

## MOVING AND ADAPTIVE GRID METHODS FOR COMPRESSIBLE FLOWS

P-16

Jean-Yves Trépanier and Ricardo Camarero  
Department of Mechanical Engineering  
École Polytechnique de Montréal, C.P. 6079, Succ. Centre-Ville,  
Montréal, Québec, Canada, H3C 3A7

## SUMMARY

This paper describes adaptive grid methods developed specifically for compressible flow computations. The basic flow solver is a finite-volume implementation of Roe's flux difference splitting scheme on arbitrarily moving unstructured triangular meshes. The grid adaptation is performed according to geometric and flow requirements. Some results are included to illustrate the potential of the methodology.

## INTRODUCTION

A large number of engineering flow problems are concerned with the numerical simulation of unsteady compressible flows in complex geometries with moving boundaries. Examples are internal gas dynamics with pistons, external flows with bodies in relative motion (store separation, etc.). Our own motivation was related to the prediction of the internal flow in a circuit-breaker, which involves electrodes and piston in relative motion [1].

The computational tools required to tackle these type of problems are still a research area. Only from the grid point of view, different schools can be found ranging from overset structured grids to global unstructured grid remeshing at each time step, and the research is still very active in this domain.

Our own approach is to use an unstructured triangular grid. This choice was driven by many factors. First, triangular grids offers a great flexibility in gridding complex geometries with various length scales; second, their potential for automation and adaptation is clear; third, it simplifies the coding of the flow solver which has no special cases to handle. From this choice, we also select to perform adaptation by modifying the grid with local actions because in many problems, only a small portion of the grid need to be modified when adaptation is done. The flow solver also need to take into account properly the grid motion and this was assured using an ALE version of Roe's flux difference splitting scheme.

This technology has enable the investigation of various adaptation strategies, including a novel shock fitting approach where the discontinuities are captured at the interfaces of two triangles.

The paper is organised as follows : we first give a description of the grid management algorithm, followed by a few words about the moving grid flow solver. We then present various adaptive strategies using grid relocation and grid enrichment. Finally, some conclusions are drawn.

## GRID MANAGEMENT

### Temporal Evolution of the Grid

A set of curves serves to describe the geometry and its evolution is described in terms of the velocity of these curves. The temporal evolution of the grid is performed in two steps. First, one computes the effect of the moving curves on the grid and second, a smoothing term is added. According to this, the velocity of the grid nodes can be represented by:

$$w = w_g + w_s$$

where  $w_g$  is the geometric grid velocity and  $w_s$  the smoothing grid velocity. Both of these terms must respect the boundary conditions defined by the movement of the curves.

The geometric term  $w_g$  depends on the two types of curve-node interaction considered: Dirichlet and Neumann. In a Dirichlet curve-node interaction, the velocity of the grid nodes that lie on a moving curve is set equal to the velocity of the curve. There is no relative motion of the nodes with respect to the curve. In a Neumann curve-node interaction, the nodal velocity is set equal to the normal component of the curve velocity at the position of the grid node. This is the minimal constraint which can be imposed on the grid node in order to remain on the curve.

The last situation to be considered is the curve-curve interaction. This happens when a grid node is located at the intersection of two curves. This type of node will be constrained to remain on the intersection of the two curves, and its velocity is simply set equal to the velocity of the intersection of the two curves.

In addition, a smoothing of the grid velocity is performed which consists in assigning to the internal nodes the mean velocity of their direct neighbors. This procedure is repeated for a few iterations which normally is also limited to some selected nodes located near the moving curves. The final result is a diffusion-like operator which smoothes out the large variations in grid velocity and that was found effective for the type of computations that were conducted.

The purpose of the smoothing term  $w_s$  is to produce an additional smoothing of the transient grid evolution by improving the grid quality by considering the node displacement. The new position of the grid nodes is obtained as the average of the position of their neighbors. The velocity of the grid nodes is then calculated by dividing the node translation by a time interval, which is chosen to be the non-dimensional time scale of the problem. When this action is applied on nodes located on a Neumann-type boundary curve, the resulting smoothing grid velocity  $w_s$  must be tangential to it. Consequently the normal component of  $w_s$  is dropped out and the nodes will slide on this curve.

### Grid Generation

The generation of stretched triangular grids will be performed using an incremental algorithm which uses local actions on the grid to obtain, from a given triangulation, a new triangulation with the required properties. The different local actions on the grid are driven by a definition of the quality of the triangles and the different procedure will have different roles towards the reaching of the objective. For the purpose of clarity, the quality will first be defined for an isotropic, or non-stretched, grid and then generalized to an arbitrarily stretched triangulation.

Let us assume that we have a list of nodes  $\mathbf{N}$  and assume that we also have an element list  $\mathbf{T}$  giving the connectivity of the triangulation. A triangular element is defined by three points  $r_1, r_2$

and  $\mathbf{r}_3$  in a counter-clockwise direction, while its side vectors are defined by  $\Delta\mathbf{r}_1$ ,  $\Delta\mathbf{r}_2$  and  $\Delta\mathbf{r}_3$ . We then define:

$$\begin{aligned} A &= \sqrt{3}\Delta\mathbf{r}_1 \times \Delta\mathbf{r}_2 \\ B &= \frac{1}{2} \sum_{i=0}^3 \Delta\mathbf{r}_i \cdot \Delta\mathbf{r}_i \end{aligned} \quad (1)$$

$A$  is proportional to the Jacobian of the triangle (twice its area) while  $B$  is the so-called potential energy of the triangle [2]. The dimensionless quantity

$$Q = \frac{A}{B} \quad (2)$$

varies from zero to one can be used as a measure of the equilaterality of the triangle.

### Definition of Stretching

The stretching of any triangle can be simply defined by considering its transformation into an equilateral triangle. Such a transformation is built from a rotation of the system of axis to a new axis  $(x', y')$  followed by a scaling by a factor  $1/E$  in the direction  $x'$ . This couple  $(E, \theta)$  can thus be used as the definition and measure of triangle stretching.

### Quality of Stretched Triangles

The quality of a stretched triangle can now be computed. First, one has to define an objective which, in the isotropic case, was implicitly an equilateral triangle. One thus needs a spatial distribution of stretching amplitude and orientation which is considered as data from the mesh generation point of view. The quality of stretched grids can now be measured: we first apply the transformation with the couple  $(E_o, \theta_o)$  to a triangle and compute the quality of the resulting triangle in the transformed plane as:

$$\bar{Q} = \frac{\bar{A}}{\bar{B}}$$

### Local Actions on the Triangulation

Several basic actions on the triangulation are performed to make the grid closer to the objective. A remeshing algorithm based on the successive application of these operators is described in ref. [3] for isotropic grids.

The refinement of the grid is obtained through triangle subdivision. A triangle requiring to be refined is branched into two triangles by cutting it on its longest side. The lengths of the side of the triangle are measured in the transformed plane.

The coarsening of the grid is performed through node removal followed by a local remeshing. The removal of a node in the triangulation leaves an open polygon. This polygon is then retriangulated by recursively removing from it the triangle with the highest quality, until only four nodes are left. The placement of the last diagonal is performed according to the algorithm of diagonal swapping, described below. This process is influenced by the stretching requirements by retriangulating the open polygon in the transformed plane.

The swapping of diagonal is a well known technique to obtain a Delaunay triangulation from an existing triangulation [4]. It consist in examining each pair of adjacent triangles and to select from the two possible configurations for the diagonal side, the one which maximize the minimum angle of the triangulation. This procedure is repeated until no more swapping occurs. Generalized Delaunay triangulation have been proposed which introduces some notion of space transformation [5, 6, 7, 8]. In the current implementation, each valid diagonal side is examined for swapping and the configuration which maximize the minimum quality is choosed, where the qualities are measured in the transformed plane.

A coarse-cure procedure has been implemented, which examines the triangulation and marks triangles with a small quality (below 0.4) as "bad" triangles. Then, for each bad triangle, the node opposed to the longest side( measure in the transformed plane) is deleted.

### **The Remeshing Algorithm**

The goal of the remeshing algorithm is to produce a triangulation meeting these required area and this, starting from the current triangulation and using the basic tools previously described. The proposed remeshing algorithm is described in ref. [9]. The first step determines which triangles requires refinement or coarsening and a corresponding code is attributed to each triangle. In practice, these actions are discrete operations on the grid and some care must be taken in setting the triangle code to avoid possible oscillations in the remeshing process, i.e. to insure a quasi-smooth grid convergence. To do so, one has first to evaluate the average performance of the two basic operators, the refinement and the coarsening. The refinement operator produces triangles of area half of their parent. The coarsening operator which, in the average, will operate on nodes surrounded by 6 triangles, produces new triangles areas of about 1.5 times the average parent area. From this basic data, the code on triangles have been set according to the following inequalities:

IF actual area  $> 3/2$  required area THEN set a refinement code  
IF actual area  $< 3/4$  required area THEN set a coarsening code

### **Geometric Requirements**

The computation of flows in complex geometries with moving boundaries must also take into account the geometric requirements. As discussed in ref.[10], these requirements are governed by two different aspects of the computational domain: the curvature of the bounding curves and the proximity of the various parts of the domain.

An automatic method for computing these requirements has been described in ref [10]. The method defines a reference grid density which respects th geometric requirements from both the curvature and proximity point of view.

## **Adaptivity and Flow Coupling**

### **Error Estimation**

The principle of estimating the error by projection of the solution in a higher order subspace has been used in the present work. Starting from an existing flow solution, which is piecewise constant in each triangle, a projection to a piecewise linear solution is performed using the technique of Barth [11]. The error in the solution in each triangle is then estimated to be the integration of the difference

between the linear and constant solutions. In addition, since the error is estimated to be proportional to the grid size (for the first order implementation of the Roe scheme), the required areas are obtained by scaling.

Some peculiarities of compressible flow solutions must nevertheless be taken into account when one tries to use directly this type of error estimation. In practice, and due to the intrinsic nature of compressible flow solutions, very high ratios of minimum and maximum required areas for triangles will be obtained. This results in extremely small triangles in regions of high gradients of the solution and these will reduce strongly the convergence of the computation. To overcome this problem, some limits on smaller and larger triangles in the computational domain must be imposed.

## Grid Control Strategy

We propose to start by devising a initial grid for the solution process, which will be called the "reference grid". The limits on the smallest and the largest triangle area are then specified in terms of a fraction of the reference grid, and are thus locally defined. For computations in geometries which need very high ratios of initial grid sizes, this approach is more flexible than the specification of absolute minimum and maximum sizes. Another advantage of this approach is the ability to deal with both the geometric and flow grid requirements during a transient solution process.

The frequency of remeshing is determined by two conditions: the geometric requirement and the flow requirement. A geometric remeshing is performed each time that the  $\Delta t_{\text{Grid}}$ , which is defined as the minimum time interval needed to reduce one of the triangle areas by one half [12], is reached. In this grid adaptation step, very few triangles are normally involved. A fluid remeshing is carried out after a certain number of iterations on the flow solution. The frequency of this action is determined by a user controlled variable, as are the minimum and maximum area ratio limits.

Details about the grid management algorithms can be found in refs. [3, 9, 13, 14, 10].

## FLOW SOLVER

The mathematical model describing an inviscid thermally nonconducting perfect gas is given by the Euler system, which can be written for a general moving (or non-moving) reference frame in integral form as:

$$\frac{\partial}{\partial t} \int_{V(t)} U dV + \oint_{S(t)} \mathbf{n} \cdot \mathbf{F} dS = \int_{V(t)} f dV \quad (3)$$

where  $U^T = [\rho, \rho \mathbf{u}, \rho E]$  is the vector of dependent variables, with  $\rho$ , the density,  $\mathbf{u}$ , the fluid velocity, and  $E$  the specific energy. The term  $\mathbf{F}^T = [\rho(\mathbf{u} - \mathbf{w}), \rho(\mathbf{u} - \mathbf{w})\mathbf{u} + \mathbf{I}p, \rho(\mathbf{u} - \mathbf{w})E + \mathbf{u}p]$  is the flux tensor, where  $\mathbf{w}$  is the mesh velocity,  $p$  is the pressure, and  $\mathbf{I}$  is the unit tensor. The variable  $\mathbf{n}$  indicates the outward unit vector normal to the boundary. The symbol  $f$  denotes external sources from the physics or from the axisymmetric formulation. In this case,  $f^T = [0, \mathbf{e}_y p/y, 0]$ . These conservation laws are completed by the equation of state  $p = (\gamma - 1)\rho E$ .

The associated discrete approach to the above integral equations is referred to as a Finite-Volume method. For the case of non-moving meshes, with no source terms, and using an explicit procedure, the variables  $U^{n+1}$  are updated by:

$$U^{n+1} = U^n - \frac{1}{V^n} \left( \sum F_i(Q_i) \right) \quad (4)$$

where  $F_i$  represents the discrete flux through a face  $B_i$  during a time interval  $\Delta t$  and  $V^n$ , the volume of the cell.

Among the different possibilities to obtain the flux vectors on the cell faces, the methods based on the solution of a set of local Riemann problems are used frequently. As their exact solution is costly, several approximate alternatives have been proposed, one of these being introduced by Roe [15].

### Roe's Scheme for Moving Grids

The ingenuity of Roe's procedure relies on the definition of an average state  $A^*$  which approximates the Jacobian  $A = \frac{\partial F}{\partial u}$  of the equation  $\frac{\partial U}{\partial t} = \frac{\partial F}{\partial x} = A \frac{\partial U}{\partial x}$ . This average state can be obtained on the basis of the quadratic character of the variables:  $\sqrt{\rho}$ ,  $\sqrt{\rho u}$ ,  $\sqrt{\rho v}$ ,  $\sqrt{\rho h}$ . Using this information, averaged right eigenvectors  $e_k$ , eigenvalues  $\lambda_k$ , and wave strengths  $\alpha_k$  can be obtained. Then it is possible to define the flux at a face, say,  $i+1/2$  as:

$$F_{i+\frac{1}{2}} = \frac{1}{2}[F_{i+1} + F_i - |\Delta F|_{i+\frac{1}{2}}] \quad \text{with} \quad |\Delta F|_{i+\frac{1}{2}} = \sum \alpha_k |\lambda_k| e_k \quad (5)$$

This method can be extended to moving grids in a simple manner. For example, for a grid node moving with a velocity  $w$ , the wave speed  $\lambda_1 = (u - a)$  (where  $a$  is the speed of sound), becomes:  $\lambda_1 = (u - a - w)$ . On the other hand the flux  $F(u)$  now transforms to  $F(u - w)$ . In this respect there are two fundamental remarks to be done. First, this modification *only affects the convective terms*. Second, the grid motion is characterized by the face velocity which is defined by:

$$w = \frac{\Delta V}{S \Delta t} \quad (6)$$

where  $S$  represents the face area at a given time, and  $\Delta V$  the volumetric increment along a face. Details of this fundamental approach are given in [12].

Applying these ideas, the updated variable  $U^{n+1}$  can be computed by:

$$U^{n+1} = \frac{V^n}{V^{n+1}} \left[ U^n - \frac{1}{V^n} \left( \sum F_i (Q_i - \Delta V_i) \right) \right] \quad (7)$$

It can be realized that the term in brackets corresponds to the advanced flow variable  $U^{n+1}$  computed after Eq. 4, with the term  $Q_i$  modified to  $Q_i - \Delta V_i$ . More details concerning the extended Roe's scheme for moving grids can be found in ref. [16].

## ADAPTIVE METHODS

### Grid Optimization

A spring smoothing scheme is derived by minimizing the function that represents the total potential energy of the triangulation given by:

$$\Phi = \sum_{\mathbf{T}} \kappa B \quad (8)$$

where  $\kappa$  is a penalty for the spring system. The Laplacian smoothing scheme is obtained by using  $\kappa = 1$ . However, the penalty can be accomplished differently for a better control of the grid. Such a penalty was introduced by Kennon and Anderson [2] to treat the case of non-convex domains and

$\kappa = 1/A$  was used. In the current work, the spring constants are choosed as a function of the qualities, such that:

$$\kappa = \phi(Q) \quad (9)$$

where  $\phi(Q)$  is a function that depends only on the quality of the triangle. Since the Eq. 8 is continuous with respect to the position of the nodes  $\mathbf{r}$ , a minimum of the function will be found when the gradient of Eq. 8 with respect to  $\mathbf{r}$  is zero.

### Optimization of Stretched Triangles

Following the previous discussions and definitions, a generalized form of the non-linear spring system given by Eq. 8 and 9 can be obtained by the minimization of the function:

$$\Phi = \sum_{\mathbf{T}} \phi(\overline{Q}) \overline{B} \quad (10)$$

### Minimization Methodology

The previously defined optimization problem is then solved using the gradient method of steepest descent. It is well known that this simple algorithm can converges very slowly but it is sufficient to test the formulation of the problem and the implementation of some more sophisticated optimization strategies are reported to a future work.

Details about the various adaptive strategies can be found in refs. [17, 18]

### Example

The optimization strategy is applied to the computation of a shock reflection problem. The initial grid and solution are reproduced on Fig. 1. The shock is diffused over two or three cells. Starting from this solution, requirements on grid stretching and orientations have been set according to the gradient of the density.

Figure 2 illustrate the final grid and solution. A comparison of the optimized and initial grid is presented on Fig. 3 where it beomes evident that this type of adaptation is a serious alternative to grid refinement.

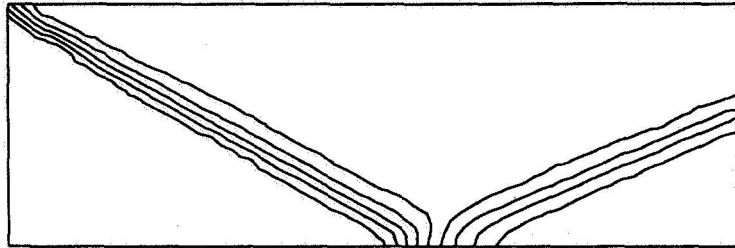
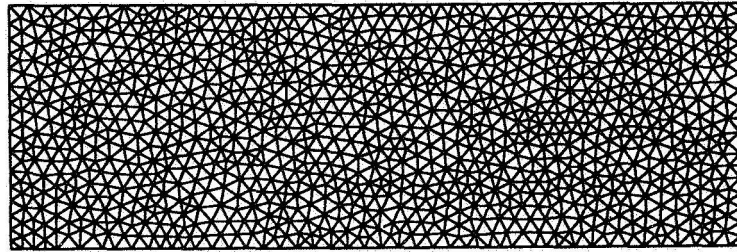


Figure 1: Initial grid and isoMach lines.

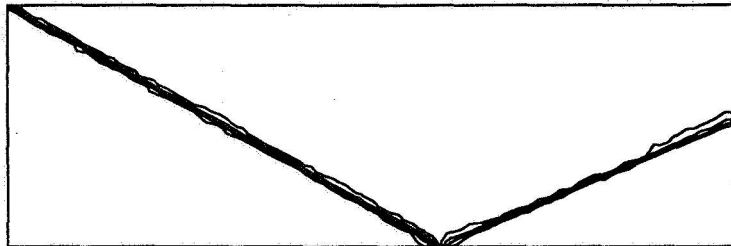
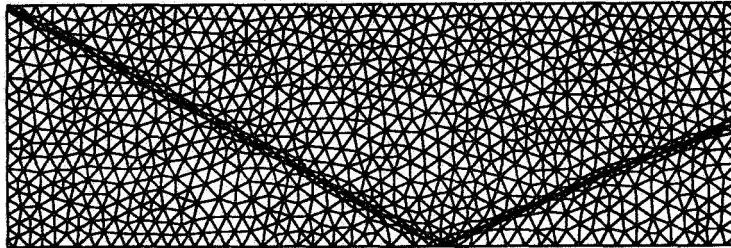


Figure 2: Optimized grid and isoMach lines.



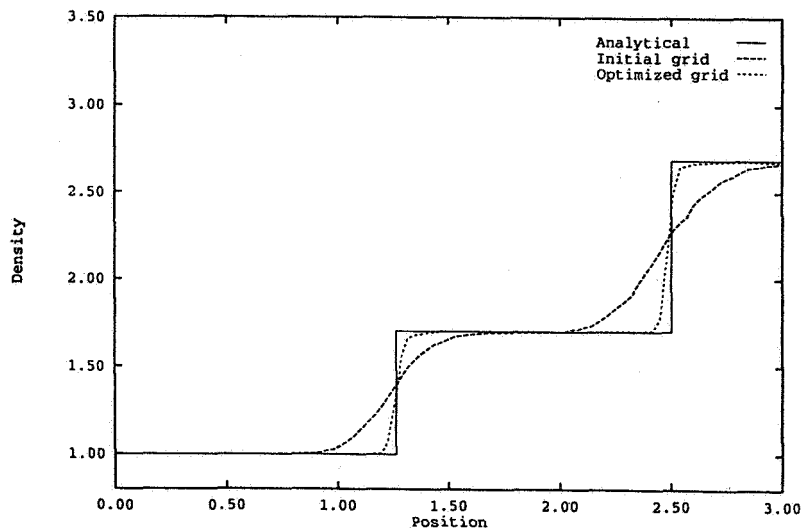


Figure 3: Density profiles for the initial and optimized grids.

### Shock Fitting

The adaptation to discontinuities such as shock waves has been a persistent problem in the numerical simulation of compressible flows. The weakness of the shock capturing approach is that the shocks are captured over several grid points. While the sharpness of these discontinuities can be improved by refining the grid in the vicinity of these regions, this leads to computing costs due the large number of elements and the decrease of the global time step particularly in 2 or 3D space dimension. The shock-fitting approach and recent variants such as the  $\lambda$ -scheme have fared better on the second count with some severe limitations on the topology of interaction patterns.

Resolving these problems requires addressing some fundamental and technological issues relating to the correct computation of shock discontinuities and their detection. Until recently it was felt that applying the Rankine-Hugoniot relations was the only way to achieve the exact jump conditions. With the Roe scheme, it is possible, although this is not generally appreciated, to obtain the exact jump provided the shock and the cell face are aligned. The problem of shock detection and tracking does not have rigorous foundations and is still largely based on heuristics. However a recent study [19] has proposed a model for the wave propagation phenomena. In this model three basic waves are identified and relations to compute the directions and strengths of these from the basic variables are given. It is possible, to extract information from the flow field this model to align locally the mesh. Coupled with a grid adaptivity algorithm, it is felt this model can produce the local grid alignment to allow the correct jump calculation and sharp shock resolution by the Roe scheme.

In the present section, a methodology is described to perform these tasks. The basic idea in the present method is to adapt dynamically the mesh to fit discontinuities in the flow. This involves

two fundamental capabilities: the first one is to detect accurately the various wave patterns of the flow; the second one is to perform the required actions on the grid to align it with discontinuities. The adjustment of the grid must be carried out without perturbing the solution because the method is to be applied to unsteady flows and because the convergence of the process will be improved if unphysical perturbations are avoided.

In the present work, feature detection is performed using the wave model proposed by Roe. This model has demonstrated its accuracy to capture oblique shock waves as well as contact discontinuities. A full description of this wave model is found in Ref. [19]. Application of this wave model requires the flow gradients. In the present finite-volume scheme, the flow properties are piecewise constants for the first order scheme and piecewise linear for the second order scheme and for both schemes are stored at the centers of the triangles. The flow gradients are computed at the triangle center using a standard Gauss quadrature involving all the triangles sharing a common node with the considered triangle.

### **Identification of the Flow Features**

The wave model used in this study is based on a superposition of linear waves and is not capable of representing genuinely non-linear waves and discontinuities. But this is not critical because it is not used for that purpose. What is required is the detection of a dominant wave and its angle. On the other hand, when the model is applied in regions of discontinuities such as shocks or slip lines, a correct physical behaviour will be captured by the model. More specifically, a shock wave will be seen as a strong acoustic wave, a slip line will be represented by a shear wave and a moving contact discontinuity as an entropy wave. This correspondance is at the basis of the detection algorithm.

The detection process involves the filtering of the waves which comprises two operations. First, the weak waves are discarded, based on the relative strength of each wave. The criteria for this step has been fixed at ten percent of the maximum wave intensity over the whole domain. Second, only one wave needs to be selected for each triangle. In this case, a wave is retained if it has a strength of an order of magnitude greater than the other waves in the same element.

After this process, most of the triangles will have their waves discarded, except triangles near discontinuities, dividing the whole triangulation in two groups: the active group comprising triangles with only one strong wave and the non-active group comprising triangles without a dominant wave.

### **Adaptation**

The grid management is a critical aspect of the algorithm. It is performed with three basic actions: i) orientation of some edges of the triangulation to align them perpendicular to the wave direction; ii) translation of the edges to follow moving discontinuities; and iii) removal of ill shaped triangles. In addition a grid adaptivity procedure can be superimposed on these algorithms.

*Orientation of the edges* The orientation of the edges is obtained from the output of the feature detection phase of the method based on the wave model. As described in section 3.2, this is a set of triangles for which a dominant wave has been identified. Only the orientation of this wave is used to modify the orientation of the grid.

The list of triangles in the active group is converted into a list of active edges. For each triangle, an edge is selected which is the most perpendicular to the main wave; and the edge becomes active only if it is selected by its two neighbor triangles. With this list of active edges, an attempt is made to orient these edges perpendicularly to the dominant. As conflicting requirements can result from different wave directions, this is carried out globally through an optimisation procedure. A

function that represents the vector product between the normalized wave orientation vector and the normalized side vector is constructed. The minimization is performed using a gradient method based on the steepest descent technique.

*Translation of the edges* The translation is performed to move the edges directly on the shock or slip lines. One is reminded that the third condition on Roe's average state matrix is another form of the Rankine- Hugoniot condition. This is obtained by adding at each node the velocity of the main wave in the direction detected by the wave model. For this action, the velocity of the main wave is taken from the flux eigenvalues provided by Roe's scheme which are more accurate than the wave speeds computed by the wave model, because of the Gauss quadrature required for the latter. For steady discontinuities the movement converges to an accurate positioning of the edges directly on the discontinuities. For unsteady flows, the velocity obtained at each node is the sum of two velocities, one of which follows the normal movement of the discontinuity and the other that rotates the edge about the discontinuity.

## Flow Over a Wedge

This first test case will be used to illustrate how the method works. It consists of a Mach 2 flow incident over a 10 degree wedge. The effect of the various actions involved in the process of grid adaptation will be investigated in a systematic way. The starting point of the adaptation process is the grid and solution represented on fig. 4. The oblique shock wave is captured by the scheme and extends over approximately two to three cells. In a first computation, the method was used without cure and adaptation. This means that the grid connectivity remains unchanged as the grid nodes move. After a few time steps, the grid motion and the optimization phase of the algorithm have almost succeeded in aligning the grid with the shock wave. This is illustrated on fig. 5 together with the current grid velocity, as computed by the algorithm.

After a few hundred time steps, some triangles tend to degenerate along the shock line, as shown in fig. 6. This is attributable to the translation grid velocities which attempts to bring grid lines from both side of the shock to the shock position directly. At this point, the algorithm almost stops due to the time step limitation given by the CFL criteria. The solution obtained is represented on fig. 6 in the form of a step function of the Mach number. It can be appreciated on this figure that even if the algorithm stops because of a degenerated triangle, the overall solution is improved compared to the initial solution. However, further improvements are straightforward if one now allows for some cure action of the grid.

In a second computation, grid cure was allowed while adaptation was still unused. The actions on the grid are thus limited to node removal and the result is that the number of grid nodes will be reduced as the grid is cured.

The results obtained with this procedure are shown on Fig. 7. One can see the ability of the method to align grid lines with the shock. However, as the grid becomes coarser, the ability of the wave model to correctly indicate the wave angle becomes problematic. It is thus suggested to complement the coarse cure method by a local remeshing including refinement.

The third result to be presented thus allows some kind of refinement of the grid. However, to simplify the analysis, the control of the remeshing is based purely on geometrical data, i.e. the flow has no influence on this refinement. The refinement criteria was the following: a triangle is refined if its area is 1.5 times the reference area value, which is the value of the triangle over of the initial grid at the same spatial location. The refinement is performed by dissecting the triangle along its longest side. This insures that the grid size distribution will remains almost identical to the initial grid. The

resulting grid and Mach number distribution are presented on Fig. 8. The shock is clearly identified on the grid itself and the Mach number distribution is sharply discontinuous.

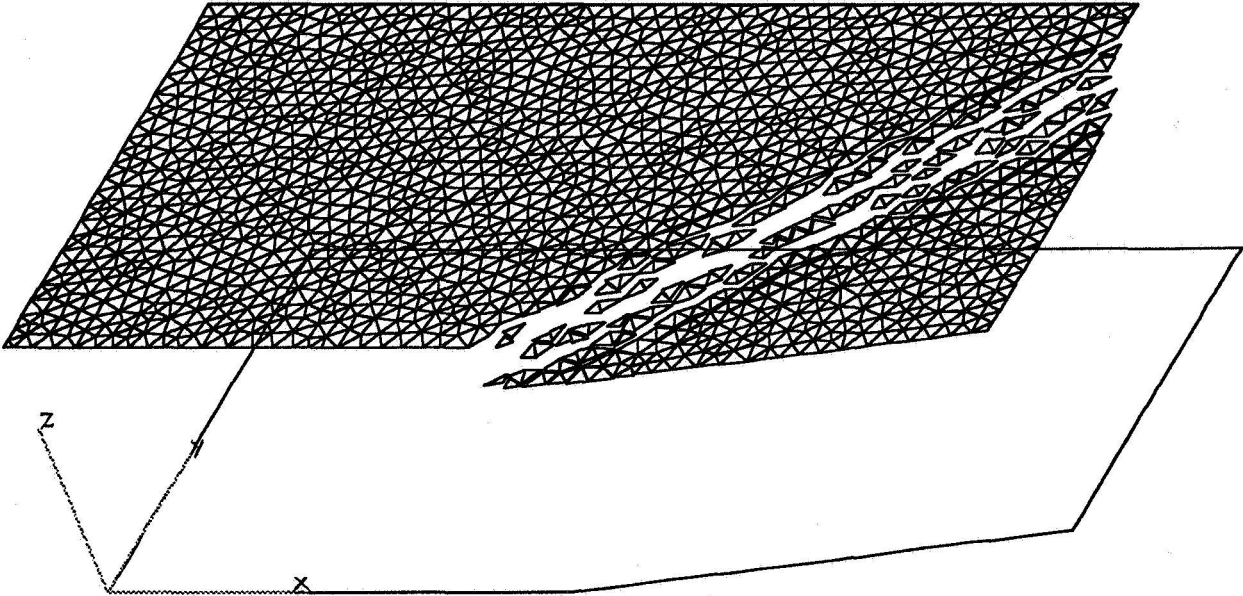


Figure 4: Initial grid and Mach graph.

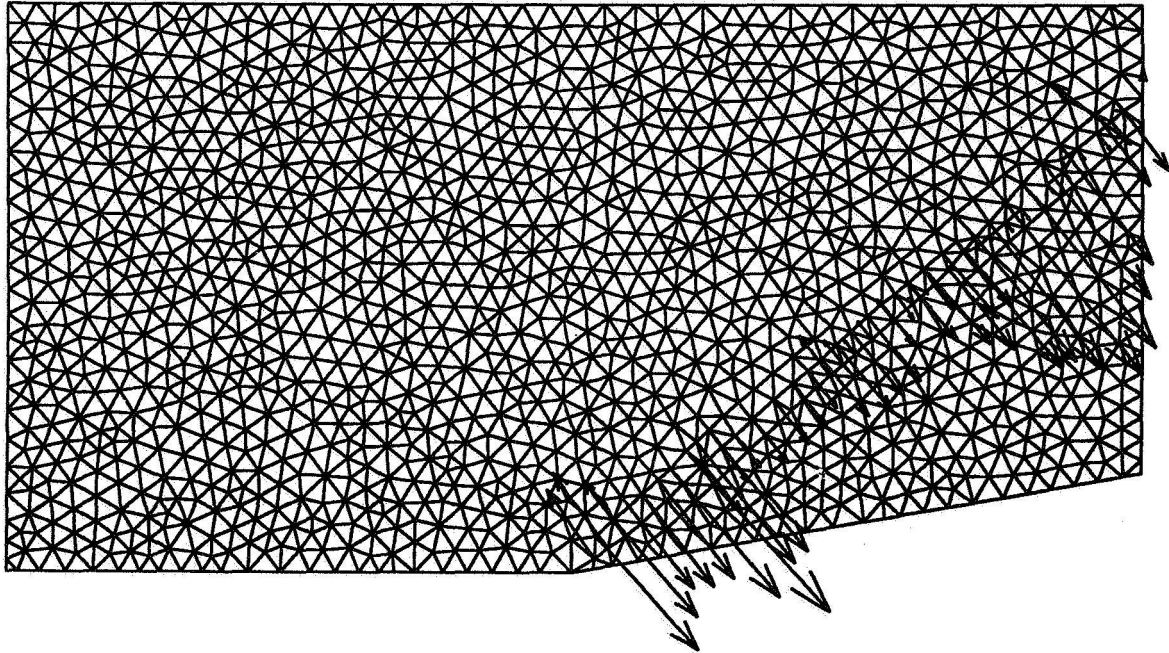


Figure 5: Grid and grid velocities after a few time steps.

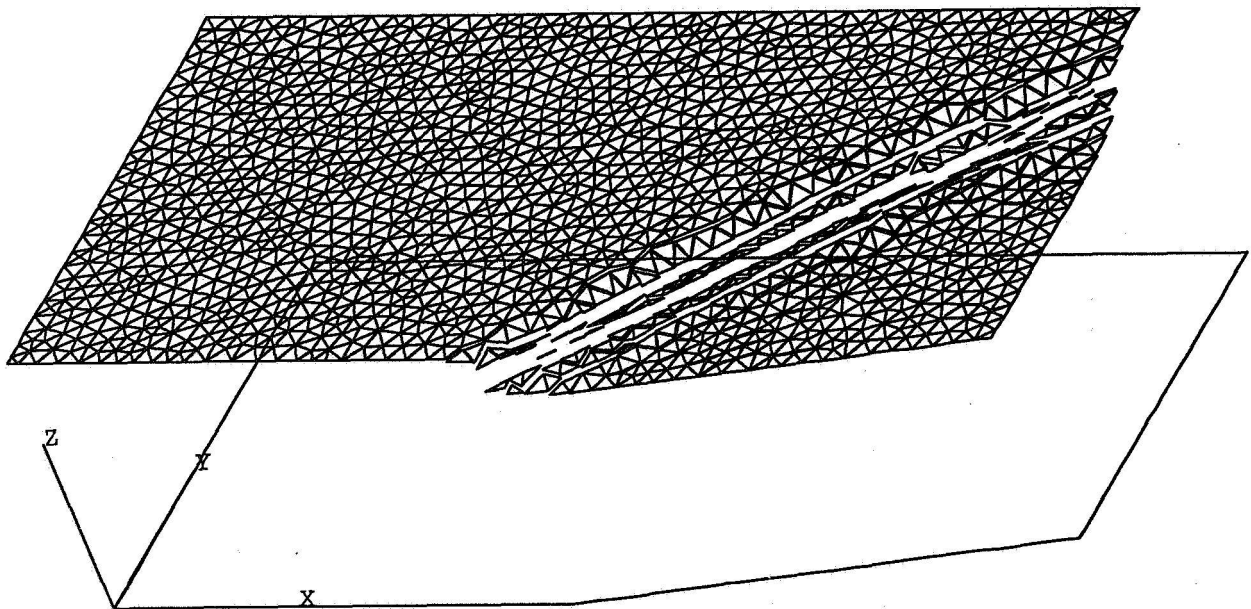


Figure 6: Grid and Mach graph without cure and adaptation.

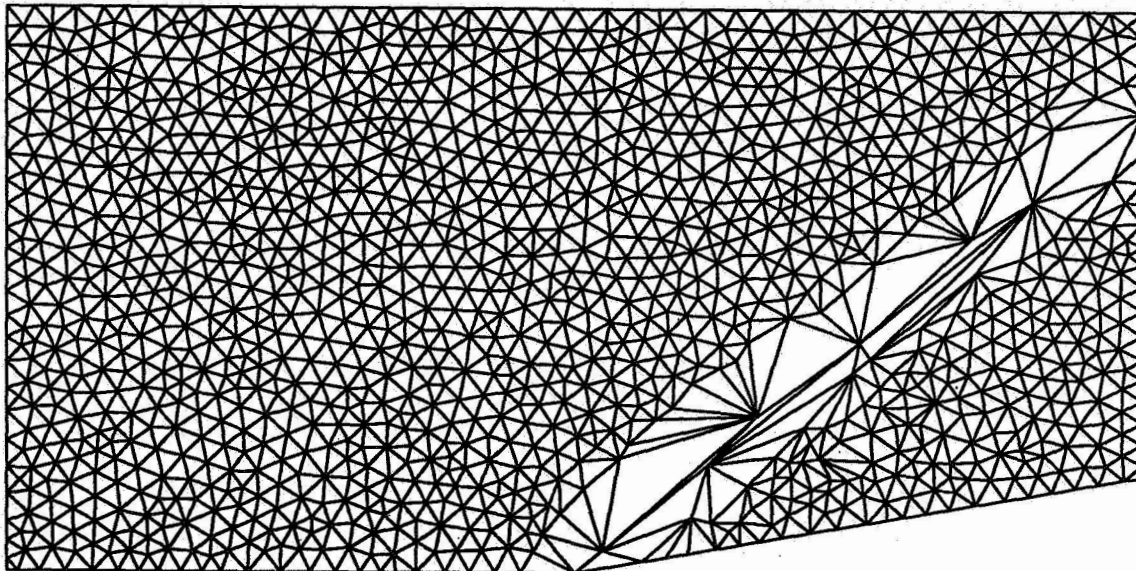


Figure 7: Grid obtained using the cure action.

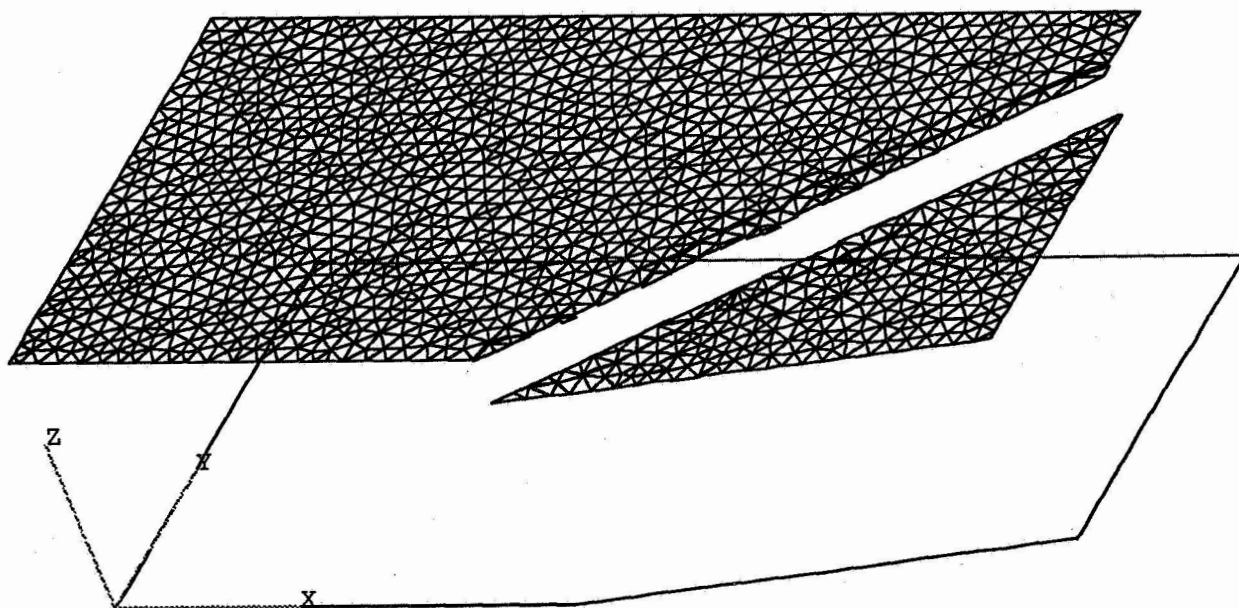


Figure 8: Grid and Mach graph with cure and adaptation.

## CONCLUSION

This paper has presented a set of methods for the computation of complex 2D compressible flows in domains with moving boundaries. It has been shown that the complete methodology provides a comprehensive tool for the solution various problems of engineering.

However, more work still need to be done on specific aspects to improve the accuracy and reliability of the method. More specifically, our future work will be concentrated on :

- develop a conservative interpolation algorithm for coarsening operations
- develop a more rigourous error estimator to drive the adaptation process
- quantify the performance of our adaptive methods

## ACKNOWLEDGMENTS

The authors thanks the National Science and Engineering Research Council (NSERC) of Canada for their financial support of this work through operational grants.

## REFERENCES

- <sup>1</sup> X.D. Zhang, J.Y. Trépanier, and R. Camarero. Modelling and computation of arc-flow interaction in circuit-breakers. *International Journal of Computational Fluid Dynamics*, 2:41–64, 1994.
- <sup>2</sup> S.R. Kennon and D.A. Anderson. Unstructured grid adaptation for non-convex domains. In S. Sengupta et al., editors, *The second Int. Conf. on Num. Grid generation in CFD*. Pineridge Press Limited, 1988.
- <sup>3</sup> J.Y. Trépanier, M. Reggio, M. Paraschivoiu, and R. Camarero. Unsteady Euler solutions for arbitrarily moving bodies and boundaries. In *The AIAA 30<sup>th</sup> Aerospace Science meeting*, 1992. AIAA Paper 92-0051, January 6-9, Reno, USA.
- <sup>4</sup> C.L. Lawson. Software for c1 interpolation. In *Mathematical software III*, pages 166–194. Academic Press, 1977.
- <sup>5</sup> D.J. Mavriplis. Adaptive mesh generation for viscous flows using Delaunay triangulation. *Journal of Computational Physics*, 90:271–291, 1990.
- <sup>6</sup> M.G. Vallet. Generation de maillages anisotropes adaptés, application à la capture de couches limites. Technical report, INRIA, Rocquencourt, 1990.
- <sup>7</sup> E.F. D’azevedo and R.B. Simpson. On optimal interpolation triangle incidences. *Siam J. Sci. Stat. Comput.*, 10(6):1063–1075, 1989.
- <sup>8</sup> T.J. Barth. On unstructured grids and solvers. In *Computational fluid dynamics*. VKI Lectures Series 1990-03, 1990.

- <sup>9</sup> J.Y. Trépanier and H. Yang. Algorithms for adaptive discretization based on triangular grids. Technical report, Ecole Polytechnique de Montréal, 1992.
- <sup>10</sup> J.Y. Trépanier, M. Reggio, and R. Camarero. Automated geometric-based mesh requirement for adaptive flow computations. In *The AIAA 31<sup>th</sup> Aerospace Science Meeting*, 1993. AIAA Paper 93-0674, January 11-14, Reno, USA.
- <sup>11</sup> T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *AIAA Paper 89-0366*, 1989.
- <sup>12</sup> H. Zhang, M. Reggio, J.Y. Trépanier, and R. Camarero. Discrete form of the GCL for moving meshes and its implementation in CFD schemes. *Computers and Fluids*, 22(1):9–23, 1993.
- <sup>13</sup> J.Y. Trépanier, H. Zhang, M. Reggio, and R. Camarero. Adaptive and moving meshes for the computation of unsteady compressible flows. In A.S. Arcilla et al., editors, *The Third International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Barcelona, Spain, June 3-7 1991. North-Holland.
- <sup>14</sup> Y. Lauzé, R. Camarero, and H. Yang. Interactive generation of structured/unstructured surface meshes with adaptivity. In A.S. Arcilla et al., editors, *The third Int. Conf. on Num. Grid generation in CFD and related fields*, Barcelona, Spain, June 3-7 1991. North-Holland.
- <sup>15</sup> P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, October 1981.
- <sup>16</sup> J.Y. Trépanier, M. Reggio, H. Zhang, and R. Camarero. A finite volume method for solving the Euler equations on arbitrary Lagrangian-Eulerian grids. *Computers and Fluids*, 20(4):399–409, 1991.
- <sup>17</sup> J.Y. Trépanier, M. Reggio, and R. Camarero. Adaptation and optimization of triangular meshes for inviscid and viscous flow computations. In *7<sup>th</sup> IMACS International Conference on Computer Methods for Partial Differential Equations*, 1992. June 22-24, New-Brunswick, NJ, USA.
- <sup>18</sup> H. Zhang and J.-Y. Trépanier. An algorithm for the optimization of directionally stretched triangulations. *International Journal for numerical methods in engineering*, 37:1481–1497, 1994.
- <sup>19</sup> P.L. Roe. Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics. *Journal of Computational Physics*, 63:458–476, 1986.