implemented, re-allocating grid points along those directions where the error is high.

THEORETICAL ASPECTS

Although the formulation will be presented for two-dimensional (2-D) flows, because it is

easier to understand how the algorithm works, this method was also implemented to deal with three-dimensional (3-D) flows.

Brackbill and Saltzman (1982) presented a 2-D mesh generation technique with mesh quality and mesh adaption control based on measures of properties obtained from the mapping between a uniform mesh in the computational space, $\xi - \eta$, and a suitable mesh in the physical space, $x_1 - x_2$. In order to improve computational efficiency and the reliability of this method Carcaillet et al. (1986) and Kennon and Dulikravich (1986) adopted a more heuristic formulation for the local adaptation problem.

Consider a typical cell, formed by four elements in the two-dimensional case (in a 3-D case the cell would be formed by eight elements), as shown in Fig. 1, where $P_{ij} = P(x_{ij}, y_{ij})$ is a common node belonging to the four elements forming the cell, which is linked to the other nodes by straight segments defined as position vectors r_{ij} .

A measure quantifying the local smoothness, denoted by SM_{ij} , is given by the sum of the square values of the differences between the areas of elements forming the typical cell (Carcaillet et al., 1986). The sum will be zero if all adjacent elements have the same area.

The four position vectors with origin at the node P_{ij} are used to form four dot products, which are squared and summed to control orthogonality of the typical cell (Carcaillet et al., 1986), denoted by OR_{ij} . The dot products are chosen so that their sum is zero if the grid is locally orthogonal.

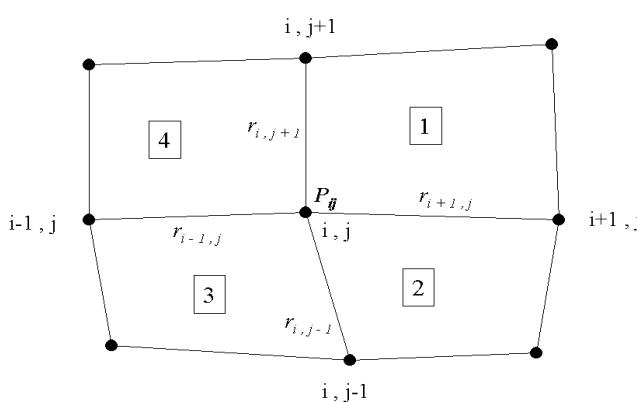


Figure 1. Typical cell defined for the two-dimensional case

Then, local smoothness SM_{ij} and local orthogonality OR_{ij} are given by:

$$SM_{ij} = (A_1 - A_2)^2 + (A_2 - A_3)^2 + (A_3 - A_4)^2 + (A_4 - A_1)^2 \quad (4)$$

$$OR_{ij} = (r_{i+1,j} \cdot r_{i,j+1})^2 + (r_{i,j-1} \cdot r_{i+1,j})^2 + (r_{i-1,j} \cdot r_{i,j-1})^2 + (r_{i,j+1} \cdot r_{i-1,j})^2 \quad (5)$$

where A_k is the area of the k elements forming the typical cell.

The cell area control of a typical cell is calculated with the following expression:

$$VC_{ij} = A_{ij} \cdot W_{ij} \quad (6)$$

where A_{ij} is the total area of the cell and W_{ij} is a monitoring function, which gives positive values and is evaluated at the node P_{ij} .

The choice of the weight function W_{ij} in the volume control is a very important aspect, because this parameter indicates where the adaption process will take place. It may be observed that greater values of W_{ij} correspond to decreasing values of the typical cell area (or volume in a 3-D case) and vice versa.

The global objective function F is obtained by a weighted linear combination of local measures of mesh quality (local smoothing and local orthogonality) and the local volume control of a typical cell. Then, the global objective function is given by:

$$F = \sum_{i=1}^m \sum_{j=1}^n \left[\delta \cdot \frac{OR_{ij}}{OR_{\max}} + (1-\delta) \frac{SM_{ij}}{SM_{\max}} + \beta \cdot VC_{ij} \right] \quad (7)$$

with $0 \leq \delta \leq 1$ and $0 \leq \beta \leq 1$, where δ and β are weighting parameters to control orthogonality, smoothing and cell area (or volume for 3-D problems), while OR_{\max} and SM_{\max} are the largest values of OR_{ij} and SM_{ij} , respectively, in order to ensure values of the same order in Eq. (7); m and n are the number of nodes in the i and j directions, respectively.

The global objective function may be written as a function of the vector T , which contains the nodal physical coordinates in a natural order:

$$T = \{(x_{ij}, y_{ij}): 1 \leq i \leq m, 1 \leq j \leq n\} \quad (8)$$

where m and n are the total number of nodes in the i -direction and j -direction, respectively.

Then, in Eq. (7), the objective function may be written as $F = F(\mathbf{T})$, which has $2 \times n \times m$ variables. The unconstrained minimization of $F(\mathbf{T})$ is carried out using the Fletcher-Reeves conjugate gradient method (Press et al., 1992). Actually, some restrictions are set to the motion of nodes belonging to the boundaries.

For the three-dimensional case, smoothness and orthogonality measures used by Kennon and Dulikravich (1986) were adopted. The corresponding expressions are given by the following respective expressions:

$$\begin{aligned} SM_{ijk} = & (r_{i+1,j,k} \cdot r_{i+1,j,k}) + (r_{i,j-1,k} \cdot r_{i,j-1,k}) + \\ & + (r_{i-1,j,k} \cdot r_{i-1,j,k}) + (r_{i,j+1,k} \cdot r_{i,j+1,k}) + \quad (9) \\ & + (r_{i,j,k+1} \cdot r_{i,j,k+1}) + (r_{i,j,k-1} \cdot r_{i,j,k-1}) \end{aligned}$$

and

$$\begin{aligned} OR_{ijk} = & (r_{i+1,j,k} \cdot r_{i,j+1,k})^2 + (r_{i,j-1,k} \cdot r_{i+1,j,k})^2 + \\ & + (r_{i-1,j,k} \cdot r_{i,j-1,k})^2 + (r_{i,j+1,k} \cdot r_{i-1,j,k})^2 + \\ & + (r_{i+1,j,k} \cdot r_{i,j,k-1})^2 + (r_{i,j-1,k} \cdot r_{i,j,k-1})^2 + \quad (10) \\ & + (r_{i-1,j,k} \cdot r_{i,j,k-1})^2 + (r_{i,j+1,k} \cdot r_{i,j,k-1})^2 + \\ & + (r_{i+1,j,k} \cdot r_{i,j,k+1})^2 + (r_{i,j-1,k} \cdot r_{i,j,k+1})^2 + \\ & + (r_{i-1,j,k} \cdot r_{i,j,k+1})^2 + (r_{i,j+1,k} \cdot r_{i,j,k+1})^2 \end{aligned}$$

ERROR ESTIMATION

As error estimation is only useful as an indication of the relative error between two successive meshes, a computationally simple error estimation was adopted in order to improve the mesh adaptation strategy.

Consider a one-dimensional problem, where the variable ρ is approximated by ρ_h , using linear interpolation functions. The total error in the

interval $[0, h_e]$, where h_e is the length of the element e , is given by (Peraire et al., 1987):

$$E_e^{rms} = \left\{ \int_0^{h_e} \frac{E_e^2}{h_e} d\xi \right\}^{1/2} = \frac{1}{\sqrt{120}} h_e^{1/2} \left| \frac{d^2 \rho_h}{dx^2} \right|_e \quad (11)$$

where rms means root mean square and ξ is the element local coordinate. The optimized mesh is defined as the mesh where E_e^{rms} is uniformly distributed over the element domain, i.e. $E_e^{rms} = C$, where C is a constant and positive tolerance specified by the user.

For a two or a three-dimensional problem, the second derivative of ρ_h with respect to a direction defined by the versor V is given by:

$$\frac{\partial^2 \rho_h}{\partial V^2} = V^T \mathbf{H} V \quad (12)$$

where \mathbf{H} is the Hessian matrix.

As ρ_h is interpolated with a linear function, the second derivative of cannot be represented. In order to obtain the second derivative of at a node I , a weak formulation was employed (similar to that of Ait-Ali-Yahia et al., 1996) and the following expression was obtained:

$$\left. \frac{\partial^2 \rho_h}{\partial x_j^2} \right|_I = \frac{\left[- \int_{\Omega_I} \left(\frac{\partial \phi^T}{\partial x_j} \phi \right) d\Omega \right] \left[\frac{\partial \rho_h}{\partial x_i} \right] + \left[\int_{\Gamma_I} \phi^T \phi n_j d\Gamma \right] \left[\frac{\partial \rho_h}{\partial x_i} \right]}{\left[\int_{\Omega_I} \phi^T \phi d\Omega \right]} \quad (13)$$

where ϕ is a vector containing the elements shape functions, Ω_I is the area (or volume in the 3-D case) of the elements sharing the node I and Γ_I is the corresponding boundary. The index I varies from 1 to the total number of nodes in the finite element mesh, n_j represents the cosine of the angle formed by a normal axis to Γ_I with the coordinates axis x_j .

A lumped mass matrix to represent the denominator in Eq. (13) was used. The first derivatives of ρ_h are nodal values and were obtained using a smoothing process based in the mean square method. Matrix \mathbf{H} may be diagonalized and, in this case, Eq. (12) may be written as:

$$\frac{\partial^2 \rho_h}{\partial V^2} = V^T \mathbf{R} \mathbf{A} \mathbf{R}^T V \quad (14)$$

where A is a diagonal square matrix with the eigenvalues of matrix H and R contains the corresponding eigenvectors. As the error must be a positive value, the second derivative of ρ_h in any direction has the following restriction:

$$\left| \frac{\partial^2 \rho_h}{\partial V^2} \right| = |V^T H V| \leq V^T \bar{H} V \quad (15)$$

where the modified Hessian matrix \bar{H} is given by:

$$\bar{H} = R |A| R^T \quad (16)$$

In Eq. (16), $|A|$ contains the absolute values of the eigenvalues. For example, in a 2-D problem, absolute values of the second derivatives of ρ_h in the directions \vec{e}_1 and \vec{e}_2 are given by the absolute values of the eigenvalues contained in $|A|$, where the system (\vec{e}_1, \vec{e}_2) is obtained rotating an angle γ the global system (\vec{e}_1, \vec{e}_2) with the rotation matrix. In this case, V contains the base vectors of the system (\vec{e}_1, \vec{e}_2) .

The criterion of mesh adaptation for a one-dimensional case, taking into account Eq. (11) and a uniform distribution of the error over the element domain, is given by:

$$h_e^2 \left| \frac{d^2 \rho_h}{dx^2} \right|_e = C \quad (17)$$

and extending this concept to a 2-D or a 3-D case, taking into account Eq.(15), Eq. (18) is obtained:

$$h_e^2 V^T \bar{H} V = C = d^2 \quad (18)$$

In this work, the error is uniformly distributed on element edges and in this case h_e represents the length of an element edge and V is a versor in the same direction of this specific edge. The term $V^T \bar{H} V$ is defined as the Riemann metric. As \bar{H} can be obtained for each end of the element edge, mean values are used. Then, in Eq. (18), \bar{H} may be substituted by \bar{H}_m , being the Hessian matrix evaluated at the mid point of the straight segment representing the element edge.

Then, an optimized mesh is obtained when the lengths of all edges in the Riemann metric are equal to $d = \sqrt{C}$, where d represents the directional error. In this work, VC_{ij} in Eq. (7) is defined as

being d^2 , given by Eq. (18), instead of the value given in Eq. (6).

NUMERICAL APPLICATIONS

For all test cases investigated in this section, the specific mass was the variable used for the error estimate, and the adaptive process was applied when a small value of the residual value corresponding to ρ was reached. When the adaptive process was applied, an ALE description (Löhner, 2001) was used. The velocity components of the nodes were calculated with $w_1 = \Delta x / \Delta t$, $w_2 = \Delta y / \Delta t$ and $w_3 = \Delta z / \Delta t$, where Δx , Δy and Δz are the components of node displacement and Δt is the time interval. The new nodal coordinates are obtained after applying relaxation coefficients θ_x , θ_y and θ_z to the increments Δx , Δy and Δz , respectively. Each relaxation coefficient varies between 0 and 1.

Although in the present work only two-dimensional examples are presented, they were simulated with a three-dimensional solver taking one layer of elements in the z -direction, which is perpendicular to the flow direction. The following values were adopted for the weighting parameters in Eq. (7): $\delta = 0.5$ and $\beta = 1.0$. For more details see Bono (2004). Fluid properties are considered constant, with $\gamma = 1.4$.

Example 1: A steady supersonic flow over a ramp

A steady supersonic flow over a ramp forming an angle of 16° with the horizontal axis, was analyzed. This example tests certain features of the algorithm, including the resolution of the oblique shock and its proper angle. The free-stream flow has the following properties: Mach number $M_\infty = 3.0$ and specific mass $\rho = 1.0$. The domain is discretized into a mesh of 2322 nodes and 1092 elements.

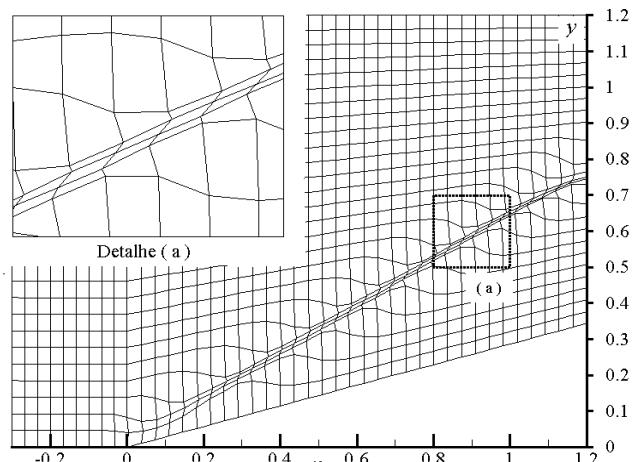


Figure 2. Final mesh after the adaption process

In Fig. 2 the final mesh is shown; it can be observed that elements and nodes are concentrated in the region where shock wave exists. This result agrees well with the analytical values reported by Anderson (1991).

Distribution of the specific mass for $y = 0.6$ and $0 \leq x \leq 1.2$ is shown in Fig. 3, where the difference between the gradients obtained with the initial and the final mesh may be observed.

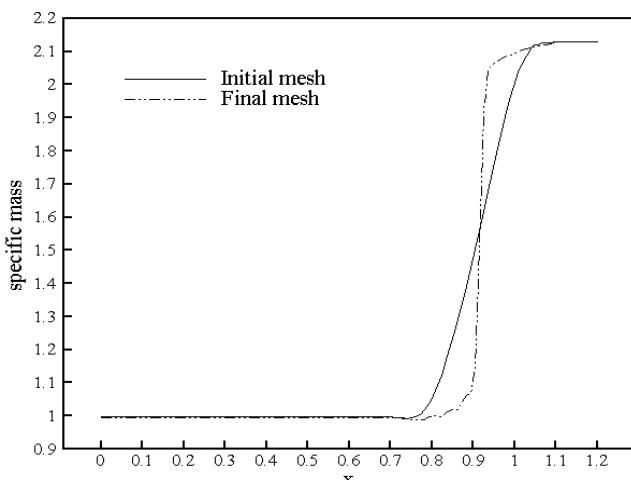


Figure 3. Specific mass distribution over a ramp

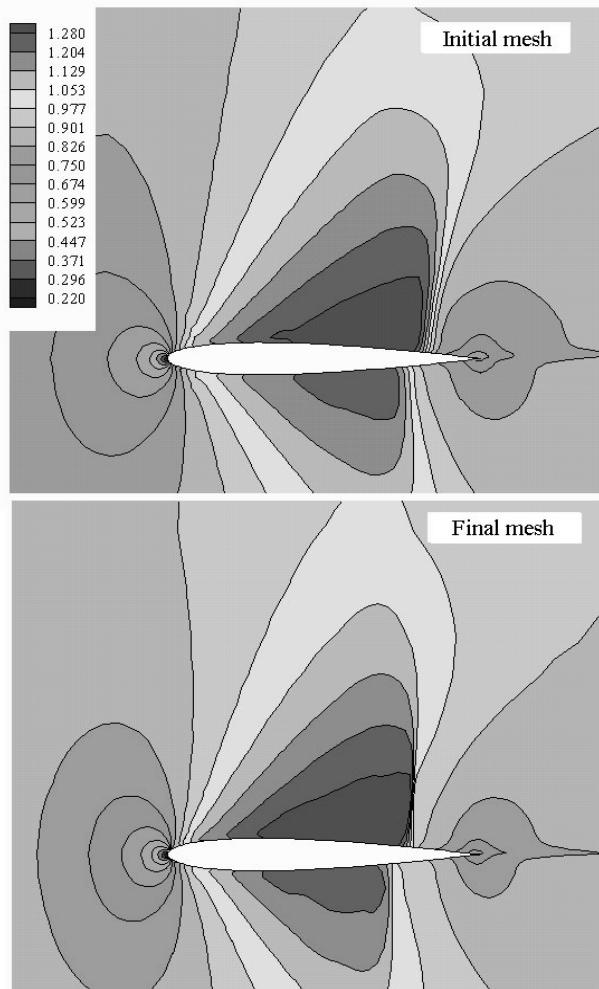


Figure 4. Distribution of the Mach number

Example 2: Transonic flow around a NACA 0012

In this test case, a NACA 0012 airfoil is placed into a transonic flow with $M_\infty = 0.8$ and an angle of attack equal to 1.25 deg. The mesh has 9940 nodes and 4800 elements. The Mach number distribution corresponding to the initial mesh and final mesh are presented in Fig. 4. The latter is similar to that presented by Baumann et al. (1992), but the initial mesh was well refined in the region where the shock wave is located.

Example 3: Supersonic flow over a forward facing step

In this example, the current methodology is applied to a steady supersonic flow with $M_\infty = 3$ over a forward facing step. The initial mesh, formed by uniformly distributed elements, has 8386 nodes and 4032 elements.

The final mesh with some details is presented in Fig. 5 and the specific mass distribution for $y = 0.4$ and $-0.4 \leq x \leq 0.4$ is shown in Fig. 6. The distributions of the Mach number for both meshes are shown in Fig. 7. Results are close to those presented by Löhner et al. (1985).

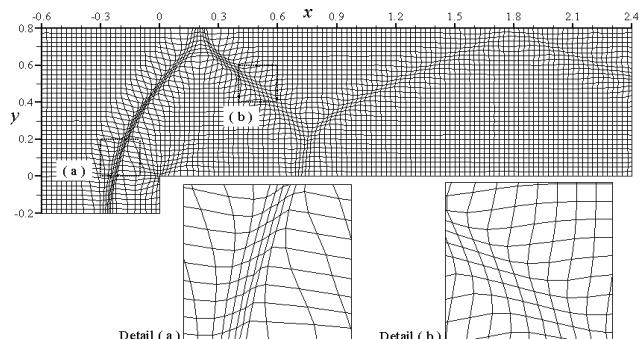


Figure 5. Final mesh after the adaption process

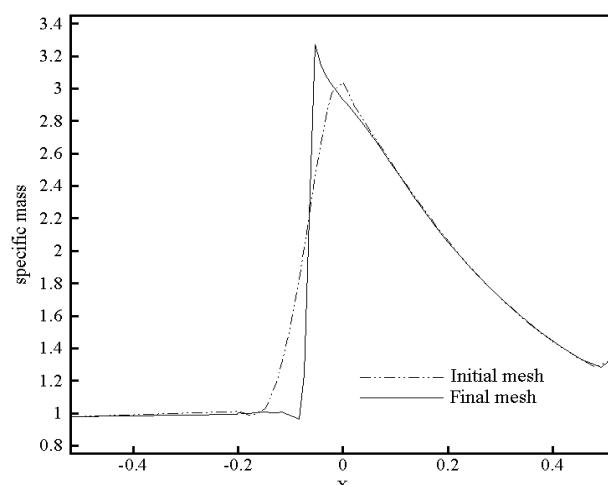


Figure 6. Specific mass distribution

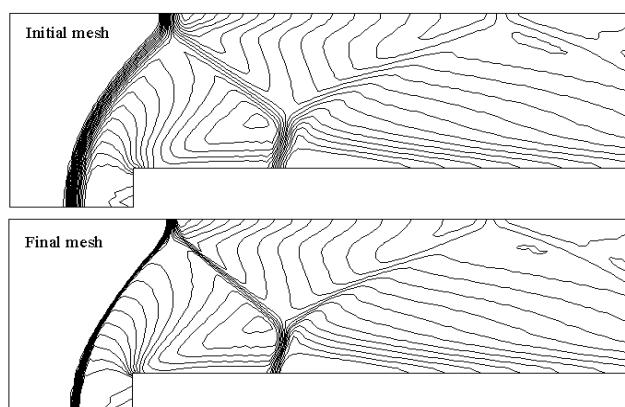


Figure 7. Mach number distribution

Example 4: Supersonic flow past a double ellipse

The supersonic flow ($M_\infty = 2$) with an angle of attack of 20 deg past a double ellipse is analyzed. The semi-axes of the large ellipse are 60 cm and 15 cm, and the semis-axes of the small ellipse are 35 cm and 25 cm. The mesh has 9432 nodes and 4550 elements.

In Fig. 8 the initial and the final mesh are shown and it may be observed that the elements are aligned with the shock wave maintaining a high quality mesh. Distributions of the specific mass for both meshes are shown in Fig. 9; it is observed that the adaptive method improves results in regions with large gradients.

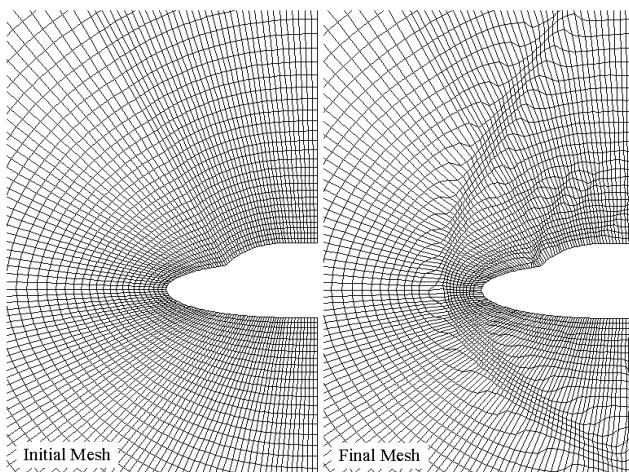


Figure 8. Initial and final mesh

CONCLUSIONS

A methodology to adapt finite element meshes to simulate compressible flows with strong shock waves is presented in this work. The method is characterized by an error estimation measured in the elements edges using a Riemann metric, which is defined employing the Hessian matrix.

An optimization procedure is used to preserve, as much as possible, mesh smoothness and orthogonality.

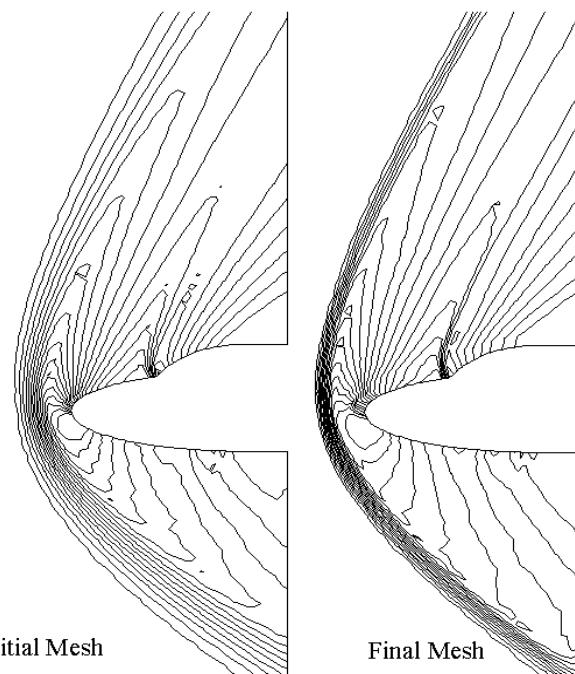


Figure 9. Specific mass distribution

The focus of the present paper is the presentation of the adaptive method and its validation through a comparative study of four classical numerical test problems. Good results for transonic and supersonic flows were obtained, showing that substantially improved results may be obtained when using the adaptive procedure in comparison to those obtained with the initial mesh (even when very fine meshes were used). It is important to highlight that meshes with good quality were attained for the four studied cases, due to smoothing and orthogonality control.

REFERENCES

Ait-Ali-Yahia, D., Habashi, W. G., and Tam, A., 1996, A Directionally Adaptive Methodology Using an Edge-Based Error Estimate on Quadrilateral Grids, *International Journal for Numerical Methods in Fluids*, Vol. 23, pp. 673-690.

Anderson, J. D., 1991, *Fundamentals of Aerodynamics*, 2nd Edition, McGraw-Hill, USA.

Azarenok, B. N., and Tang, T., 2005, Second-order Godunov-type Scheme for Reactive Flow Calculations on Moving Scheme, *Journal of Computational Physics*, Vol. 206, pp. 48-80.

Baumann, C. E., Storti, M. A., and Idelsohn, C., 1992, A Petrov-Galerkin Technique for the Solution of Transonic and Supersonic Flows, Computer Methods in Applied Mechanics and Engineering, Vol. 95, pp. 49-70.

Bono, G., 2004, Adaptação via Movimento de Malhas em Escoamentos Compressíveis, M.Sc. Thesis, PROMEC, Universidade Federal do Rio Grande do Sul, Brazil. (*in Portuguese*)

Brackbill, J. U., and Saltzman, J. S., 1982, Adaptive Zoning for Singular Problems in Two Dimensions, Journal of Computational Physics, Vol. 46, pp. 342-368.

Cao, W., Huang, W., and Russell, R. D., 1999, An r-Adaptive Finite Element Method Based upon Moving Mesh PDEs, Journal of Computational Physics, Vol. 149, pp. 221-244.

Cao, W., Huang, W., and Russell, R. D., 2001, Comparison of Two-dimensional r-Adaptive Finite Element Methods Using various Error Indicators, Mathematics and Computers in Simulation, Vol. 56, pp. 127-143.

Carcaillet, R., Dulikravich, G. S., and Kennon, S. R., 1986, Generation of Solution-adaptive Computational Grids Using Optimization, Computer Methods in Applied Mechanics and Engineering, Vol. 57, pp. 279-295.

Gnoffo, P. A., 1983, A Finite-volume, Adaptive Grid Algorithm Applied to Planetary Entry Flowfields, AIAA Journal, Vol. 21, pp. 1249-1254.

Kennon, S. R., and Dulikravich, G. S., 1986, Generation of Computational Grids Using Optimization, AIAA Journal, Vol. 24, pp. 1069-1073.

Kessler, M. P., and Awruch, M. A., 2004, Analysis of Hypersonic Flows Using Finite Elements with Taylor-Galerkin Scheme, International Journal for Numerical Methods in Fluids, Vol. 44, pp. 1355-1376.

Li, R., Tang., T., and Zhang, P., 2002, A Moving Mesh Finite Element Algorithm for Singular Problems in Two and Three Space Dimensions, Journal of Computational Physics, Vol. 177, pp. 365-393.

Löhner, R., Morgan, K., and Zienkiewicz, O. C., 1985, An Adaptive Finite Element Procedure for Compressible High Speed Flows, Computer Methods in Applied Mechanics and Engineering, Vol. 51, pp. 441-465.

Löhner, R., 2001, *Applied Computational Fluid Dynamics Technique: An Introduction Based*

on Finite Element Methods, John Wiley & Sons Ltd., USA.

McRae, D. S., 2000, *r*-Refinement Grid Adaptation Algorithms and Issues, Computer Methods in Applied Mechanics and Engineering, Vol. 189, pp. 1161-1182.

Nakahashi, K., and Deiwert, G. S., 1987, Self-adaptive-grid Method with Application to Airfoil Flow, AIAA Journal, Vol. 25, pp. 513-520.

Palmeiro, B., 1994, An Attraction-repulsion Mesh Adaption Model for Flow Solution on Unstructured Grids, Computers Fluids, Vol. 23, pp. 487-506.

Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., 1987, Adaptive Remeshing for Compressible Flow Computations, Journal of Computational Physics, Vol. 72, pp. 449-466.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1992, *Numerical Recipes in Fortran 77*, 2nd Edition, Cambridge University Press, USA.

Soni, B. K., Koomullil, R., Thompson, D. S., and Thornburg, H., 2000, Solution Adaptive Grid Strategies Based on Point Redistribution, Computer Methods in Applied Mechanics and Engineering, Vol. 189, pp. 1183-1204.