AIAA JOURNAL
Vol. 36, No. 5, May 1998

Certifiable Computational Fluid Dynamics Through Mesh Optimization

W. G. Habashi,* J. Dompierre,† and Y. Bourgault†
Concordia University, Montreal, Quebec H3G 1M8, Canada
M. Fortin‡
Université Laval, Ste-Foy, Quebec G1K 7P4, Canada
and
M.-G. Vallet§
National Research Council, Boucherville, Quebec J4B 6Y4, Canada

The accuracy and reliability of computational fluid dynamics (CFD) are addressed by proposing a novel, efficient, and generic mesh optimization approach. By using an appropriate directional error estimator, coupled with an effective mesh adaptation technique that is tied closely to the solver, it can be demonstrated that, for each flow condition and geometry combination, a controllable error level and an optimal mesh can be obtained. It is further demonstrated that such an optimal mesh can be reached from almost any reasonable initial grid and, more astonishingly, that the order of accuracy of well-posed numerical algorithms has a considerably reduced impact on solution accuracy if the mesh is well adapted. Thus, the proposed approach can be considered a first step toward user-, mesh-, and solver-independent, and thus certifiable, CFD.

I. Introduction

THE past 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 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 The conventional empirical way of the component/process design–building–testing cycle has been superseded almost altogether by ever more (accurate and reliable?) computer simulations, leading to analyses showing features and details that are difficult, expensive, or impossible to measure or visualize experimentally. 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, in a test cell, or under 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 in multidisciplinary contexts such as conjugate heat transfer (flow and heat transfer), aeroelasticity (flow and structures), aeroacoustics (flow and noise), aerothermodynamics (flow and combustion), aeroicing (flow and ice accumulation over lifting surfaces), and aeroelectromagnetics (stealth aircraft).

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 may not necessarily be a given that, for a specific physical situation, solving exactly the same set of equations, all codes (or even most of them) yield the same solution or even close answers. Thus, designers, users, and others have become increasingly critical in assessing solution quality, accuracy, and reliability.

Such discrepancies are due to several reasons that must be individually identified before one can hope to assess them, reduce them, or altogether eliminate them. Let us 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 is selected to best, or at least in the most cost-effective manner, simulate a given physical class of problems.

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 observations) and the numerical answers. We address this second level of error. After physical assumptions have been made and a set of partial differential equations (PDEs) with appropriate boundary conditions has been defined, the question to be posed is, Why can't one obtain a controllably and certifiably accurate answer to the solution of these PDEs? The errors made in solving that set, i.e., the difference to be anticipated between the exact solution of the PDEs and the numerical solution of the discretized equations, are generally due to the following culprits: 1) error resulting from the discretization of a continuum problem over a finite grid; 2) additional grid-related errors, i.e., arbitrary and hence inappropriate grid distribution and various geometric approximations; 3) computer round-off error; 4) deviation from reality by the addition of stability and convergence enhancers such as artificial viscosity, damping, smoothing, and upwinding; and 5) incomplete convergence of the code. (Surprisingly, it is rare to see convergence curves, to machine accuracy, in journal papers.)

In fact, rather than limit ourselves to rhetorically analyze the reasons for the lack of accuracy, we address what we believe are some of the steps needed to maintain a handle on the numerical error and control it. We specifically address the often-ignored issue of arbitrariness of mesh generation and how, by strategically adapting the mesh to the evolving solution, a mesh optimization approach can be shown to be the leading contender to overcome the CFD accuracy problem, by simultaneously 1) yielding solutions that are discretization-method-independent; 2) yielding solutions that are initial grid-independent, hence user-independent; and 3) drastically reducing, and in some cases altogether eliminating, the need for artificial stability and convergence enhancers.

Received Oct. 22, 1996; revision received Aug. 4, 1997; accepted for publication Aug. 25, 1997. Copyright © 1997 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Director, CFD Laboratory, Department of Mechanical Engineering. Associate Fellow AIAA.

†Research Associate, CFD Laboratory, Department of Mechanical Engineering.

‡Director, GIREF, Département de Mathématiques et de Statistique.

§Research Professional, Industrial Materials Institute.

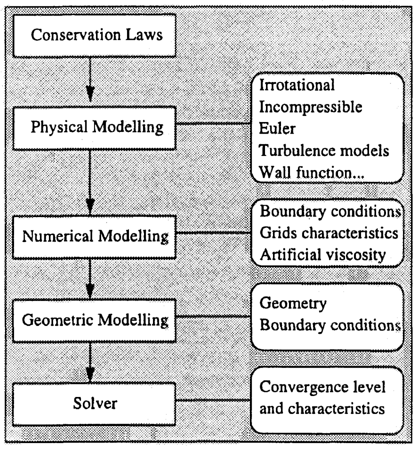


Fig. 1 Steps in constructing a CFD code.

Usually, the first step 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 is the use of fine grids as much as possible to resolve physical features such as boundary layers, flow separation, recirculation regions, and shocks. Taking this to a sometimes implausible limit, mesh-independence seekers use finer and finer meshes until the solution no longer changes.With most complex multiscale and multiphysics phenomena, attaining such mesh 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 meshes necessary to carry out CFD analyses are currently arbitrarily generated, reflecting only 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 meshes may have undetected geometric pathologies, such as skewness (a concept redefined in this paper) and degeneration (which may not considerably adversely affect some schemes), that introduce unwanted mesh-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 thus can be concluded that urgent emphasis should be placed on relieving the user from mesh decisions and that mesh improvements should be carried out through an automated adaptive process. Recent years thus have 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 (unknown) solution and alters either the mesh or the discretization to obtain a target precision level. Guided by an equidistribution principle that there is no justification for seeking a more accurate solution in some regions at the expense of others, adaptation efforts aim at producing a balanced error distribution or quality.

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

An alternative approach is proposed, which is to seek solutions on anisotropic meshes where equal resolution is achieved by a technique that improves results without necessarily increasing the overall number of mesh points, something that is hardly possible with standard isotropic adaptation techniques. 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 finite element method (FEM) and finite volume method (FVM) solvers also is studied and 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 computation through a wide range of Mach and Reynolds numbers.

II. Error Estimators for Mesh Adaptation

Generally, studies have aimed at determining an upper and a lower bound for the error based on an estimator that then is used to add or delete mesh points to equidistribute the error.[1–4] The basic idea underlying the proposed error estimator is the estimate

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

where $\Pi_h u$ denotes the interpolant of the solution in the finite element space, C is a constant depending only on the PDE solved, and $\|\cdot\|$ denotes a functional norm to measure the error. This inequality means that controlling the interpolation error will result in a control of the error on the (finite element, finite volume, finite difference, etc.) solution itself.

The problem of finding an optimal interpolant has been addressed by D'Azevedo and Simpson[5,6] for a piecewise linear finite element approximation. The case of higher-order interpolants has not received much attention, although the actual theory for linear elements has already been used to efficiently improve results for higher-order methods.[7] D'Azevedo and Simpson's premise is that the error on a piecewise linear interpolation is bounded by a quadratic term, as in a Taylor-series expansion. From Lagrangian interpolation, the error $E(x)$ at a point x is given by

$$|E(x)| = |[(x^2/2) - (xh/2)]g''| \le (h^2/8)|g''|$$

whenever $x = 0$ and 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, globally 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 two dimensions, the mesh that minimizes the interpolation error for a given number of nodes thus would yield equilateral triangles but with edge lengths measured using the second derivatives of the solution. A solution rapidly varying in one direction, such as across a shock wave, would have a much larger second derivative in that direction than the orthogonal one, smoothly and naturally introducing anisotropy in the length measure and in the mesh.

III. Mesh Adaptation Tools

The adaptation loop is explained briefly 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 is thought to be more cost-effective than regenerating the grid, especially during the late stages where there are dwindling differences between the current mesh and the previous one. The attempts in using these same edge-based error estimates for generating a new mesh[8,9] at each adaptation step require a larger time, and anisotropy may be harder to impose than through local mesh modifications.

The 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 then are computed on this mesh and kept unchanged during the 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 outcome is then an adapted mesh and a corresponding solution interpolated on it from the original one on the background mesh.

A. Local Improvements

All mesh modifications are made local and, for a triangular unstructured mesh, four local operations are used: 1) adding a node at midedge if the error is above the target one, 2) removing a node if the error is smaller than the target one on all surrounding edges, 3) moving a node to equidistribute the error on its neighboring edges, and 4) swapping the diagonal of the quadrilateral formed by any pair of adjacent triangles to more equidistribute the error estimator over the edges of neighboring elements.

In adapting a structured grid, the operations are restricted to node movement, so that the grid structure is preserved.[10] Because of the connectivity constraints in that case, the goal of having the edge error estimates equally distributed must be somewhat relaxed. The optimal grid then is defined as one with minimum energy, using a spring network analogy.

B. Adaptation-Loop Strategies

Particular attention has to be paid to the various 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. Alternating refinement and coarsening, however, 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 criteria governing the local operations therefore must be set in such a way that the overall process converges.

Nodal movement is thus a crucial tool in the current approach. In fact, whereas all other processes are discontinuous, with one choosing to act or not on the basis of a criterion, node movement is a continuous process. It thus performs surgical improvements after each discontinuous process.

After many tests, the following algorithm has been found to be the most appropriate:

1) Smooth the mesh after estimating the error by alternatively

a) moving all the nodes iteratively,

b) swapping all of the edges until convergence.

2) Adapt the mesh by iterating the following loop:

a) Refine all edges above a threshold error estimate, and then move the nodes.

b) Remove all nodes whose edges have an error estimate below a threshold value, and then move the nodes.

c) Swap the edges until convergence, and 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.

More details on each local improvement technique and the looping strategy can be found in Ref. 7.

C. Coupling Mesh Adaptation with the Solver

The goal of an optimization approach is not only to perform some operation on the mesh to get better results but to converge the adaptive mesher and solver to an optimal solution on an optimal mesh, in other words, to the solution with the lowest error on a mesh with a specified number of nodes. 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} then is 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 the following of 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 invariably better results are obtained when the mesh and the solution are made to converge in a coupled manner. Typically, about 20 adaptive steps have been used to reach a converged steady flow, starting from a uniform initial solution.

IV. Evaluation of Efficiency of 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 grid is established using numerical examples.

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

A. Convergence of an Adaptation Loop

First, the approach is demonstrated as a postprocessing step. A test case is presented for viscous laminar flow around a NACA 0012 airfoil with a freestream Mach number of 2.0 and a Reynolds number of 10^4 (Fig. 2, middle). The absolute error level demanded is 0.2. The adapted mesh (Fig. 2, right) has no resemblance to the initial one (Fig. 2, left), 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 five iterations.

With convergence demonstrated, the remaining task is to characterize the resulting mesh. Figure 3 indicates that the edge error estimate is more equally distributed as adaptation proceeds, finally yielding a Gaussian distribution respecting the specified target error limits. The ratio of the maximum to minimum value of the error decreases from 5000 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 results in an increase of the maximum), the algorithm of Sec. III.B converges toward a mesh that can be considered adapted for the flow conditions at hand.

B. Convergence of the Coupled Adaptation–Solution Cycle

We consider the convergence of the total number of nodes on successive adapted meshes to be, for all intents and purposes, an indication of the convergence of the coupled problem.

Table 1 Convergence of local improvements during one adaption loop

Refinement		Coarsening		Swapping	
No. of edges	%	No. of nodes	%	No. of 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

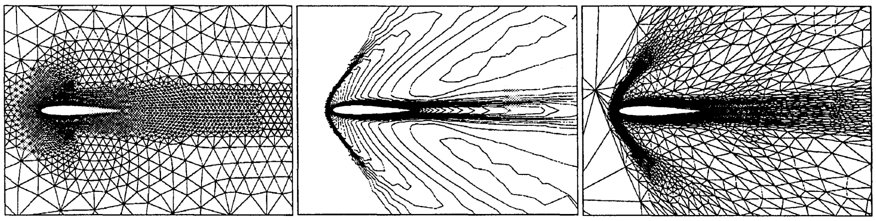


Fig. 2 Laminar flow around a NACA 0012 airfoil: left) initial mesh; middle) error estimator derived from second derivatives of Mach-number field computed on the initial mesh; right) final mesh at the end of an adaptation loop.

As in Sec. IV.A, a viscous laminar flow around a NACA 0012 profile with a freestream Mach number of 2.0 and a Reynolds number of 10^4 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 on Fig. 4 are presented in Fig. 5.

A surprising fact is that a vigorous initial refinement occurs, followed by a gradual decrease in the number of nodes to an asymptotic value. The few first meshes being poorly adapted, the solutions are polluted with spurious oscillations. An overrefinement of the meshes is called upon at the beginning to correct the solutions and detect the yet-smeared salient features of the flow. When the solution emerges and the main features have been detected, the mesher gradually reduces the number of nodes, bringing to bear all of the adaptation techniques cited earlier.

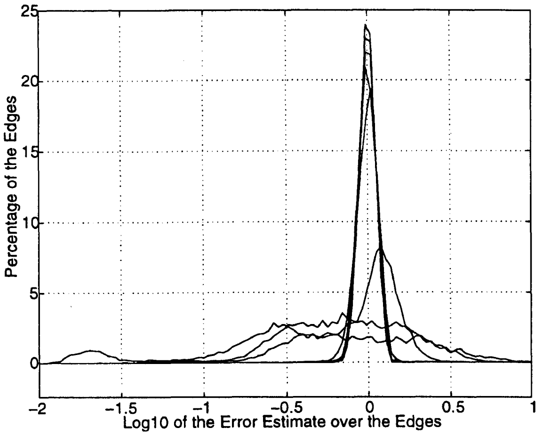


Fig. 3 Distribution of edge error over the mesh at different iterations of an adaptation loop.

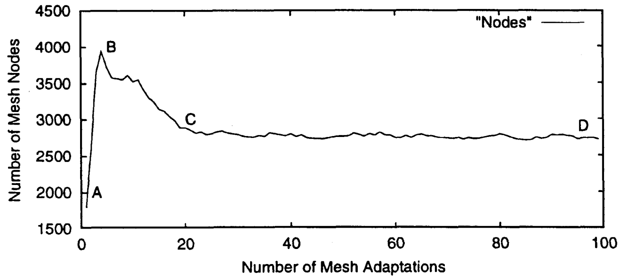


Fig. 4 Total number of nodes vs adaptation–solution cycles for flow over a NACA 0012 airfoil at M = 2.0 and Re = 10^4.

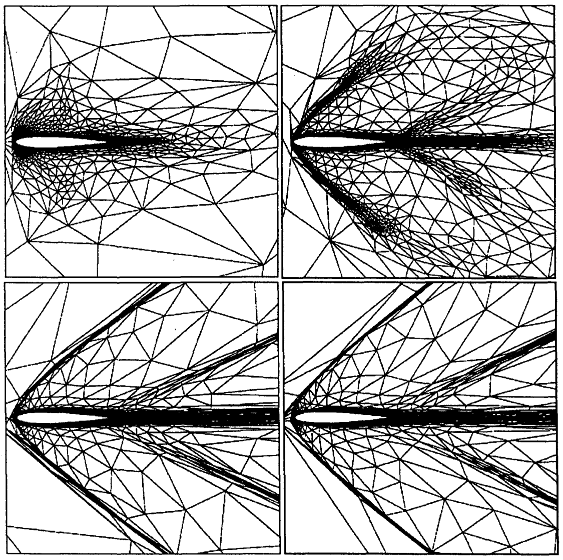


Fig. 5 Adapted meshes after 1, 4, 20, and 100 mesher–solver cycles (steps A, B, C, and D in Fig. 4).

This particular feature of the present approach makes it easier to start the solution using relatively coarse meshes, thereby shaving considerable time and effort from the mesh generation stage. Obviously, the solver may have to be adjusted, e.g., with extra artificial viscosity, to be robust enough to go through the initial difficult phases in which it is forced to solve on such coarse and sometimes inappropriate grids.

V. Validation of Proposed Adaptation Technique

To investigate the validity of the overall adaptation strategy, a simple and standard test case is used, i.e., the flow over a NACA 0012 airfoil at $M = 2$, $Re = 10^3$, and angle of attack (AOA) = 10 deg, namely test case A4 of the GAMM Workshop.[11] The temperature on the wall is constant and equal to $T_{\text{wall}} = T_0 = T_\infty [1 + (\gamma - 1)M^2/2]$, being the stagnation temperature. Unfortunately, T_∞ is not given in Ref. 11, but through the Sutherland law $\mu(T) = \mu_\infty(T/T_\infty)^{3/2}(T_\infty + 110)/(T + 110)$ it is expected that the value of T_∞ has only a small effect on the results. When T_∞ is increased from 50 to 293 K, the influence on the lift coefficient is found to affect only the third digit and the second digit for the drag coefficient. In the computations shown here, T_∞ is fixed as 83 K and T_{wall} as 149.4 K.

To study the convergence of the numerical solution to the exact solution while increasing the number of nodes, the target error criterion was slightly reduced every five mesher–solver steps. The effect of that manipulation was a slow increase in the number of nodes of the mesh from approximately 1200 to 7400 in 90 mesher–solver steps. Figure 6 shows the meshes and the corresponding Mach contours at steps 34, 64, and 90 with 1768, 3615, and 7397 nodes, respectively.

Figure 7 demonstrates that refining the mesh fourfold does not allow detection of any new features of the flow solution and that all features have been captured with the coarse mesh. Hence, in this particular case, adding nodes only increases the accuracy of the solution without qualitatively changing its character. The most accurate values obtained on the finer mesh were 0.34092 for the lift coefficient and 0.28299 for the drag coefficient. They compare well with the values of 0.3427 and 0.3388 for the lift coefficient and reasonably well with the values 0.2535 and 0.2515 for the drag coefficient given by Bristeau et al.[11]

Figure 8 shows a superposition of the C_p and C_f curves for the 1768-, 3615-, and 7397-node meshes. There is no noticeable change in the C_p curves, but the C_f curve is more sensitive to the quality of the mesh, especially for the extremum values. This confirms the general trend of the drag being much more difficult to predict than the lift.

Figure 9 shows the evolution of the lift coefficient as a function of the number of nodes. The most accurate value, 0.34092, is set as the exact value of the lift coefficient. The percentage of error is given by 100|(lift − 0.34092)/0.34092|. Figure 9 shows the log of the percentage of error on the lift coefficient as a function of the number of nodes. With approximately1000 nodes, the error is about 1%, as opposed to 0.1% and 0.01% for meshes with 4000 and 8000 nodes, respectively. This also means that, for a lift coefficient of 0.34092, the first three significant digits are accurate, the fourth may not be as accurate, and the fifth may be meaningless.

The comments about the lift coefficient also apply to the drag coefficient, as can be seen in Fig. 10. In the latter case, the percentage of error is given by 100|(drag − 0.28299)/0.28299|. It is also noticeable that the drag coefficient is about 10 times less accurate than the lift coefficient.

Up to this point, it has been shown that the mesher, the solver, and the mesher–solver coupling converge. The effectiveness and the accuracy of the approach also have been demonstrated. It remains to analyze the dependence of the final adapted mesh and solution on the initial mesh and on the flow solver used. These two points are investigated in Secs. V.A and V.B.

A. Independence of Optimal Solution from Initial Mesh

It is desirable to verify whether the final adapted mesh is unique. The fundamental question asked is whether for flow over, or in, a given geometry there is a unique mesh that should be used for each

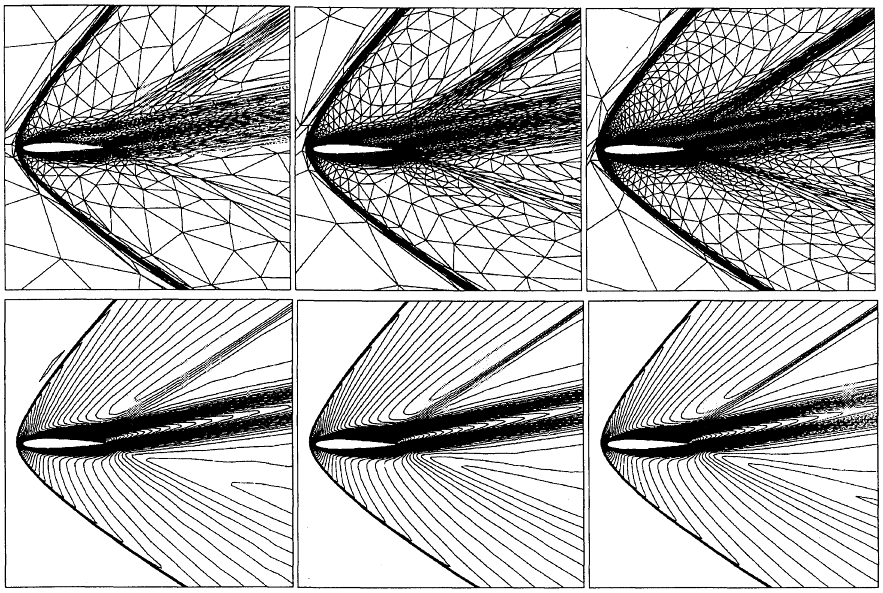


Fig. 6 Meshes with 1768 (left), 3615 (center), and 7397 (right) nodes and corresponding Mach contours for a viscous flow around a NACA 0012 at $M = 2, Re = 10^3$, and AOA = 10 deg.

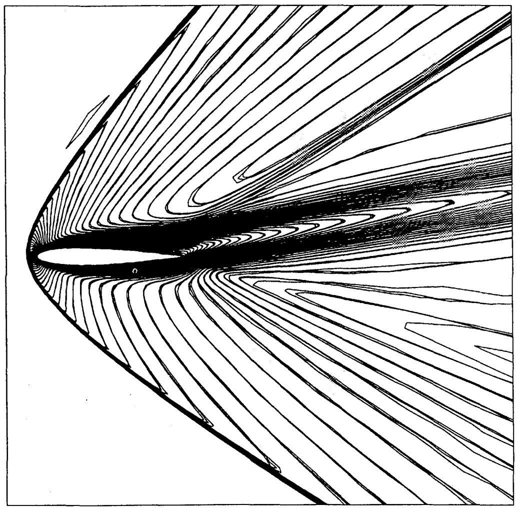


Fig. 7 Superimposition of the three solutions of Fig. 6: Mach contours with $\Delta M = 0.05$.

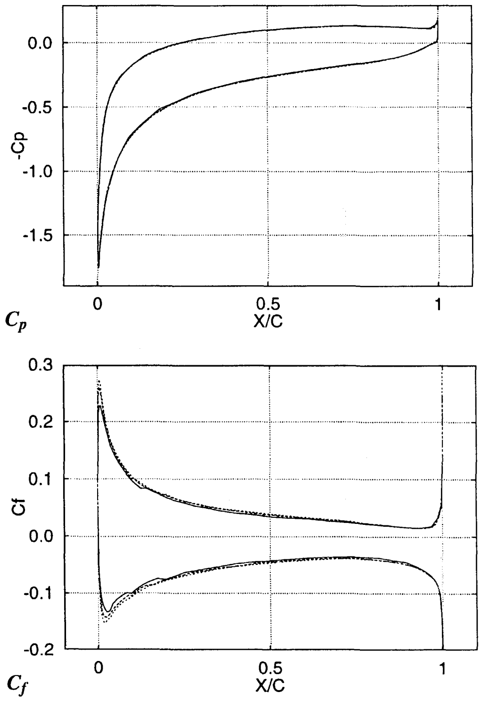
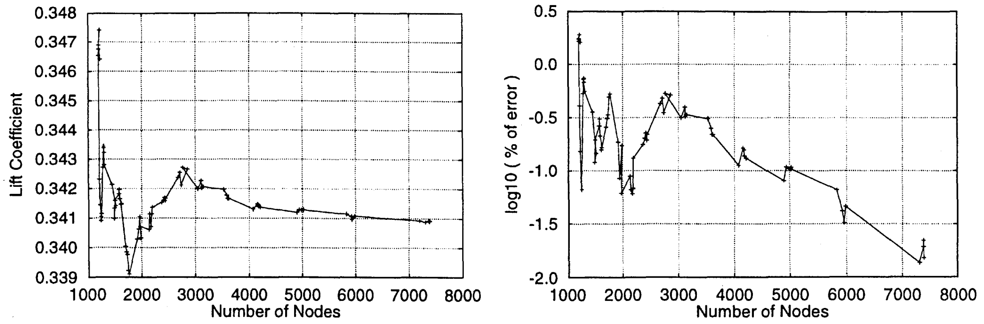


Fig. 8 Superposition of the C_p and C_f curves for meshes with 1768, 3615, and 7397 nodes.

freestream condition and whether the current mesh optimizer can yield that mesh. Intuitively, the answer is yes; the scheme has been proven to converge for a single adaptation step as a postprocessor and also when tightly coupled in a mesher–solver loop.

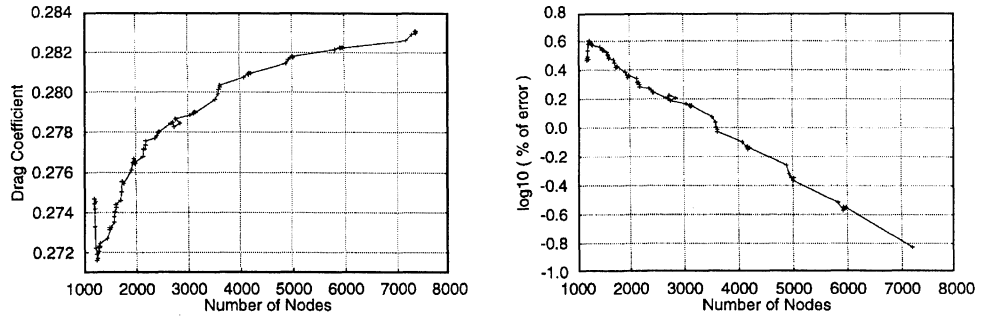
To demonstrate the uniqueness point conclusively, the same problem as in the preceding section is solved on three vastly different meshes. In Fig. 11 (left), a common mesh is used, with refinement around the airfoil, embedded in a much coarser mesh away from the airfoil. Figure 11 (center) shows a very coarse mesh in which only 18 points are used to represent the airfoil. Finally, Fig. 11 (right) shows an intentionally arbitrary and counterintuitive mesh, with thousands of nodes in the upper half and only dozens in the lower half, although the flow conditions are symmetric. The corresponding solutions, obtained using the finite volume code of Mohammadi[12] and Mohammadi and Pironneau,[13] are also shown. It is interesting to view in Fig. 11 (bottom) the improvement of the three initial very diffuse solutions to a very sharp and crisp final optimal one. In fact, the final solutions look identical simply because the final adapted meshes are all similar to the adapted one of Fig. 5. Figure 12 shows the initial and final distributions of the friction coefficient. The left-hand side shows how the initial distributions are different from one another and altogether incorrect. The right-hand side shows how the three superposed results are indistinguishable. The percentages of edges having a given error (log 10 of error) is represented in Fig. 13. 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, on a logarithmic scale, from the initial one.

This example convincingly demonstrates that there is reason to believe that mesh-independent results could come soon and easily because one can save considerable time by starting from a quickly generated coarse mesh and letting the coupled solver–adaptation procedure generate the correct and unique grid. This also may 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 making an educated guess, let alone of definitely knowing, what is the most appropriate mesh for a given geometry at various flow conditions. It also means that results can be made repeatable; users wanting to duplicate some results have only to specify the error level desired, letting the unique mesh corresponding to this error level be determined by the automatic mesher.



Evolution of lift coefficient with adaptation **Convergence of lift coefficient with adaptation**

Fig. 9 Convergence of lift coefficient as a function of the number of nodes.



Evolution of drag coefficient with adaptation **Convergence of drag coefficient with adaptation**

Fig. 10 Convergence of drag coefficient as a function of the number of nodes.

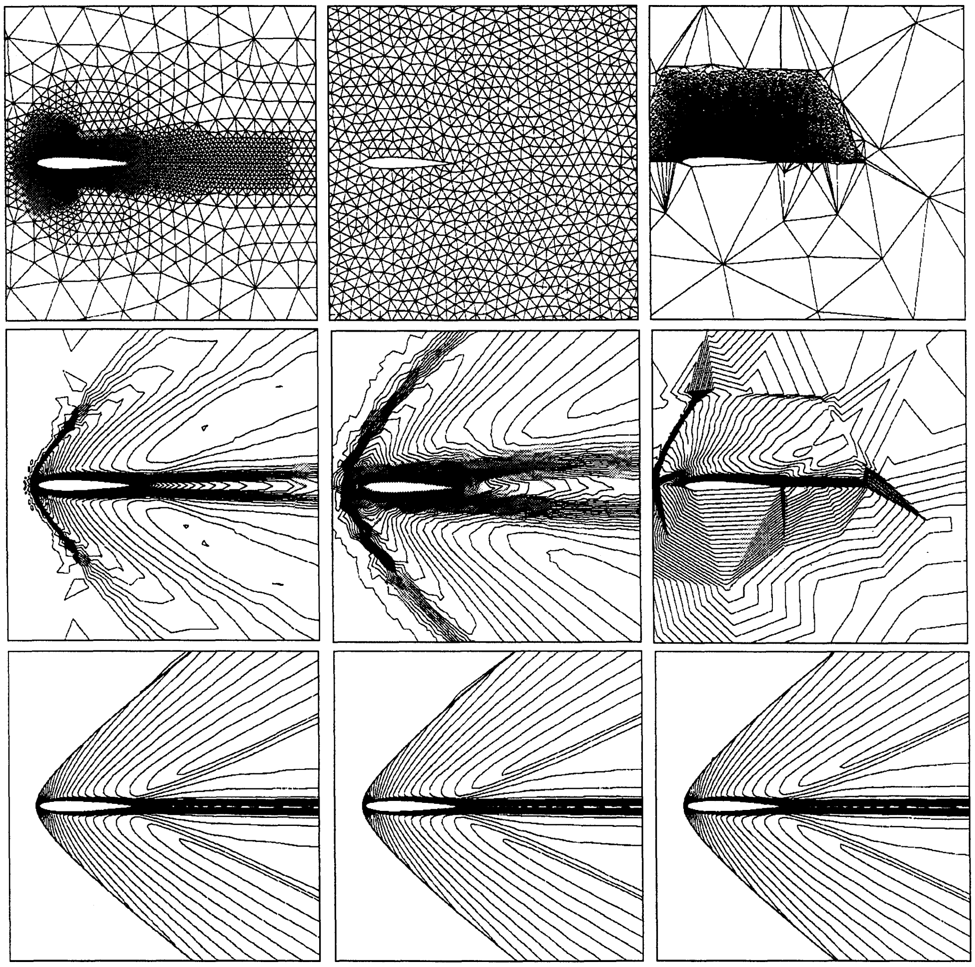
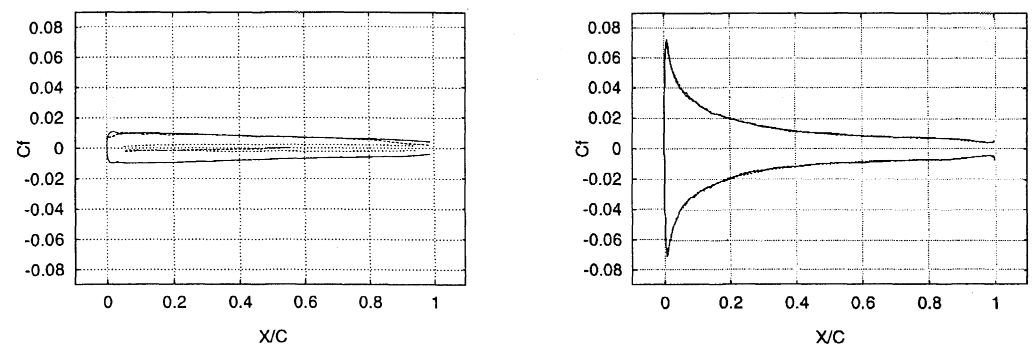


Fig. 11 Initial meshes (top), solutions (middle), and final optimal solutions (bottom) after 125 solver–adaptation cycles starting from the three initial meshes: $M = 2$ and $Re = 10^4$ (from left to right, fine, coarse, and arbitrary meshes).



Initial C_f (Extrados and Intrados) for three test cases

Final C_f (Extrados and Intrados) for three test cases

Fig. 12 Different and incorrect C_f on initial meshes (left) and identical C_f on optimal meshes (right) for the three test cases.

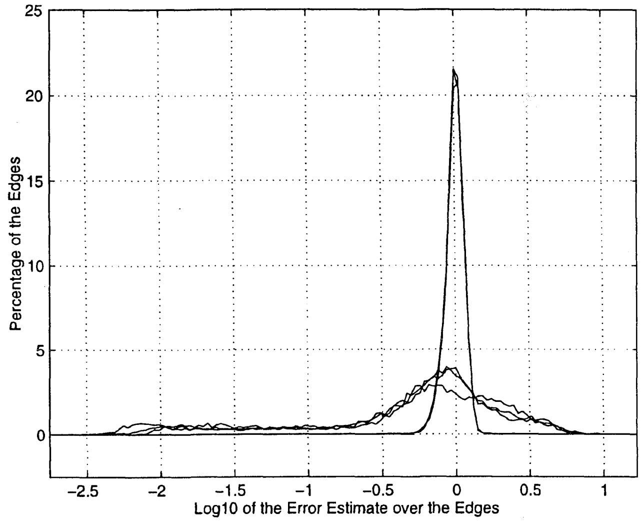


Fig. 13 Percentages of element edges vs error level for the three test cases at the first and final mesh optimization steps.

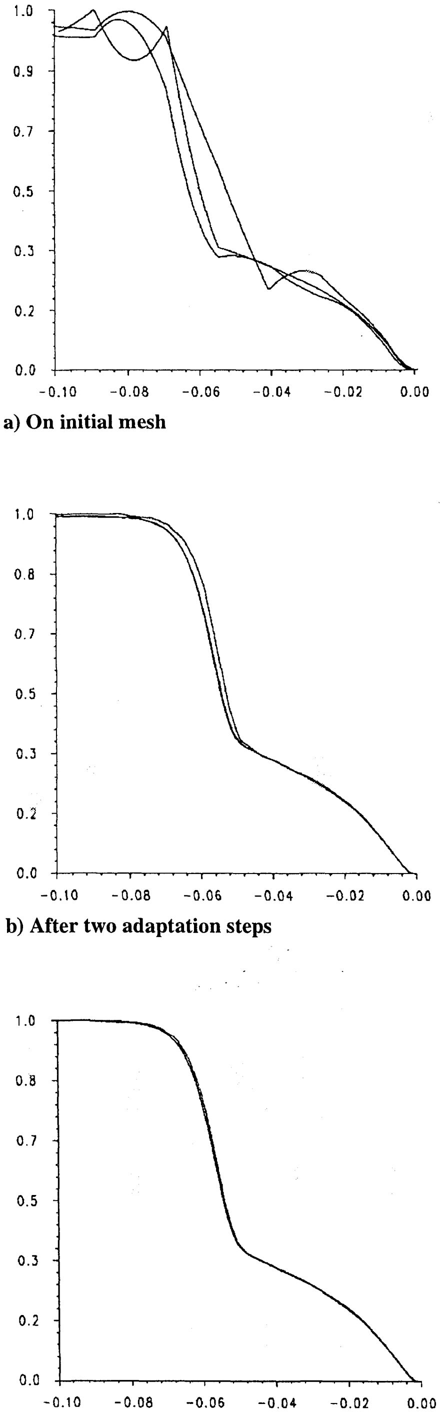
At this stage of the paper an obvious question emerges: Can the obtuse and skewed elements of volumes of the grids incommodate and blow up any solver? The answer is most definitely yes! Should such a distorted (optimal grid) be used directly in any solver, it may indeed blow up, indicating a very poor-quality grid. However, the blowup is due to poor matching of solver and grid. It is thus primordial to couple adaptation and solver from the start so that they develop in unison. With such an approach, most solvers will tolerate even the seemingly distorted anisotropic grid, which in effect consists of equilateral triangles or squares in the selected error norm. Thus the concept of grid skewness based on geometry, against which most CFD textbooks and many papers make dire warnings, is meaningless unless related to the equations being solved. It is no wonder, therefore, that very little quantification of these warnings has ever been published because the same apparent grid skewness could have vastly different devastating effects or none (if skewness is a result of mesh being adapted to solver), depending on whether one is solving the Euler, Stokes, Navier–Stokes, or perhaps the linear strain–stress equations.

B. Independence of Optimal Solution from Solver

It is desirable to verify the dependence, or independence, of the optimization scheme on the solver used. To this effect, numerical solutions obtained with three different finite element solvers are presented. All solvers are based on a primitive-variables formulation of the Navier–Stokes equations and originally were developed at a considerable effort to study the conservation aspect and also to bridge the gap between finite element and finite volume formulations in that respect. 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.



a) On initial mesh

b) After two adaptation steps

c) After four adaptation steps

Fig. 14 Cuts of the horizontal velocity component u over a NACA 0012 at $M = 2$ and $Re = 5 \times 10^2$, given by three solvers.

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.

For more details on the solvers, see Refs. 14–16.

A supersonic laminar flow at $M = 2$ over a NACA 0012 airfoil 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 tracked. Figure 14 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. During remeshing, the P1/P2 solver converges a bit faster to the correct shock position; after four remeshings, the viscous shock profiles of all methods completely agree.

Surprisingly, the use of mixed FEM seems to dispense with the addition of any extra artificial viscosity to stabilize the pressure at low Reynolds numbers on the original grids and at higher Reynolds numbers on the adapted meshes.

Another interesting fact is that mesh adaptation also improves local conservation because the solution of the locally conservative method agrees with those of the other schemes. It has been extensively demonstrated, using other examples, that identical solutions can be reached using vastly different finite element and finite volume formulations, but these cannot all be covered here for the sake of brevity.[7]

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 may turn out to be slightly more efficient in terms of temporary precision during adaptation and/or of overall computing time, but the final meshes always have the same statistical distribution 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 and in Ref. 7 for finite element and finite volume computations of laminar viscous flows. As a result, a good solution is a question more of meshers than of solvers.

VI. Conclusions

A promising approach to the control of accuracy of CFD calculations is presented, and it is a step toward certifiable CFD results. It points the way to the fact that not enough attention may have been paid to the use of proper meshes when solving fluid flow problems and to the fact that the task of defining such proper meshes is insurmountable unless an automatic way is formulated. 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 or coarsen and swap or move the mesh in such a way as to equate the lengths of all edges scaled in terms of second derivatives, as explained in Sec. II. This approach, unlike most, does not increase the number of mesh points disproportionately with mesh adaptation.

A criticism of this approach, as with any new technology that may go against the grain of common beliefs, is that it is limited (here) to laminar flows and to two-dimensional applications. However, these results are only a small aspect of a major joint project between Concordia and Laval Universities, where turbulent-flow and three-dimensional applications have already been made, with the result that the technology is already in use in demanding industrial situations. It was thought, however, that it would be more productive to limit the test cases demonstrated here to simple ones that can be reproduced by interested users or by those who may question such a technology. Thus difficulties that could be caused, for example, by the unexpected vagaries of different developers implementing the same turbulence model could be avoided.

It is the first conclusion that the concept of grid quality in terms of evaluating its skewness is meaningless if dissociated from the equations being solved, as well as from the flow conditions.

It is also the conclusion that it is time to stop worrying about evaluating the accuracy of results and to start using methods that can guarantee a specified level of accuracy. It has been demonstrated not only that accuracy is controllable by using mesh adaptation but that optimal (best and unique) meshes, over which most of the CFD panoply of stabilization artifices either are not needed or are greatly diminished in importance, can be automatically determined. 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 meshes. This somewhat removes the thunder from the increasingly complicated, high-accuracy schemes designed in the past decade to fight a numerical evil that vanishes once an optimal mesh is used and gives hope for obtaining higher accuracy from simpler schemes. Of course, this question has to be addressed in more detail.

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 be achieved only by an increase in the order of polynomial approximation, starting with linear, quadratic (Simpson), cubic, etc., and quickly becoming cost-prohibitive. On the other hand, Gauss–Legendre integration asks the more logical 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 meshes presented here to recalculate some of their results, using their own codes, and verify the veracity of the present conclusions. This invitation is addressed especially to the multitude of mesh-independent results presented in the literature and which now can be obtained, with more certifiable accuracy, on much coarser, intelligent grids. Cataloging such a database of simpler grids would give code developers a much more cost-effective way of testing their approaches and, even better, would isolate modeling limitations from numerical ones.

Acknowledgments

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada and Fonds pour la Formation des Chercheurs et l'Aide à la Recherche for Operating, Strategic, and Team grants under which this work was supported and the Centre de Recherches Mathématiques de l'Université Montréal for its partial support of a postdoctoral fellowship to J. Dompierre. Thanks are also due to Sylvain Boivin of the Université du Québec à Chicoutimi for allowing the use of his P1/(P1-iso-P2) Navier–Stokes solver and to Bijan Mohammadi, who has made accessible through the World Wide Web his NSC2KE code, developed at the Institut National de Recherche en Informatique et en Automatique, France.

References

[1] Babuška, I., and Rheinboldt, W. C., "Error Estimates for Adaptive Finite Element Computation," *SIAM Journal of Numerical Analysis*, Vol. 15, No. 4, 1978, pp. 736–754.

[2] Babuška, I., and Rheinboldt, W. C., "A Posteriori Error Analysis of Finite Element Solution for One-Dimensional Problems," *SIAM Journal of Numerical Analysis*, Vol. 18, 1981, pp. 565–589.

[3] Verfürth, R., "A Posteriori Error Estimates for Nonlinear Problems: Finite Element Discretizations of Elliptic Equations," *Mathematics of Computations*, Vol. 62, 1994, pp. 445–476.

[4] Verfürth, R., "A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques," *Journal of Computational and Applied Mathematics*, Vol. 50, 1994, pp. 67–86.

[5] D'Azevedo, E. F., and Simpson, R. B., "On Optimal Interpolation Triangle Incidences," *SIAM Journal on Scientific and Statistical Computing*, Vol. 10, No. 6, 1989, pp. 1063–1075.

[6] D'Azevedo, E. F., and Simpson, R. B., "On Optimal Triangular Meshes for Minimizing the Gradient Error," *Nümerische Mathematik*, Vol. 59, No. 4, 1991, pp. 321–348.

[7] Habashi, W. G., Fortin, M., Ait-Ali-Yahia, D., Boivin, S., Bourgault, Y., Dompierre, J., Robichaud, M. P., Tam, A., and Vallet, M.-G., "Anisotropic Mesh Optimization: Towards a Solver-Independent and Mesh-Independent CFD," *Computational Fluid Dynamics*, VKI Lecture Series 1996-06, von Kármán Inst. for Fluid Dynamics, Brussels, Belgium, 1996.

[8] Castro-Díaz, M. J., and Hecht, F., "Error Interpolation Minimization and Anisotropic Mesh Generation," *Ninth International Conference on Finite Elements in Fluids, New Trends and Applications* (Venice, Italy), edited by M. Morandi Cecchi, K. Morgan, J. Périaux, B. A. Schrefler, and O. C. Zienkiewicz, Inst. of Applied Computational Mechanics, 1995, pp. 1139–1148.

[9] Castro-Díaz, M. J., Hecht, F., and Mohammadi, B., "Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations," *4th International*

Meshing Roundtable (Albuquerque, NM), 1995, pp. 73–85.

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

[11] Bristeau, M. O., Glowinski, R., Mantel, B., Périaux, J., and Rogé, G., "Numerical Simulation of Compressible Navier–Stokes Flows," *A GAMM-Workshop, Notes on Numerical Fluid Mechanics 18*, Vieweg, Braunschweig, Germany, 1987.

[12] Mohammadi, B., "Fluid Dynamics Computation with NSC2KE, a User-Guide," rel. 1.0, Inst. National de Recherche en Informatique et en Automatique, TR RT-0164, Rocquencourt, France, May 1994.

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

[14] Boivin, S., "A Numerical Method for Solving the Compressible Navier–Stokes Equation," *Impact of Computing in Science and Engineering*, Vol. 1, 1989, pp. 64–92.

[15] Boivin, S., and Fortin, M., "A New Artificial Viscosity Method for Compressible Viscous Flow Simulations by FEM," *International Journal of Computational Fluid Dynamics*, Vol. 1, 1993, pp. 25–41.

[16] Bourgault, Y., "Méthodes des Éléments Finis en Mécanique des Fluides: Conservation et Autres Propriétés," Ph.D. Thesis, Dèpt. de Mathématiques et de Statistique, Univ. Laval, Québec, PQ, Canada, 1996.

J. Kallinderis
Associate Editor