Journal of Computational Physics **176,** 40–69 (2002)
doi:10.1006/jcph.2001.6967, available online at http://www.idealibrary.com on IDEAL®

Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows

David A. Venditti and David L. Darmofal

Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 37-442, Cambridge, Massachusetts 02139
E-mail: venditti@mit.edu

Received March 28, 2001; revised November 5, 2001

This paper presents an error estimation and grid adaptive strategy for estimating and reducing simulation errors in integral outputs (functionals) of partial differential equations. Adaptive criteria are derived using an adjoint-based error correction technique that relates the local residual errors of both the primal and adjoint solutions to the global error in the prescribed functional. This relationship allows local error contributions to be used as indicators in a grid adaptive strategy designed to produce specially tuned grids for accurately estimating the chosen functional. In this paper, attention is limited to two-dimensional inviscid flows using a standard finite volume discretization, although the procedure may be readily applied to other types of multidimensional problems and discretizations. Numerical examples demonstrate that the proposed adaptive procedure compares favorably in terms of accuracy, efficiency, and robustness relative to a commonly used gradient-based adaptive method.

Key Words: grid adaptation; error estimation; functional correction; adjoint; integral output.

1. INTRODUCTION

Computation simulations of complex aerodynamic flows have become integral components of the design process in the aerospace industry due to their potential to provide valuable physical data thereby reducing the need for costly wind tunnel testing. Increasing reliance on computer simulations necessitates a corresponding increase in the accuracy and reliability of the underlying algorithms. Furthermore, large-scale simulations must be made more affordable so that their potential benefits can be fully realized within the design cycle.

A well-known strategy for minimizing the cost of a computational simulation while achieving a given level of accuracy is grid adaptation [3–5, 7, 13, 17, 18, 20]. The basic

0021-9991/02 $35.00

premise is to locally enrich the computational grid in regions which most adversely affect the accuracy of the final solution while coarsening the grid in more benign regions to prevent incurring unnecessary computational costs. A major difficulty in achieving definite improvements using adaptation for Euler and Navier–Stokes calculations is the lack of a reliable error indicator. For example, a common strategy is to adapt to certain physical features of the flow, such as shock waves, boundary layers, wakes, slip lines, or stagnation points, by employing indicators based on large flow gradients [3, 5, 7, 13, 18]. The assumption here is that regions of larger gradients are associated with regions of larger error. Unfortunately, continuous local refinement of the dominant flow features does not necessarily guarantee that certain measures of the global error will simultaneously be reduced. In some cases this procedure may even lead to incorrect results [18].

Another concern is the choice of the error measure. Even within the setting of finite-element-discretized elliptic problems, where rigorous and robust error estimators have been available for decades [1, 16, 20], one can argue that a global error norm based directly on the solution and its derivaties may not be optimal within an engineering context. An alternate approach to making error estimation more relevant to engineering applications is to assess the error made in predicting an integral quantity representing an engineering output. For example, in aerodynamic applications, outputs of interest include the lift, drag, and moment coefficients on aircraft configurations. These are all expressible as surface integrals over portions of the domain boundary. Other examples include the total losses across a blade row in a gas turbine engine or the total heat flux to a turbine blade. In combustion modeling one is often concerned with the total production of nitrous oxides. When modeling soil contamination, a key quantity could be the net seepage of a pollutant into an aquifer. In electrochemical simulations of the behavior of sensors, the quantity of interest is the total current flowing into an electrode [14].

Within the context of finite element methods for fluid dynamics, error estimation and bounding procedures for functional outputs have been outlined by Becker and Rannacher [4], Larson and Barth [8], and Machiels *et al.* [10]. They are based on concepts already known from structural finite element methods [1, 2]. These procedures invoke the concept of duality, in which an equivalent dual (adjoint) formulation of the primal problem is exploited. The primary benefit of invoking the dual problem, in the context of grid adaptation, is that the error in a chosen functional can be directly related to local residual errors of the primal solution through the adjoint variables. This property elucidates the potential for devising optimal grid adaptive strategies designed to produce specially tuned grids for maximizing the accuracy of a particular functional.

Pierce and Giles [14] have developed an adjoint-based error correction technique for functional outputs. This technique extends superconvergence properties, automatically inherent in many finite element methods, to cover numerical results obtained from any numerical method, including finite difference, finite volume, or finite element without natural superconvergence. Moreover, the technique can also be used to improve the accuracy of superconvergent functionals obtained from finite element methods by constructing smoother, higher-order interpolants of the primal and dual solutions [14].

In this paper, a novel adjoint-based error correction and grid-adaptive methodology is presented for reducing simulation errors in specified functional outputs. The error correction procedure is a discrete analogue of the Pierce and Giles [14] technique. The correction terms and adaptive indicators used in this work have already been incorporated into an adaptive scheme for the quasi-one-dimensional Euler equations [17]. This paper focuses

on multidimensional applications of the same theory with particular emphasis on functional outputs of the two-dimensional Euler equations. To the authors' knowledge, this is the first application of adjoint-based correction and adaptation for a finite volume discretization of the Euler equations in multiple dimensions.

We begin by outlining the adjoint error correction procedure. The procedure involves estimating the error in the functional with respect to its value on a globally refined fine grid. Solutions on the fine grid are not required. The only auxiliary computations are functional and residual evaluations on the fine grid, and the solution of a linear adjoint problem on the original coarse grid. Next, we proceed to outline the adaptive procedure. The adaptive algorithm strives to improve the quality of the aforementioned error estimate by locally refining the coarse grid in select regions of the domain. At the next iteration, a new fine grid is defined with respect to the newly adapted coarse grid and the process is repeated until convergence. Convergence is defined in terms of local and global adaptation parameters and a prescribed error level for the computed functional.

Several numerical test cases are provided to demonstrate the effectiveness of the error estimation and adaptation procedure. These include subsonic and transonic flows (ranging from $M_\infty = 0.26$ to $M_\infty = 6.0$) around single-element and multiple-element airfoil configurations. The functionals chosen are those of primary interest in aeronautical applications: lift, drag, and moment coefficients. The proposed adaptive procedure is compared with a commonly used gradient-based method [3, 5, 18]. The proposed method is shown to compare favorably in terms of accuracy, efficiency, and robustness relative to the gradient-based method.

2. ERROR ESTIMATION PROCEDURE

In this section, we present a concise description of the adjoint error estimation and correction procedure used in this work. The impetus for this procedure is the correction technique of Pierce and Giles [14].

Consider a coarse grid, Ω_H, as being an affordable grid with respect to available computing resources and allowable solution times. The parameter, H, represents a characteristic length associated with the grid such as the average edge length in a finite element triangulation or the average grid spacing in a finite difference approximation. We are interested in estimating an integral quantity, $f(U)$, where U is the solution of the system of partial differential equations (PDEs) under consideration. The approximation of this integral on Ω_H using a prescribed quadrature rule is denoted by $f_H(U_H)$, where U_H is the corresponding discrete solution obtained on Ω_H using a specified discretization method.

Although $f_H(U_H)$ can be computed affordably, it may not be sufficiently accurate for the intended application. Now consider a fine grid, Ω_h, that may be prohibitively expensive for the purposes of estimating $f(U)$; however, if a discrete solution, U_h, were obtained on Ω_h, the corresponding discrete integral, $f_h(U_h)$, would be sufficiently accurate for our purposes. We require that Ω_h be uniformly embedded within Ω_H except at the boundaries where Ω_h is made to conform to the continuum boundary, $\partial\Omega$. For example, if Ω_H were a triangulation, one way of constructing Ω_h would be to subdivide each of the triangles of Ω_H into an integer number, N^2, of self-similar triangles where $N = (H/h)$. The fine-grid boundary nodes could then be repositioned to coincide with $\partial\Omega$. Figure 1 shows a patch of elements in a typical coarse grid and two possible fine grids corresponding to $N = 2$ and $N = 7$. Note

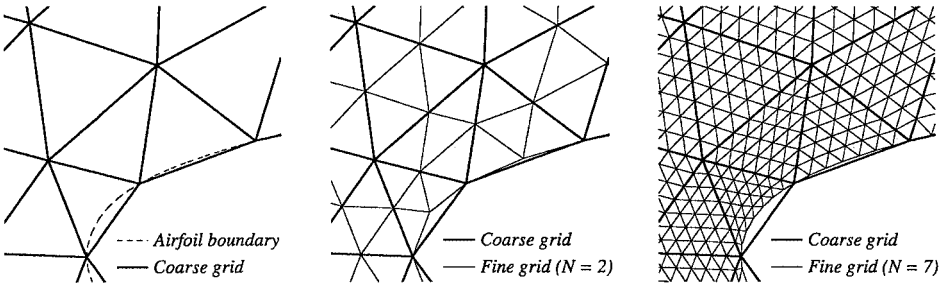

FIG. 1. A typical coarse grid and two possible fine grids that may be used in the error estimation procedure. Left: patch of coarse-grid elements near the airfoil boundary; center: superimposed fine grid corresponding to $N = 2$; right: superimposed fine grid corresponding to $N = 7$. Note that the fine grids conform to the original airfoil boundary.

that a hierarchy of fine grids corresponding to $N = 2, 3, \ldots$ could be constructed in this way and that each grid in the hierarchy would be completely characterized by N with respect to the original coarse grid.

If Ω_H were a three-dimensional grid composed of tetrahedra, a slightly different approach would be required since it is not possible to subdivide a tetrahedron into self-similar tetrahedra. Alternatively, a 1 : 12 refinement ratio can be achieved by adding nodes to the midpoints of each of the edges of the original tetrahedron and by adding one more node to its centroid. The application of this systematic refinement algorithm to each tetrahedron in the coarse grid would yield a legitimate fine grid for the purposes of the error estimation procedure. Recursive repetition of this process would result in a hierarchy of grids where each successive grid would be embedded within the previous one. As in the two-dimensional case, each fine grid should be made to conform to the original continuum boundary, $\partial\Omega$.

For the moment, let us consider a single fine grid, Ω_h. Our goal is to obtain an accurate estimate of $f_h(U_h)$ without ever solving on the fine grid. The point of departure in this analysis is an expansion of $f_h(U_h)$ about the coarse-grid solution

$$f_h(U_h) = f_h\left(U_h^H\right) + \left.\frac{\partial f_h}{\partial U_h}\right|_{U_h^H}\left(U_h - U_h^H\right) + \cdots. \tag{1}$$

The row vector, $\partial f_h/\partial U_h|_{U_h^H}$, contains the linear sensitivities of the fine-grid functional with respect to the fine-grid solution vector evaluated using the coarse-grid solution. The column vector, U_h^H, represents the coarse-grid solution expressed on the fine grid via some consistent reconstruction or projection operation, viz,

$$U_h^H \equiv I_h^H U_H. \tag{2}$$

An appropriate definition for I_h^H depends on the underlying discretization method [17]. In the case of a finite difference scheme, I_h^H could represent higher-order interpolation through the computed nodal values of the finite difference solution. In the case of a finite element discretization, U_h^H could be obtained from the coarse-grid basis functions directly or from a higher-order projection [14]. The projection operators used in this work are described in Section 4.5.

The nonlinear residual operator representing the system of equations arising from some discretization of the original partial differential equations on the fine grid is denoted by

$$R_h(U_h) = 0. \tag{3}$$

It is assumed here that $R_h(U_h)$ represents an integral statement such as one that would arise naturally from a finite element or finite volume formulation. A typical finite difference stencil would need to be scaled by an appropriate volume term (or area term in two dimensions) so that the residual became analogous to an integral expression.

Linearizing (3) about the coarse-grid solution yields

$$R_h(U_h) = R_h\left(U_h^H\right) + \left.\frac{\partial R_h}{\partial U_h}\right|_{U_h^H}\left(U_h - U_h^H\right) + \cdots, \tag{4}$$

where $\partial R_h/\partial U_h|_{U_h^H}$ is the Jacobian of the fine-grid system of equations evaluated using the projected coarse-grid solution. Symbolically, we can invert (4) (assumed nonsingular due to well-posedness) to obtain an approximation of the error vector,

$$\left(U_h - U_h^H\right) \approx -\left[\left.\frac{\partial R_h}{\partial U_h}\right|_{U_h^H}\right]^{-1} R_h\left(U_h^H\right). \tag{5}$$

Substituting (5) into (1) yields an estimate for the functional,

$$f_h(U_h) \approx f_h\left(U_h^H\right) - \left(\Psi_h|_{U_h^H}\right)^T R_h\left(U_h^H\right), \tag{6}$$

where $\Psi_h|_{U_h^H}$ is the discrete adjoint solution satisfying

$$\left[\left.\frac{\partial R_h}{\partial U_h}\right|_{U_h^H}\right]^T \Psi_h|_{U_h^H} = \left(\left.\frac{\partial f_h}{\partial U_h}\right|_{U_h^H}\right)^T. \tag{7}$$

Equation (6) is an estimate of $f_h(U_h)$. It is exact for linear residuals and functionals. To compute this estimate would require the solution of the adjoint problem on the fine grid, which is undesirable. Instead, $\Psi_h|_{U_h^H}$ is replaced by the interpolated coarse-grid adjoint,

$$\Psi_h^H \equiv J_h^H \Psi_H, \tag{8}$$

where J_h^H is a projection operator analogous to I_h^H in (2). The coarse-grid adjoint, Ψ_H, is obtained from the solution of the discrete adjoint equations on the coarse grid,

$$\left[\frac{\partial R_H}{\partial U_H}\right]^T \Psi_H = \left(\frac{\partial f_H}{\partial U_H}\right)^T. \tag{9}$$

The projection operation in (8) requires that the discrete adjoint Ψ_H be consistent with the true analytical adjoint. This may have implications regarding the imposition of certain boundary conditions in the definition of the primal residual operator [16].

Substituting the interpolated coarse-grid adjoint into (6) yields the final, computable estimate as

$$\tilde{f}_h(U_H) = f_h\left(U_h^H\right) - \left(\Psi_h^H\right)^T R_h\left(U_h^H\right). \tag{10}$$

The coarse grid, Ω_H, should be fine enough to capture the essential features of the problem under consideration (that is, the corresponding discrete solution, U_H, should be within the asymptotic range) in order for the truncated expansions, (1) and (4), to be uniformly valid.

It should be noted that the correction formula given by (10) is a discrete analogue of the Pierce and Giles correction term [14]. The current derivation utilizes the notion of a fine grid and appeals directly to the nonlinear, discrete residual equations, whereas the Pierce and Giles derivation is cast in a continuous framework. In the continuous approach, the vector inner product of the adjoint and the discrete residual in (10) is replaced by an integral inner product of the adjoint and the residual of the governing partial differential equations using approximate adjoint and primal solutions. Both approaches have been shown to produce superconvergent functional estimates. Further discussion comparing the two approaches, including implementation issues, is provided in reference [17].

3. ADAPTATION PROCEDURE

In this section, an adaptive strategy is proposed that is designed to improve the accuracy of the computable error estimate in (10). This is in contrast to other adaptive schemes that attempt to optimize the computational grid with respect to maximizing the accuracy of the functional directly [4]. One variant of the latter approach could be based exclusively on (10), where it is evident that the error in the functional can be expressed as a weighted sum of the local residual errors with the adjoint variables as the weighting functions. These local error contributions could be used as indicators in a grid adaptive strategy designed to yield nearly optimal grids for computing the chosen functional. Unfortunately, this approach could lead to erroneous requests to the grid generator for refinement and/or coarsening in regions where the adjoint solution is not sufficiently resolved. We wish to reduce this risk by deriving a more conservative criterion for adaptation that is based on both the primal and adjoint residual errors. The proposed adaptive strategy involves reducing and equidistributing the value of an adaptation parameter throughout the computational domain. We will demonstrate that reducing the proposed adaptation parameter leads to improvements in the quality of the error estimates. In practice, it also leads to improvements in the base value of the functional before correction.

3.1. Error in the Computable Correction

We begin by examining the remaining error in the functional after correction. Equation (6) can be decomposed in the following manner,

$$f_h\left(U_h^H\right) - f_h(U_h) \approx \underbrace{\left(\Psi_h^H\right)^T R_h\left(U_h^H\right)}_{\text{Computable correction}} + \underbrace{\left(\Psi_h|_{U_h^H} - \Psi_h^H\right)^T R_h\left(U_h^H\right)}_{\text{Error in computable correction}}. \tag{11}$$

Alternatively, (11) may be recast as

$$f_h\left(U_h^H\right) - f_h(U_h) \approx \underbrace{\left(\Psi_h^H\right)^T R_h\left(U_h^H\right)}_{\text{Computable correction}} - \underbrace{\left\{R_h^\Psi\left(\Psi_h^H\right)\right\}^T \left[\left.\frac{\partial R_h}{\partial U_h}\right|_{U_h^H}\right]^{-1} R_h\left(U_h^H\right)}_{\text{Error in computable correction}}, \tag{12}$$

where R_h^Ψ is the adjoint residual operator defined by,

$$R_h^\Psi(\Psi) \equiv \left[\left.\frac{\partial R_h}{\partial U_h}\right|_{U_h^H}\right]^T \Psi - \left(\left.\frac{\partial f_h}{\partial U_h}\right|_{U_h^H}\right)^T. \tag{13}$$

A third form of the error in the computable correction (ECC), which is dual to that given in (11), can be obtained using (4) and (12). Specifically,

$$f_h\left(U_h^H\right) - f_h(U_h) \approx \underbrace{\left(\Psi_h^H\right)^T R_h\left(U_h^H\right)}_{\text{Computable correction}} + \underbrace{\left\{R_h^\Psi\left(\Psi_h^H\right)\right\}^T\left(U_h - U_h^H\right)}_{\text{Error in computable correction}}. \tag{14}$$

The proposed adaptive strategy is based on reducing the ECC, thereby improving the accuracy of the computable correction. It is evident from (11), (12), and (14) that the ECC can be written in at least three different forms. According to (12), reducing the local residual errors in both the primal and adjoint solutions simultaneously would lead to a reduction in the ECC. Adapting on both residuals seems advantageous with respect to the robustness of the procedure. In using this form, however, one must address the issue of how to decrease the two residuals simultaneously during the adaptive procedure. In general, the units of the primal and adjoint residuals will be different and their magnitudes could vary significantly. A viable adaptive scheme must ultimately combine the two residuals into a single adaptation parameter for each cell or element in the grid. For the purposes of adaptation, (11) and (14) provide more convenient forms of the ECC from which to work with. In particular, the ECC can be expressed as the inner product of the adjoint solution-error and the primal residual-error, (11), or as the inner product of the adjoint residual-error and the primal solution-error, (14). When nonlinear terms are neglected, these two inner products are equal. The magnitudes of their corresponding components are comparable and their units are equal to those of the functional, $f(U)$. This illustrates the duality between the primal and adjoint residual operators. If nonlinear effects are accounted for, a duality gap, $D \neq 0$, will exist between the two inner products. By retaining nonlinear terms in (4), one can obtain an expression for the duality gap,

$$\begin{aligned} D &\equiv \left(\Psi_h|_{U_h^H} - \Psi_h^H\right)^T R_h\left(U_h^H\right) - \left\{R_h^\Psi\left(\Psi_h^H\right)\right\}^T\left(U_h - U_h^H\right), \\ &= -\left(\Psi_h|_{U_h^H} - \Psi_h^H\right)^T W, \end{aligned} \tag{15}$$

where W is a vector containing quadratic forms of the primal error. An explicit expression for W is given in reference [17]. The proposed adaptive procedure is based on reducing and equidistributing the magnitudes of the components of each of the inner products on the right-hand side of (15). In addition to improving the quality of the computable correction, this will lead to a reduction in the magnitude of the duality gap, and hence, a reduction in the nonlinear contribution to the functional error.

The form of the ECC shown in (11) is essentially what is used by Becker and Rannacher [14] in their output-based adaptive strategy. The addition of the dual ECC term, shown in (14), in the current approach is a natural way of incorporating the adjoint residual error into the adaptive scheme. It is expected that utilizing information from both the primal and adjoint residuals will lead to a more robust adaptive algorithm.

3.2. Adaptive Criteria

The philosophy adopted for the current adaptive framework is to allow the user the freedom to specify control parameters that are most relevant within a practical engineering design context. One or more of these parameters will ultimately define the termination criteria for the adaptive algorithm. The parameters that may be prescribed *a priori* in the current strategy include: (1) a desired remaining error level for the functional of interest; (2) an upper limit on the total number of degrees of freedom in the simulation; (3) an upper limit on the total computing time; and (4) an upper limit on the total number of adaptive iterations. One additional parameter is required that is not directly related to the aforementioned needs of the user: a lower bound on the local element size due to machine round-off issues.

In the proposed adaptive strategy, we seek to reduce and equidistribute the value of an adaptation parameter throughout the computational domain while simultaneously monitoring and reducing an upper bound on the estimated error in the functional of interest. Consider the operation of computing an inner product over the fine grid, Ω_h, embedded within Ω_H. For each coarse-grid element, k, there are N^2 (in two dimensions) fine-grid elements over which a partial inner product must be computed. For each fine-grid node, $l(k)$, within element k, there are four subcomponents (for the two-dimensional Euler equations) to the primal and adjoint residual vectors corresponding to the mass, x-momentum, y-momentum, and energy conservation equations. Equation (15) suggests the following definition for the adaptation parameter, ε_k, at element k:

$$\varepsilon_k = \frac{1}{2}\sum_{l(k)} \Big\{ \Big| \big[Q_h^H \Psi_H - L_h^H \Psi_H\big]_{l(k)}^T \big[R_h\big(L_h^H U_H\big)\big]_{l(k)} \Big| + \Big| \big[Q_h^H U_H - L_h^H U_H\big]_{l(k)}^T \big[R_h^\Psi\big(L_h^H \Psi_H\big)\big]_{l(k)} \Big| \Big\}. \tag{16}$$

In this last expression, a term of the form, $[V_h]_{l(k)}$, for some generic vector, V_h, on Ω_h, refers to the 4×1 subvector (component) of V_h corresponding to the fine-grid node, $l(k)$, within the coarse-grid element, k. The summation in (16) is over all fine-grid nodes within the kth coarse element. For those nodes on the boundary of k, a partial contribution may be implied depending on how the nodal residual is defined for the particular discretization. For example, in the case of a vertex-based finite volume discretization, the fractional contribution from a node on the boundary of element k is proportional to the fraction of the associated control volume within that element. The operators L_h^H and Q_h^H are interpolation operators that map coarse-grid vectors onto the fine grid via linear and quadratic interpolation, respectively (see Section 4.5).

Let ε denote the summation of ε_k over all elements in Ω_H. That is,

$$\varepsilon = \sum_k \varepsilon_k. \tag{17}$$

It is evident from (11) and (14) that if the absolute values signs had been absent in (16), the summation of the corresponding elemental quantities over Ω_H would have yielded an estimate of the remaining error in the predicted functional after correction. Thus, ε, as presently defined, represents an upper bound on this error estimate. In light of this, a global adaptation parameter can be defined as

$$\eta_g = \frac{\varepsilon}{e_o}, \tag{18}$$

where e_o is a user-specified desired error level. The global error criterion is satisfied if $\eta_g \leq 1$ while further refinement is required if $\eta_g > 1$. Using this measure alone would lead to uniform grid refinement only. To complete the h-refinement strategy a local adaptation parameter must also be defined. An appropriate local error measure can be determined by invoking the principle of error equidistribution [3, 19]. In particular, an attempt is made to equidistribute ε_k over all elements in the domain. In this vein, a local adaptation parameter is defined as

$$\eta_k = \frac{\varepsilon_k}{\bar{e}_o}, \tag{19}$$

where $\bar{e}_o = e_o / N_e$ is the target error for each element and N_e is the total number of elements in the current grid, Ω_H. The local criterion is satisfied if $\eta_k \leq 1$ while further refinement of element k is indicated if $\eta_k > 1$. Finally, at each adaptive iteration, a new desired element size, $\tilde{H}_k$ is computed from the old one, H_k, according to

$$\tilde{H}_k = H_k \left(\frac{1}{\eta_g \eta_k} \right)^{\omega}. \tag{20}$$

The parameter ω controls how aggressively each subsequent grid is refined during the iterative adaptive process. An inappropriately large choice for ω would lead to over-refinement, resulting in a suboptimal final grid with more elements than is necessary for the prescribed level of error. Conversely, an inappropriately small value would lead to under-refinement, which would prolong the adaptive process by increasing the total number of adaptive iterations to convergence. An appropriate value for ω can be deduced by examining the asymptotic convergence rates of the global and local adaptation parameters. Numerical results indicate (see Section 5.2) that $\eta_g \sim \eta_k \sim \mathcal{O}(\bar{H}^p)$ with $2 \leq p \leq 3$, where $\bar{H}$ is an average element size in the grid. These results were obtained by monitoring the convergence of η_g and η_k on a series of uniformly refined grids for a subsonic test case without geometric irregularities. A heuristic for determining ω is to render the new element size independent of its current size; that is, to choose ω such that $\tilde{H}_k \sim \mathcal{O}(1)$. In the present case we assumed a convergence of $\eta_g \eta_k \sim \mathcal{O}(H_k^4)$ yielding a heuristic value of $\omega = 1/4$, which was used in all the output-based adaptive simulations in this paper. This simple convergence rule is not valid near singularities, discontinuities, or geometric irregularities. Nevertheless, the adaptive procedure exhibited robust grid-convergence, usually terminating after three or four adaptive iterations. The algorithm used to refine the grid is described in Section 4.6.

4. NUMERICAL IMPLEMENTATION

The proposed error estimation and grid adaptive procedure is implemented using a standard, unstructured grid, finite volume discretization of the two-dimensional Euler equations.

4.1. Governing Equations

The time-dependent Euler equations are expressions of the principles of mass, momentum, and energy conservation for an inviscid, compressible gas. In two dimensions these equations may be written as

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{21}$$

where U, F, and G are given by

$$U = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{Bmatrix}, \quad F = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \end{Bmatrix}, \quad G = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \end{Bmatrix}. \tag{22}$$

In these expressions, ρ is the mass density; u and v are, respectively, the x- and y-components of the gas velocity; p is the static pressure, and E is the total energy. The system is closed with the equation of state for an ideal gas,

$$p = \rho(\gamma - 1)\left(E - \frac{u^2 + v^2}{2}\right), \tag{23}$$

where γ is the ratio of specific heats.

4.2. Flow Solver

The base solver used for the simulations in this investigation is FUN2D: a fully implicit, unstructured grid, vertex-based, finite volume code developed by Dr. W. K. Anderson, formerly of NASA Langley. The Osher upwind scheme [6] is used to evaluate fluxes at cell interfaces. A backward-Euler time stepping method is used to drive the solution to a steady state. An approximate solution of the linear system of equations formed at each time step is obtained using several iterations of a point-iterative Gauss–Seidel-type method. Further details pertaining to the flow solver can be found in reference [12].

4.3. Adjoint Solver

After each flow solution in an adaptive run, the coarse-grid adjoint is obtained by solving the linear system of equations given in (9). Although this system can be solved directly using GMRES [15], a time-like derivative is added and the solution is obtained by marching in time, much like the flow solver,

$$\left\{\frac{A_H}{\Delta t} + \left[\frac{\partial R_H}{\partial U_H}\right]^T\right\}(\Delta\Psi_H)^n = \left(\frac{\partial f_H}{\partial U_H}\right)^T - \left[\frac{\partial R_H}{\partial U_H}\right]^T (\Psi_H)^n, \tag{24}$$

where $(\Psi_H)^{n+1} = (\Psi_H)^n + (\Delta\Psi_H)^n$ and A_H is a diagonal matrix containing the control-volume areas associated with each node in Ω_H. The time term can be used to increase the diagonal dominance of the system for cases in which GMRES alone would tend to stall. Generally, this results in a more robust adjoint solver. Preconditioning is achieved using a point-iterative scheme similar to that used in the flow solver. Further details concerning the adjoint solver can be found in reference [12].

4.4. Functionals

The functionals chosen for the present study are those of common interest in aeronautical applications: the lift, drag, and moment coefficients. These outputs are all expressible as surface integrals over parts of the domain boundary. Let $\partial\Omega'$ denote that portion of the

continuum boundary over which aerodynamic forces are to be computed, such as the surface of an airfoil or the walls of a channel. The resultant force $\vec{F}_R$ acting on $\partial\Omega'$ due to the inviscid flow is given by

$$\vec{F}_R \equiv \int_{\partial\Omega'} p\,\vec{\mathbf{n}}\,ds, \tag{25}$$

where $\vec{\mathbf{n}}$ is the unit normal vector on $\partial\Omega'$. The lift L and drag D are defined as the components of $\vec{F}_R$ perpendicular and parallel, respectively, to the free-stream velocity, $\vec{V}_\infty$,

$$L \equiv |\vec{F}_R| \sin\theta, \tag{26}$$

$$D \equiv |\vec{F}_R| \cos\theta, \tag{27}$$

where θ is the angle between $\vec{V}_\infty$ and $\vec{F}_R$. In two dimensions, an implicit definition for θ that accounts for the proper sign convention is

$$\sin\theta \equiv \left(\frac{\vec{V}_\infty}{|\vec{V}_\infty|} \times \frac{\vec{F}_R}{|\vec{F}_R|} \right) \cdot \vec{k},$$

where $\vec{k}$ is the unit vector pointing out of the page. The total moment exerted on $\partial\Omega'$ about $\vec{r}_0 = (x_0, y_0)$ is given by

$$M \equiv \int_{\partial\Omega'} (\vec{r} - \vec{r}_0) \times p\,\vec{\mathbf{n}}\,ds \cdot \vec{k}, \tag{28}$$

where $\vec{r} = (x, y)$ is the position vector measured with respect to the origin. Finally, the lift, drag, and moment coefficients are defined, respectively, as

$$C_L \equiv \frac{L}{q_\infty c}, \tag{29}$$

$$C_D \equiv \frac{D}{q_\infty c}, \tag{30}$$

$$C_M \equiv \frac{M}{q_\infty c^2}, \tag{31}$$

where $q_\infty = \rho_\infty V_\infty^2/2$ is the free-stream dynamic pressure and c is a reference length such as the airfoil chord length.

4.5. Projection Operators

In order to compute the error estimate given by (10), we must define the projection operators I_h^H and J_h^H introduced in (2) and (8). These operators map coarse-grid primal and adjoint solutions, respectively, onto the fine grid. For the purposes of computing error estimates, a quadratic projection operator, Q_h^H, is used for both the primal and adjoint solutions. That is, $I_h^H \equiv J_h^H \equiv Q_h^H$. The projection is performed by constructing a quadratic interpolant over each coarse-grid element using a local least squares procedure to be described below.

To compute the adaptation parameter defined in (16), we use the same quadratic projection operator, Q_h^H, and a linear projection operator, L_h^H. The linear operator represents simple linear interpolation over each coarse-grid element.

The following method is used to calculate the quadratic projection operator, Q_h^H. Consider a generic scaler function, ϕ, and its derivatives, ϕ_x and ϕ_y, sampled at the nodes of the coarse grid. Over each coarse-grid element, k, we can establish the following linear distributions

$$\bar{\phi} = \sum_{i=1}^{3} N_i \phi^i, \tag{32}$$

$$\bar{\phi}_x = \sum_{i=1}^{3} N_i \phi_x^i, \tag{33}$$

$$\bar{\phi}_y = \sum_{i=1}^{3} N_i \phi_y^i, \tag{34}$$

where ϕ^i, ϕ_x^i, and ϕ_y^i are the sampled values at the three vertices of element k, and N_i are the standard shape functions for a linear triangle [19]. A quadratic interpolant of ϕ over element k can be represented as

$$\tilde{\phi} = \sum_{j=1}^{6} \tilde{N}_j \tilde{\phi}_j, \tag{35}$$

where $\tilde{N}_j$ are the shape functions for a quadratic triangle [19], and $\tilde{\phi}_j$ are the unknown coefficients to be determined. The least squares problem can be cast as the following unconstrained minimization: find $z^* = \{\tilde{\phi}_1^*, \tilde{\phi}_2^*, \ldots, \tilde{\phi}_6^*\}^T$ such that $\Lambda(z^*) = \min_z \Lambda(z)$, where

$$\Lambda(z) = \int_{\Omega_k} [(\tilde{\phi}(z) - \bar{\phi})^2 + (\tilde{\phi}_x(z) - \bar{\phi}_x)^2 + (\tilde{\phi}_y(z) - \bar{\phi}_y)^2]\, d\Omega. \tag{36}$$

Once z^* has been determined for element k, it is substituted back into (35) yielding the desired interpolant. The procedure is repeated for all elements in the coarse grid.

4.6. Grid Refinement

The original grids (prior to adaptive refinement) were generated using AFLR2: an advancing-front/local-reconnection, unstructured, triangulator developed by Dr. D. Marcum at Mississippi State University [11]. During the adaptive runs, a Delaunay node insertion and local retriangulation algorithm is used for refining the grids. The algorithm attempts to satisfy a prescribed element size distribution such as the one implied by (20). If more than 50% of the elements in the current grid are flagged for refinement, then the desired element size distribution obtained from (20) is scaled so that no more than half of the elements are refined from one adaptive iteration to the next. This measure may be triggered in the early stages of an adaptive run if the prescribed error level for the functional is very low relative to the resolution of the initial grid.

Grid coarsening is not performed in the current study. Requests for larger element sizes are simply ignored by the refinement algorithm. Coarsening could be introduced into the adaptive procedure, if desired, with minor modifications to the proposed adaptive strategy.

Various grid quality measures are incorporated as postprocessing steps after the initial Delaunay refinement: edge swapping is used to reduce the maximum node degree in the triangulation to 7, a node removal algorithm is used to remove nodes of degree lower than 5, and smoothing of select regions of the grid is performed to improve element aspect ratios.

5. NUMERICAL RESULTS

Several inviscid test cases are presented to demonstrate the effectiveness of the proposed grid adaptive procedure. Comparisons are made with a commonly used gradient-based adaptive method [3, 5, 18]. The proposed output-based scheme is shown to compare favorably in terms of accuracy, efficiency, and reliability relative to the gradient-based scheme.

5.1. Gradient-Based Adaptive Method

A gradient-based adaptive method is implemented for the purpose of comparatively assessing the performance of the proposed output-based method. The adaptive indicator is based on the local magnitude of the second derivatives in either the computed pressure or Mach number scaled with H_k^r, where H_k is the local element size and $r = 2$ or 3 [3, 5, 18]. The elemental second derivatives are estimated using the quadratic resconstruction procedure described in Section 4.5. The adaptive criterion requires that the value of the indicator be less than some predetermined threshold value for all elements. In the absence of any rigorous relationship to the functional error, the threshold values used in this work are chosen heuristically depending on the field quantity being adapted on and the value of r. While the threshold value influences the extent to which the grid is refined, it does not change the general refinement pattern of the final grid. Hence, meaningful comparisons with the output-based adaptive patterns are possible. If the adaptive indicator for an element is less than the predetermined threshold value, the element is flagged for refinement and the desired new element size is set to half of its original size. The gradient-based method is implemented using the same grid refinement algorithm used for the proposed output-based scheme (as described in Section 4.6). Further implementation details can be found in reference [18].

5.2. $M_\infty = 0.38$ Gaussian Bump

The first test case is subsonic flow in a channel of height 10 and length 20 with an inlet Mach number of $M_\infty = 0.38$. The shape of the lower wall is a Gaussian of amplitude 0.5 centered at the mid-length of the channel. The purpose of this test case is to examine convergence rates of various quantities associated with the base solver and error estimation procedure. No adaptation is performed for this test case. A hierarchy of eight embedded grids is constructed for the purposes of conducting the convergence tests. Figure 2 shows a plot of the computed Mach number distribution on the finest grid. Three functionals are considered: the coefficients of drag, lift, and moment acting on the walls of the duct. The moment is computed with respect to the origin. Figure 3 shows plots of the base error in these functionals and the remaining error after correction. The remaining error is measured with respect to fine-grid values corresponding to $N = 2$ (see Section 2). The base value of the drag converges like $\mathcal{O}(h^3)$, whereas the lift and moment coefficients appear to converge at a slightly lower rate. In all cases, the remaining error after correction is observed to be superconvergent,

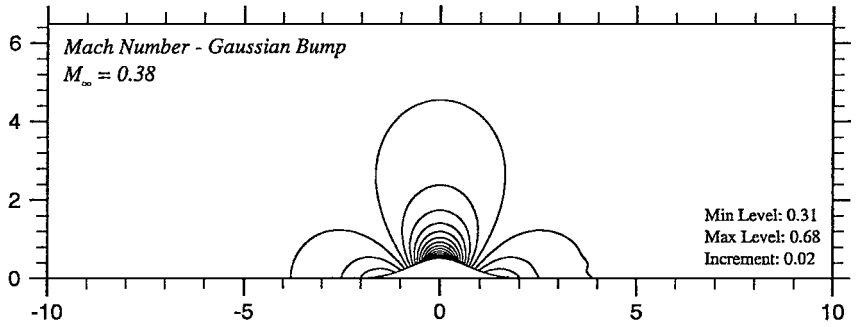

FIG. 2. Gaussian bump test case: $M_\infty = 0.38$, $\alpha = 0^\circ$. Computed Mach number distribution.

exhibiting approximately $\mathcal{O}(h^4)$ convergence. Figure 4 shows convergence plots of the global and local adaptation parameters defined in (18) and (19), respectively. The local adaptation parameter η_k is evaluated over an interior element located approximately 0.5 units above the peak of the Gaussian bump, and over a boundary element bordering the peak. The local adaptation parameter on the boundary is observed to converge essentially as $\mathcal{O}(h^2)$, one order slower than in the interior. Even slower convergence would be expected near singularities, discontinuities, or geometric irregularities such as sharp corners. The global adaptation parameter exhibits approximately $\mathcal{O}(h^{2.5})$ convergence. The convergence rates of these parameters are used as a guide in determining an appropriate value for the relaxation parameter ω that appears in (20).

5.3. M_∞ = 0.95 NACA 0012 Airfoil

This test case involves inviscid, transonic flow past a NACA 0012 airfoil [18] with a free-stream Mach number of $M_\infty = 0.95$, and angle of attack, $\alpha = 0^\circ$. The drag coefficient is chosen as the functional of interest in this case. Figure 5 shows plots of the computed Mach number distribution for a well-resolved solution. The left plot is a blow-up near the airfoil; the right plot illustrates some of the flow features further away from the airfoil. In this test case, the flow accelerates as it passes over the leading edge section through the sonic line to supersonic speeds. The flow then encounters an oblique shock at the trailing edge after which it remains at a low supersonic Mach number. Further downstream of the airfoil, a weak normal shock occurs lowering the Mach number to subsonic conditions.

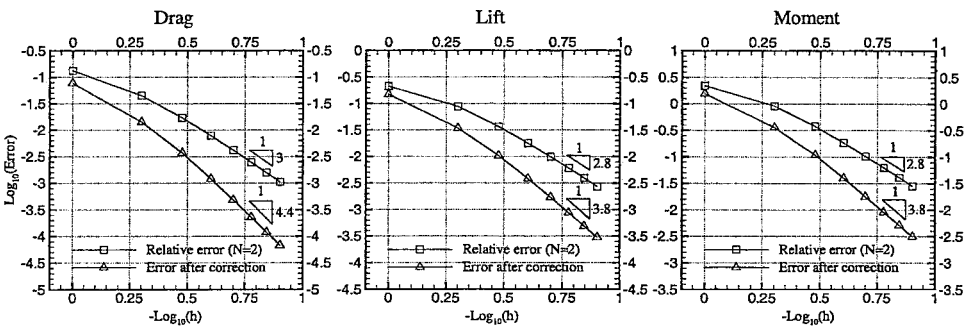

FIG. 3. Gaussian bump test case: $M_\infty = 0.38$, $\alpha = 0^\circ$. Convergence plots of the error in the functional and the remaining error after correction. Errors are measured with respect to fine-grid values corresponding to $N = 2$.

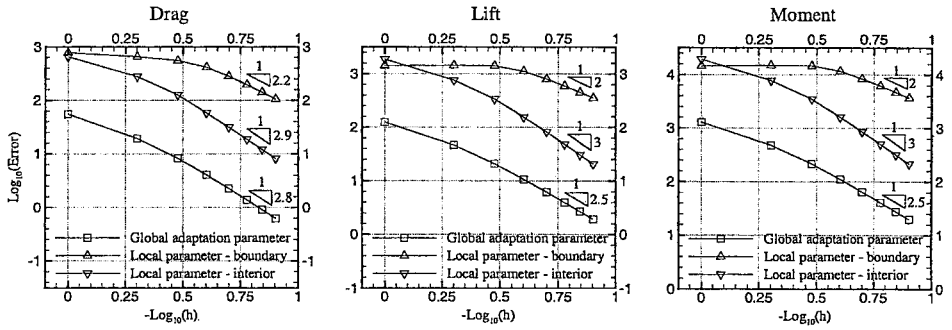

FIG. 4. Gaussian bump test case: $M_\infty = 0.38$, $\alpha = 0^\circ$. Convergence plots of the global and local adaptation parameters as defined in (18) and (19), respectively.

Figures 6 and 7 show adaptive results using the proposed output-based adaptive procedure. Desired error levels of $e_o = 0.01$ and $e_o = 0.0005$ were specified, respectively. The drag is plotted versus the total number of nodes in the corresponding grid. Also shown is the corrected drag on each grid after using the proposed error estimation procedure to correct the base value. For comparison, the drag convergence on a series of uniformly refined grids is also plotted (upper-most curve). These grids were generated by subdividing each of the edges in the original grid into n equal segments for values of $n = 2, 3, \ldots, 8$. The original grid in the hierarchy of uniformly refined grids is identical to the starting grid in the adaptive run. The extrapolated value in the figure is obtained using a Richardson extrapolation of the three finest, uniformly refined grids.

In both adaptive cases the desired error levels are surpassed, the drag predictions converge to the correct value, and the adaptive process self-terminates after only two or three iterations. Note that the grid is refined near the leading edge region where the flow is rapidly accelerating. There is also some modest refinement near the trailing edge of the airfoil. Conversely, the oblique shock and the normal shock downstream of the airfoil are not resolved by the adaptive algorithm.

Also of interest is the manner in which the adaptive procedure converges for different requested error levels, e_0. Comparing Figs. 6 and 7, we see that when e_0 is relatively large (e.g., $e_0 = 0.01$), the adaptive algorithm adds only a limited number of nodes (less then

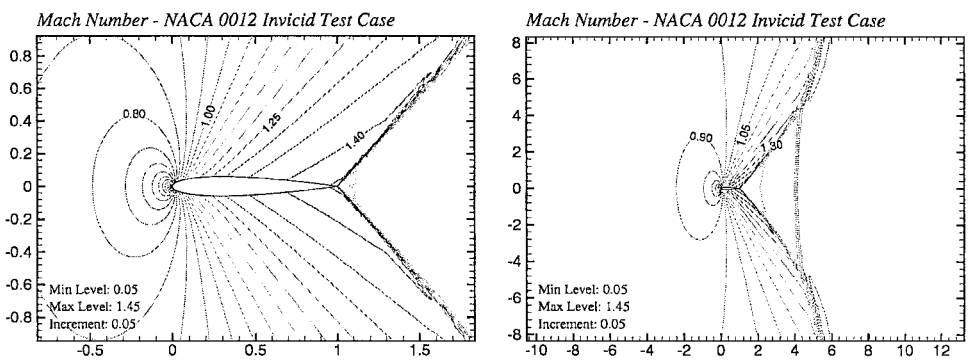

FIG. 5. NACA 0012 test case: $M_\infty = 0.95$, $\alpha = 0^\circ$. Computed Mach number distributions. Left: blow up near the airfoil; right: more distant view.

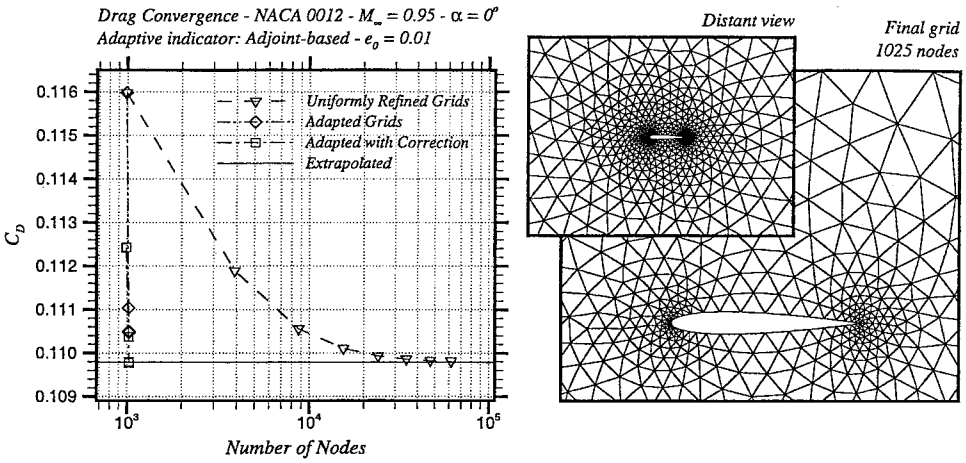

FIG. 6. NACA 0012 test case: $M_\infty = 0.95, \alpha = 0^\circ$. Left: error convergence in the computed drag during a typical adaptive run. The proposed output-based adaptive algorithm is used. A requested error level of $e_o = 0.01$ was prescribed. Right: final adapted grid.

50 new nodes were added) to quickly raise the desired accuracy. However, when higher accuracy is requested (i.e., lower values of e_0), the adaptive procedure tends toward more global refinement. For $e_0 = 0.0005$, the initial refinement more than triples the grid size, adding over 2000 new nodes.

Figure 8 shows plots of the initial (top left) and final (top right) grids in a typical adaptive run using the proposed output-based scheme. Below each of these are the corresponding distributions of the adaptation parameter ε_k. It is evident that the adaptation parameter was effectively reduced during the adaptive process to a level below the minimum contour level on the plot.

Many adaptive strategies in use today employ flow gradient or curvature-based indicators for grid refinement [3, 18]. Such indicators are very sensitive to shocks and would, therefore,

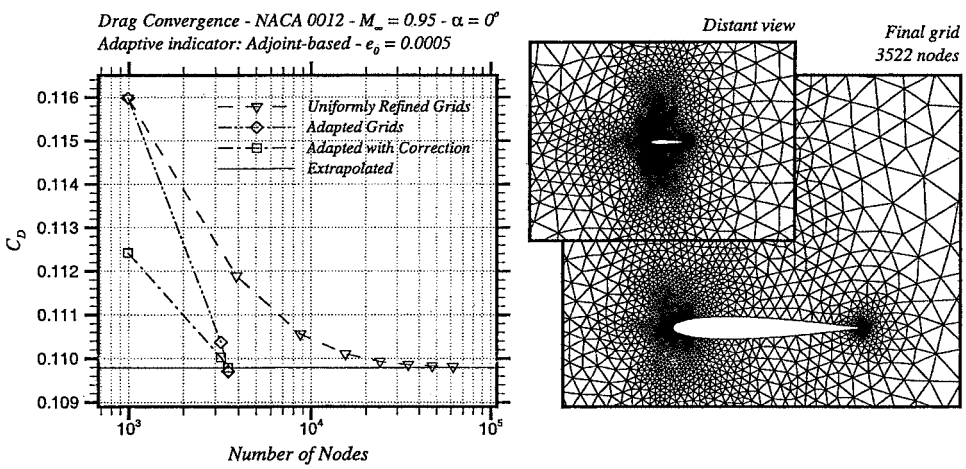

FIG. 7. NACA 0012 test case: $M_\infty = 0.95, \alpha = 0^\circ$. Left: error convergence in the computed drag during a typical adaptive run. The proposed output-based adaptive algorithm is used. A requested error level of $e_o = 0.0005$ was prescribed. Right: final adapted grid.

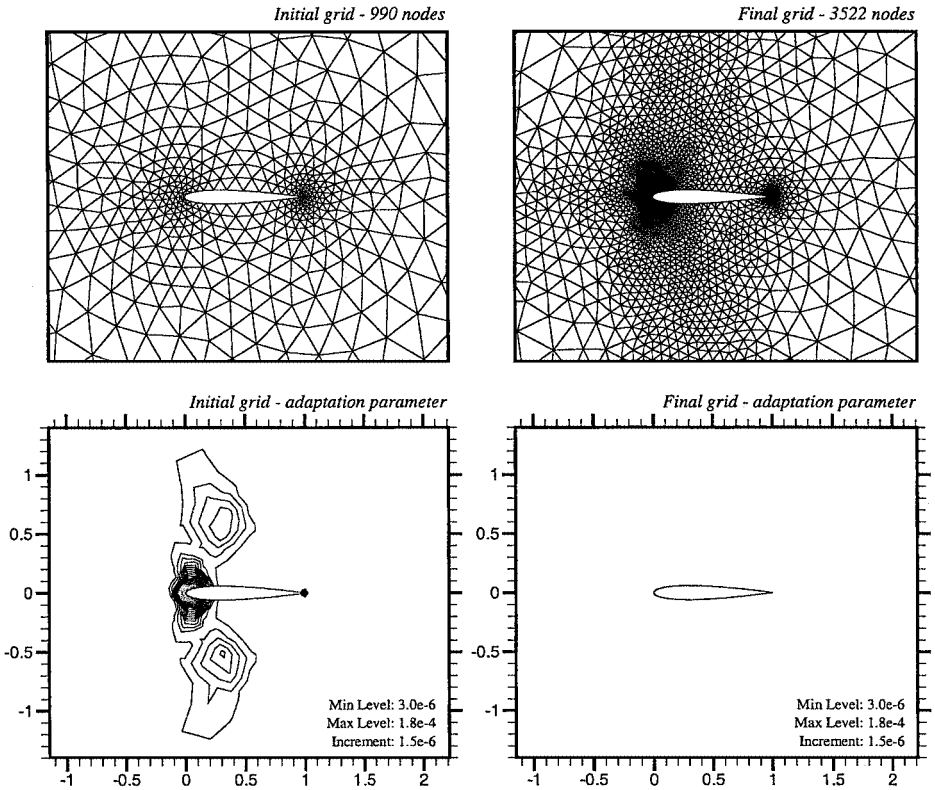

FIG. 8. NACA 0012 test case: $M_\infty = 0.95$, $\alpha = 0^\circ$. Top left: original grid. Bottom left: adaptation parameter, ε_k, on the original grid. Top right: final adapted grid using the proposed output-based adaptive method with a prescribed error level of $e_o = 0.0005$. Bottom right: adaptation parameter on the final grid.

aggressively refine the grid in these regions. For many common functionals of interest in aeronautical applications, continual refinement of shocks and other isloated flow features would not necessarily enhance the accuracy of the predicted functionals. In some cases, this type of refinement could even lead to incorrect results [18].

We demonstrate the performance of gradient-based adaptive indicators for comparison with our output-based approach. Implementation details are given in Section 5.1 and reference [18]. The left-hand plot in Fig. 9 shows the convergence of the computed drag on each grid during a typical adaptive run. The adaptive indicator is based on the magnitude of the second derivatives in the pressure scaled with H_k^2 where H_k is the local element size. It is evident that both the corrected and uncorrected values of the drag are converging to an erroneous value somewhat less than the extrapolated value. Furthermore, the adaptive process did not show signs of self-terminating and was, therefore, halted after eight adaptive iterations. The right plot in Fig. 9 shows an intermediate grid in the adaptive sequence. The pressure-based indicator, being extremely sensitive to the shock structure, continued to refine the grid near the shocks despite the lack of convergence in the drag.

Figure 10 is analogous to Fig. 9 in almost every regard except that the adaptive indicator used in this run is based on the magnitude of the second derivatives in the pressure scaled with H_k^3 instead of H_k^2. This is recommended as an improvement in reference [18]. This time, the predicted values for the drag are converging to an erroneous value significantly

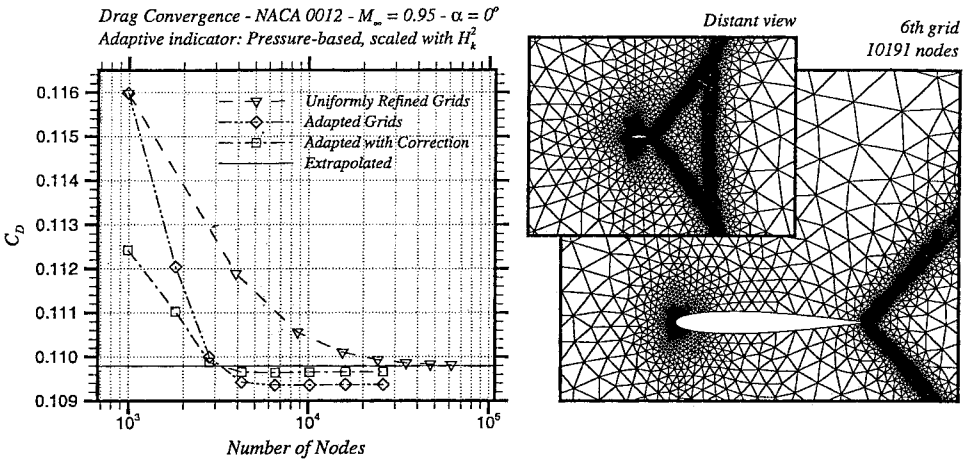

FIG. 9. NACA 0012 test case: $M_\infty = 0.95, \alpha = 0^\circ$. Left: error convergence in the computed drag during a typical adaptive run. A gradient-based adaptive strategy is used [18] with an indicator based on the magnitude of the second derivatives in the pressure scaled with H_k^2 where H_k is the local element size. Right: intermediate grid in the adaptive run.

larger than the extrapolated value, however, it appears as though the adaptive process itself would have eventually terminated had it not been stopped after eight iterations. Note that the functional corrections in Figs. 9 and 10 are providing a significant improvement in accuracy. Nevertheless, the corrected values are still not converging to the extrapolated value of the drag.

Finally, Fig. 11 shows analogous results using an indicator based on the magnitude of the second derivatives in the Mach number scaled with H_k^2. As with the pressure-based scheme, the predicted drag converges to an erroneous value slightly less than the extrapolated value. Furthermore, the adaptive process does not appear to be terminating. The Mach

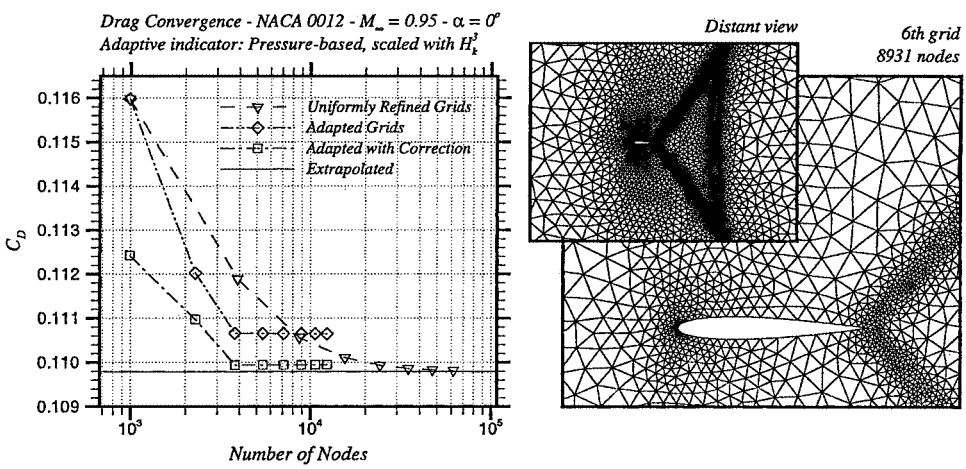

FIG. 10. NACA 0012 test case: $M_\infty = 0.95, \alpha = 0^\circ$. Left: error convergence in the computed drag during a typical adaptive run. A gradient-based adaptive strategy is used [18] with an indicator based on the magnitude of the second derivatives in the pressure scaled with H_k^3 where H_k is the local element size. Right: intermediate grid in the adaptive run.

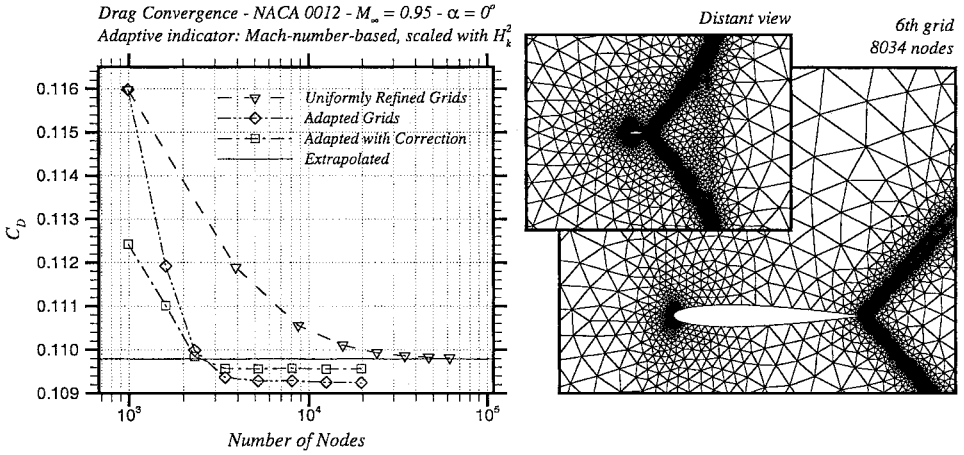

FIG. 11. NACA 0012 test case: $M_\infty = 0.95$, $\alpha = 0^\circ$. Left: error convergence in the computed drag during a typical adaptive run. A gradient-based adaptive strategy is used [18] with an indicator based on the magnitude of the second derivatives in the Mach number scaled with H_k^2 where H_k is the local element size. Right: intermediate grid in the adaptive run.

number indicator continues to refine the grid near the shocks without any corresponding improvement in the accuracy of the drag.

5.4. M_∞ = 6.0 NACA 0012 Airfoil

The next test case involves inviscid, supersonic flow past a NACA 0012 airfoil with a free-stream Mach number of $M_\infty = 6.0$, and angle of attack, $\alpha = 0^\circ$. The drag coefficient is chosen as the functional under consideration. In particular, we are interested in the performance of the proposed adaptation algorithm in the presence of strong shocks. The output-based adaptive results are shown in Fig. 12. Convergence of the drag to the desired error limit (in this case $e_0 = 0.001$) is achieved with under 4000 nodes. By comparison, the uniform refinement cases have still not converged even after 60,000 nodes. The adaptation occurs around the body in general but is most significant at the leading edge and at the bow shock in the leading edge region.

Pressure-based adaptive results are shown in Fig. 13. The adaptive algorithm does not appear to be self-terminating after the fifth adaptive iteration and the drag is exhibiting oscillatory convergence. Refinement for the pressure-based scheme is much more uniform between the bow shock and the airfoil, in contrast to the output-based method, which favors leading-edge refinement. As one would expect, the leading-edge is a dominant region for determining aerodynamic performance.

5.5. M_∞ = 0.4, α = 5° NACA 0012 Airfoil

In this case, we consider subsonic flow past a NACA 0012 airfoil with a free-stream Mach number of $M_\infty = 0.4$, and an angle of attack of $\alpha = 5^\circ$. The functional of interest is the moment coefficient taken about the leading edge. The output-based adaptive results are shown in Fig. 14. Convergence of the moment to the desired error tolerance of $e_0 = 0.001$ is achieved in three to four adaptive iterations with a final converged grid containing approximately 9000 nodes.

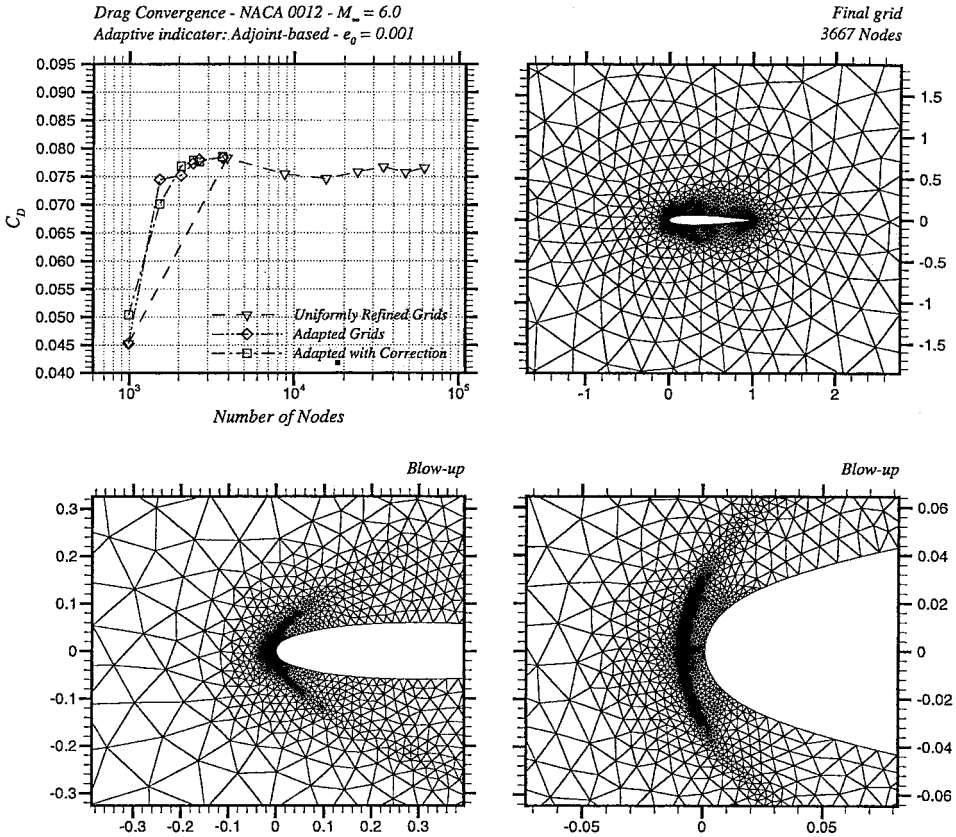

FIG. 12. NACA 0012 test case: $M_\infty = 6.00$, $\alpha = 0^\circ$. Upper left: error convergence in the computed drag during a typical adaptive run. The proposed output-based adaptive algorithm is used. A requested error level of $e_o = 0.001$ was prescribed. Upper right: final adapted grid. Lower: blow-up of airfoil leading edge region.

Pressure-based adaptive results are shown in Fig. 15. The computed moment coefficient fluctuates erratically during the pressure-based adaptive process and does not appear to be converging to any particular value. Furthermore, the adaptive process does not show signs of terminating after the eighth iteration. In contrast, the adjoint-based adaptive scheme provides a stable, monotonic convergence of the moment to the extrapolated value and self-terminates once the requested error level has been surpassed.

5.6. Advanced Energy Efficient Transport (EET) Three-Element Airfoil

The next test case is inviscid, subsonic flow past the advanced energy efficient transport (EET) three-element airfoil [9]. Adaptive simulations were performed for a free-stream Mach number of $M_\infty = 0.26$, and an angle of attack of $\alpha = 8^\circ$. The functional of interest in this case is the lift coefficient for the airfoil (i.e., all three-elements). Figure 16 shows the computed Mach number contours and streamlines for this test case.

The results for the output-based adaptation are shown in Fig. 17 for an error level of $e_o = 0.05$. The adaptive algorithm surpasses the requested error level and self-terminates at a final grid of less than 9000 nodes. In comparison, the uniform-grid solutions attain about the same error level at approximately 70,000 nodes.

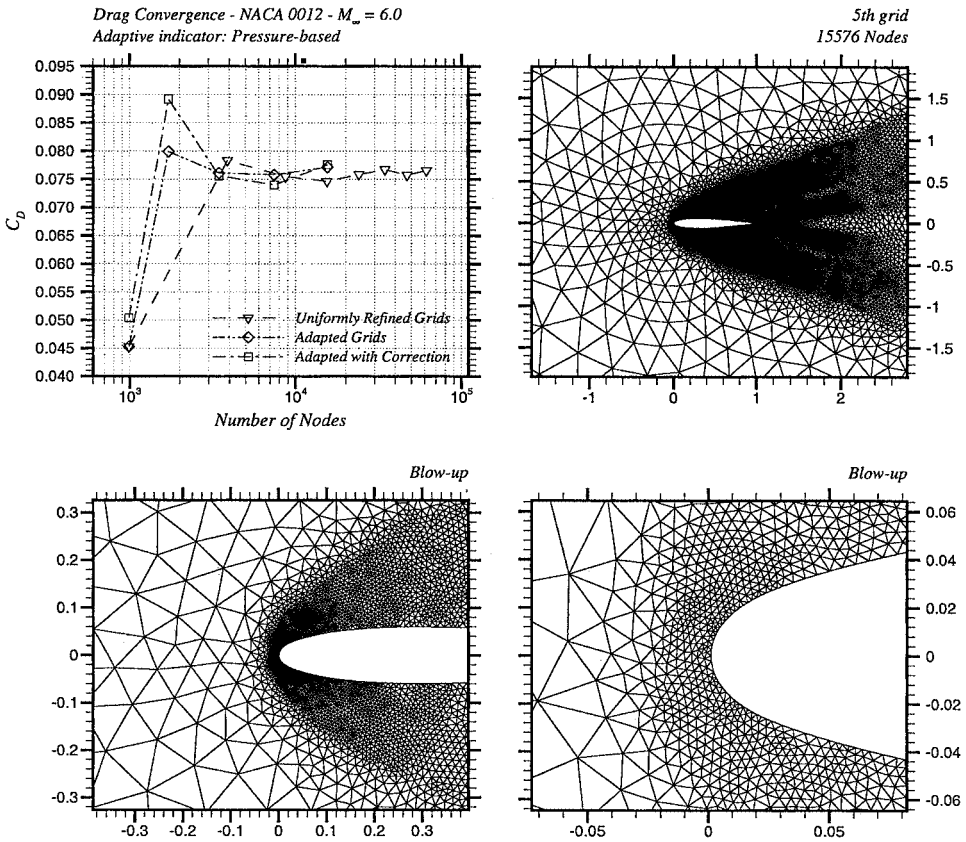

FIG. 13. NACA 0012 test case: $M_\infty = 6.00$, $\alpha = 0^\circ$. Upper left: error convergence in the computed drag during a typical adaptive run. A gradient-based adaptive strategy is used [18] with an indicator based on the magnitude of the second derivatives in the pressure scaled with H_k^2 where H_k is the local element size. Upper right: final adapted grid. Lower: blow-up of airfoil leading edge region.

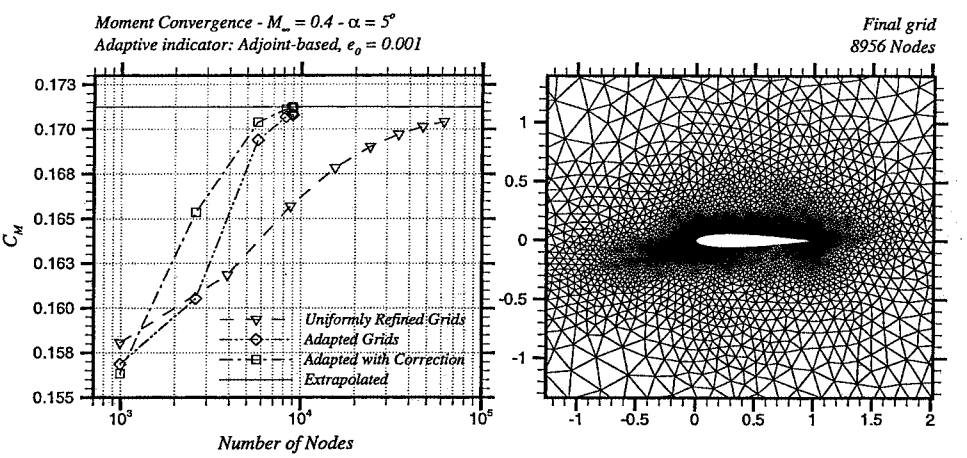

FIG. 14. NACA 0012 test case: $M_\infty = 0.4$, $\alpha = 5^\circ$. Left: error convergence in the computed leading-edge moment during a typical adaptive run. The proposed output-based adaptive algorithm is used. A requested error level of $e_o = 0.001$ was prescribed. Right: final adapted grid.

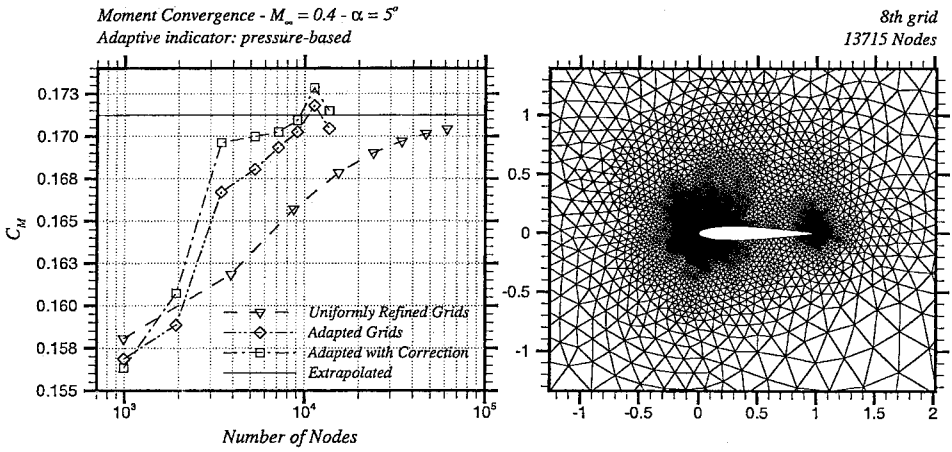

FIG. 15. NACA 0012 test case: $M_\infty = 0.4, \alpha = 5^\circ$. Left: error convergence in the computed leading-edge moment during a typical adaptive run. A gradient-based adaptive strategy is used [18] with an indicator based on the magnitude of the second derivatives in the pressure scaled with H_k^2 where H_k is the local element size. Right: final adapted grid.

Figure 18 shows adaptive results using the pressure-based adaptive method. Even though the flow field is smoothly varying (i.e., no shocks), the (uncorrected) life appears to be converging to an erroneous value and never achieves the level of accuracy of the final output-based grid. Note that the error estimation procedure dramatically improves the lift error but still does not quite match the apparent asymptotic lift value of the uniform-grid results. Based on visual inspections of the grids in Figs. 17 and 18, we attribute the lack of

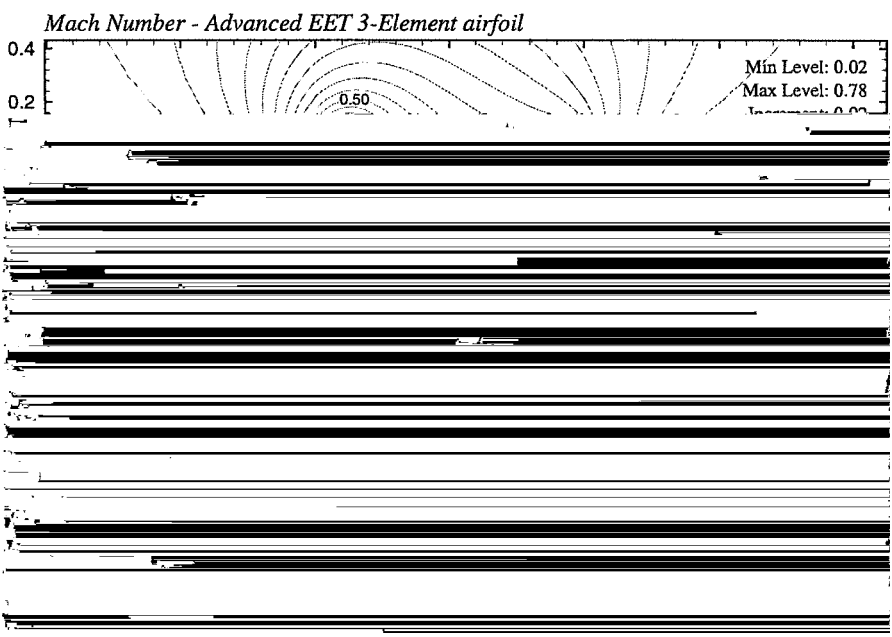

FIG. 16. Advanced EET 3-element airfoil test case: $M_\infty = 0.26, \alpha = 8^\circ$. Computed Mach number distribution and streamlines.

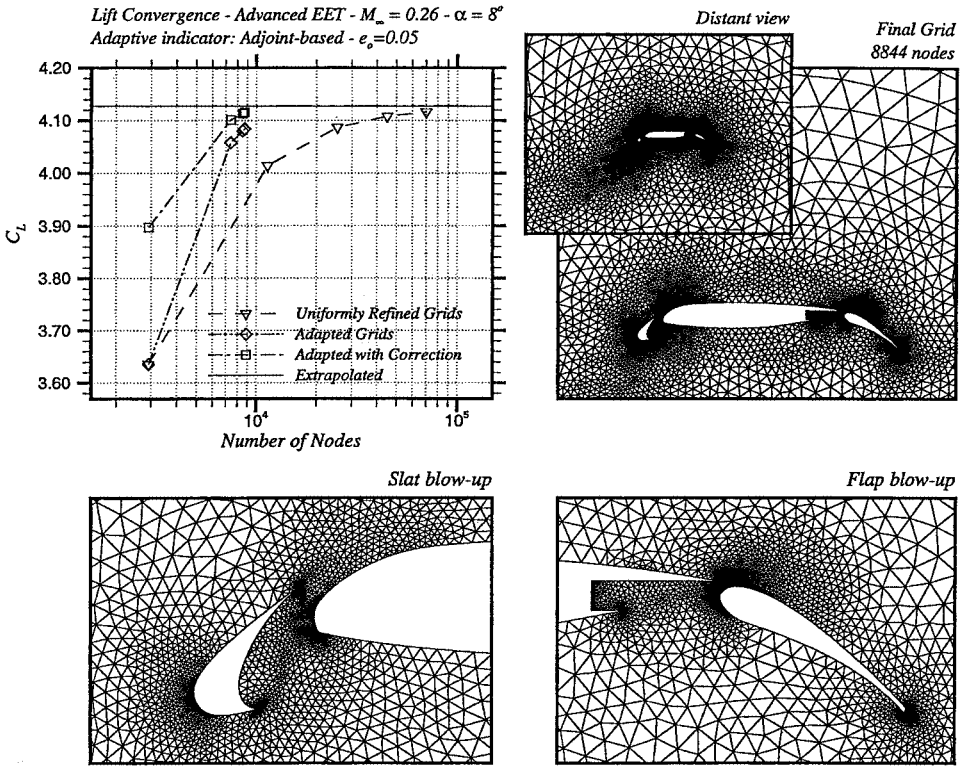

FIG. 17. Advanced EET 3-element airfoil test case: $M_\infty = 0.26$, $\alpha = 8^\circ$. Upper left: error convergence in the computed lift during a typical adaptive run. The proposed output-based adaptive algorithm is used. A requested error level of $e_o = 0.05$ was prescribed. Upper right: final adapted grid. Lower: blow-up near slat and flap.

convergence in the lift using the pressure-based scheme to a lack of grid resolution in the following regions: the rear portion of the suction side of the main element, the pressure side of the main element, and the pressure side of the flap. During the last four pressure-based adaptive iterations, there is little or no grid refinement in these regions. Most of the nodes are being added near the geometric irregularities of the three elements (i.e., at the sharp corners).

5.7. M_∞ = 3.0 Two-Element Airfoil

In this test case, we study the interactions of two airfoils in a $M_\infty = 3.0$ free-stream. The left plot of Fig. 19 shows the Mach number distribution for this flow. The airfoils are positioned such that the bow shock of the upstream airfoil interacts with the surface of the downstream airfoil. The functional is chosen to be the pressure drag on the downstream airfoil. The right plot of Fig. 19 shows a contour plot of the adjoint x-momentum variable based on the lower-airfoil drag. The results for the output-based and pressure-based algorithms are shown in Figs. 20 and 21, respectively. The output-based approach (with an error tolerance of $e_0 = 0.001$) converges the lower airfoil drag in less than 4000 nodes while the pressure-based approach requires over 37,000 nodes to achieve the same accuracy. The differences in adaptation are striking with the pressure-based approach excessively refining all shock waves including those which do not play any role in determining the lower airfoil

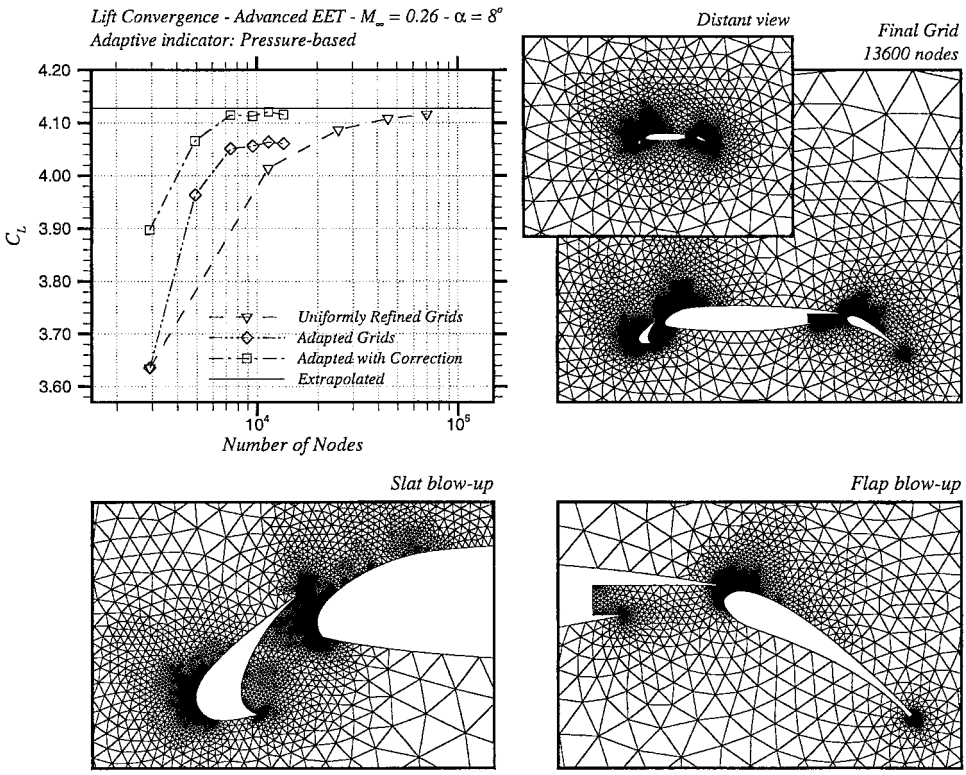

FIG. 18. Advanced EET 3-element airfoil test case: $M_\infty = 0.26$, $\alpha = 8^\circ$. Upper left: error convergence in the computed lift during a typical adaptive run. A gradient-based adaptive strategy is used [18] with an indicator based on the magnitude of the second derivatives in the pressure scaled with H_k^2 where H_k is the local element size. Upper right: final adapted grid. Lower: blow-up near slat and flap.

drag (e.g., the upper surface shocks on the upstream airfoil). Also, the pressure-based refinement of the leading-edge of the two airfoils is nearly identical, whereas the output-based approach refines only those features which dominate the lower airfoil drag. Specifically, the leading-edge region (including the bow shock) of the lower airfoil, the oblique shock from

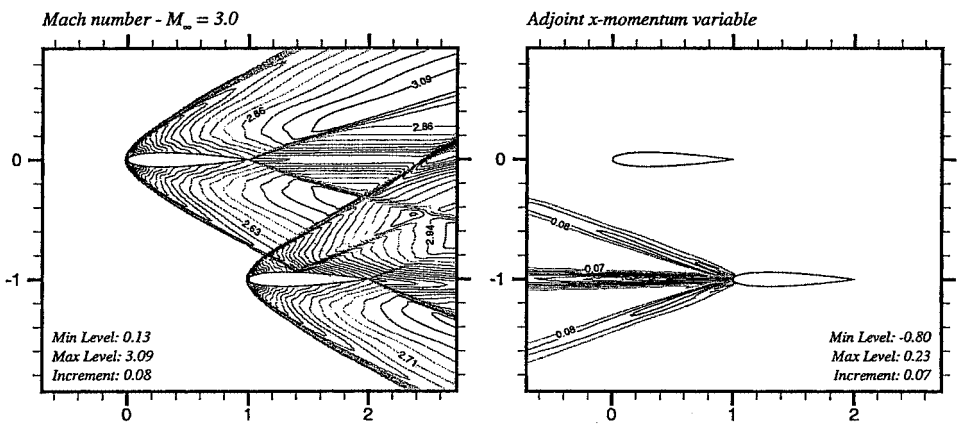

FIG. 19. Two airfoil (NACA 0012) test case: $M_\infty = 3.00$, $\alpha = 0^\circ$. Left: computed Mach number distribution. Right: adjoint x-momentum variable based on the lower-airfoil drag.

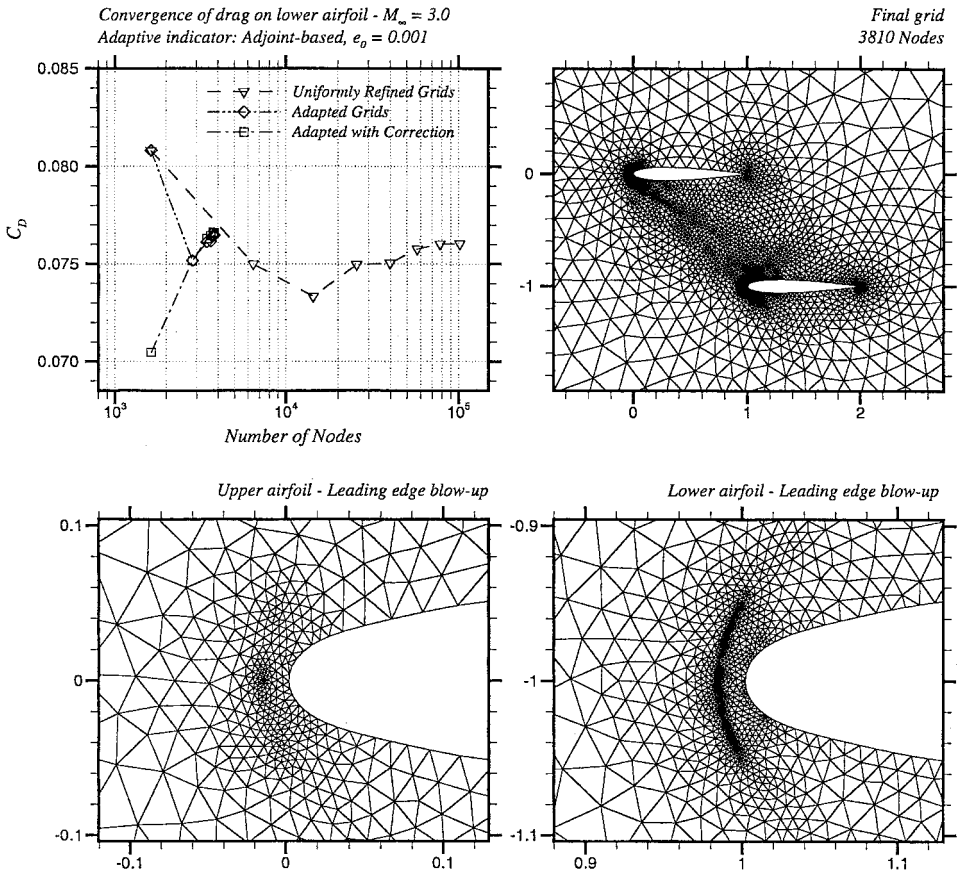

FIG. 20. Two airfoil (NACA 0012) test case: $M_\infty = 3.00$, $\alpha = 0^\circ$. Upper left: error convergence in the computed drag on the *downstream* (i.e., lower) airfoil during a typical adaptive run. The proposed output-based adaptive algorithm is used. The adjoint is based on the lower-airfoil drag. A requested error level of $e_o = 0.001$ was prescribed. Upper right: final adapted grid. Lower: blow-up of leading edge regions.

the upper airfoil, and the leading-edge region of the upper airfoil are adapted. None of the trailing edge shocks are refined as they play little role in the drag.

5.8. M_∞ = 0.8, α = 1.25° NACA 0012 Airfoil

The last test case is transonic flow past a NACA 0012 airfoil with a free-stream Mach number of $M_\infty = 0.8$, and an angle of attack of $\alpha = 1.25^\circ$. This flow exhibits a strong shock on the upper surface of the airfoil and a relatively weaker shock on the lower surface. The functional of interest is chosen to be the lift coefficient.

Output-based adaptive results are shown in Figs. 22 and 23 for prescribed error levels of $e_o = 5\%\ C_L$ and $e_o = 2.5\%\ C_L$, respectively. Pressure-based adaptive results are presented in Fig. 24. It is evident that the lift converges poorly on the uniformly refined grids for this test case. This is attributed to inadequate grid resolution in the trailing edge region of the seed grid. Note that this seed grid is also the starting grid in each of the adaptive runs. For this test case only, the extrapolated lift value in the convergence plots of Figs. 22–24 is based on a different set of uniformly refined grids. The alternate seed grid is chosen as the final adapted grid in the $e_o = 2.5\%\ C_L$ output-based run; it contains 5214 nodes and is shown in Fig. 23. This grid is globally refined twice

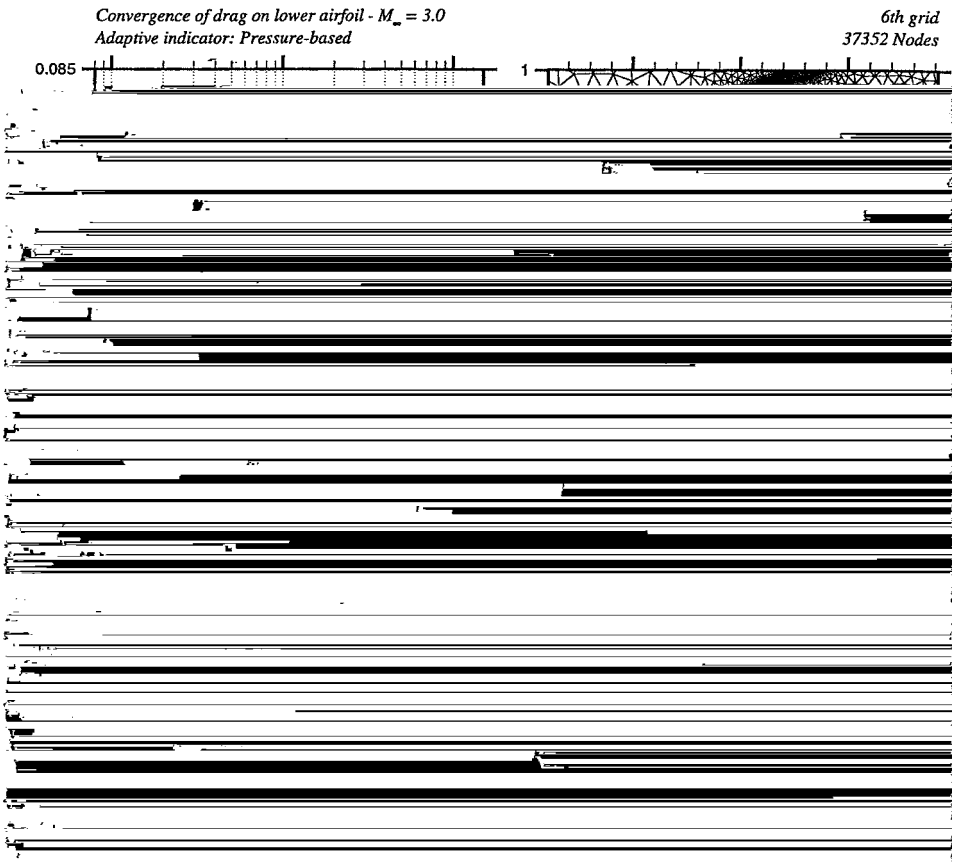

FIG. 21. Two airfoil (NACA 0012) test case: $M_\infty = 3.00$, $\alpha = 0^\circ$. Upper left: error convergence in the computed drag on the *downstream* (i.e., lower) airfoil during a typical adaptive run. A gradient-based adaptive strategy is used [18] with an indicator based on the magnitude of the second derivatives in the pressure scaled with H_k^2 where H_k is the local element size. Upper right: final adapted grid. Lower: blow-up of leading edge regions.

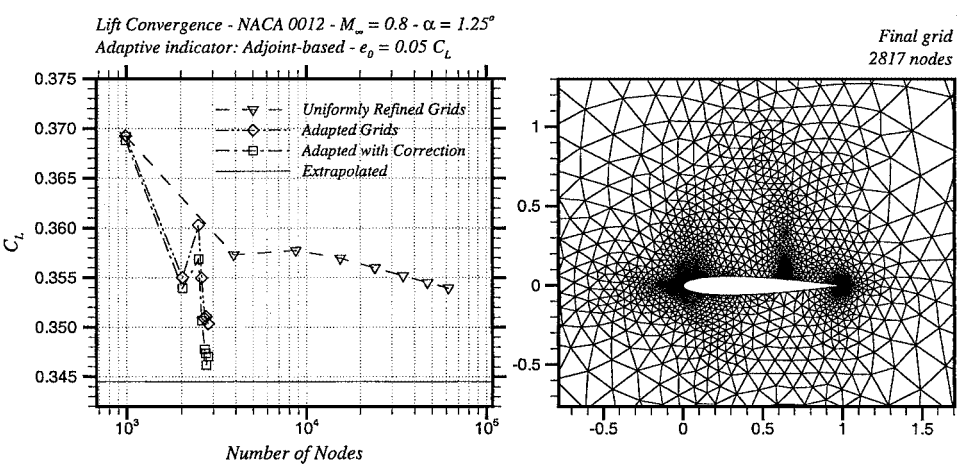

FIG. 22. NACA 0012 test case: $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Left: error convergence in the computed lift coefficient during a typical adaptive run. The proposed output-based adaptive algorithm is used. A requested error level of $e_o = 5\%\ C_L$ was prescribed. Right: final adapted grid.

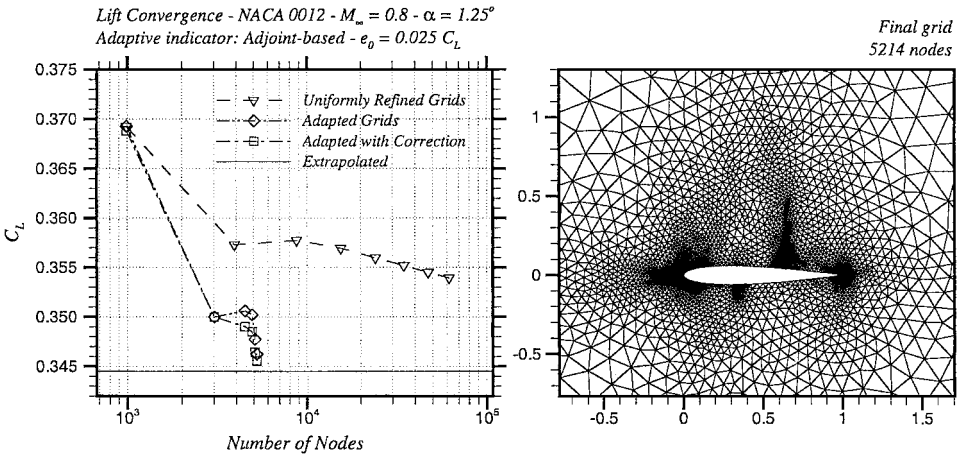

FIG. 23. NACA 0012 test case: $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Left: error convergence in the computed lift coefficient during a typical adaptive run. The proposed output-based adaptive algorithm is used. A requested error level of $e_o = 2.5\%\ C_L$ was prescribed. Right: final adapted grid.

yielding two additional fine grids corresponding to $n = 2$ and $n = 4$. The $n = 4$ grid contains 81840 nodes. The final lift value is obtained by performing a Richardson extrapolation on the three grids in the series. The implied convergence rate from the extrapolation is first order.

It is apparent from Figs. 22 and 23 that the output-based algorithm terminated well within the prescribed error limits for the lift. Note that the weak shock on the lower surface is only slightly refined in the $e_o = 5\%\ C_L$ case. Evidently, refining the lower shock is not as important as refining the leading edge, upper shock region, and trailing edge for the given error tolerance. When the tolerance is lowered to $e_o = 2.5\%\ C_L$, however, it becomes beneficial to resolve the weaker shock as well.

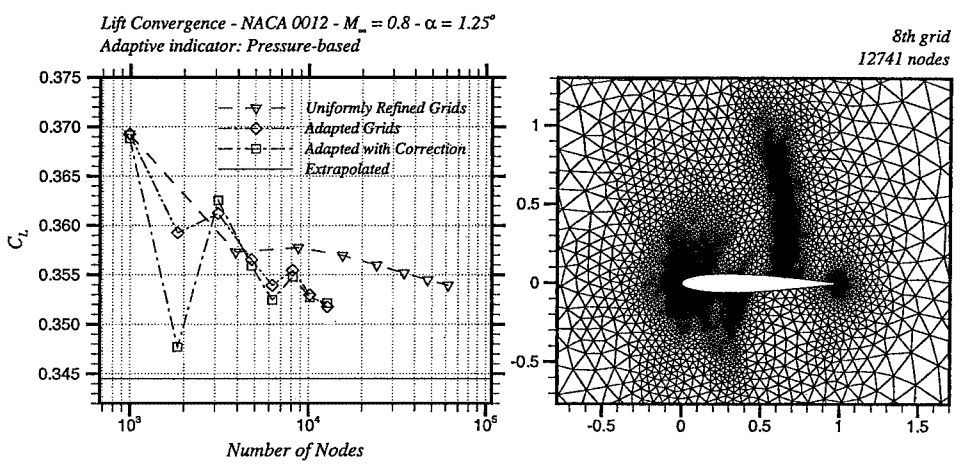

FIG. 24. NACA 0012 test case: $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Left: error convergence in the computed lift coefficient during a typical adaptive run. A gradient-based adaptive strategy is used [18] with an indicator based on the magnitude of the second derivatives in the pressure scaled with H_k^2 where H_k is the local element size. Right: final adapted grid.

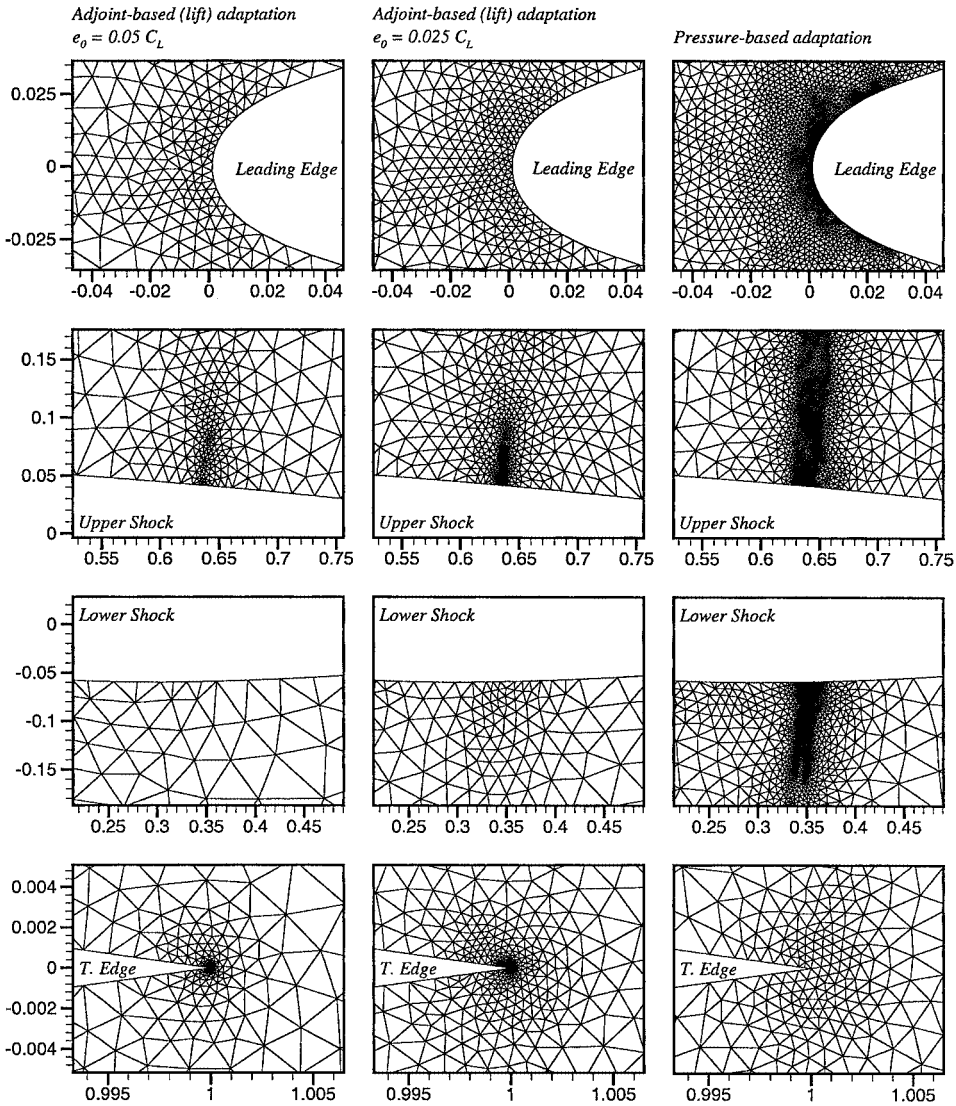

FIG. 25. NACA 0012 test case: $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Comparison of final adapted grids using the proposed output-based method with $e_o = 5\%\ C_L$ (left), $e_o = 2.5\%\ C_L$ (middle), and the pressure-based method (right). Top row: blow-up of leading edge region; 2nd row: shock region on suction side; 3rd row: shock region on pressure side; bottom row: trailing edge region.

We see from Fig. 24 that the pressure-based adaptive method tends to refine the shock structure further away from the airfoil surface, where it is less important to have high resolution with respect to the lift.

Figure 25 shows blow-ups of the final refined grids for each of the adaptive runs. The pressure-based scheme refines the leading edge and shock regions significantly more than does the output-based method; however, it comparatively underrefines the trailing edge. In light of the convergence results, we conclude that the resolution of the trailing edge region is critical for computing an accurate lift in this test case. This is further elucidated in Fig. 26 where plots of the x-momentum adjoint variable are shown. The magnitude of the adjoint is plotted using a logarithmic contour distribution. It is evident from the right-hand plot that

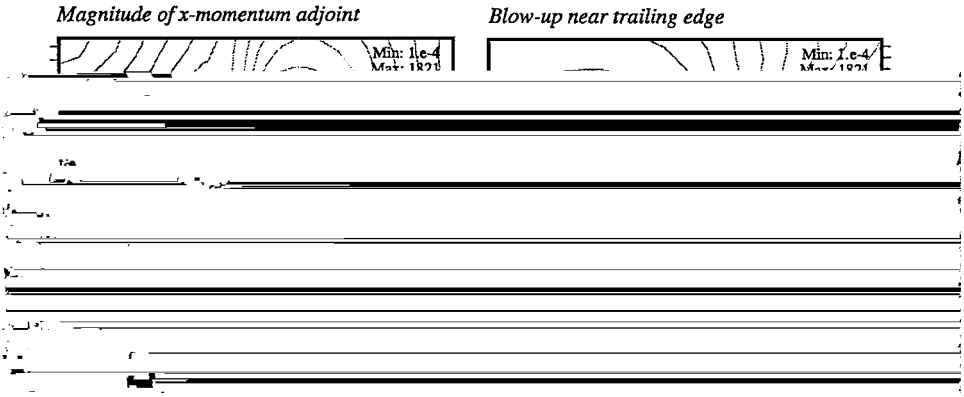

FIG. 26. NACA 0012 test case: $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Adaptive adjoint solution using the proposed output-based method with $e_o = 2.5\%\ C_L$. Left: magnitude of the computed x-momentum adjoint variable based on the lift. Right: blow-up near trailing edge.

the magnitude of the adjoint is very large at the trailing edge, implying that the functional error is extremely sensitive to local residual errors in the trailing edge region. Other features of interest include a singularity in the adjoint along the stagnation streamline and a weak discontinuity upstream of the primal shock on the upper surface.

6. CONCLUSION

An error estimation and grid adaptive procedure was presented for improving the accuracy of specified integral outputs such as lift or drag. The procedure is based on a discrete adjoint formulation in which the estimated error in the functional can be directly related to the local residual errors of both the primal and adjoint solutions. The effectiveness of the adaptive/estimation procedure was demonstrated by application to several two-dimensional, inviscid test cases. Comparisons were made with a commonly used gradient-based adaptive scheme. In many cases, the gradient-based scheme either failed to self-terminate or produced erroneous values for the predicted functional at termination. For every test case considered, the proposed output-based scheme succeeded in self-terminating and surpassed the prescribed accuracy level for the chosen functional.

Future work will include the development and implementation of output-based adaptive procedures for Navier–Stokes computations. Many important viscous problems encountered in industry involve multiple scales and isolated flow features such as shocks, wakes, and boundary layers. It is not always clear, even for the most experienced practitioners of CFD, how the grid should be adapted in order to enhance the accuracy of important outputs while maintaining computational efficiency. It is evident that some of the most commonly used gradient-based adaptive algorithms cannot achieve this reliably. The potential benefit of a robust output-based adaptive scheme is that these outputs can be predicted accurately, efficiently, and in an automated manner, saving valuable time and resources during the design phase.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the National Science Foundation (ACI-9896341), The Boeing Company, and NASA Langley (NAG1-2275) during various stages of this research. Thanks are due to

Dr. W. K. Anderson and Dr. E. J. Nielsen for their valued support in using and modifying FUN2D. Insightful comments and suggestions were offered by Dr. J. T. Thomas of NASA Langley and by Professor M. B. Giles of the Oxford Computing Laboratory during the course of this work. The use of AFLR2, developed by Professor D. L. Marcum of Mississippi State University, is greatly appreciated. The first author also acknowledges FCAR (Fonds pour la Formation de Chercheurs et l'Aide à la Recherche) for its support.

REFERENCES

1. M. Ainsworth and J. T. Oden, *A posteriori* error estimation in finite element analysis, *Comput. Meth. Appl. Mech. Eng.* **142**, 1 (1997).
2. I. Babuška and A. Miller, The post-processing approach in the finite element method—Part 1: Calculation of displacements, stresses and other higher order derivatives of the displacements, *Int. J. Numer. Meth. Eng.* **20**, 1085 (1984).
3. T. J. Baker, Mesh adaptation strategies for problems in fluid dynamics, *Finite Elements Anal. Design* **25**, 243 (1997).
4. R. Becker and R. Rannacher, Weighted *a posteriori* error control in finite element methods, in *Proceedings of ENUMATH-97, Heidelberg* (World Scientific, Singapore, 1998), pp. 621–637.
5. J. C. Carette, *Adaptive Unstructured Mesh Algorithms and SUPG Finite Element Method for Compressible High Reynolds Number Flows*, Ph.D. thesis (von Karman Institute for Fluid Dynamics, Belgium, 1997).
6. S. R. Chakravarthy and S. Osher, Numerical experiments with the Osher upwind scheme for the Euler equations, *AIAA J.* **21**, 1241 (1983).
7. W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M. G. Vallet, Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part I: General principles, *Int. J. Numer. Meth. Fluids* **32**, 725 (2000).
8. M. G. Larson and T. J. Barth, *A posteriori Error Estimation for Discontinuous Galerkin Approximations of Hyperbolic Systems*, NAS Technical Report NAS-99-010 (1999).
9. J. C. Lin and C. J. Dominick, *Optimization of an Advanced Design Three-Element Airfoil at High Reynolds Numbers*, Technical Paper 95-1858 (AIAA Press, Washington, DC, 1995).
10. L. Machiels, J. Peraire, and A. T. Patera, *A posteriori* finite element output bounds for the incompressible Navier–Stokes equations: Application to a natural convection problem, submitted for publication.
11. D. L. Marcum and N. P. Weatherill, Unstructured grid generation using iterative point insertion and local reconnection, *AIAA J.* **33**(9), 1619 (1995).
12. E. J. Nielsen and W. K. Anderson, Recent improvements in aerodynamic Optimization on Unstructured Meshes, Technical Paper 2001-0596 (AIAA Press, Washington, DC, 2001).
13. J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz, Adaptive remeshing for compressible flow computations, *J. Comput. Phys.* **72**, 449 (1987).
14. N. A. Pierce and M. B. Giles, Adjoint recovery of superconvergent functionals from PDE approximations, *SIAM Rev.* **42**(2), 247 (2000).
15. Y. Saad and M. H. Schultz, GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
16. G. Strang and G. J. Fix, *An Analysis of the Finite Element Method* (Wellesley–Cambridge Press, Wellesley, MA, 1988).
17. D. A. Venditti and D. L. Darmofal, Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow, *J. Comput. Phys.* **164**, 204 (2000).
18. G. P. Warren, W. K. Anderson, J. T. Thomas, and S. L. Krist, Grid convergence for adaptive methods, Technical Paper 91-1592 (AIAA Press, Washington, DC, 1991).
19. O. C. Zienkiewicz, *The Finite Element Method*, 3rd ed. (McGraw–Hill, London, 1977).
20. O. C. Zienkiewicz and J. Z. Zhu, A simple error estimation and adaptive procedure for practical engineering analysis, *Int. J. Numer. Meth. Eng.* **24**, 337 (1987).