INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **39**:675–702 (DOI: 10.1002/fld.357)

Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part III. Unstructured meshes

Julien Dompierre[1], Marie-Gabrielle Vallet[1], Yves Bourgault[2], Michel Fortin[3] and Wagdi G. Habashi[4,*,†]

[1] *Centre de Recherche en Calcul Appliqué, 5160, boul. Décarie, Montréal Que., Canada H3X 2H9*
[2] *Department of Mathematics and Statistics, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5*
[3] *Groupe Interdisciplinaire de Recherche en Éléments Finis, Université Laval, Québec, Canada G1K 7P4*
[4] *CFD Lab, Department of Mechanical Engineering, McGill University, 688 Sherbrooke West, Suite 715 Montreal, Que., Canada H3A 2S6*

SUMMARY

The present paper is the third article in a three-part series on anisotropic mesh adaptation and its application to two- and three-dimensional, structured and unstructured meshes. This third paper concerns the application of the full adaptation methodology to 2-D unstructured meshes, including all four mesh modification strategies presented in Part I, i.e. refinement/coarsening, edge swapping and node movement. The mesh adaptation procedure is validated through a careful monitoring of a single adaptation step and of the solution–adaptation loop. Independence from the initial mesh and from the flow solver is illustrated. The efficiency of the overall methodology is investigated on relevant laminar and turbulent flow benchmarks. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: edge-based error estimator; anisotropic mesh adaptation; unstructured meshes; coupled solution–adaptation

1. INTRODUCTION

The general principles of a novel and cost-effective anisotropic mesh adaptation methodology have been laid out in Part 1 [1] of the present four-part series on anisotropic mesh adaptation. The three principal components of the anisotropic adaptation procedure, namely the error

* Correspondence to: W. G. Habashi, CFD Lab, Department of Mechanical Engineering, McGill University, 688 Sherbrooke Street West, 7th Floor, Montreal, Que., Canada H3A 2S6.
† E-mail: wagdi.habashi@mcgill.ca

Contract/grant sponsor: NSERC.
Contract/grant sponsor: FCAR.
Contract/grant sponsor: Centre de Recherches Mathématiques de l'Université de Montréal (CRM).

Received 2 October 2001
Revised 24 January 2002

estimate, the mesh operations and the adaptation criterion, are based on element edges. This permits to reorient the elements and prescribe a large degree of anisotropy along directions with mild variation versus those with strong variation.

Part II [2] of the series addressed structured grids, for which only node movement was allowed. A thorough validation of the mesh movement algorithm was undertaken. In particular, it has been shown that the algorithm is reversible. This means that, starting from highly distorted triangular and rectangular grids defined on triangular or rectangular domains, the algorithm is able to recover a perfectly uniform mesh if a uniform second derivative is specified. Moreover, in Part II, numerical tests on standard benchmarks for compressible flows showed the cost-effectiveness of the present anisotropic adaptation approach.

The adaptation of structured grids Part II [2] is restricted to only mesh movement, severely restricting its potential. On the other hand, while the greater flexibility of unstructured meshes has been trumpeted, this has usually been tempered by an accompanying loss of accuracy. In Part III, the full power of the proposed adaptation method will be unleashed on unstructured meshes, with all four mesh modification strategies combined and driven by the same error estimate. It will be shown that not only does mesh adaptation palliate the deficiencies of unstructured meshes but, surprisingly, the final adapted mesh and solution turn out to be the same, independent of the starting mesh. An equally important surprise is that the mesh, and the solution, at least in the tested runs, turn out to be independent of the solution algorithm used. This demonstrates heuristically that many of the advantages of higher accuracy schemes are noticeable mostly when meshes are inappropriate, but that adapted meshes are much less impervious to showcasing such advantages. In other words, on a properly adapted grid, low and high order schemes should give equally good solutions, that is the physically relevant solution of the PDEs.

In Section 2 of the paper, the mesh adaptation procedure is validated through a careful monitoring of a single adaptation step of the inner loop and of the coupled solution–adaptation external loop. In this section, the independence from the initial mesh and from the flow solver is also illustrated. In Section 3, the efficiency of the overall methodology is investigated on relevant laminar and turbulent flow benchmarks. In Section 4, the floating-point precision is shown to be the main limiting factor for two-dimensional (2-D) mesh adaptation.

2. CONVERGENCE AND UNIQUENESS OF ADAPTED MESHES

For completeness, we recall the error estimator used in our mesh adaptation scheme. The error $e(\boldsymbol{x}_I - \boldsymbol{x}_J)$ on the edge connecting the node $\boldsymbol{x}_I$ and $\boldsymbol{x}_J$ is given by

$$e(\boldsymbol{x}_I - \boldsymbol{x}_J) = \int_0^1 \sqrt{(\boldsymbol{x}_I - \boldsymbol{x}_J)^{\mathrm{T}} \boldsymbol{M}(l)(\boldsymbol{x}_I - \boldsymbol{x}_J)}\, \mathrm{d}l \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{x})$ is the absolute value of the Hessian matrix $\boldsymbol{H}(\boldsymbol{x})$ of the solution, in practice an approximation of this matrix obtained through a post-processing of the numerical solution u_h. Details are presented in Part I of this series [1].

Moreover, we briefly repeat the adaptation algorithm:

1. Smooth the mesh after estimating the error by alternatively:
   (a) swapping all the edges until convergence
   (b) moving all the nodes iteratively.
2. Adapt the mesh by iterating over the following loop:
   (a) refine all edges above a threshold error estimate,
   (b) swap the edges until convergence, then apply node movement,
   (c) remove all nodes whose edges have an error estimate below a threshold value,
   (d) swap the edges until convergence, then apply node movement.
3. Finally, smooth the mesh by repeating loop 1 before solving the equations again, starting from an interpolated solution.

Given an error field over a given mesh, the algorithm adapts the mesh so that the size and the orientation of the elements meet the specified error level as uniformly as possible. While all local modification techniques tend to equi-distribute the edge error, they do it in slightly different ways. For instance, threshold values for refinement and coarsening have slightly different impact, and hence have to be set so that both methods collaborate instead of competing, with one local modification technique eventually destroying the improvement brought by the other. Typical threshold values for coarsening and refinement are, respectively, 0.6 and 1.4 of the average edge error estimator over the mesh. For example, an edge is refined if the value of the error estimator on that edge is more than 1.4 times the average edge error estimator over the mesh. The closer to 1 these threshold values are, the more uniformly distributed will be the error estimator, the better will be the mesh but, of course, the larger will be the number of global adaptation loops needed to adapt the mesh.

To build confidence in this method, one must check the convergence of the adaptation algorithm itself: i.e. with a prescribed error field or, in other words, for a single step of the solution–adaptation process, the last iterations of the loop 1–3 presented above must produce negligible changes to the mesh. The next validation step is to show that the coupled solution–adaptation process is convergent. At every flow solution–mesh adaptation cycle, the mesh and the error field are updated, building a sequence of meshes and solutions. Of course, it must be verified that the whole iteration process should reach a converged state. Moreover, this converged state, if achievable, must be dictated only by the physics of the flow, not by the particular flow solver or the initial mesh at hand. As with any iterative process, the independence from the initial data, in our case the initial mesh, must be achieved. These are all important questions that need to be addressed by any practical mesh adaptation method.

In the following, it will be demonstrated that the mesh adaptation converges at each single step of an iteration, i.e. when the solution is frozen and adaptation is used only as a post-processor, and also converges when it is coupled to the solver during iteration. We will also demonstrate the independence of the adapted mesh from the solver used and also its independence from the initial mesh.

2.1. Convergence of an adaption cycle, with solution frozen

We first investigate the efficiency of the adaptation loop 1–3 for a single step of the solution–adaptation process. On a given initial mesh, a flow solution is computed once and this ‘frozen’

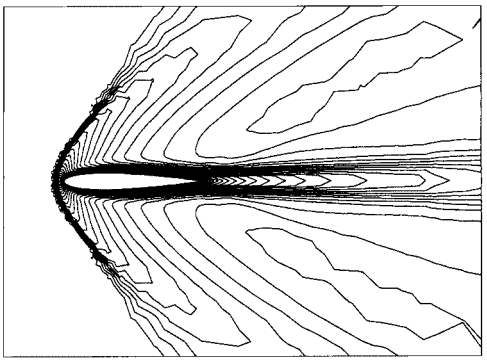

Figure 1. The error estimator is derived from second derivatives of the Mach number field shown here.

solution is used to calculate the matrix $\boldsymbol{M}$ in Equation (1) on the initial or background mesh. Starting from the initial mesh, a new mesh is iteratively generated with the loop 1–3, computing the error estimator on the edges of the adapted mesh using Equation (1), but with the matrix $\boldsymbol{M}$ reinterpolated from the background mesh. As the matrix $\boldsymbol{M}$ is frozen on the background mesh, the error field is prescribed and the loop 1–3 should involve less and less work as iterations proceed and the variance of the edge error estimator over the adapted mesh decreases.

A test case is presented for which a mesh is adapted to a viscous laminar flow around a NACA 0012 profile with a freestream Mach number of 2.0 and a Reynolds number of 10 000 (Figure 1). The Navier–Stokes equations for compressible flows are solved, namely

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot \tau = 0$$

$$\rho \frac{\partial e}{\partial t} + \rho \mathbf{u} \cdot \nabla e + p \nabla \cdot \mathbf{u} - \nabla \mathbf{u} : \tau - \nabla \cdot (\kappa \nabla e) = 0$$

where ρ, $\mathbf{u}$, p and e stand for the density, velocity, pressure and internal energy of the fluid, respectively. The variable τ is the usual viscous stress tensor for a compressible fluid (with the number Re^{-1} hidden in) and κ is the renormalized heat diffusion coefficient. No-slip and adiabatic wall conditions are used on the body.

This test case, and laminar flow in particular, are chosen to avoid any idiosyncrasy of turbulence modelling implementation. The adapted mesh has no resemblance to the initial one (Figure 2), despite the fact that it is deduced from it using successive local alterations. The number of alterations is represented in Table I at each iteration of the overall adaptation loop and indicates that changes become negligible after only 5 iterations.

With the convergence demonstrated, the next step is to characterize the converged state. The statistical graph of Figure 3 indicates that the edge error estimate concentrates around the target error, with a more even distribution, as adaptation proceeds. In particular, the ratio

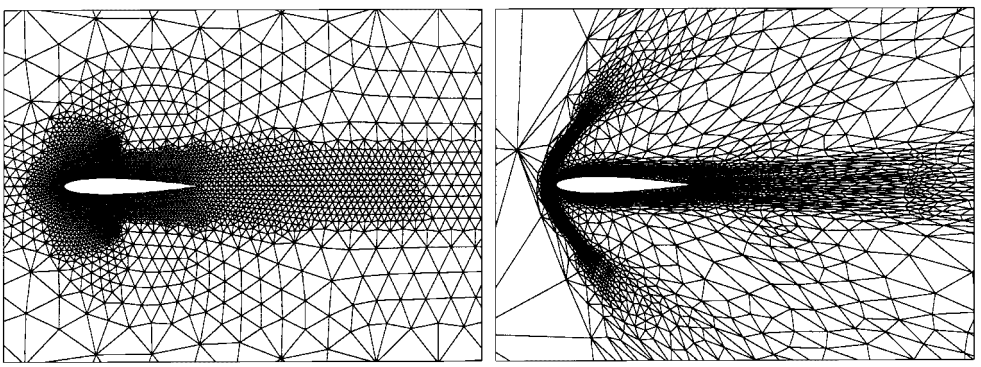

Figure 2. Initial mesh (also used as the background mesh to support the error estimator) and converged adapted mesh, with solution frozen.

Table I. Convergence of local improvements during one mesh adaptation.

Refinement		Coarsening		Swapping	
Edges	%	Nodes	%	Edges	%
1671	15.32	2225	41.54	2697	24.73
721	7.77	290	7.53	102	1.10
62	0.59	69	1.90	45	0.43
25	0.24	25	0.70	17	0.16
19	0.18	14	0.39	14	0.13
9	0.09	13	0.36	11	0.10
10	0.10	10	0.28	8	0.08
8	0.08	8	0.22	10	0.10
12	0.11	10	0.28	2	0.02
7	0.07	5	0.14	4	0.04

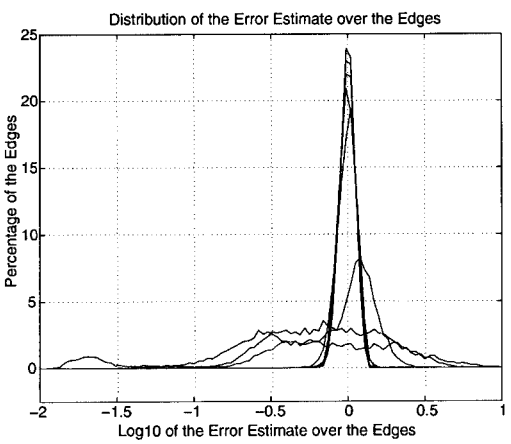

Figure 3. Distribution of the edge error over the mesh at different iterations of the mesh adaptation process.

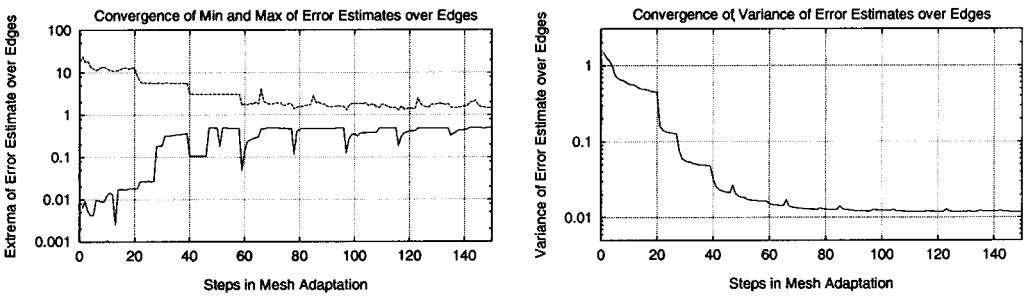

Figure 4. Evolution of the extremum values and the variance of the edge error during one cycle of mesh adaptation.

between the maximum and minimum values decreases from 5000 to 3 (Figure 4, left) and the standard deviation is reduced by two orders of magnitude (Figure 4, right).

Even if the convergence indicators are not absolutely monotonic (refinement results in a decrease of the minimum error, and coarsening in an increase of the maximum) the algorithm of Section 2 converges towards a mesh that can be considered optimal for the flow solution at hand.

2.2. Convergence of the coupled solution–adaptation cycle

Now that we have build some confidence in the adaptation method of loop 1–3, we next investigate the convergence of the coupled solution–adaptation process. At each solution–adaptation cycle, a background mesh and its initial solution are provided. The flow solver first computes the best flow solution on that background mesh, for example by reducing the residual of the discrete Navier–Stokes equations. This new solution and the background mesh are sent to the mesher, namely loop 1–3, that generates a new adapted mesh and the flow solution reinterpolated on that mesh. These constitute the output of the solution–adaptation cycle and the input of the next cycle. The whole process can be initiated with the mesh obtained through a mesh generator and an unphysical constant flow solution. In this section, we monitor the solution–adaptation cycles for a typical test case, being understood that this test case is representative of all the experiments done so far.

The topology of the meshes changes in a discrete and radical way. As a consequence, apart from a close monitoring of the coupling strategy, no common metric or norm can be used to quantify the convergence of the meshes. To assess the coupling strategy, we will consider the convergence of the number of nodes on successive adapted meshes as an indication of the convergence of the coupled problem. Since the number of nodes of an adapted mesh is controlled by the target error, if the number of nodes stabilizes, it is an indication that the coupling strategy converges to a mesh with a prescribed global error on the solution.

The example of Section 2.1 above is again used to illustrate this point. Figure 5 shows the number of nodes of the successive meshes as the solution–adaptation loop proceeds. An indication of convergence is the leveling-off of the number of nodes after a certain number of remeshing steps. The meshes at steps A, B, C, and D marked in Figure 5 are presented in Figure 6.

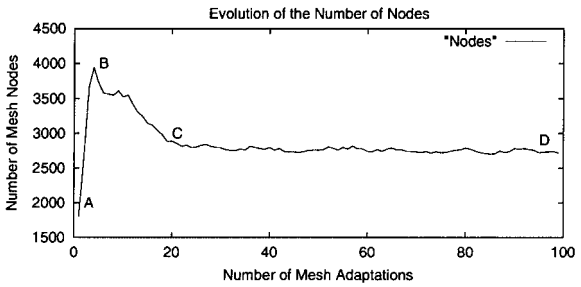

Figure 5. Total number of nodes versus adaptation cycles for the flow over a NACA 0012 at $Ma = 2.0$ and $Re = 10\,000$.

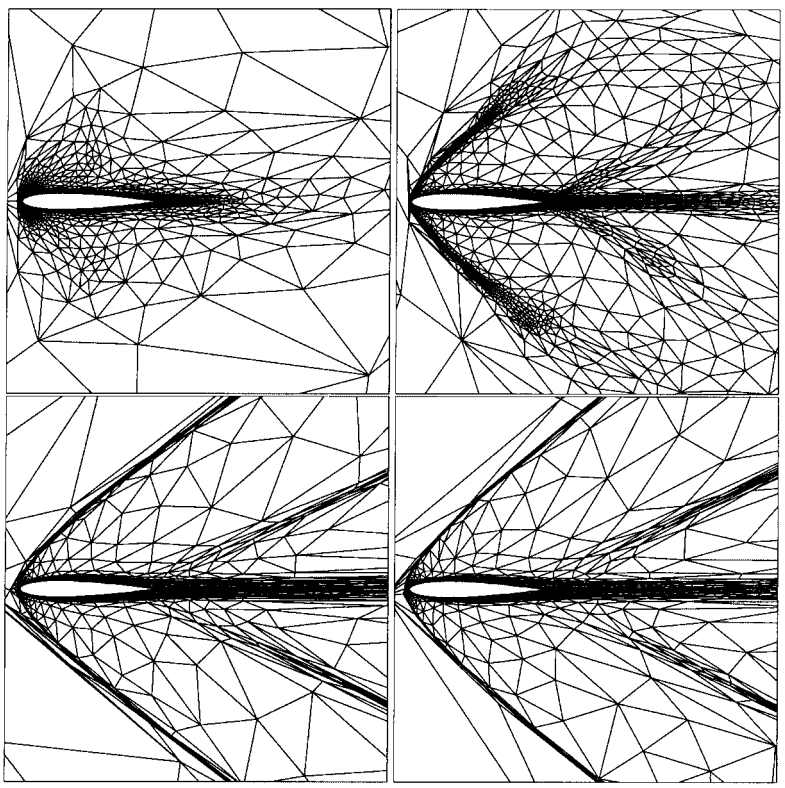

Figure 6. Adapted meshes after 1, 4, 20 and 100 iterations (left-right, top-bottom) of the coupled solution–adaptation loop.

Initially, the number of nodes increases, followed by a gradual decrease to the asymptotic value. Since the few first meshes are not well adapted, the solution is polluted with spurious oscillations. An over-refinement of the mesh results in order to define the salient features of the flow. When the solution improves, the adaptation gradually reduces the number of nodes, bringing to bear all the techniques presented above.

2.3. Independence of adapted solution from initial mesh

It is desirable to verify whether the mesh adaptation is unique. The question asked is whether for flow over or in a given geometry, there is a unique mesh that should be used for each freestream condition? In addition, whether the current mesh adaptation can yield that mesh? Intuitively, the answer is yes, as the scheme has been shown to converge for a single adaptation step as a post-processor and also when tightly coupled in a solution–adaptation loop.

To demonstrate this uniqueness property conclusively, the same problem is solved on vastly different meshes. The flow conditions are similar to those of the previous section. In Figure 7, a common mesh is used, with refinement around the airfoil, embedded in a much coarser mesh away from the airfoil. The figure also shows a very coarse mesh in which only 18 points are used on the airfoil. Finally, it shows an intentionally counter-intuitive mesh, with thousands of nodes in the upper half and only three points on the body in the lower half, although the problem is symmetric. Each of the three figures also shows the corresponding results, using FVM.

It is interesting to follow in Figures 8–10, the evolution of adapted meshes and solutions started from the three meshes. The figures indicate a nearly similar final mesh and an identical final solution. It is even more interesting to view in Figure 11, the progressive improvement of the three initial solutions towards a very sharp and crisp final solution. In Figure 12 the Mach contours are compared: they fall on top of one another. Figure 13 shows the initial and final distributions of the friction coefficient. The left side shows that the initial distributions are quite different, and all incorrect. The right side of the figure shows how the three superposed results are indistinguishable. The percentage of edges having a given error (log 10 of error) is represented in Figure 14. The initial three meshes have a very large error band. After adaptation, the three solutions give the same Gaussian error distribution whose maximum error has been reduced an order of magnitude from the initial one.

This example convincingly demonstrates that there is reason to hope that mesh-independent flow solutions can be obtained in a reasonable time and without considerable effort, by starting from a coarse and arbitrary mesh and let the adaptation procedure ‘generate’ the correct mesh. This may also then lead to user-independent results as the meshing decisions are taken away from the user, who in the first place has no way of guessing, let alone knowing, what is the most appropriate mesh for a given geometry, at various flow conditions. It will also mean that results will be reproducible, as a user wanting to duplicate some previous results has only to specify the error level desired, since the mesh corresponding to it is unique and will be obtained as part of the solution. A rational basis of comparison and benchmarking between commercial codes can also ensue, as the methodology allows one to compare them on equal footing, ridding them of any proprietary bells and whistles that may have been used to make them converge, often to only a qualitative answer.

2.4. Independence of final adapted solution from solver

In this section, the current solution–adaptation strategy is shown to lead to independence of the final result from the flow solver used, i.e. *different solvers give* (*almost*) *identical final results.* One solver may be more efficient in terms of precision and computing time, but the final meshes statistically have the same node distribution and nearly the same number of nodes. As a result, the final adapted solutions obtained with the different solvers are identical, as demonstrated here for laminar flows with low to moderate Reynolds numbers.

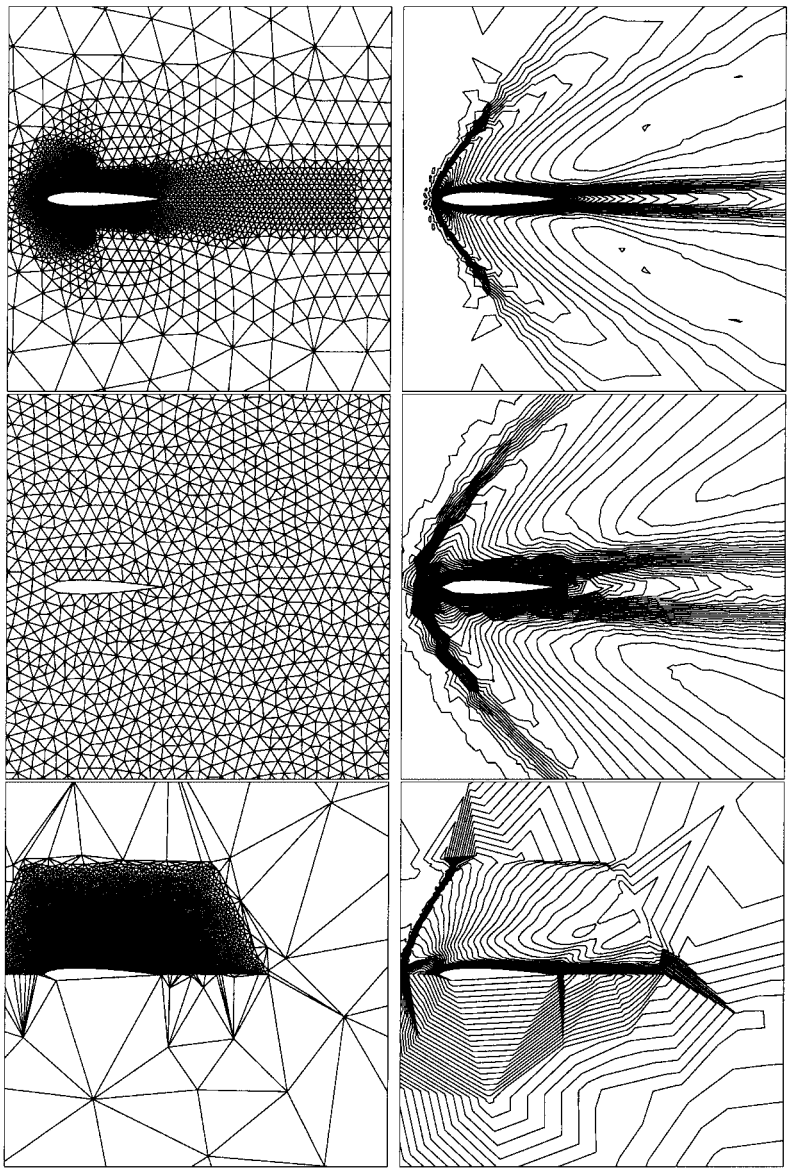

Figure 7. Solution on a fine mesh (top), a coarse mesh (middle) and an arbitrary, counter-intuitive mesh (bottom).

The test problem, solved by FEM and FVM codes, is compared. The two codes are used as black boxes, with no modification to their source. The finite element code is that of Boivin [3]: it solves the non-conservative form of the Navier–Stokes equations, for the primitive variables ρ, u, v and T. It uses mixed triangular elements (P1/P1-iso-P2) with a 'nearly' quadratic interpolation of velocities, i.e. with velocity expressed linearly over triangles but such that

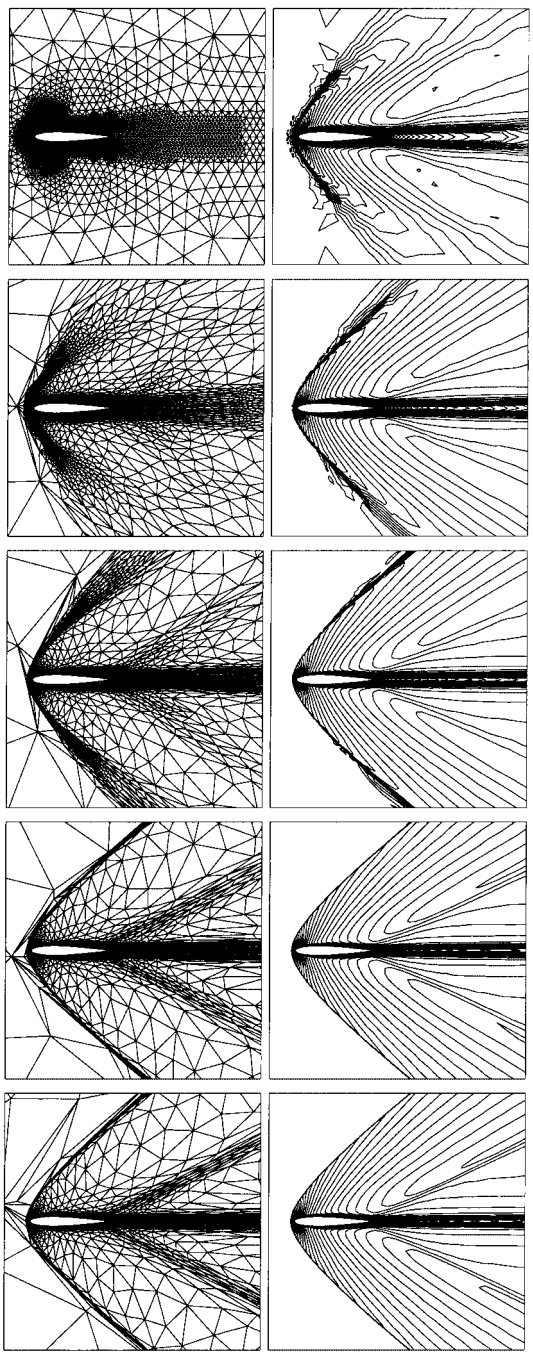

Figure 8. Meshes and solutions at adaptation steps 0, 1, 2, 5 and 10, starting from the fine mesh.

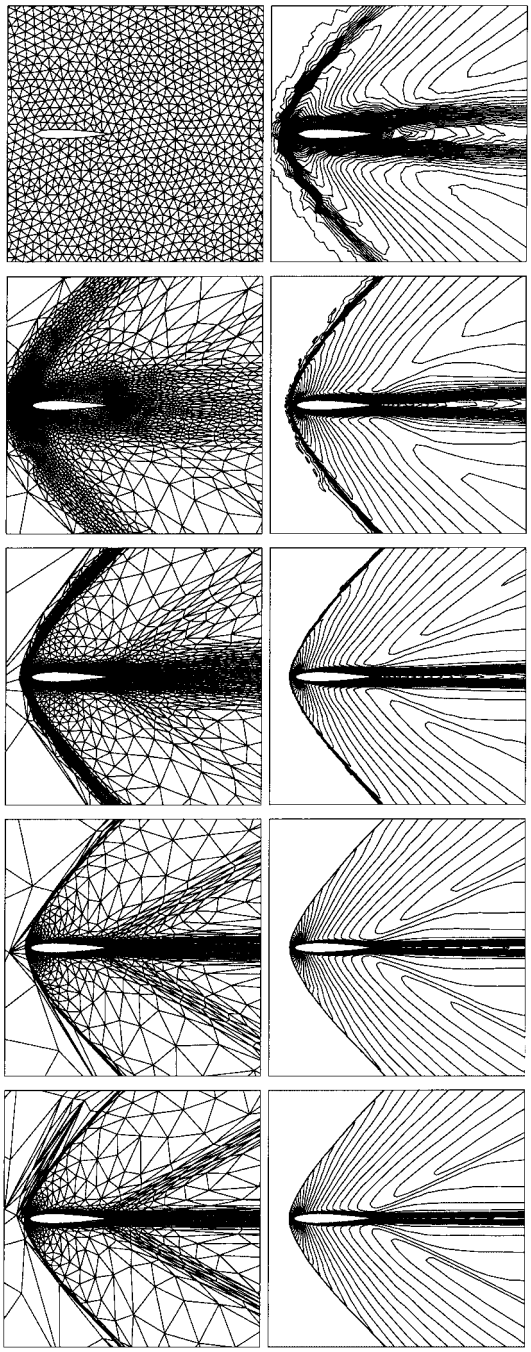

Figure 9. Meshes and solutions at adaptation steps 0, 1, 2, 5 and 10, starting from the coarse mesh.

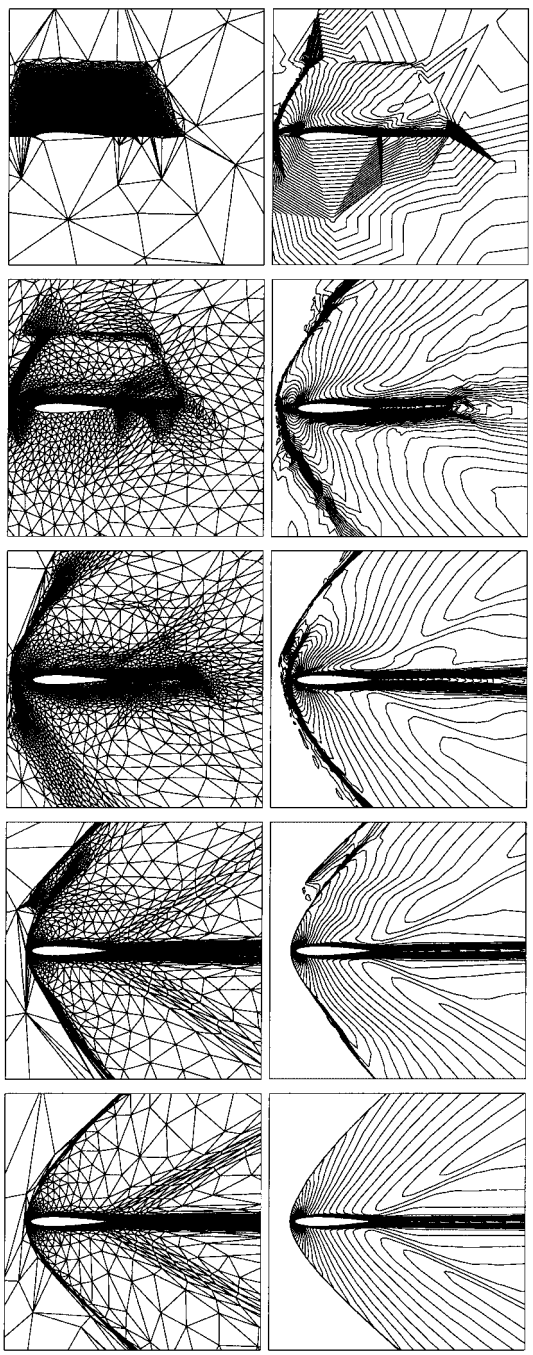

Figure 10. Meshes and solutions at adaptation steps 0, 1, 2, 5 and 10, starting from the counter-intuitive mesh.

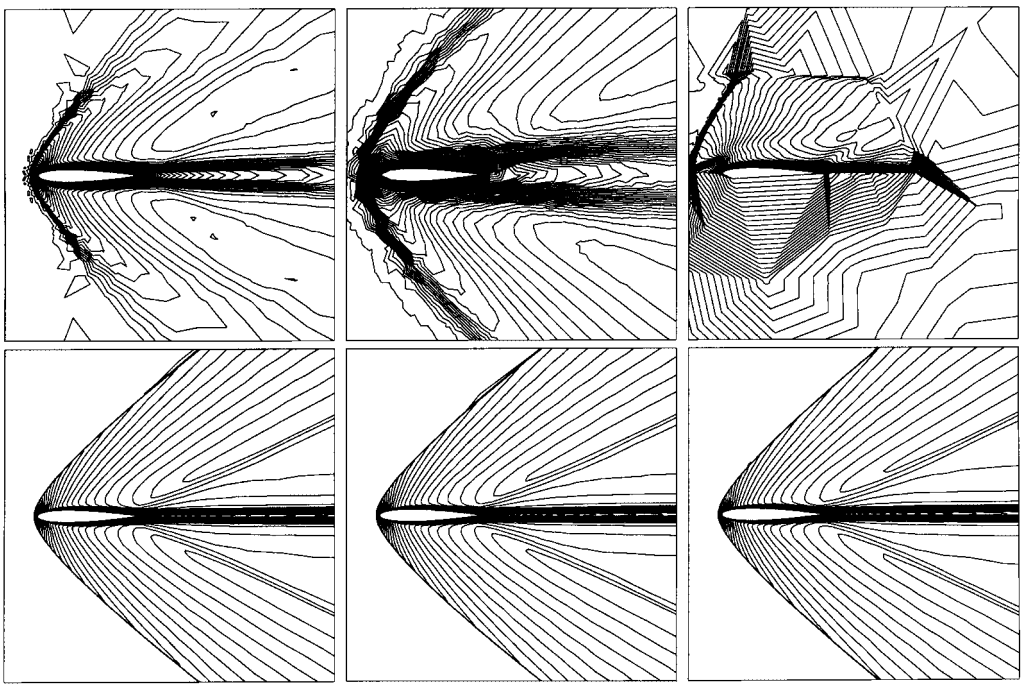

Figure 11. Comparison between the initial and adapted solutions (125 adaptation steps), for the fine, coarse and counter-intuitive meshes.

each of the four velocity triangles form a large one at the vertices of which density and temperature are defined. Local time stepping is used with an implicit Gear scheme of second order to advance in time. The set of linear equations at each time step is solved by a non-linear GMRES solver. The code is programmed in double precision.

The finite volume code is NSC2KE, developed by Bijan Mohammadi [4] and available on the Web. It solves the Navier–Stokes equations in conservative form for the conservative variables (ρ, ρu, ρv and ρE) and uses linear interpolation for all. Several solvers are available, among them a Roe scheme, an Osher scheme and kinematic schemes of orders 1 and 2. An explicit four-step Runge–Kutta scheme is used to advance in time, with local time stepping. The code is written in single precision.

Results are presented at $Re = 500$ because, at such a low Reynolds number, the shock is thick and the convergence behaviour is easily seen. The conclusion remains the same at higher Reynolds numbers.

Having established that the algorithms are substantially different and involve many features that by themselves merit a comparative study (FVM versus FEM, equal order interpolation versus mixed elements, primitive versus conservative variables, different artificial viscosity, etc.), Figure 15 shows the large differences in the two solutions when the codes are first run on the initial mesh. Of course, one can probably tweak the parameters (artificial viscosity coefficients, etc.) in both solvers to improve the solutions, but still some differences will remain. On the other hand, Figure 16 compares the final adapted solution from the two codes.

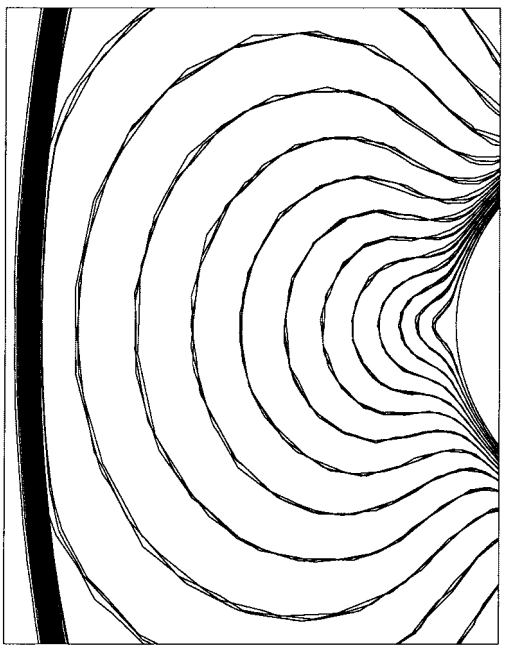

Figure 12. Superposition of the three final solutions at the leading edge.

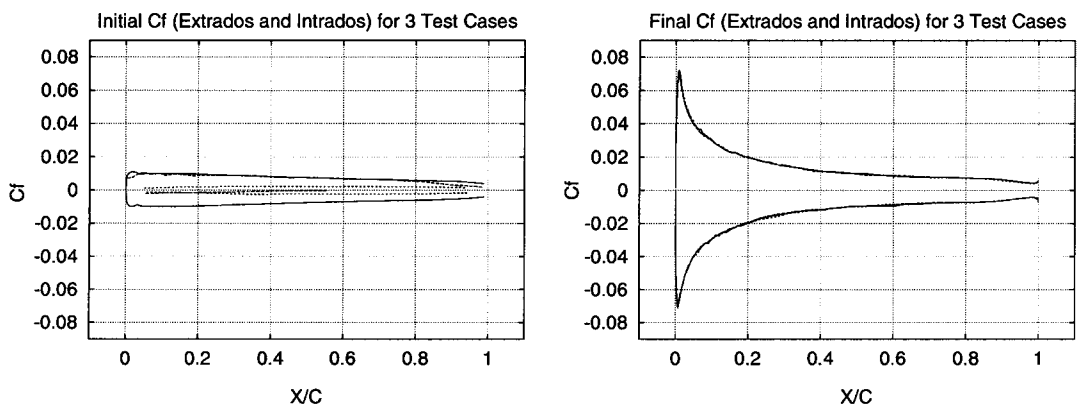

Figure 13. C_f on initial meshes (left) and final meshes (right) for the three test cases.

Surprisingly, they are not only similar; they are for all intents and purposes identical. A comparison of the Mach contours shows them to be amazingly similar. A blowup of the solution at the leading edge shows, on the left side of Figure 17 the superposition of the Mach number contours, and on the right side a split screen showing in the top half the FEM upper solution and in the bottom half the FVM solution. Their juncture at the centreline hardly shows a kink. In Figure 18, a horizontal cut at the leading edge shows that the shocks

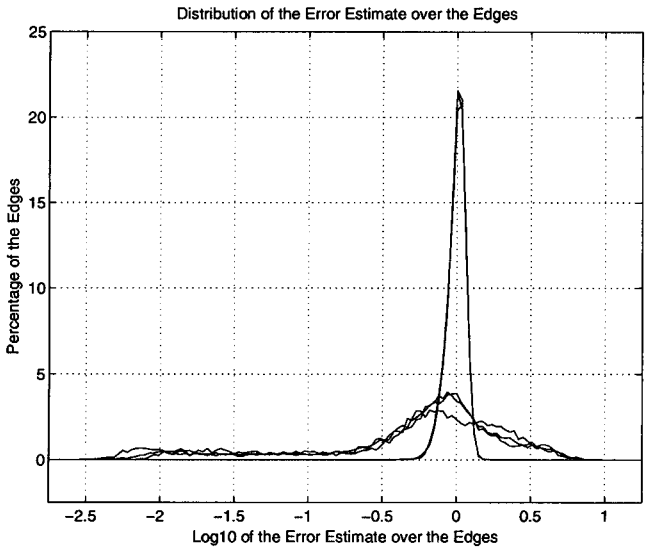

Figure 14. Percentage of the element edges versus the error estimate over the edges for the three test cases, at the first and final mesh adaptation steps.

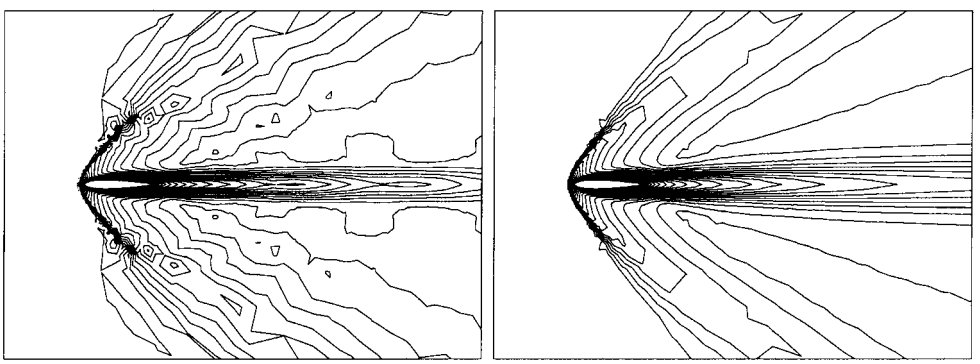

Figure 15. FEM (left) and FVM (right) solutions on the initial mesh ($Ma = 2$, $Re = 500$).

obtained by both codes are identical. Derived quantities such as the pressure and friction coefficients are compared in Figure 19. One must be reminded that the figure on left is the superposition of four curves (one each for FEM, FVM, suction and pressure surfaces), while the second figure is the superposition of two results.

It must also be mentioned that the mixed finite element code of Boivin did not require any upwinding to be added, and it can therefore be assumed that the adapted mesh lowered any upwinding needed in the Mohammadi code way down as the mesh was adapting, proving at the end that a more elaborate upwinding scheme will show a difference on unadapted meshes but may make little or no difference on an adapted one.

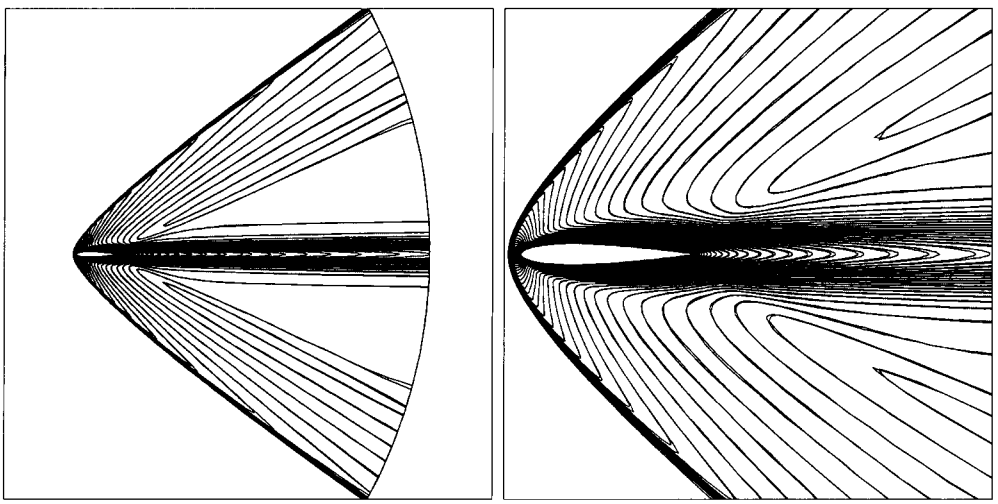

Figure 16. Superposition of FEM and FVM solutions ($Ma=2$, $Re=500$).

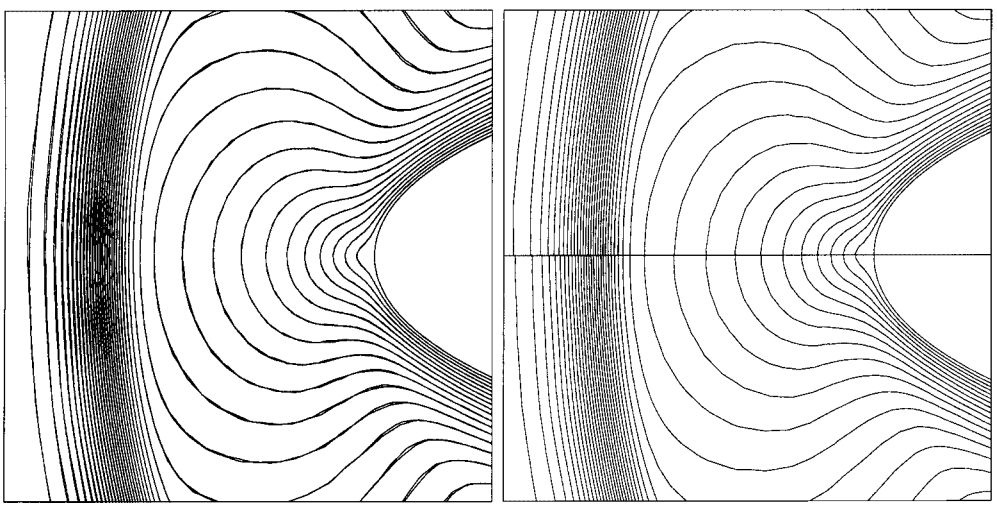

Figure 17. Zoom on the leading edge. Left: superposition. Right: (top) FEM, (bottom) FVM.

The preceding example is at least convincing enough to create further interest in researching this point. It brings one to question the major effort of the last decade into more refined algorithms, all the while ignoring the impact of the meshes on the numerical solution.

3. VALIDATION WITH LAMINAR AND TURBULENT FLOWS

A number of viscous flow test cases will be presented spanning subsonic to hypersonic Mach numbers.

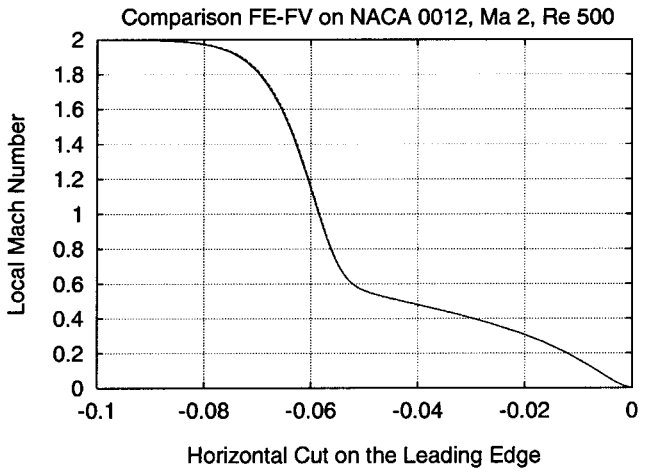

Figure 18. Horizontal cut through the shock wave for FEM and FVM.

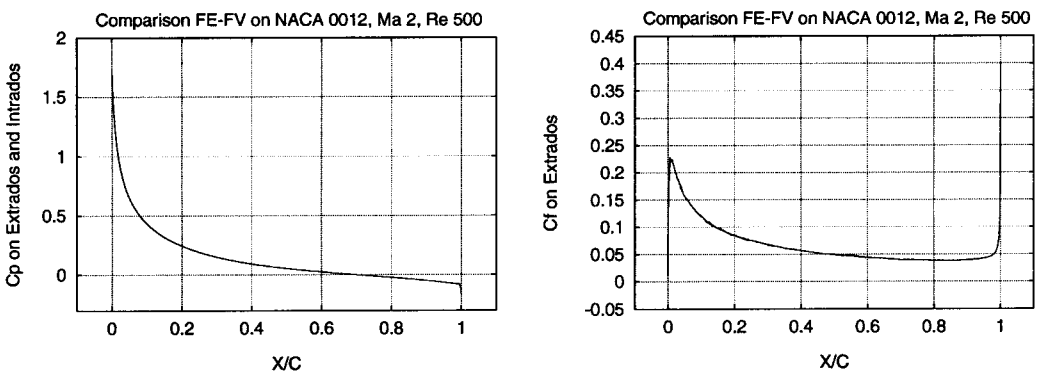

Figure 19. Comparison of C_p (left) and C_f (right) for FEM and FVM.

3.1. *Laminar flow*

One of the problems all CFD developers have to contend with is that of spurious oscillations. Generally speaking, oscillations may appear during the solution of flow equations for two reasons. First, the approximation scheme may be unstable. For example, while solving the Stokes or Navier–Stokes equations, an equal-order approximation of the velocity and pressure gives rise to spurious oscillations (see, e.g. Reference [5] or [6]). As long as the discretization scheme does not satisfy the inf–sup condition of stability, some terms must be added to the equations to eliminate oscillations and ensure convergence. Oscillations can also appear in convection dominated flow computations, even with a stable approximation. They are mainly caused by meshes that are too coarse to represent the convected solution in areas of steep gradients. The simplest cure is to add a diffusive term into the equations. Artificial

viscosity and upwinding methods are often combined with oscillation detectors to selectively add diffusion, while preventing excessive smearing of the solution.

Mesh adaptation proves to be a good way to altogether reduce the amount of artificial viscosity needed. More than that, we postulate that any artificial viscosity is bad viscosity and that very little, and sometimes none, should be needed, provided that:

(1) the discretization scheme is stable,
(2) and the mesh can be adapted enough to capture physical features.

Of course, adaptation has its own limitations and cannot cure all spurious oscillations, but it has a stabilizing effect. This concept is nicely illustrated by the following example: a compressible viscous solution is computed by the Boivin [3] code on a NACA 0012 at a freestream Mach number of 2 and Reynolds numbers ranging from 125 to 32 000.

The mesh adaptation algorithm makes it possible to compute external viscous supersonic flows with *absolutely no artificial viscosity*, at least up to a Reynolds number of 32 000. We did not try to further increase the Reynolds number because the flow computation becomes prohibitively expensive, due to the unsteadiness of the wake at Reynolds above 32 000 (Figures 20 and 21).

To show that the computed solutions contain only the amount of natural viscosity prescribed by the fluid, without any extra artificial viscosity, cuts are taken through the boundary layer and through the shock wave (Figures 22 and 23, left). In Figure 22 (right), it can be seen that the thickness of the boundary layer varies exactly as the square root of the Reynolds number and in Figure 23 (right), that the slope of the viscous shock is exactly proportional to the Reynolds number. This is in perfect agreement with the laws of fluid mechanics and shows that, using adapted meshes, the added numerical viscosity can be nearly nil and the viscosity nearly equal to the physical one.

3.2. Turbulent flow

In the previous sections, the mesh optimization methodology was shown to be capable of generating meshes for flows with shocks. In this section, it will be applied to automatically generate meshes for turbulent flow calculations.

Meshes for turbulent flow computations usually cluster the nodes in a region of thickness 1×10^{-3} chord close to the wall. Also, the first node off the wall should be around 1×10^{-5}–1×10^{-6}, even closer when a low Reynolds number turbulence model is used. The handmade meshes that satisfy these requirements usually are very stretched, dense across the boundary layer and sparse along the body. It is essentially the same kind of meshes used to track shocks, but parallel to the walls.

With the results presented below, we only intent to show that our mesh adaptation captures the features of the turbulent flow boundary layers using highly anisotropic triangles. All our computations arc based on the κ–ε turbulence model using a wall-law boundary layer formulation, as implemented in the code NSC2KE developed by Bijan Mohammadi [4].

3.2.1. AGARD AR138 test case. The test case AGARD AR138 is used to test the mesh adaptation methodology for turbulent flow. This test case is a NACA 0012 at $Ma = 0.5$, $Re = 2.89 \times 10^6$ and $\mathrm{AoA} = 0°$. Figure 24 shows the adapted mesh and Mach contours on an 11 116-node mesh with a distance from the wall for the first layer of about 1.0×10^{-5}.

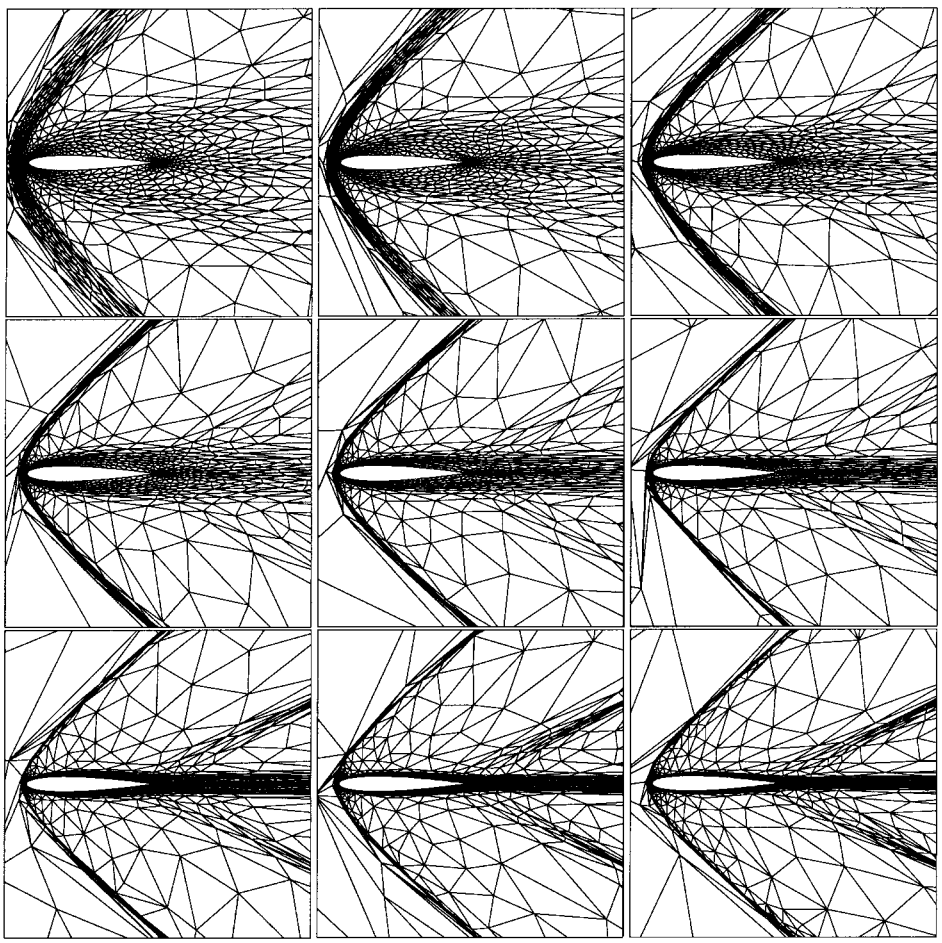

Figure 20. Adapted meshes for flow at $Ma = 2.0$ and $Re = 125$, 250, 500, 1000, 2000, 4000, 8000, 16 000 and 32 000.

Figure 25 shows a zoom of the mesh and the Mach contours in the boundary layer and Figure 26 shows a comparison of the experimental and numerical pressure coefficients. This test case demonstrates that the mesh optimization methodology is suitable for turbulent flows, although it might be argued that Euler would also have given a good pressure distribution. Effectively, the wake is predictable when the angle of attack is zero and, since there is no separation of the boundary layer, there exist some formulae to predict its thickness. Then, it is easy to build a C-grid around the wing with a specified node distribution such that turbulent flow is easily computed. But, in a more general case, with complex geometries such as multiple element airfoils, the boundary layer and the wake are less predictable and a handmade turbulent flow mesh is more difficult to generate. The test case in the next section will demonstrate the suitability of the mesh adaptation methodology in these more complex situations.

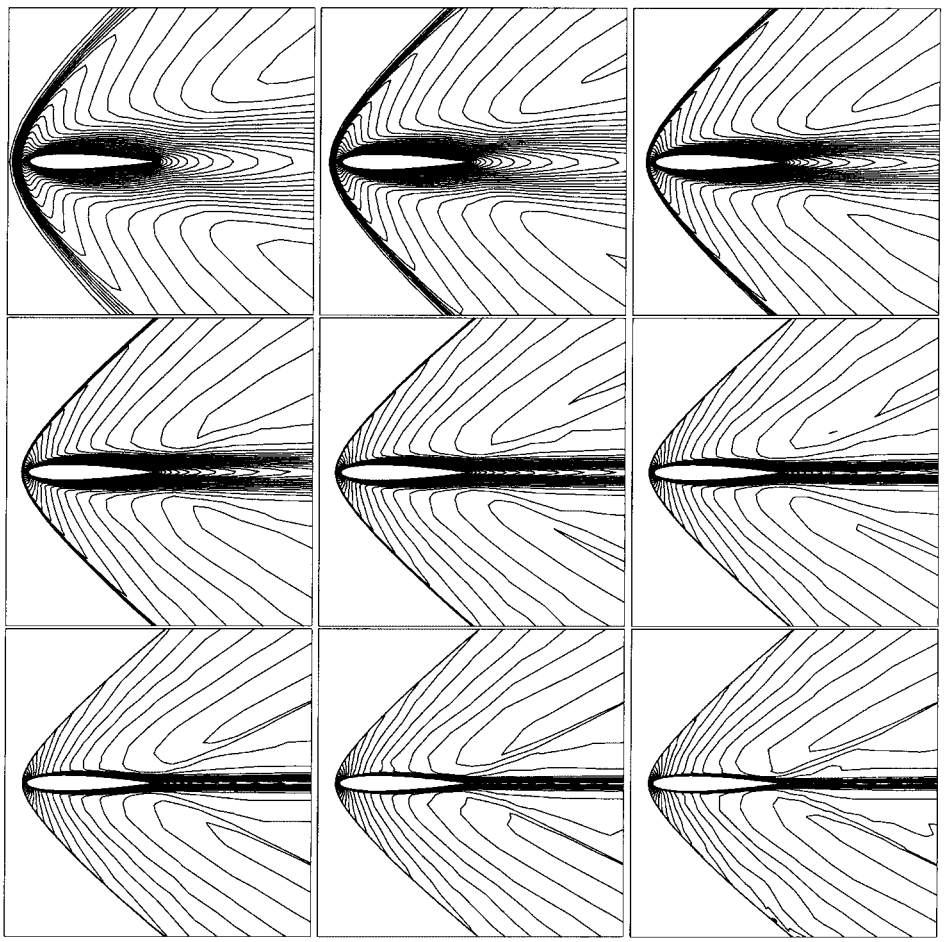

Figure 21. Mach contours for flow at $Ma=2.0$ and $Re=125,\ 250,\ 500,\ 1000,\ 2000,\ 4000,\ 8000,\ 16\,000$ and $32\,000$.

3.2.2. 2-D car. The following test case is a 2-D generic car scaled down by a factor of 2.5. The wind tunnel experiments are at $Re=3\times10^6$ and the exact Mach number for the experiments is not specified, but $Ma=0.1$ seems a reasonable choice for a car. The car is 10 cm over a floor that begins 0.88 chord length before the leading edge of the car. The flow is characterized by a duct flow between the car and the floor, by a separated region over the rear window which has a large recirculation. This situation, while still geometrically simple, is much more complex to mesh than the NACA 0012 of the previous section. The reason is that, contrarily to the AGARD AR138 test case, the flow features are less predictable and one is left with making arbitrary decisions.

Figure 27 shows a general view of the adapted mesh and Mach contours for the car over a moving ground. The mesh has 11 072 nodes with the smallest distance to the wall around 5×10^{-6}. Figure 28 shows the initial mesh and Mach contours. This mesh has 10 050 nodes

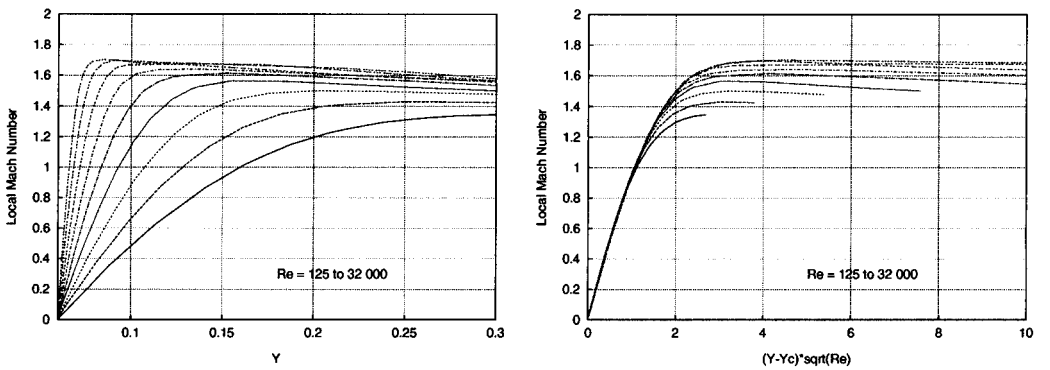

Figure 22. Cut of the local Mach number through the boundary layer at $x = 0.3$ (left) non-dimensionalization of the cut (right).

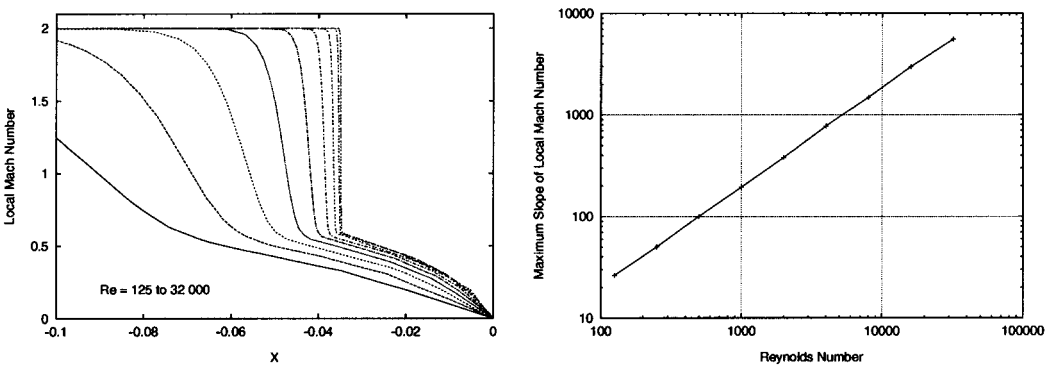

Figure 23. Cut of the local Mach number through the bow shock along the x-axis (left). Maximum slope of the local Mach number over the cut through the viscous shock wave versus Reynolds number (right).

and was automatically generated. On the same figure, the adapted mesh with 13 648 nodes clearly shows small and stretched triangles in the thin boundary layer close to the body.

Figure 29 shows a zoom of the mesh and the corresponding Mach contours between the car and floor, demonstrating clearly the advantages of anisotropy. As can be seen in Figure 30, the comparison between the experimental and numerical pressure coefficients, before and after adaptation, demonstrates the effectiveness of the method.

4. ROUNDOFF ERROR AS A LIMITING FACTOR

To illustrate a limiting factor of the method, we use test case III.3 of the Workshop on Hypersonic Flows for Reentry Problem [7] held in Antibes, France in 1990 and 1991. The geometry of this test case is a flat plate 0.25 m long, with a 0.10 m ramp at 15°. The leading edge is a 15° downward ramp and the thickness of the body is 0.02 m (Figure 31). The wind

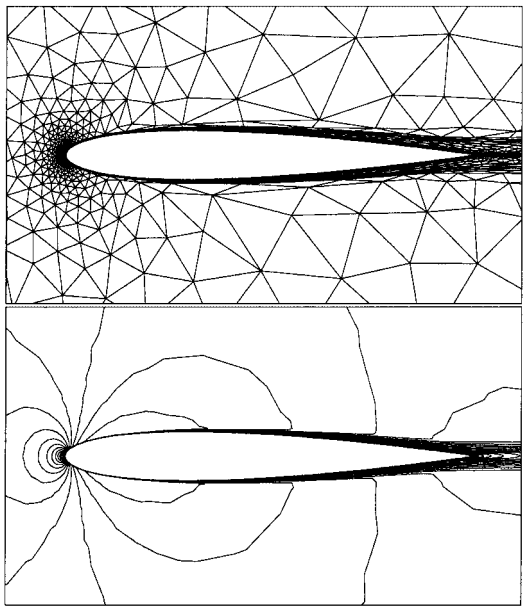

Figure 24. Adapted mesh and Mach contours for NACA 0012 at $Ma = 0.5$, $Re = 2.89 \times 10^6$ and $\mathrm{AoA} = 0^\circ$.

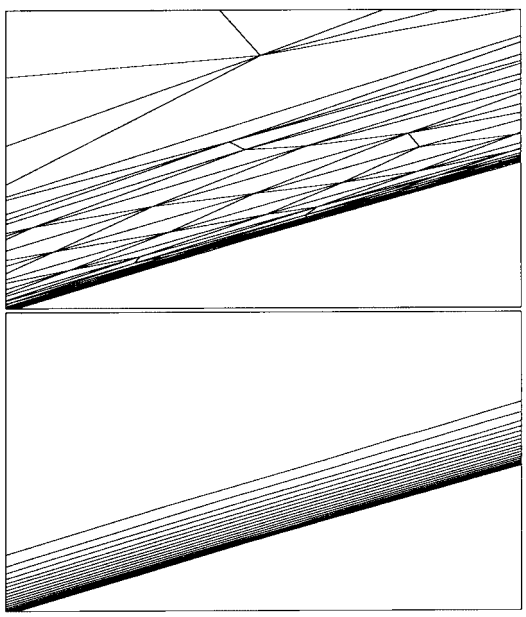

Figure 25. Zoom on the boundary layer for test case AGARD AR138.

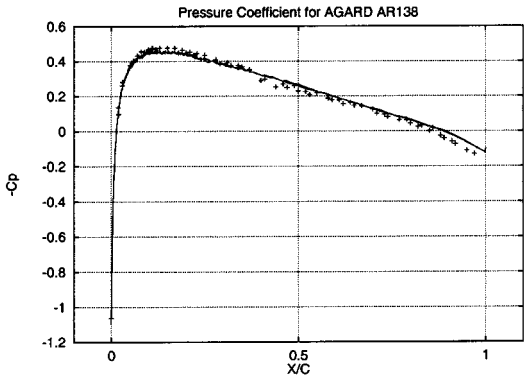

Figure 26. Comparison of the experimental (+) and numerical (solid) pressure coefficients for test case AGARD AR138.

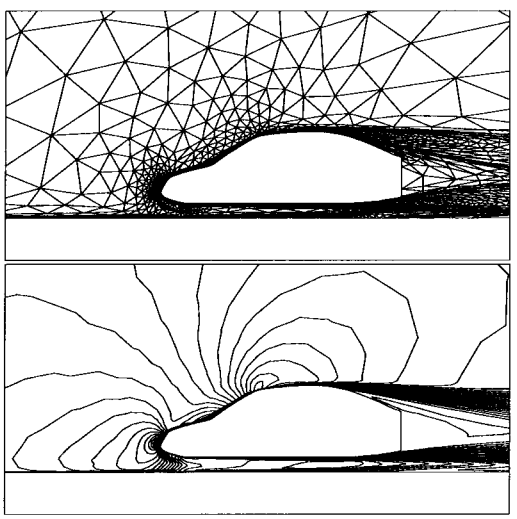

Figure 27. Adapted mesh and Mach contours for 2-D generic car over a moving ground at $Ma=0.1$ and $Re=3\times10^6$.

tunnel experiments [7] at $T_\infty=50\,\mathrm{K}, T_{\mathrm{wall}}=290\,\mathrm{K}$, $Ma=10$ and $Re=9.1\times10^6$ show a laminar non-reactive flow field and therefore, modelling with the laminar Navier–Stokes equations is well suited. This is a difficult test case because of thin boundary layers and shock-boundary layer interaction arising from high Reynolds and Mach numbers. The flow is computed using the Boivin FEM code [3].

Local Mach number contours are shown in Figure 31. To get this solution, we start from an arbitrary coarse mesh and a uniform flow initial solution. With this initial mesh and solution, the solver needs quite a bit of additional artificial viscosity. As the solution–adaptation iterations proceed, the artificial viscosity is progressively decreased. For $Re<10^5$, one is able

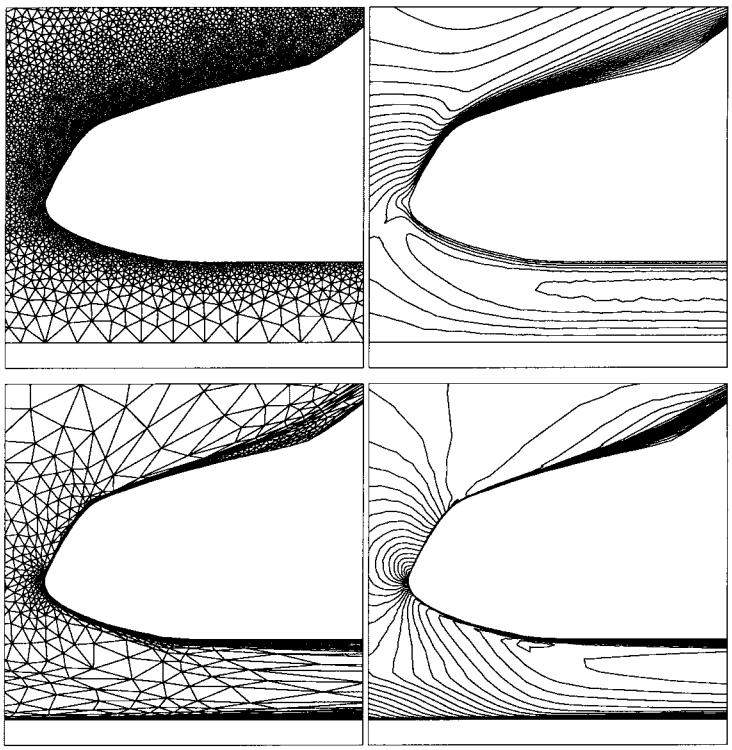

Figure 28. Initial and adapted meshes, and the corresponding Mach contours, around the 'leading edge' of a 2-D generic car over a flat plate at $Ma = 0.1$ and $Re = 3 \times 10^6$.

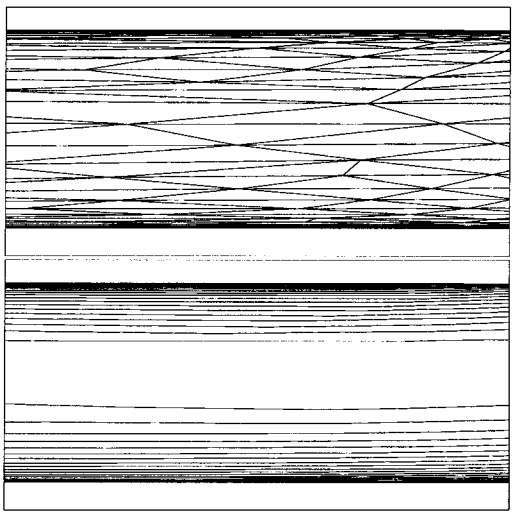

Figure 29. Zoom of the mesh and Mach contours between the floor and a 2-D generic car at $Ma = 0.1$ and $Re = 3 \times 10^6$.

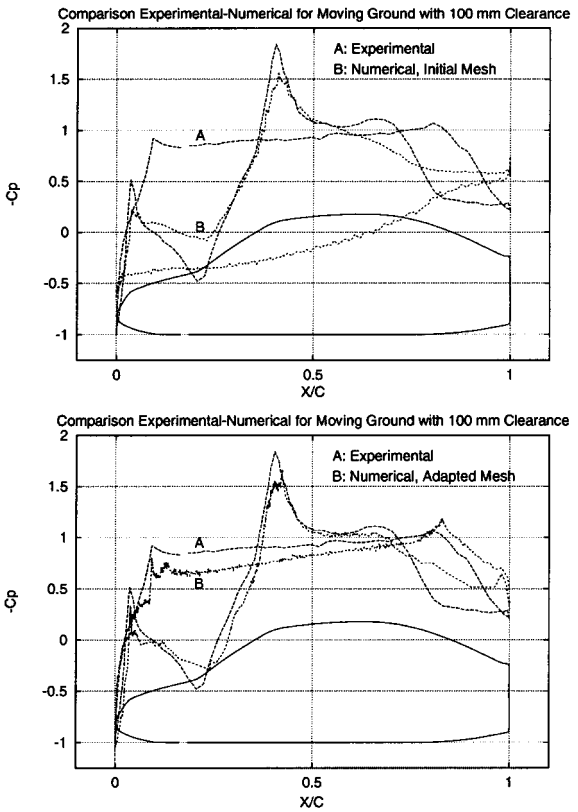

Figure 30. Comparison of the experimental and numerical pressure coefficients without and with mesh adaptation.

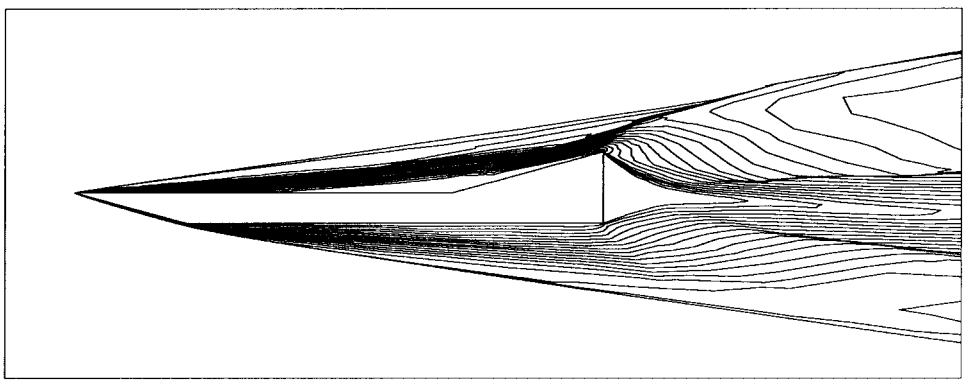

Figure 31. Flat plate and 15° ramp of the EHDB test case III.3. The figure shows Mach contours from 0.0 to 10.0 with $\Delta Ma = 0.5$.

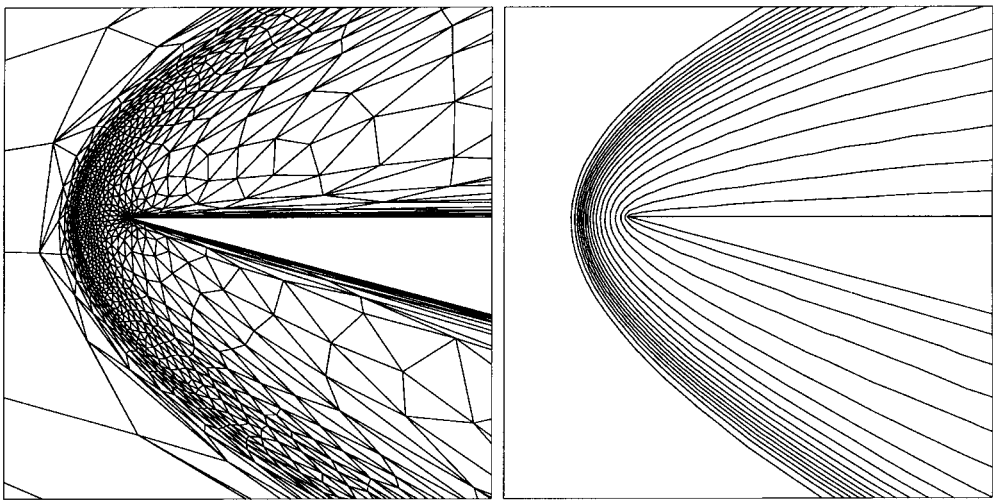

Figure 32. Zoom on the leading edge of the mesh and Mach contours.

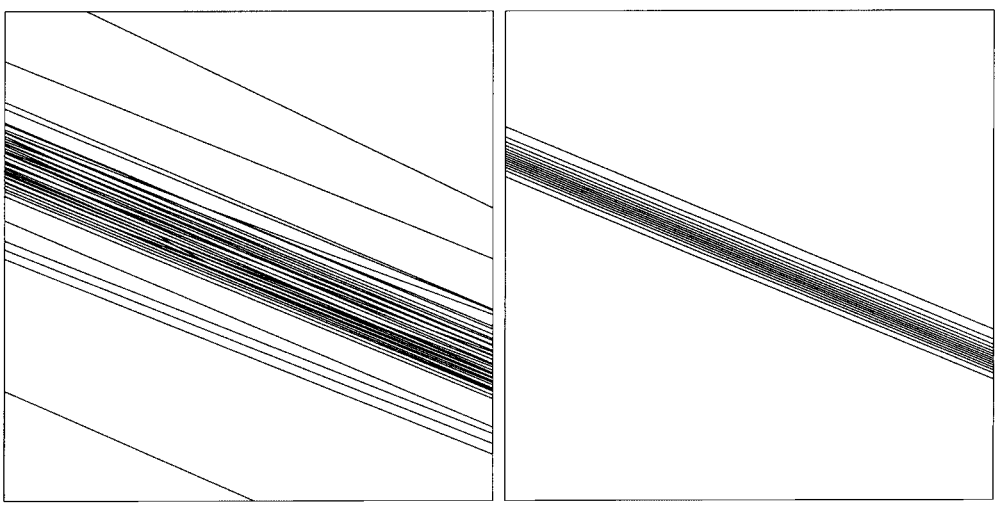

Figure 33. Zoom through mesh at the lower shock, and resulting Mach contours.

to completely eliminate the artificial viscosity. For $Re > 10^5$, there always remains the need for some artificial viscosity.

Figures 32 and 33 show some features of the mesh and the solution for $Re = 10^6$, while Figure 32 is a zoom at the leading edge. There are large triangles before the shock and very small triangles within the shock and at the leading edge. The area ratio of the largest to the smallest triangle, in the whole domain, is 10^{14}. With the standard h-method adaptation, which

consists of dividing a triangle into four triangles, 23 refinement steps would have been needed to reach the same level of adaptation. The mesh is therefore considered strongly adapted.

The ratio between the length of the longest edge to the length of the shortest edge is 10^7. It is important, though stunning, to think that in the chosen metric, all these edges are approximately of the same length. The shortest edge is at the leading edge and its length is around 10^{-8} m. While this makes sense for a geometry with a perfect singularity at the leading edge, the real test object in a wind tunnel could not be made to have such a sharp leading edge and therefore, some discrepancy can be anticipated with the experimental data.

The left-hand side picture in Figure 33 is a zoom of the mesh on the lower shock and the right-hand side picture shows the corresponding isovalues of the Mach number. Fifteen Mach contours are shown from Mach 10 (at the bottom) to Mach 5 (at the top). The theory states that the thickness of the shock is inversely proportional to the Reynolds number. With $Re > 10^6$, the aspect ratio of the triangles in the shock reaches 10^{-6}. The shock is crisp, perfectly uniform and oscillation-free because the mesh was built to correspond to the solution that it should represent.

Had the mesh been isotropic, there would be oscillations because the edges would not be aligned with the shock. And, in passing, there would also have been thousands and thousands more triangles!

For the Reynolds number (9.1×10^6) of this test case, the slope of the viscous shock should be 10^7 and the shape of the triangles should be approximately inversely proportional. Such stretched triangles cannot be constructed in double precision because the three nodes of the triangle would be so close to a straight line that the calculation of some characteristic functions on the triangle, for example the shape factor of the triangle in the mesher or oscillation limiters in the solver, would be swamped by roundoff error. So, for high Reynolds numbers, we cannot build the mesh to represent the solution of the equations if co-ordinates are only in double precision, and therefore some artificial viscosity is needed to capture the solution on such a non-optimal mesh. We believe that at such a conjuncture the approach has reached a (temporary) limit for 2-D flows. Pushing the limit of the stretching factor to machine accuracy is certainly feasible in double precision, but this would require a careful analysis of the floating-point error propagation for each formula based on the triangle vertex co-ordinates in the mesher and the solver.

5. CONCLUSIONS

An anisotropic mesh adaptation method has been proposed. An edge-based error estimate drives four local mesh modification techniques, namely refinement/coarsening, edge swapping and node movement, to obtain elongated elements along the directional features of the flow, such as shocks, boundary layers, slip lines, wakes, etc.

The full power of the method has been demonstrated for 2-D unstructured meshes, using a number of numerical examples. In particular, it was shown that:

(1) Given an input mesh and a numerical solution on that mesh, the combination of the four modification techniques leads to a convergent mesh modification algorithm.
(2) The iterative coupling of the mesh adaptation and the flow solution converge to a final mesh.

(3) This final mesh depends on the flow conditions but is independent, except for slight variations, from the flow solver and from the initial mesh.

The method has also been proven to be efficient for laminar and turbulent flows over several geometries. For 2-D steady flows, the main limitation of the method seems to be the roundoff error caused the floating-point representation of the mesh point co-ordinates.

ACKNOWLEDGEMENTS

The authors would like to thank NSERC and FCAR for Operating and Strategic grants under which this work was supported and the Centre de Recherches Mathématiques de I'Université de Montréal (CRM) for its partial support of a Post-Doctoral Fellowship to Dr J. Dompierre. Thanks are also due to Prof. Sylvain Boivin of the Université du Québec à Chicoutimi, for allowing the use of his P1/(P1-iso-P2) Navier–Stokes solver [3], and to Prof. Bijan Mohammadi of the Université de Montpellier 2, for allowing the use of his NSC2KE flow solver [4].

REFERENCES

1. Fortin M, Habashi W, Vallet M-G, Dompierre J, Bourgault Y, Ait-Ali-Yahia D. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles. *International Journal for Numerical Methods in Fluids* 2000; **32**:725–744.
2. Ait-Ali-Yahia D, Baruzzi G, Habashi W, Fortin M, Dompierre J, Vallet M-G. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part II: structured grids. *International Journal for Numerical Methods in Fluids* 2002; **39**:657–673.
3. Boivin S. A numerical method for solving the compressible Navier–Stokes equation. *Impact of Computing in Science and Engineering* 1989; **1**:64–92.
4. Mohammadi B. Fluid dynamics computation with NSC2KE, a user-guide, release 1.0. *Technical Report RT-0164*, Institut National de Recherche en Informatique et en Automatique (INRIA), May 1994.
5. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods.* Springer Series in Computational Mathematics. Springer: Berlin, 1991.
6. Girault V, Raviart P-A. *Finite Element Methods for Navier–Stokes Equations*: *Theory and Algorithms.* Springer: Berlin, 1980.
7. Désidéri J-A, Glowinski R, Périaux J (eds). *Hypersonic Flows for Reentry Problems*. Springer: Berlin, 1991–1993.

