Certifiable computational fluid dynamics through mesh optimization

Habashi, W. G.; Dompierre, J.; Bourgault, Y.; Fortin, M.; Vallet, M. -G.

Publisher's version / Version de l'éditeur:
https://doi.org/10.2514/2.458
AIAA Journal, 36, 5, pp. 703-711, 1998-05

Soumis à l'AIAA Journal

IMI96AP-50302-71085-G
CNRC 40575

Certifiable CFD through Mesh Optimization

W.G. Habashi**, M. Fortin†,
J. Dompierre*¶, M-G. Vallet‡ and Y. Bourgault*

* CFD Laboratory, Dept. of Mech. Eng., Concordia University, Montreal, Canada

† Dept. of Math. and Stat., Université Laval, Ste-Foy, Canada

¶ Centre de Recherches Mathématiques, Université de Montréal, Montreal, Canada

‡ Industrial Materials Institute, National Research Council, Boucherville, Canada

Abstract

This paper addresses the problems of CFD accuracy and reliability by proposing an efficient mesh optimization approach. By developing an appropriate directional error estimator, coupled with an effective mesh adaptation technique closely tied to the solver used, it can be demonstrated that for each flow condition and geometry combination an optimal grid can be obtained. It is further demonstrated that such an optimal grid can be reached from almost any reasonable initial mesh and that the order of accuracy of well-posed numerical algorithms have a minimal impact on solution accuracy if the grid is well adapted. Thus, the proposed approach can be considered a first step towards user-, mesh- and solver-independent, and thus certifiable, CFD.

1 Introduction

The last decade has seen Computational Fluid Dynamics (CFD) become the method of choice in the design of many aerospace, automotive and industrial components and processes in which fluid

*Associate Fellow, AIAA

or gas flows play a major role. Phenomenal advances have been made in this discipline with a corresponding meteoric growth in its use for an ever increasing number and range of applications. The conventional empirical way of the component/process design-building-testing cycle has been almost altogether superseded by ever more (accurate and reliable?) computer simulations, leading to analyses showing features and details that are difficult and/or expensive to measure or visualize. These simulation capabilities have led to first-off designs, in which prototypes are seldom built and tested, but concepts and ideas are parametrically investigated on computers, with only the computationally-selected final configuration verified in a wind tunnel, test cell or full-scale laboratory conditions before becoming the product.

Thus, an increased competitiveness has set in in the development or acquisition of (reliable?) tools to reduce the lead time needed to bring more refined CFD-based high performance designs to the market. Already, the two main challenges of the next few years for CFD have been defined as being its use in inverse design or automatic optimization and its utilization in multi-disciplinary contexts such as conjugate heat transfer (flow and heat transfer), aero-elasticity (flow and structures), aero-acoustics (flow and noise), aero-thermodynamics (flow and combustion), aero-icing (flow and ice accumulation over lifting surfaces) and aero-electromagnetics (stealth aircraft), etc.

Yet, one of the major but most mundane concerns facing CFD today is the assessment and enhancement of the quality of the solutions obtained, be they from proprietary or commercial codes. The various modeling approximations and algorithmic decisions made in building a simulation code make it such that it is not necessarily a given that for a specific physical situation, solving exactly the same set of equations, all codes, or even most of them, necessarily yield the same solution or even close answers.

Such discrepancies are due to several reasons that must be individually identified before one can hope to eliminate or altogether reduce them. The lead article by Unmeel Mehta in this AIAA Journal issue spells out in great details the sources of such errors, and ways and means of identify-

ing them in view of eliminating or avoiding them.

In the present article, we first distinguish between the modeling and numerical errors in order to clear a path to considerably reducing the latter. The process of building a CFD code is illustrated in Fig. 1. Starting from the fundamental laws of physics, a set of governing equations and corresponding boundary conditions will be selected to best, or at least in the most cost-effective manner, simulate a given physical class of problems.

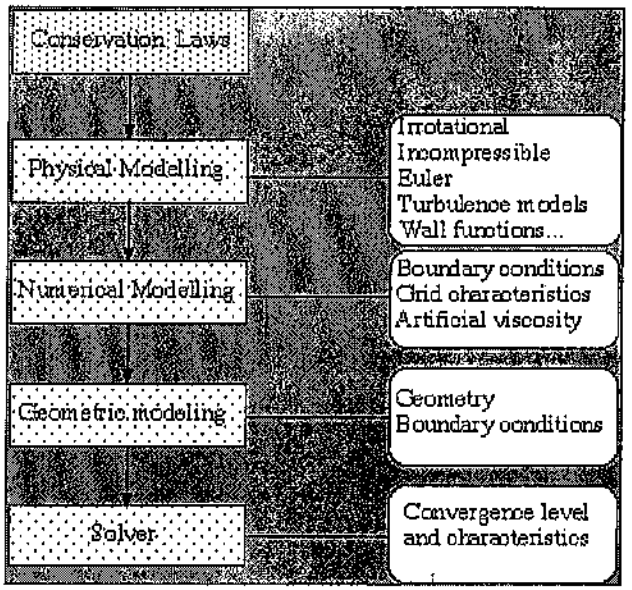


Figure 1: Steps in constructing a CFD code

At that stage, it must be realized that modeling has disconnected physics from numerics and that a certain discrepancy is to be expected between reality (interpreted as measurements and/or observation) and the numerical answers. We address in this paper this second level of error. After physical assumptions have been made and a set of partial differential equations with appropriate boundary conditions has been defined, the question to be posed is why one cannot obtain a very accurate and certifiably unique answer to the solution of these partial differential equation. The errors made in solving that set, i.e. the difference to be anticipated between the exact solution of the partial differential equations and the numerical solution of the discretized equations are, in

general, due to the following culprits:

1. Error of discretization and roundoff,

2. Other grid related errors: arbitrary and hence innapropriate grid distribution and various geometry approximations,

3. The addition of stability and convergence enhancers such as artificial viscosity, damping, smoothing, upwinding, etc,

4. Less than complete convergence of the code (it is rare to see convergence curves, too machine accuracy, in papers).

In fact, rather than limit ourselves to rhetorically analyse the reasons for the lack of accuracy, we will address in this paper what we believe are the steps needed to maintain a handle on the numerical error and control it. We will specifically address the issue of the current arbitrariness of grid generation and how, by strategically adapting the grid to an evolving solution, a mesh optimization approach can overcome this problem, while simultaneously:

1. Yielding solutions that are discretization method-independent,

2. Yielding solutions that are initial grid-independent,

3. Considerably reducing, if not altogether eliminating, the need for stability and convergence enhancers.

The first step usually made in improving solutions is to note that CFD phenomena are characterized by regions of steep gradients of the flow variables, embedded in regions where these variables vary more smoothly. The general trend in large-scale CFD problems of current interest is the use of a large number of grid points to as much as possible resolve physical features

such as boundary layers, flow separation, recirculation regions and shocks. Taking this to a sometimes impossible limit, mesh-independence seekers use finer and finer grids until the solution no longer changes. With most complex multi-scale and multi-physics phenomena, attaining such grid independence in three dimensions, even for those best equipped in terms of computers, can be a chimera. This brute force approach to mesh independence is due to the fact that the necessary grids to carry out CFD analyses are currently arbitrarily generated, only reflecting a user's engineering sense of where to concentrate points, within given constraints of memory, boundary conditions or mesh aspect ratio quality. Furthermore, in the worst cases, these grids may have undetected geometrical pathologies such as skewness and degeneration, that introduce unwanted grid-dependent errors in flow solutions. For these various reasons it is not unexpected that different solutions could be obtained by various users of the same code, on the same flow problem, using the same overall number of intuitively generated mesh points.

It can thus be concluded that urgent emphasis should be placed on relieving the user from mesh decisions and that grid optimization should be carried out through an automatic adaptive process. Recent years have thus seen a rapid development of such adaptation methods based on a posteriori error estimates: given a computed solution, one assesses its quality or more precisely measures the error with respect to the exact solution and alters either the mesh or the discretization to obtain a target precision level. Guided by an equi-distribution principle that there is no justification for getting a more accurate solution in some regions at the expense of others, adaptation efforts aim at producing a balanced error distribution or quality.

Current grid refinement strategies suffer from several problems, not least among them is the fact that they may easily lead to an uncontrolled increase in the number of nodes, in the search for higher precision or even in the search for a more uniform error distribution. Both refinement and redistribution methods produce nearly isotropic meshes since their aim is to make the length scales of each element essentially the same in all directions. These methods are therefore optimal mostly

for flow field regions possessing large gradients in all spatial directions but can stand improvement for regions with highly directional flow features such as shocks, boundary layers, wakes, vortices and slip lines.

In this paper an alternative approach is proposed, which is to seek solutions on anisotropic meshes where more resolution is selectively introduced along directions of rapidly changing error estimates, hence without necessarily increasing the overall number of mesh points.

In the following, the anisotropic adaptation criterion is presented and the adaptive strategy and its implementation for triangular meshes is demonstrated. The strategy of coupling the adaptive library to FEM and FVM solvers is also studied. This anisotropic strategy is validated through the monitoring of the adaptation loop and convergence of the overall solver-adaptation cycle. Finally, the efficiency of the strategy is demonstrated on examples of external flow computations through a range of Mach and Reynolds numbers.

2 Error Estimators for Grid Adaptation

Recent years have seen a rapid development of a posteriori adaptation methods based on error estimates. Given a computed solution u_h, one wants to assess its quality, more precisely measure the error with respect to the exact (unknown) solution and from this measure, adapt the mesh or the method to obtain a uniform, or at least improved, precision level.

The construction of error estimators is thus a crucial point in this process. It is clear that, the exact solution being unknown, such estimates can only be indicators. There have been [2, 3, 4, 14, 15] some formal analyses of properties that should be expected from a good estimate. Generally speaking, these formal analyses aim at determining an upper and a lower bound for the error based on an error estimator, the error estimator being thus usable to add or delete mesh points to *equidistribute* the error.

The basic idea underlying the error estimator proposed here is the *a priori* estimate

$$E = \|u - u_h\| \leq C\|u - \Pi_h u\|$$

where $\Pi_h u$ denotes the interpolate of the solution in the finite element space, C is a constant depending only on the PDE solved without any minimal or maximal angle condition to satisfy and $\|\cdot\|$ denotes a usual functional norm to measure the error. This inequality means that controlling the interpolation error will indeed result in a control of the error on the (FE, FV, FD, etc) solution itself.

The problem of finding an optimal interpolant has been addressed by Simpson and D'Azevedo [10, 11] for a piecewise linear finite element approximation. The case of higher order interpolants has not yet received much attention, although the actual theory for linear element has already been used to efficiently improve results for higher order methods. The starting point of Simpson and D'Azevedo's theory is the idea that the error on a piecewise linear interpolation is bounded by a quadratic term, as in a Taylor series expansion. More precisely, from the elementary theory of Lagrange interpolation, it is well known that the interpolation error $E(\bar{x})$ at a point $\bar{x}$ is given by

$$|E(\bar{x})| = \left|\left(\frac{\bar{x}^2}{2} - \frac{\bar{x}h}{2}\right)g''\right| \leq \frac{h^2}{8}\left|g''\right|, \tag{1}$$

whenever $x = 0$ and $x = h$ are used as interpolation nodes and g'' is the second derivative of the interpolated function at some point. The same reasoning can be applied on the edge of the meshes, showing that measuring the error on an edge is directly related to the estimation of the second derivative along that edge. Then comes again the idea that globaly minimizing the interpolation error is equivalent to equidistributing this error or a scaled measure of it, namely the edge length squared times the second derivative of the solution. In 2-D, the mesh that minimizes the interpolation error for a given number of nodes would thus yield only equilateral triangles, but with edge length measured using the second derivative of the solution. A solution rapidly varying in one direction, such as a shock wave, would have a large second derivative in this direction and a small

one in the orthogonal direction, naturally introducing anisotropy in the length measure and in the mesh.

3 Mesh Adaptation Tools

The adaption loop is briefly explained in this section. To obtain a mesh with all edges having approximately the same value of the error estimator, the mesh is recursively modified. This approach is thought to be cheaper than generating new meshes, especially at the late stages where there are few differences between the current mesh and the starting one.

Input for the adaptive step is thus a *background mesh*, with a solution vector and a description of the domain boundaries. Second derivatives of the solution are then computed on this mesh and kept unchanged during the entire adaptation loop. Local modifications are iteratively performed until the error estimates, constantly reinterpolated using the derivative field on the background mesh, are nearly equal on all edges of the current mesh. The output is then an adapted mesh with a solution interpolated on it from that on the background mesh.

3.1 Local Improvements

All mesh modifications are local and for a triangular unstructured mesh, four local operations are used:

- adding a node at mid-edge if the measured error is above the target error,
- removing a node if the error is smaller than the target error on all surrounding edges,
- moving a node to equidistribute the error on its neighboring edges,
- swapping the diagonal of the quadrilateral formed by any pair of adjacent triangles in order to equilibrate the error estimator over the edges of neighboring two elements.

If adapting a structured mesh, the operations are restricted to node movement, so that the mesh structure is preserved [1]. Due to the connectivity constraints in that case, the goal of having the edge error estimates equally distributed must somewhat be relaxed. The optimal grid is then defined as one with minimum energy, using analogy with a spring network.

3.2 Adaption Loop Strategies

Particular attention has to be paid to the different criteria driving the adaptation process. For example, the refinement process converges in few sweeps over all edges, and the same is true for mesh coarsening. But, alternating refinement and coarsening can loop infinitely when the threshold value for cutting an edge into two, and those for removing an edge to create greater ones, are too close to each other. All the criteria governing the local operations must therefore be set in such a way that the overall process converges.

The node movement is a crucial tool in the current approach. In fact, while all other processes are discontinuous: one chooses to do something or nothing depending on whether a criterion is above a threshold value, node movement is a continuous adaptive process. It can thus perform surgical improvements after a discontinuous process.

After many tests, the following algorithm is thought to be the most appropriate:

1. Smooth the mesh after estimating the error by alternatively

   (a) moving all the nodes iteratively

   (b) swapping all the edges until convergence.

2. Adapt the mesh by iterating the following loop

   (a) refine all edges above a threshold error estimate, then move the nodes,

(b) remove all nodes whose edges have an error estimate below a threshold value, then move the nodes,

(c) swap the edges until convergence, then move the nodes.

3. Finally, smooth the mesh by repeating loop 1 before solving the equations again, starting from a reinterpolated solution on the new grid.

3.3 Coupling Mesh Adaptation to a Solver

The goal of an optimization approach is not only to perform a few operations on the mesh and get better results, but to converge the adaptive mesher and solver to an optimal solution, on an optimal mesh. This is done by coupling the solver with the mesher in the following cycle:

> Given (M_n, S_n), a mesh and a solution on this mesh at step n, the mesher produces a new mesh M_{n+1} and a solution $S_{n+1/2}$, the reinterpolation of S_n on M_{n+1}. A solution S_{n+1} on M_{n+1} is then obtained with the solver starting with $S_{n+1/2}$ as an initial guess. The iterations go over until convergence is reached.

A close coupling provides a maximum of flexibility to the mesh and permits to follow the evolution of the solution during the iterative resolution. This is done by frequent adaptation. In fact, it is useless to completely converge the nonlinear equations on an intermediate mesh, and better results are obtained when the mesh and the solution are made to converge in a coupled manner. Typically, about 100 adaptive steps have been used to reach a converged steady flow, starting from a uniform initial solution.

4 Evaluation of the Efficiency of the Proposed Adaptation Technique

The efficiency of the proposed adaptation technique is first demonstrated by a careful monitoring of the adapted mesh and solution evolution, at a single adaptation cycle and in an adaptation/solution loop.

Next, the independence of the final adapted mesh and solution from the starting mesh is established using numerical exemples.

Finally, it is demonstrated that using the proposed mesh adaptation approach, widely different schemes will practically get identical answers, showing the independence of the adaptation technique from any well-posed flow solver and the wide range of applicability of the method.

4.1 Convergence of an Adaptation Loop

A test case is presented for which a mesh is adapted for a viscous laminar flow around a NACA 0012 profile with a freestream Mach number of 2.0 and a Reynolds number of 10000 (Fig. 2). The optimal mesh has no resemblance to the initial one (Fig. 3), despite the fact that it is deduced from it using successive local alterations. The number of alterations is represented in Table 1 at each iteration of the loop and indicates that changes become negligible after 5 iterations.

With some convergence demonstrated, the question is to characterize the converged mesh. Figure 4 indicates that the edge error estimate is more equally distributed as adaptation proceeds, finally yielding a Gaussian distribution respecting the target error limits specified. The ratio of the maximum to minimum value of the error decreases from 5 000 to 3 and the standard deviation is reduced by two orders of magnitude.

Even if the convergence indicators are not monotonic (refinement results in a decrease of the minimum error and coarsening in an increase of the maximum) the algorithm of § 3.2 converges

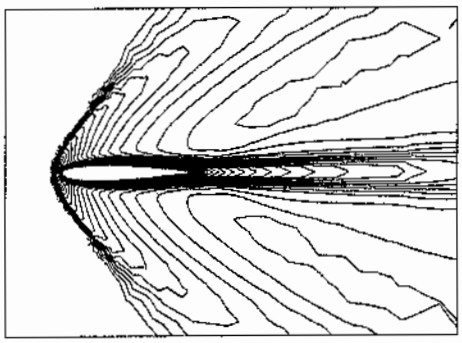


Figure 2: The error estimator is derived from second derivatives of the Mach number field shown here.

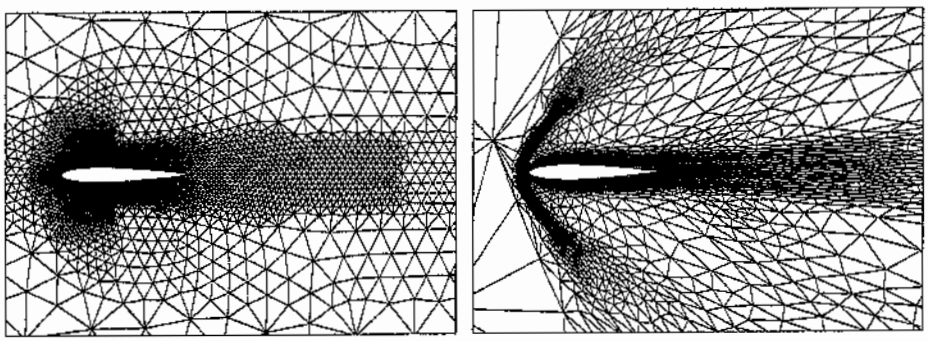


Figure 3: Initial mesh and final mesh at the end of an adaptation loop.

towards a mesh that can be considered adapted for the flow conditions at hand.

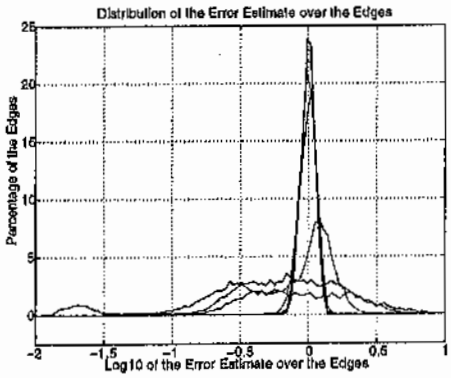


Figure 4: Distribution of the edge error over the mesh at different iterations of an adaptation loop.

Table 1: Convergence of local improvements during one adaptation loop.

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

4.2 Convergence of the Coupled Adaptation+Solution Cycle

We will consider the convergence of the total number of nodes on successive adapted meshes as an indication of the convergence of the coupled problem.

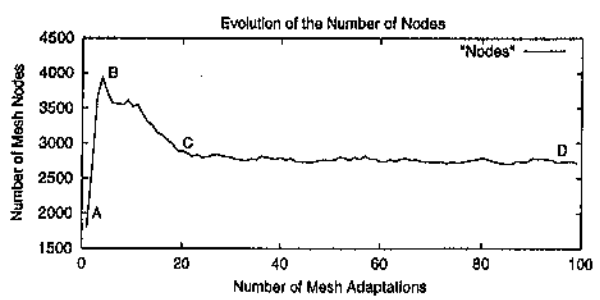


Figure 5: Total number of nodes versus adaptation+solution cycles for the flow over a NACA 0012 at $Ma = 2.0$ and $Re = 10\,000$.

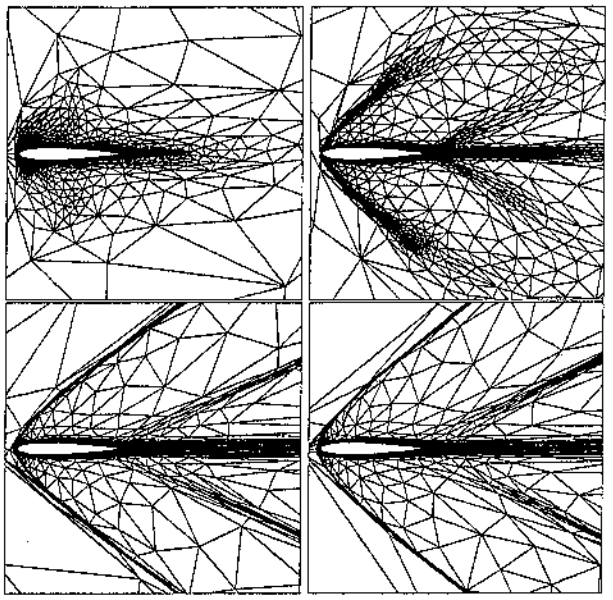


Figure 6: Adapted meshes after 1, 4, 20 and 100 mesher/solver cycles (left-right, top-bottom).

As in § 4.1 above, a viscous laminar flow around a NACA 0012 profile with a freestream Mach number of 2.0 and a Reynolds number of 10000 is used to illustrate this point, but starting with a different coarse initial mesh. 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 on Fig. 5 are presented on Fig. 6.

A surprising fact is a violent initial refinement, followed by a gradual decrease of the number of nodes to an asymptotic value. The few first meshes being not well adapted, the solutions are polluted with spurious oscillations. An over-refinement of the meshes results at the beginning in order to correct the solutions and define the salient features of the flow. When the solution improves and the main features are detected, the mesher reduces gradually the number of nodes, bringing to bear all the techniques presented above. This particular feature of the present approach makes it easier to start the solution using rapidly generated coarse grids.

4.3 Independence of Optimal Solution from Initial Mesh

It is desirable to verify whether the final adapted mesh is unique. The question asked is if for flow over, or in, a given geometry there is a unique grid that should be used for each freestream condition, and whether the current mesh optimizer can yield that grid. Intuitively, the answer is yes, as the scheme has been proven to converge for a single adaptation step as a post-processor and also when tightly coupled in a mesher-solver loop.

To demonstrate the uniqueness point conclusively, the same problem as the previous section is solved on three vastly different grids. The flow conditions are similar to the previous section but the Reynolds number is chosen as 10,000. In Fig. 7 left, a common grid is used, with refinement around the airfoil, embedded in a much coarser grid away from the airfoil. Figure 7 center shows a very coarse grid in which only 10 points are used to represent the airfoil. Finally, Figure 7 right shows an intentionally arbitrary and counter-intuitive mesh, with thousands of nodes in the upper half and only hundreds in the lower half, although the problem is symmetric. Each of the three figures has below it the corresponding result, using the finite volume code of B. Mohammadi [12, 13]. It is interesting to view in Fig. 7 bottom the improvement of the three initial solutions towards a very sharp and crisp final solution. As a matter of fact, the final solutions look the same simply because the final adapted meshes are all similar to the adapted one of Fig. 6. Figure 8 shows the initial and final distributions of the friction coefficient. The left side shows how the initial distributions are different, and incorrect. The right side of the figure shows how the three superposed results are indistinguishable. The percentages of edges having a given error (log 10 of error) is represented in Fig. 9. The initial three meshes have a very large error band. After adaptation, the three solutions give the same Gaussian error distribution, with a maximum error reduced 10 times from the initial one.

This example convincingly demonstrates that there is reason to hope that mesh-independent results could come soon and easily, as one can save considerable time by starting from arbitrary

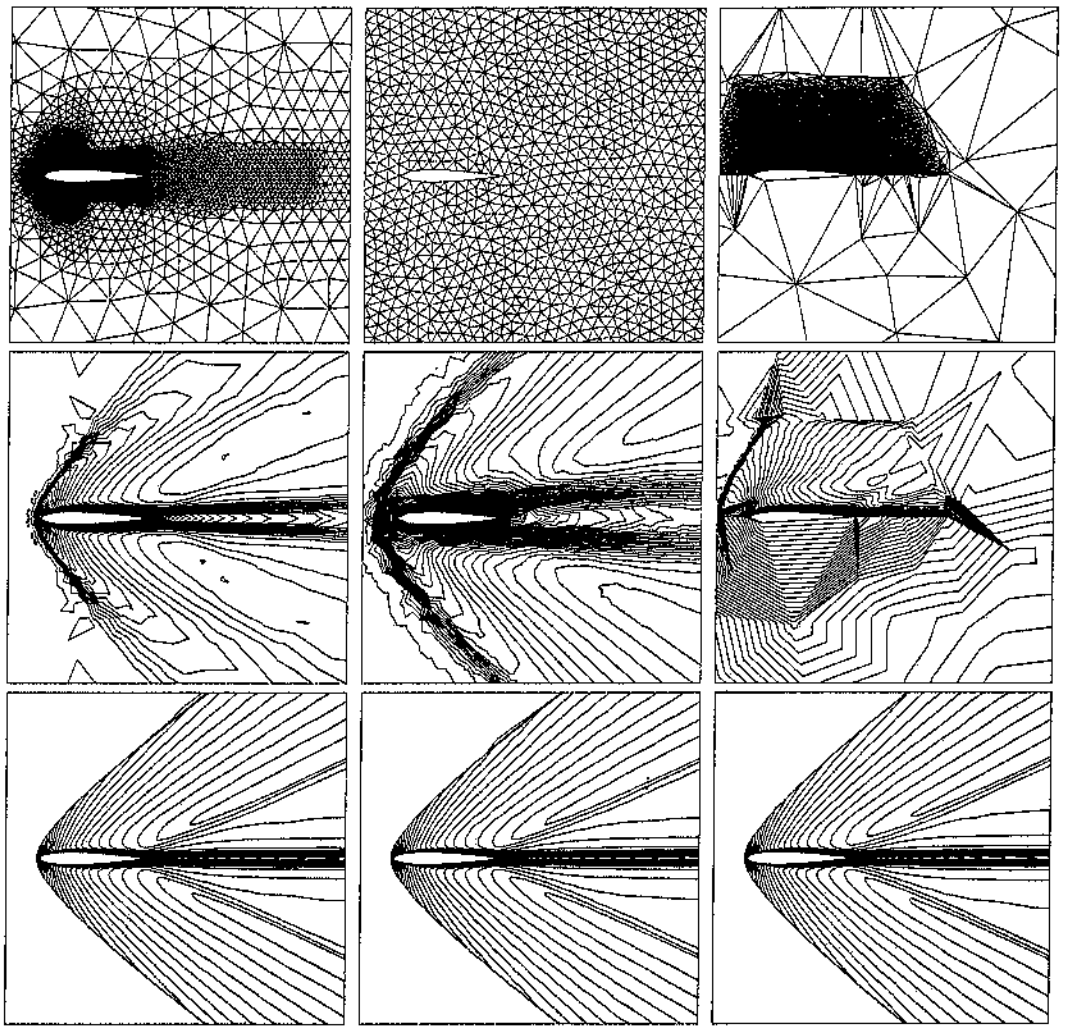


Figure 7: Initial meshes (top) and solutions (middle), and final optimal solutions (bottom) after 125 solver+adaptation cycles starting from the 3 initial grids (from left to right, fine, coarse and arbitrary grids)

meshes and let the adaptation procedure "generate" the correct and unique grid. This may also then lead to user-independent results as the gridding decisions are taken away from the user, who in the first place has no way of making an educated guess, let alone definitely know, what is the most appropriate grid for a given geometry at various flow conditions. It will also mean that results will be repeatable, as users wanting to duplicate some results have only to specify the error level desired, letting the unique grid corresponding to this error level be determined by the mesher.

4.4 Independence of Optimal Solution from Solver

A characteristic of the current remeshing-resolution strategy is the independence of the final result from the solver used i.e. different solvers give almost the same final result. One solver turn out to be more efficient in terms of precision and computing time, but the final meshes always have the same aspect and nearly the same number of nodes. More than that, the final solutions obtained with the different solvers are all identical, at least as demonstrated here for finite element computations of a laminar viscous flow. As a result, a good solution is more a question of meshers than of solvers.

To justify this claim, numerical solutions obtained with three different finite element solvers will be presented. All solvers are based on a primitive variables formulation of the Navier-Stokes equations. The three different mixed finite element solvers are labeled as follows

P1/(P1-iso-P2): linear elements for density and temperature, P1-iso-P2 elements for the velocity;

P1/P2: linear elements for density and temperature, quadratic elements for the velocity;

Conservative P1/P2: linear elements for temperature, quadratic elements for the velocity and (P1 + bubble) elements for density to ensure local conservation of mass, element by element. This particular formulation addresses the conservation claim of finite volume methods vs finite element methods.

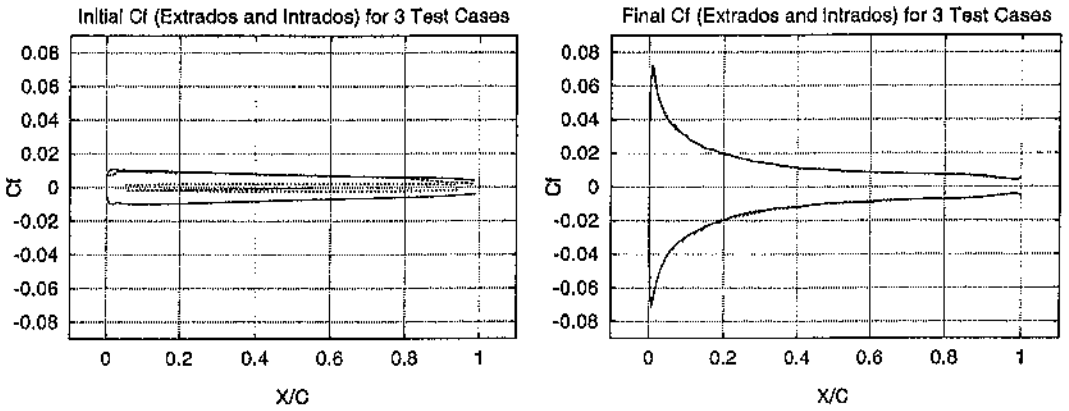


Figure 8: Different and "incorrect" C_f on initial meshes (left) and identical C_f on optimal meshes (right) for the three test cases

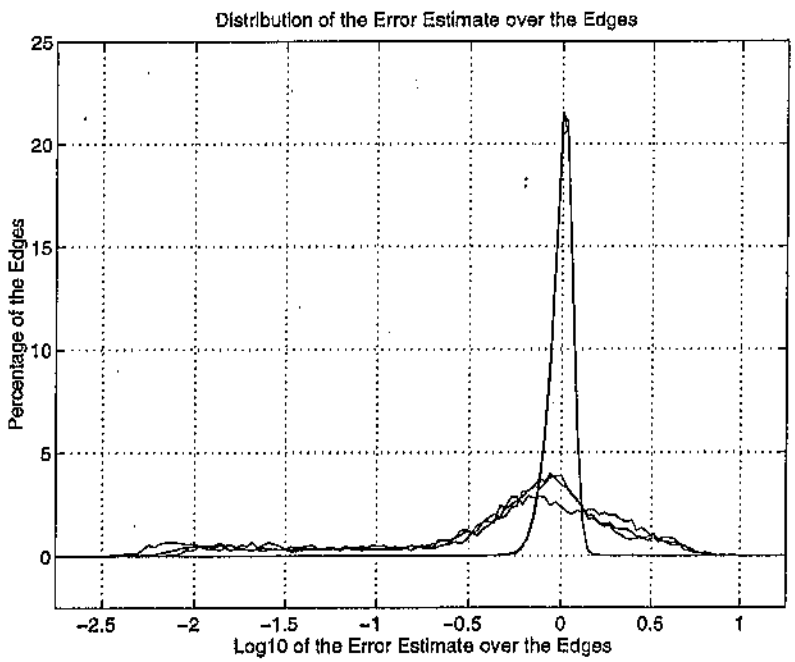
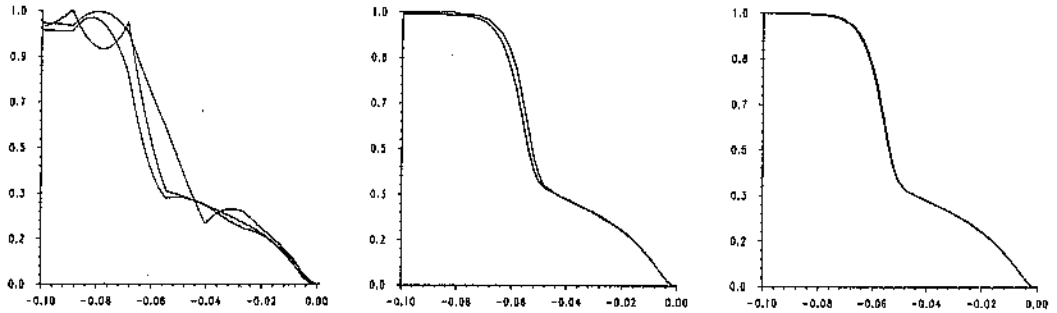


Figure 9: Percentages of the element edges versus error level for the three test cases, at the first and final mesh optimization steps

The use of mixed FEM dispenses with the addition of any extra artificial viscosity to stabilize the pressure at low Reynolds numbers or on adapted meshes. For more details on the solvers, see [5, 6, 7, 8, 9].



A B C

Figure 10: Cuts of the horizontal velocity component u over a NACA 0012 at $Ma = 2$ and $Re = 500$, given by three solvers: (A) on initial mesh,(B) after two remeshing steps and (C) after four remeshing steps.

A supersonic laminar flow at $Ma = 2$ over a NACA 0012 was used as a test case. Results are presented at $Re = 500$ because, at such a low Reynolds number, the shock is thick and the convergence behavior is easily seen. Figure 10 presents cuts of the horizontal velocity component in front of the airfoil leading edge before remeshing. The solutions are substantially different and it really is not obvious which one is best. After four remeshings, the viscous shock profiles completely agree. Note that while remeshing, the P1/P2 solver converges a bit faster to the correct shock position.

Another surprising fact is that mesh adaptation improves local conservation, as the solution of the locally conservative method agrees with those of the other schemes. Identical conclusions can be reached using vastly different finite element and finite volume formulations as well as for higher Reynolds numbers, but cannot all be covered here for the sake of brevity.

5 Conclusions

This paper presents a promising approach to the control of accuracy of CFD calculations and is a step towards certifiable CFD results. It points the way to the fact that not enough attention may have been paid to the use of proper grids when solving fluid flow problems and to the fact that the task of defining such proper grids is insurmountable unless an automatic way is formulated. In this paper this is done by using a generic directional mesh adaptation strategy, based on estimating the truncation error on mesh edges and coupling it to the flow solver to dynamically refine/coarsen/swap/move the mesh in such a way as to equate the lengths of all edges scaled in terms of second derivatives. This approach, unlike most, not only does not increase the number of mesh points disproportionately with mesh adaptation, but may result in optimal grids coarser than the initial one.

It is the conclusion of this paper that not only is accuracy controllable by using mesh adaptation, but that optimal (best and unique) grids can be automatically determined, over which most of the CFD panoply of stabilization artifices are either not needed or greatly diminished in importance. Interestingly, the current work also points to the fact that most well-posed stable numerical schemes, be they of first or second order, give practically the same answers on optimal grids, removing somewhat the thunder from under complicated high-accuracy schemes designed to fight a numerical evil that is no longer there once the optimal grid is used and giving hope for higher accuracy from simpler schemes. Of course, this question has to be addressed in more details.

The increased cost effectiveness should not be surprising if one makes the analogy to the integration of a function under a curve by Newton-Cotes and Gauss-Legendre methods. In the first approach, higher accuracy can only be achieved by an increase of the order of polynomial approximation, starting at linear, quadratic (Simpson), etc., quickly becoming cost-prohibitive. On the other hand, Gauss-Legendre integration asks the more intelligent question of how to select sample

points, with corresponding weights, to "exactly" integrate a polynomial of order 2n+1, with only n points.

Finally, a standing offer is made to the CFD community to use some of the optimal grids presented in this paper to recalculate some of their results, using their own codes, and verify the veracity of the present conclusions.

Acknowledgments

The authors would like to thank NSERC and FCAR for Operating and Strategic grants under which this work was supported and the CRM (Centre de Recherches Mathématiques de l'Université Montréal) for its partial support of a Post-Doctoral Fellowship to J. Dompierre. Thanks are also due to Dr. Sylvain Boivin for allowing the use of his P1/(P1-iso-P2) Navier-Stokes solver [5, 6, 7, 8] and to Dr. Bijan Mohammadi who has made accessible through the WWW his code NSC2KE [12, 13] developed at INRIA, France.

References

[1] D. Ait-Ali-Yahia, W. G. Habashi, A. Tam, M.-G. Vallet and M. Fortin. A directionally-adaptive finite element method for high-speed flows. In *32nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, number AIAA--96–2553, Lake Buena Vista, FL, July 1996.

[2] I. Babuška and W. C. Rheinboldt. Error estimates for adaptive finite element computation. *SIAM J. Num. Anal.*, 15(4):736–754, 1978.

[3] I. Babuška and W. C. Rheinboldt. A posteriori error analysis of finite element solution for one-dimensional problems. *SIAM J. Num. Anal.*, 18:565–589, 1981.

[4] I. Babuška, O. C. Zienkiewicz, J. Gago and A. de Oliveira, editors. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, London, 1986. John Wiley & Sons.

[5] S. Boivin. A numerical method for solving the compressible Navier-Stokes equation. *Impact Comp. Sc. Engng*, 1:64–92, 1989.

[6] S. Boivin and M. Fortin. A new artificial viscosity method for compressible viscous flow simulations by FEM. *Int. J. Comp. Fluid Dyn.*, 1:25–41, 1993.

[7] S. Boivin and M. Fortin. A nonisotropic artificial viscosity method: Application to the simulation of compressible viscous flows. *Int. J. Comp. Fluid Dyn.*, 1995.

[8] S. Boivin and M. Fortin. A class of predictor-corrector schemes: Application to the resolution of the compressible Navier-Stokes equations by FEM. *Int. J. Num. Meth. Fluids*, 1996.

[9] Y. Bourgault. *Méthodes des éléments finis en mécanique des fluides: Conservation et autres propriétés*. PhD thesis, Université Laval, Québec, Canada, 1996.

[10] E. F. D'Azevedo and R. B. Simpson. On optimal interpolation triangle incidences. *SIAM J. Sci. and Stat. Comp.*, 10(6):1063–1075, 1989.

[11] E. F. D'Azevedo and R. B. Simpson. On optimal triangular meshes for minimizing the gradient error. *Nümerische Mathematik*, 59(4):321–348, July 1991.

[12] B. Mohammadi. Fluid dynamics computation with NSC2KE, an user-guide, release 1.0. Technical Report RT-0164, Institut National de Recherche en Informatique et en Automatique, May 1994.

[13] B. Mohammadi and O. Pironneau. *Analysis of the K-Epsilon Turbulence Model*. Wiley & Sons and Masson, 1994.

[14] R. Verfürth. A posteriori error estimates for nonlinear problems. Finite element discretizations of elliptic equations. *Math. of Comp.*, 62(206):445–, April 1994.

[15] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. *J. Comp. Appl. Math.*, 50(1/3):67–, May 1994.