ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 187 (2003) 22–46

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows

David A. Venditti, David L. Darmofal *

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Room 37-442 Cambridge, MA 021139, USA

Received 19 September 2002; accepted 3 February 2003

Abstract

An anisotropic, unstructured grid adaptive method is presented for improving the accuracy of functional outputs of viscous, compressible flow simulations for general discretizations. The procedure merges output error control with Hessian-based anisotropic grid adaptation. An adjoint formulation is used to relate the estimated functional error to the local residual errors of both the primal and adjoint solutions. This relationship allows local error contributions to be used as indicators in a grid adaptive method designed to produce specially tuned grids for accurately estimating the chosen functional. Element stretching and orientation information is obtained from interpolation error estimates for linear triangular finite elements. The proposed adaptive method is implemented using a standard second-order upwind finite volume discretization, although the procedure is applicable to other types of discretizations such as the finite element method. A series of airfoil test cases, including separated, high-lift flows, are presented to demonstrate the approach; the functionals considered are the lift and drag coefficients. The proposed adaptive method is shown to be superior in terms of reliability and output accuracy relative to pure Hessian-based adaptation.
© 2003 Elsevier Science B.V. All rights reserved.

Keywords: Anisotropic; Unstructured grid adaptation; A posteriori error estimation; Functional output; Duality; Adjoint methods; Hessian; Navier–Stokes; Viscous flow; Turbulent flow; Finite volume method; Finite element method; Computational fluid dynamics; Aerodynamics

1. Introduction

Inappropriate grid resolution has been identified as one of the dominant reasons for inaccurate computational fluid dynamics (CFDs) predictions of aerodynamic forces in the published literature. Rumsey and Ying [44] have conducted a survey of recent CFD methods applied to high-lift configurations and have concluded that numerical error often supersedes modeling error as the primary reason for inconsistent lift predictions. Similarly, grid quality has been cited as a contributing factor to the undesirable level of scatter

* Corresponding author. Tel.: +1-617-258-0743; fax: 1-617-258-5143.
E-mail address: darmofal@mit.edu (D.L. Darmofal).

0021-9991/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00074-3

in the estimation of drag during a recent CFD drag prediction workshop [27]. While automatic grid adaptation has shown significant potential for improving the accuracy of CFD [1,7,12,13,22,30, 39,41,53,54,56], to date these methods have not been able to demonstrate reliable improvements in the solution accuracy.

A major difficulty in achieving definite improvements using current adaptive methods for Euler and Navier–Stokes calculations is the lack of reliable error indicators with which to drive the adaptive algorithms. For example, a common strategy has been to adapt to certain physical features of the flow, such as shock waves and boundary layers, by employing indicators based on large flow gradients or undivided differences [7,41,53]. Unfortunately, continuous local refinement of the dominant features of the flow does not necessarily guarantee a corresponding reduction in the overall discretization error. In some cases this procedure may even lead to incorrect results [53]. An alternate approach has been to use adaptive indicators based on interpolation error estimates for linear finite elements [13,22,39,53,55]. These indicators essentially adapt to the curvature of the solution and share some of the potential deficiencies associated with feature-based indicators for nonlinear flow problems.

Even within the setting of finite-element-discretized elliptic problems, where rigorous error estimates have been available for decades [2,46], one can argue that a global error norm based directly on the solution and its derivatives may not be optimal within an engineering context. The issue is even less clear for practical Euler and Navier–Stokes computations that exhibit multiple length scales and isolated flow features throughout complicated, multidimensional domains. Adaptation based on an error norm of this nature generally leads to a somewhat uniform consideration of all the features in a flow, which may not be desirable. For example, if one is interested in computing the drag on an aircraft, it may not be optimal, or even appropriate, to refine all the shocks and wakes in the flow field to the same extent.

An alternate approach to making error estimation more relevant for engineering applications is to assess the error made in predicting an integral quantity representing an engineering output. Examples include the lift and drag forces on aircraft configurations, the total heat flux to a high-pressure turbine blade, the acoustic noise levels at an airport terminal due to an aircraft taking off, or the rate of ice formation on an aircraft wing during adverse flight conditions. There has been a significant volume of research into a posteriori error analysis and optimal grid adaptation for functional outputs within the context of finite element methods for fluid dynamics. Becker, Rannacher and collaborators [8,9,11,42] have developed an optimal control approach for output-based grid adaptation within a Galerkin finite element framework. Patera, Peraire and collaborators [29,37,38] have established an implicit a posteriori procedure for computing upper and lower bounds on functional outputs of finite element solutions. Other researchers in the area include Süli and co-workers [18,24,47], Larson and Barth [25], and Formaggia et al. [16]. For general discretizations, Pierce and Giles [19–21,40] have developed an adjoint-based error correction technique for functional outputs that extends superconvergence properties, automatically inherent in many finite element methods, to cover numerical results from any numerical method, including finite difference, finite volume, or finite element without natural superconvergence.

The goal of our work is to improve the reliability, accuracy, and efficiency of CFD through the development and application of an error estimation and grid adaptive method for improving the accuracy of functional outputs. The error estimation procedure is based on an algebraic version of the Pierce and Giles correction technique. It shares the advantage of being applicable to any type of discretization method, including finite volume methods which are widely used for the simulation of compressible flows. The functional correction terms and adaptive indicators used in this work have already been incorporated into isotropic adaptive schemes for finite volume discretizations of quasi-one-dimensional [51], two-dimensional [52], and three-dimensional [36] inviscid flows, and for a Galerkin finite element discretization for low-Peclet-number convection–diffusion [50].

This paper focuses on a finite volume implementation of the proposed adaptive method for two-dimensional, viscous (laminar and turbulent), compressible flows and generalizes the approach for

anisotropic adaptation. The proposed anisotropic adaptive procedure and laminar results were first presented in [50]. The method can be viewed as a merging of output error control with Hessian-based anisotropic grid adaptation. The local element size in the grid is determined from output-based error estimates, whereas stretching and orientation information is obtained from interpolation error estimates for linear finite elements [13,22,39]. To the authors' knowledge, this work represents the first implementation of output-based grid adaptation for viscous flows within a finite volume framework. It is also the first anisotropic adaptive procedure for functional outputs of compressible Navier–Stokes simulations for either the finite volume or finite element methods, and represents the first application of adjoint error correction for Navier–Stokes outputs.

The paper is organized as follows: Section 2 presents the a posteriori error estimation/correction procedure for functional outputs that is used to drive the proposed adaptive algorithm. The presentation is cast in a general framework, without reference to a particular discretization method. The procedure is based on a discrete adjoint formulation that relates the local residual errors in the flow solution to the global error in the output. The functional error is decomposed into two contributions: a computable term and a remaining error term. The computable term approximates the leading order error in the functional to high accuracy and is, therefore, used to correct the functional explicitly. The proposed output-based adaptive methodology, described in Section 3, is designed to enhance the accuracy of the corrected functional by working to reduce the magnitude of the remaining error after correction. Local elemental contributions to the remaining error are driven towards the same value throughout the domain, while the estimated global remaining error is reduced towards a user-specified tolerance. Section 4 discusses several issues pertaining to the application of the proposed output-based adaptive methodology to viscous flows. Section 5 presents the proposed procedure for incorporating output error control into a Hessian-based framework for anisotropic adaptation. Numerical results are presented in Section 6. The proposed output-based adaptive method is applied to a series of laminar and turbulent airfoil test cases and comparisons are made with pure Hessian-based adaptation [13,22,39]. The proposed method is shown to be superior in terms of reliability and output accuracy relative to pure Hessian-based adaptation.

2. Functional error estimation

The primary goal is the robust, accurate estimation of an integral output $f(U)$ that can be written as a nonlinear functional of the solution U to a system of partial differential equations (PDEs) defined over a physical domain Ω. Consider two distinct computational grids that approximate the physical domain: a coarse grid Ω_H and a fine grid Ω_h. The parameters H and h ($H > h$) represent characteristic lengths associated with each grid such as the average edge length in a finite element or finite volume triangulation, or the average grid spacing in a finite difference approximation. The nonlinear system of discrete residual equations arising from some discretization of the original PDEs on the coarse grid is denoted $R_H(U_H) = 0$, where U_H is the corresponding discrete solution on that grid. The discrete approximation of $f(U)$ on the coarse grid using a prescribed quadrature rule is denoted $f_H(U_H)$. Analogous quantities are defined for the fine grid as summarized in Table 1.

Table 1
Nomenclature associated with coarse- and fine-grid quantities

	Coarse grid (Ω_H)	Fine grid (Ω_h)
Characteristic element size	H	h
Discrete solution	U_H	U_h
Residual equations	$R_H(U_H) = 0$	$R_h(U_h) = 0$
Functional output	$f_H(U_H)$	$f_h(U_h)$

Consider the *coarse grid* as being representative of a typical working grid. While the coarse grid may be regarded as affordable with respect to available computing resources and allowable solution times, it may not provide sufficient accuracy for the predicted output $f_H(U_H)$. The *fine grid* is considered to be a grid of improved resolution relative to the coarse grid. Solving on the fine grid would generally be expensive, however, if the discrete solution were obtained on this grid, the computed output would have increased accuracy for the particular application. In the present work, the fine grid is taken to be a uniform refinement of the coarse grid. For two-dimensional triangulations, Ω_h can be constructed by subdividing each of the triangles of Ω_H into an integer number, N^2, of self-similar triangles where $N = H/h$. At the boundaries, the fine grid is made to conform to the physical boundary $\partial\Omega$.

2.1. Functional correction

In the following development, the focus will be on deriving a correction term for the error in the fine-grid output due to a perturbation in the fine-grid solution. In the present context, the perturbed solution will be obtained by prolongating the coarse-grid solution onto the fine grid using polynomial reconstruction, as described in Section 2.2. In general, however, it could be obtained from other sources such as an asymptotic expansion, a reduced order model, or an entirely different discretization. The impetus for this procedure is the correction technique of Pierce and Giles [21,40].

Let δU_h represent a small perturbation in the primal solution. Each component of the vector δU_h corresponds to the local error in the perturbed solution relative to the exact solution of the primal residual equations on the fine grid. The perturbed solution is denoted $\tilde{U}_h$ and is defined by

$$\tilde{U}_h \equiv U_h + \delta U_h. \tag{1}$$

In practice, the fine-grid solution U_h and the error δU_h are not known, whereas, the approximate solution $\tilde{U}_h$ is assumed given. The resulting perturbations in the functional and residual operators due to the perturbation in the solution are, respectively,

$$\delta f_h \equiv f_h(\tilde{U}_h) - f_h(U_h), \tag{2}$$

and,

$$\delta R_h \equiv R_h(\tilde{U}_h) - R_h(U_h) = R_h(\tilde{U}_h). \tag{3}$$

The last equality holds due to the fact that the fine-grid solution satisfies the residual equations $R_h(U_h) = 0$. Linearizing about the perturbed solution yields,

$$\delta f_h \approx \frac{\partial f_h}{\partial U_h}\delta U_h, \tag{4}$$

and,

$$R_h(\tilde{U}_h) \approx \frac{\partial R_h}{\partial U_h}\delta U_h. \tag{5}$$

The row vector, $\partial f_h/\partial U_h$, contains the linear sensitivities of the functional with respect to the solution vector. The square matrix $\partial R_h/\partial U_h$ is the Jacobian of the residual operator. Both quantities are evaluated using the perturbed solution.

The discrete adjoint system of equations associated with the primal residual operator and functional is given by

$$\frac{\partial R_h}{\partial U_h}^{\mathrm{T}} \Psi_h = \frac{\partial f_h}{\partial U_h}^{\mathrm{T}}, \tag{6}$$

where Ψ_h is the discrete adjoint solution on the fine grid. Using (4)–(6), the perturbation in the functional can be expressed as the inner product of the adjoint solution and the primal residual error,

$$\delta f_h \approx \Psi_h^{\mathrm{T}} R_h(\tilde{U}_h). \tag{7}$$

This expression is exact for linear functionals and residuals. To compute this estimate would require the solution of the adjoint problem on the fine grid, which is undesirable. Instead, it is assumed that an approximate adjoint solution $\tilde{\Psi}_h$ is available as a substitute. In a fashion analogous to (1), we define the adjoint perturbation $\delta\Psi_h$ by

$$\tilde{\Psi}_h \equiv \Psi_h + \delta\Psi_h. \tag{8}$$

Note that the perturbation in the adjoint is independent of the perturbation in the primal. Using (7) and (8), the perturbation in the functional can be expanded as

$$\delta f_h \approx \Psi_h^{\mathrm{T}} R_h(\tilde{U}_h) = \tilde{\Psi}_h^{\mathrm{T}} R_h(\tilde{U}_h) - \delta\Psi_h^{\mathrm{T}} R_h(\tilde{U}_h). \tag{9}$$

The term, $\tilde{\Psi}_h^{\mathrm{T}} R_h(\tilde{U}_h)$, is computable given the approximate solutions $\tilde{U}_h$ and $\tilde{\Psi}_h$. This term can be used to correct the perturbed functional yielding the following estimate for the fine-grid output:

$$f_h(U_h) \approx f_h(\tilde{U}_h) - \tilde{\Psi}_h^{\mathrm{T}} R_h(\tilde{U}_h). \tag{10}$$

Supplemental discussion pertaining to computer implementation and comparisons with the Pierce and Giles correction technique can be found in [50,52].

2.2. Prolongation operators

This section describes the linear and quadratic operators, L_h^H and Q_h^H, respectively, used for prolongating the coarse-grid primal solution onto the fine grid. Polynomial interpolants are constructed over each coarse-grid element and then used to inject values directly onto the embedded fine grid nodes within that coarse-grid element. The adjoint prolongation operators $\bar{L}_h^H$ and $\bar{Q}_h^H$ are modified for extrapolation at the primal Dirichlet boundaries to account for the non-uniform character of the discrete adjoint at those boundaries, as discussed in Section 4.2.

The linear operator L_h^H represents simple linear interpolation over each coarse-grid element. The resulting interpolant matches the data at the coarse-grid nodes exactly and is, therefore, continuous across element edges. This operator is used for computing the adaptive parameters described in Section 3.2.

The quadratic prolongation operator Q_h^H is defined by a local least squares procedure [50,52]. A quadratic profile is obtained over each coarse-grid element using a local error-minimization process involving the nodal values and slopes of the function being interpolated. This process produces a piecewise quadratic interpolant that is discontinuous across element edges. Values on edges are obtained by taking the arithmetic average of the interpolant values from the two adjacent elements associated with that edge. Values at coarse-grid nodes are similarly obtained by taking the arithmetic average of the interpolant values from the patch of elements surrounding that node. The least squares problem can be solved efficiently by inverting the associated normal equations. Unfortunately, the normal equations become severely ill-conditioned on highly stretched elements resulting in numerical instabilities. To circumvent this issue, the least squares problem is solved using a singular value decomposition (SVD) algorithm [49]. The quadratic prolongation operator is used for evaluating the correction term in (10), and for computing the adaptive parameters described in Section 3.2.

3. Output-based adaptive methodology

The proposed output-based adaptive methodology is designed to compliment the functional correction procedure of Section 2.1. The goal of the adaptive algorithm is to improve the accuracy of the corrected output in (10) by reducing the remaining error after correction.

3.1. Remaining error

As shown in (9), the error in the functional can be expressed as the sum of two terms: a computable correction that can be evaluated given the approximate solutions $\tilde{U}_h$ and $\tilde{\Psi}_h$, and a remaining error term that generally cannot be evaluated without solving for quantities on the fine grid. The remaining error can be written in several different forms. Two forms that are particularly useful in the present context are,

$$\underbrace{\delta f_h}_{\text{Error}} - \underbrace{\tilde{\Psi}_h^{\mathrm{T}} \mathrm{R}_h(\tilde{\mathrm{U}}_h)}_{\text{Correction}} \approx \underbrace{-\delta \Psi_h^{\mathrm{T}} \mathrm{R}_h(\tilde{\mathrm{U}}_h)}_{\text{Remaining Error}} \approx \underbrace{-\mathrm{R}_h^{\Psi}(\tilde{\Psi}_h)^{\mathrm{T}}\, \delta \mathrm{U}_h}_{\text{Remaining Error}}, \tag{11}$$

where $R_h^{\Psi}(\cdot)$ is the adjoint residual operator defined as

$$R_h^{\Psi}(\cdot) \equiv \frac{\partial R_h}{\partial U_h}^{\mathrm{T}} (\cdot) - \frac{\partial f_h}{\partial U_h}^{\mathrm{T}}, \tag{12}$$

so that $R_h^{\Psi}(\Psi_h) = 0$. We see from (11) that the remaining error can be expressed as the inner product of the adjoint solution-error and the primal residual-error, or as the inner product of the adjoint residual-error and the primal solution-error. Neglecting nonlinear terms, these two inner products are equal. 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 the expansion for the residual operator in (5), one can obtain the following expression for the duality gap:

$$D \equiv R_h^{\Psi}(\tilde{\Psi}_h)^{\mathrm{T}}\, \delta U_h - \delta \Psi_h^{\mathrm{T}} R_h(\tilde{U}_h), = \delta \Psi_h^{\mathrm{T}} W, \tag{13}$$

where W is a vector containing quadratic forms of the primal error. An explicit expression for W is given in [50,51]. 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 (13). 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 primal form of the remaining error, $\delta \Psi_h^{\mathrm{T}} R_h(\tilde{U}_h)$, is essentially what is used by Becker and Rannacher [8,9] in their finite-element output-based adaptive strategy. The addition of the dual term in the current approach is a natural way of incorporating the adjoint residual error into the adaptive scheme. Utilizing information from both the primal and adjoint residuals is expected to lead to a more robust adaptive algorithm.

Müller and Giles [31] have adopted a somewhat different philosophy in their adaptive strategy for functional outputs. Instead of using the remaining error in the functional as an adaptive sensor, they focus directly on the correction term $\tilde{\Psi}_h^{\mathrm{T}} R_h(\tilde{U}_h)$. In principle, this leads to an adaptive algorithm for minimizing the magnitude of the correction term. However, if one is to preserve the role of this term as an accurate correction for the functional, it may not be advantageous to minimize its magnitude. Unlike the remaining error terms, the correction term is computable and generally approximates the leading order error in the functional to high accuracy (for example, see the functional convergence results in [21,40,51,52]). The strategy in the present work, therefore, is to take advantage of the improved accuracy from correcting the functional, and to use the adaptive process to further enhance this accuracy by reducing the remaining error.

3.2. Adaptive criteria and parameters

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 remaining error in the functional of interest. The iterative adaptive algorithm works to reduce this bound until it falls below a user-specified tolerance, e_0.

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 r subcomponents to the primal and adjoint residual vectors where $r = 4$ for the two-dimensional (laminar) Navier–Stokes equations (corresponding to the mass, x-momentum, y-momentum and energy conservation equations), and $r = 5$ for the Reynolds-averaged Navier–Stokes (RANS) equations and transport equation for the turbulence model. Eq. (13) suggests the following definition for the adaptation parameter, ε_k, at element k:

$$\varepsilon_k = \frac{1}{2}\sum_{l(k)} \left\{ \left| [R_h^{\Psi}(\bar{L}_h^H \Psi_H)]^{\mathrm{T}}_{l(k)} [Q_h^H U_H - L_h^H U_H]_{l(k)} \right| + \left| [\bar{Q}_h^H \Psi_H - \bar{L}_h^H \Psi_H]^{\mathrm{T}}_{l(k)} [R_h(L_h^H U_H)]_{l(k)} \right| \right\}. \tag{14}$$

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 $r \times 1$ subvector (component) of V_h corresponding to the fine-grid node, $l(k)$, within the coarse-grid element, k. The summation in (14) 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 prolongation operators L_h^H and Q_h^H map the coarse-grid primal solution onto the fine grid via linear and quadratic interpolation, respectively. The corresponding adjoint prolongation operators are denoted $\bar{L}_h^H$ and $\bar{Q}_h^H$, respectively. These operators are described in Section 2.2.

The adaptation parameter in (14) is a crude approximation of the magnitudes of the primal and dual forms of the remaining error in the functional. The intention here is not to obtain a quantitative estimate of these terms, but rather to establish an indication of how they are distributed throughout the domain. The residual operators in the expression for the adaptation parameter are evaluated using linear prolongations of the respective coarse-grid solutions. Linear prolongation is chosen over quadratic in this case because the magnitudes of the residuals tend to be larger, giving a more conservative bound on the remaining error terms. The perturbations δU_h and $\delta \Psi_h$ in (13) have been replaced by measures of the local interpolation error in the primal and adjoint solutions, respectively. In particular, setting $\delta U_h \approx Q_h^H U_H - L_h^H U_H$ and $\delta \Psi_h \approx Q_h^H \Psi_H - L_h^H \Psi_H$ amounts to the assumption that the dominant component of the local error in the primal and adjoint solutions is characterized by the interpolation error.

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

$$\varepsilon = \sum_k \varepsilon_k. \tag{15}$$

With the presence of the absolute value signs in (14), ε represents an upper bound on the estimated remaining error in the functional. In light of this, a global adaptation parameter can be defined as

$$\eta_{\mathrm{g}} = \frac{\varepsilon}{e_0}, \tag{16}$$

where e_0 is the user-specified desired error level. The global error criterion is satisfied if $\eta_{\mathrm{g}} \leqslant 1$ while further refinement is required if $\eta_{\mathrm{g}} > 1$. 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 [7,57]. 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

$$\eta_k = \frac{\varepsilon_k}{\bar{e}_0}, \tag{17}$$

where $\bar{e}_0 = e_0/N_e$ 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 $\eta_k \leqslant 1$ while further refinement of element k is indicated if $\eta_k > 1$. Finally, at each adaptive iteration, a new desired element size, H'_k is computed from the old one, H_k, according to

$$H'_k = H_k \left(\frac{1}{\eta_g \eta_k} \right)^{\omega}. \tag{18}$$

The underrelaxation parameter ω controls how aggressively each subsequent grid is refined during the iterative adaptive process. An inappropriately large value of ω would lead to oscillatory grid convergence, whereby the grid is continually over- and under-refined in an alternating fashion. Conversely, an inappropriately small value for ω 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, and by choosing ω such that $H'_k \sim \mathcal{O}(1)$. We assume a convergence rate of $\eta_g \eta_k \sim \mathcal{O}(H_k^4)$ yielding a heuristic value of $\omega = 1/4$, which is used for the laminar test cases in Section 6. Due to higher anisotropy requirements and limitations associated with the grid generator, values as low as $\omega = 1/8$ are used for the turbulent simulations in order to slow down changes in the grid from one adaptive iteration to the next. Onate and Bugeda [35] provides further discussion on grid convergence for adaptive methods.

4. Application to viscous flow

This section discusses several issues pertaining to the application of the proposed output-based adaptive methodology to two-dimensional, viscous, compressible flow simulations using an unstructured, upwind, finite volume discretization.

4.1. Flow and adjoint solvers

The governing equations for the turbulent simulations are the compressible, Reynolds-averaged Navier–Stokes (RANS) equations. The eddy viscosity is obtained using the one-equation turbulence model of Spalart and Allmaras [45] in fully turbulent mode. Laminar solutions are obtained by omitting the turbulence transport equation and setting the eddy viscosity to zero.

The flow and adjoint solvers used in this work are part of the FUN2D suite of codes [32]. The FUN2D flow solver is a two-dimensional, implicit finite volume scheme that employs unstructured grids composed of triangular elements. Non-overlapping control volumes are constructed around each node in the grid by connecting the centroid of each triangle to the midpoints of its edges. The discrete residual equations associated with each interior node are obtained by numerically integrating the steady form of the governing equations over the control volume surrounding that node. The Roe upwind scheme [43] is used to evaluate the inviscid fluxes at control volume interfaces. States immediately adjacent to the interfaces are obtained by linear extrapolation of the primitive variables from associated vertices. Limiters are not employed in the

present work. A Galerkin-type formulation is used to approximate the viscous terms. Farfield fluxes are evaluated using characteristic reconstructions. The convective terms in the turbulence model are discretized using first-order upwind differencing and the higher-order derivatives are evaluated in the same manner as in the flow solver. The solution is driven to steady state using a backward-Euler local time-stepping method. Further details pertaining to the flow solver can be found in [4]. The adjoint solver utilizes an exact linearization of the flow (primal) residual operator described above. Further details pertaining the adjoint solver can be found in [33,34].

4.2. Strong boundary conditions

The imposition of boundary conditions in a *strong sense* refers to the explicit modification of the form of the residual equations at boundary nodes relative to their counterparts in the interior. This has implications on the character of the discrete adjoint at the corresponding nodes. For RANS simulations, the issue arises when the momentum and energy residuals associated with nodes on solid boundaries are discarded in favor of imposing no-slip and specified temperature conditions, respectively. The same issue arises with the transport equation for the turbulence model when homogeneous Dirichlet conditions are applied to the dependent variable at the walls. Given a perturbation δU_h, the residual on the boundary represents the error in the Dirichlet data, whereas in the remainder of the domain the residual represents a lack of conservation. Correspondingly, the discrete adjoint takes on a different character on the boundary relative to the interior. This behavior has been observed previously [6]. Mathematical illustrations are provided in [17,50].

Fig. 1 shows contour distributions (middle and right plots) of the discrete adjoint x-momentum variable for a simulation of laminar flow past a cylinder. The corresponding grid is plotted on the left. In this case, $Re = 10$, $M_\infty = 0.38$ and the adjoint is based on the drag. The distinction between the boundary adjoint and the interior adjoint is clear from the clustering of contour lines near the boundary. The contour levels in this plot are distributed linearly within each element, however, the sharp variations within the elements adjacent to the boundary are meaningless since two entirely different quantities are being interpolated within these elements.

For the purposes of building a fine-grid representation of the discrete adjoint near strong primal boundaries, direct interpolation of the coarse-grid adjoint is not appropriate. A viable alternative is to extrapolate the interior adjoint to the boundary of the coarse grid and prolongate the extrapolant onto the fine grid. Once the prolongation is performed, the only remaining task is to postprocess the boundary adjoint on the fine grid. A formula for recovering the boundary adjoint given an approximation of the interior adjoint on the fine grid is given in [50] and is based on the derivation presented in [17].

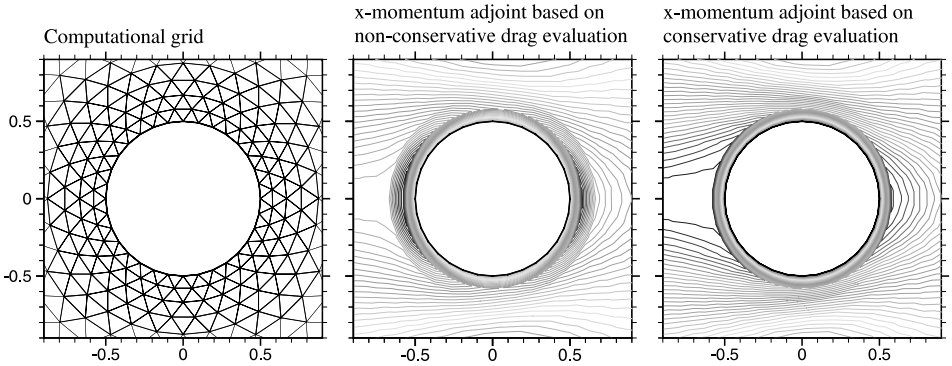

Fig. 1. Comparison of the x-momentum adjoint variable based on a non-conservative (center) and a conservative (right) evaluation of the drag for viscous flow past a circular cylinder ($Re = 10$, $M_\infty = 0.38$). The computation grid is plotted on the left.

4.3. Conservative functional evaluation

Aerodynamic outputs often involve boundary integrals of diffusive-type fluxes which involve gradients of the solution normal to the boundary. In viscous applications, the skin friction contribution to the aerodynamic forces on a body can be expressed as a contour integral involving the gradients of the velocity components at the surface of the body. One method of computing these forces is to differentiate the solution within the elements adjacent to the body, and then integrate the result. Evaluating the forces in this manner generally results in poor convergence of the output. This type of functional definition may also lead to irregularities in the adjoint variables near the corresponding boundaries resulting in compromised accuracy of the corrected functionals and diminished effectiveness of the adaptive algorithm. A more natural and accurate method of evaluating the aerodynamic forces, which is both bounded and conservative, is to define them as residual balances as described below.

Let $\bar{R}_h(U_h)$ represent the vector of residual equations prior to imposing no-slip conditions, specified temperature, and zero eddy viscosity in a strong sense at the solid boundaries. A typical way of constructing these equations in a finite volume method is to loop over all edges in the grid and add the corresponding flux contributions to the associated nodes. The resulting equations are discrete conservation statements defined over control volumes surrounding each interior node in the grid. At the boundary nodes, the momentum, energy, and turbulence model residuals are typically missing flux contributions from the wall. In the case of the momentum residuals, these missing fluxes correspond to the pressure and shear forces that would be needed in order to maintain momentum conservation over the control volumes surrounding the boundary nodes. The sum of these forces over all boundary nodes corresponds to the total pressure and skin friction forces on the body. In this vein [17], the forces can be defined as

$$f_h(U_h) = c_h^{\mathrm{T}} B_h \bar{R}_h(U_h), \tag{19}$$

where B_h is a projection matrix with unit diagonal entries on the rows where Dirichlet conditions are imposed strongly and zeros everywhere else. The vector c_h takes the appropriate component of the momentum residuals into the selected force direction (for example, lift or drag). The right hand side for the discrete adjoint system is given by

$$\frac{\partial f_h}{\partial U_h}^{\mathrm{T}} = \frac{\partial \bar{R}_h}{\partial U_h}^{\mathrm{T}} B_h c_h. \tag{20}$$

Fortunately, this expression can be evaluated using existing subroutines from the discrete adjoint code.

Fig. 1 compares the computed x-momentum adjoint variable near the boundary of a circular cylinder for a non-conservative evaluation (center plot) and a conservative evaluation (right plot) of the drag. The conditions for this laminar flow simulation are $Re = 10$ and $M_\infty = 0.38$. The computational grid is plotted on the left. In the center plot, irregularities in the adjoint are observed in the second layer of elements away from the boundary. These irregularities are not present in the right plot, illustrating that the conservative evaluation of the drag leads to an interior adjoint with better smoothness properties. The clustering of contour lines in the first layer of elements reflects the presence of the boundary adjoint, which is a separate issue as discussed in Section 4.2.

Fig. 2 compares lift and drag convergence plots for the same test case using both types of functional evaluations. These values are obtained from simulations on a series of uniformly refined grids. The farfield boundary is placed at 22.5 diameters. The coarsest grid in the series contains 800 nodes and is shown in Fig. 1. The nth grid in the series is constructed by subdividing each triangle in the coarsest grid into n^2 self-similar triangles for values of $n = 2$, 4, and 8. Boundary edges are made to conform to the circular cylinder. The lift and drag errors are plotted versus a characteristic length parameter h associated with each grid in the series. A value of $h_1 = 1$ is arbitrarily assigned to the coarsest grid. The length parameter associated

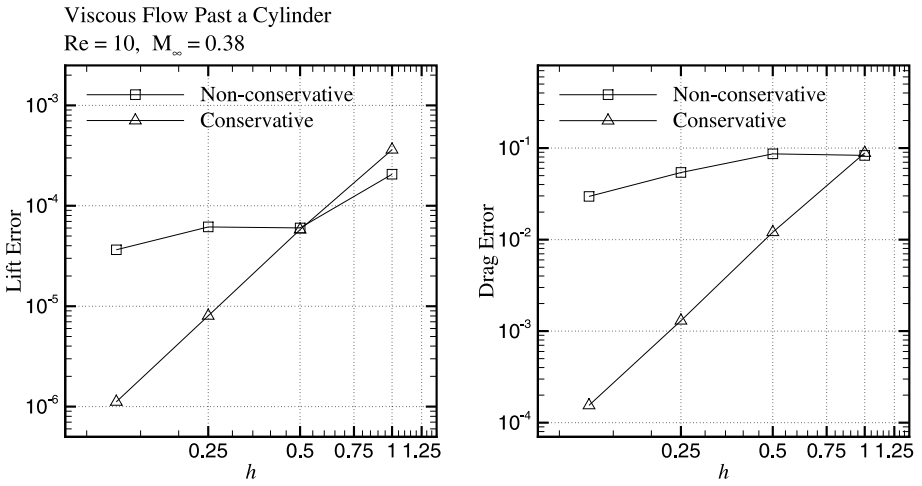

Fig. 2. Convergence of the error in the lift and drag on a series of uniformly refined grids for viscous flow past a circular cylinder ($Re = 10$, $M_\infty = 0.38$). The conservative and non-conservative evaluations of the forces are compared.

with the nth grid is given by $h_n = h_1/n$. The true lift and drag values used to measure the errors in Fig. 2 are taken as $C_L = 0$ and $C_D = 2.99165$, respectively. The true drag value is obtained as a Richardson extrapolation of the conservative drag values from the three finest grids. The implied convergence rate from the extrapolation is approximately third order. We see from Fig. 2 that the non-conservative evaluation produces essentially first-order-accurate values for the forces whereas the conservative evaluation is essentially third order.

5. Anisotropic adaptation

This section presents the proposed output-based, anisotropic adaptive procedure. The section begins by reviewing some of the key elements of anisotropic adaptation based exclusively on interpolation error estimates for linear triangular finite elements [13,22,39]. Then the incorporation of output error control into the Hessian-based framework is described.

5.1. Pure Hessian-based adaptation

Error indicators for driving anisotropic adaptation of unstructured triangular and tetrahedral grids have often been based on the interpolation error incurred by approximating continuous functions using piecewise-linear interpolation [13,22,39,55].

Consider the two-dimensional function $u(x,y)$ and a piecewise-linear approximation $u_h^L(x,y)$ of this function over a given triangulation. Assuming zero error at the nodes, the maximum interpolation error over an edge E in the triangulation, with unit tangent vector $\hat{s}$ and length h, is given by [46]

$$\max_{(x,y)\in E} \left|u(x,y) - u_h^L(x,y)\right| \leqslant \frac{1}{8} h^2 \max_{(x,y)\in E} |u_{ss}|, \tag{21}$$

In this last expression u_{ss} is the directional second derivative of u in the direction $\hat{s}$, which can be expressed as

$$u_{ss} = \hat{s}^{\mathrm{T}} \mathscr{H} \hat{s}, \tag{22}$$

where

$$\mathscr{H} = \begin{bmatrix} u_{xx} & u_{xy} \\ u_{xy} & u_{yy} \end{bmatrix}, \tag{23}$$

is the Hessian matrix of second derivatives.

A geometric interpretation of the interpolation error along an edge can be made by considering the length of that edge in a Riemannian space [10,13,15,22]. The transformation from the physical space to the Riemannian space is specified by a symmetric positive-definite matrix $\mathscr{M}$,

$$\mathscr{M} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}. \tag{24}$$

For constant $\mathscr{M}$, the generalized length $l_{\mathscr{M}}$ of edge E in the transformed space is given by

$$l_{\mathscr{M}}^2 = \hat{s}^{\mathrm{T}} \mathscr{M} \hat{s} h^2. \tag{25}$$

Note that if $\mathscr{M}$ is the identity matrix, the physical (Euclidean) length $l_{\mathrm{E}}^2 = \hat{s}^{\mathrm{T}} I \hat{s} h^2 = h^2$ is recovered.

To illustrate the directional properties of the metric $\mathscr{M}$, consider the unit phasor $\hat{e} = (x, y)^{\mathrm{T}}$ stemming from the origin in the metric space. The square of its metric length is given by

$$\hat{e}^{\mathrm{T}} \mathscr{M} \hat{e} = ax^2 + 2bxy + cy^2 = 1. \tag{26}$$

This is the equation of an ellipse centered at the origin in the physical space. The major axis is rotated by an angle θ to the x-axis, and has major and minor principal lengths h_1 and h_2, respectively. These parameters are related to the metric components (a, b, c) as follows:

$$\mathscr{M} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} = R \begin{bmatrix} 1/h_1^2 & 0 \\ 0 & 1/h_2^2 \end{bmatrix} R^{\mathrm{T}}, \tag{27}$$

where

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{28}$$

Comparing (21) and (22) with (25), the interpolation error along edge E is analogous to the square of the metric length of that edge if $\mathscr{M}$ is chosen to be a modification of the Hessian matrix. In particular, the Hessian is modified by taking the absolute value of its eigenvalues, resulting in a symmetric positive-definite matrix. Let R be the orthonormal matrix containing the eigenvectors of $\mathscr{H}$ as its columns, and let Λ be the corresponding diagonal matrix containing its eigenvalues. The symmetric Hessian is diagonalized by

$$\mathscr{H} = R \Lambda R^{\mathrm{T}}, \tag{29}$$

and the Hessian-based metric can be defined as

$$\mathscr{M}_{\mathscr{H}} = R |\Lambda| R^{\mathrm{T}}. \tag{30}$$

One possible strategy for anisotropic adaptation is to equidistribute the error along edges in the grid by equidistributing the length of the edges in the Riemannian space governed by $\mathscr{M}_{\mathscr{H}}$ [10,13,15,22]. This strategy is implemented here for the purposes of comparing the effectiveness of the output-based strategy described in Section 5.2. The algorithm attempts to equidistribute the interpolation error in the computed Mach number over each edge in the grid. Different error tolerances are achieved by scaling the metrics

associated with each element by a constant multiplicative factor κ. Larger values of κ correspond to more stringent tolerances on the estimated interpolation error. The Mach-number Hessian is obtained using the quadratic reconstruction procedure described in Section 2.2. Piecewise quadratic Mach number profiles are constructed over each element and differentiated twice yielding piecewise constant second derivatives. Nodal values of the second derivatives are obtained using an area-weighted average of the piecewise-constant values over each element surrounding the node. To avoid excessive element sizes in the farfield, the eigenvalues of the metric are bounded from below so that $h_2 \leqslant h_1 \leqslant h_{\max}$, where $h_{\max}$ is a specified maximum element length for the domain.

Adaptation based on the Mach-number Hessian alone is known to have several deficiencies. For example, curves of inflection in the Mach-number distribution may lead to inappropriate grid stretching in certain regions of the domain, and inadequate resolution of the flow field may occur in regions where the magnitude of the Hessian is close to zero. Heuristic remedies for these deficiencies have been proposed in the literature [13,55], however, no explicit steps are taken in the current implementation to deal with them. Results show (see Section 6) that the proposed output-based method is able to mitigate these effects while maintaining effective control of the functional error.

5.2. Output error control

The output-based adaptive parameters derived in Section 3.2 are incorporated into an anisotropic grid-adaptive framework for functional outputs.

The Riemannian metric introduced in Section 5.1 contains three independent pieces of information that can be used for anisotropic grid adaptation. In its original form the metric is specified by the components (a, b, c). Through diagonalization, the metric can also be characterized by the parameters (h_1, h_2, θ), as shown in (27) and (28). In the present context it is convenient to decompose the metric into the parameters (H, β, θ), which represent the local size, stretching, and orientation of the elements, respectively. These parameters are defined by

$$(H, \beta, \theta) \equiv (h_2, h_1/h_2, \theta) \tag{31}$$

In the proposed output-based adaptive method, the stretching and orientation parameters, β and θ, are obtained from the Mach number Hessian whereas the local size parameter H is determined from the adjoint criteria.

In a typical output-based adaptive simulation, several adaptive iterations are required in order to achieve grid convergence. Each iteration involves a flow and adjoint solution on the current grid, and an adaptive step in which a new grid is generated. For each element in the current grid, the new size, stretching and orientation parameters (H', β', θ') are computed for the proposed new grid. The stretching and orientation parameters, β' and θ', are obtained directly from the Hessian-based metric. The new size parameter H' is expressed as the product of an adjoint-based factor and the current element-size parameter H.

The following describes a method for determining the size parameter H associated with an element in the current grid. If it is assumed that the current grid is optimal in some metric, then the metric lengths of all the edges in the grid are constant. Correspondingly, the local metric value for an element can be approximated using (26). Specifically, consider the triangle shown in Fig. 3. For each of its edges $k \in [1, 3]$, the following holds, approximately [1]:

$$a(\Delta x_k)^2 + 2b\Delta x_k \Delta y_k + c(\Delta y_k)^2 = 1. \tag{32}$$

[1] The appropriate constant on the right-hand-side of (32) is dependent on the standard employed by the particular grid generator. In BL2D and BAMG, physical lengths are mapped onto a metric length of unity [23,26].

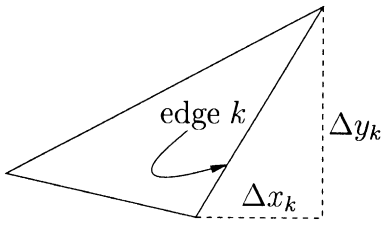

Fig. 3. Typical triangle in the current grid.

This results in the system of equations

$$\begin{bmatrix} (\Delta x_1)^2 & 2\Delta x_1 \Delta y_1 & (\Delta y_1)^2 \\ (\Delta x_2)^2 & 2\Delta x_2 \Delta y_2 & (\Delta y_2)^2 \\ (\Delta x_3)^2 & 2\Delta x_3 \Delta y_3 & (\Delta y_3)^2 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \tag{33}$$

which can be solved for the metric components (a, b, c). These components represent an average metric for the triangle. The size parameter H can be obtained from the largest singular value $\sigma_{\max}$ of the implied metric via [2]

$$H = \frac{1}{\sigma_{\max}^{1/2}}. \tag{34}$$

The approximate grid-implied metric for a tetrahedron can be obtained in an analogous fashion. In three dimensions, the Riemannian metric is a symmetric positive-definite 3×3 matrix specified by six components. Tetrahedra have six edges over which six equations analogous to (32) can be written and solved for the six unknown components.

5.3. Proposed adaptive procedure

Prior to an output-based adaptive simulation, one must specify the functional of interest $f(U)$, a desired tolerance on the functional error e_0, and a scalar field for the Hessian calculations. For the test cases in this paper, the Hessians are computed from the Mach number distribution.

A single iteration in the output-based procedure is comprised of the following steps:

1. Obtain the flow and adjoint solutions on the current grid.
2. For each element, compute the piecewise-constant Mach number Hessians using the quadratic reconstruction procedure described in Section 2.2.
3. For each element, compute the Hessian-based metric $\mathscr{M}_{\mathscr{H}}$ using (30), by diagonalizing the Hessian and taking the absolute value of its eigenvalues. Compute the desired new stretching and orientation parameters, β' and θ', using (27), (28) and (31).
4. For each element, estimate the grid-implied metric as described in Section 5.2, and compute the current element-size parameter H.
5. Compute the adjoint-based adaptation parameters η_g and η_k derived in Section 3.2. Specifically, $\eta_g = \varepsilon/e_0$ and $\eta_k = \varepsilon_k/\bar{e}_0$ for each element k, where $\bar{e}_0 = e_0/N_e$ is the element target error, N_e is the total number of elements in the current grid, $\varepsilon = \sum_k \varepsilon_k$, and ε_k is obtained from (14).

[2] The true metric is symmetric positive-definite; its eigenvalues are positive and equal to its singular values. The implied metric is not guaranteed to be positive definite, therefore, the size parameter H is computed in terms of the largest singular value, which is equal to the absolute value of the largest eigenvalue in magnitude.

6. For each element, the desired new element size H' is obtained in terms of the current size H using (18).
7. For each element, construct the new output-based metric $\mathscr{M}'$ for the next grid using the parameters (H', β', θ'). Transfer the elemental metric components to the nodes using area-weighted averages of the piecewise-constant values over the elements surrounding each node.
8. Input the nodal metrics into the anisotropic grid generator and regenerate the grid.

The iterative process is considered converged when the change in the total number of nodes from one grid to the next is on the order of 1% or less ($\sim 5\%$ for very coarse grids). If the desired error tolerance e_0 is very low relative to the resolution of the initial grid, excessive overrefinement may result in the early stages of the adaptive process. To overcome this difficulty, a modest (large) value for e_0 can be prescribed initially and then gradually ramped down to the desired error level over the course of several iterations.

The grid generators used for the laminar and turbulent adaptive simulations in this work are BL2D [10,26] and BAMG [23], respectively. These grid generators are incorporated into the current adaptive framework using shell scripts. Information regarding their usage, capabilities, and algorithms can be found in the references.

6. Numerical results

Several test cases are presented to demonstrate the performance of the proposed output-based, anisotropic adaptive method. Comparisons are made with a standard adaptive method based exclusively on the Mach-number Hessian (see Section 5.1). Pure Hessian-based adaptation is essentially what is used in [13,22,39,55] although their methodologies and implementations differ to varying degrees.

6.1. Laminar flow adaptive results

6.1.1. $Re = 5000$, $M_\infty = 0.5$, $\alpha = 3°$ NACA 0012 airfoil

Adaptive simulations are performed for laminar flow over a NACA 0012 airfoil with free stream conditions $Re = 5000$, $M_\infty = 0.5$ and $\alpha = 3°$. Under these conditions, the flow separates from the suction side of the airfoil at approximately the 50% chord position while remaining attached on the lower surface. The functionals considered for this test case are the lift and drag coefficients. The initial grid in the adaptive simulations for this case is the coarse, inviscid-style grid shown in Fig. 4. The farfield boundary is placed at 15 chords.

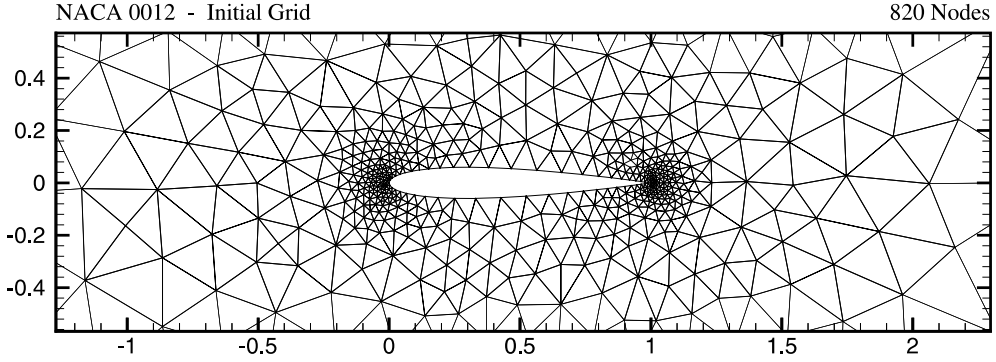

Fig. 4. NACA 0012 airfoil test case: $Re = 5000$, $M_\infty = 0.5$, $\alpha = 3°$. Initial grid in the adaptive runs for this test case.

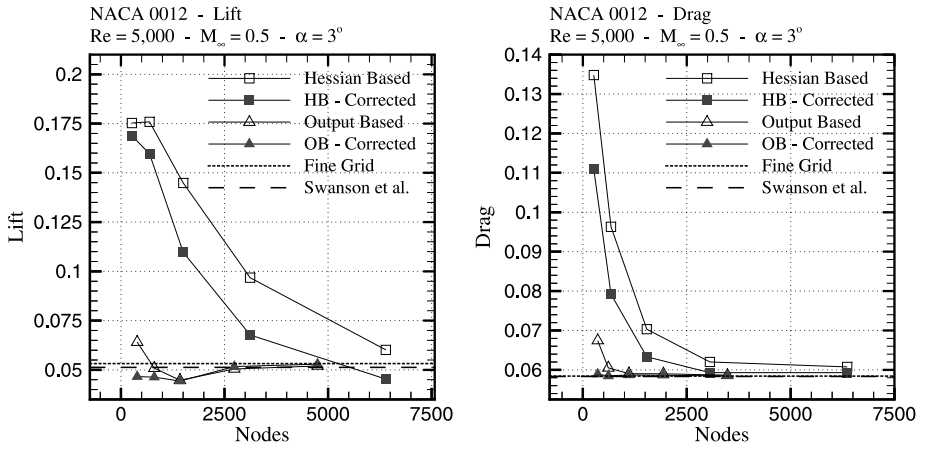

Fig. 5. NACA 0012 airfoil test case: $Re = 5000$, $M_\infty = 0.5$, $\alpha = 3°$. Plots comparing the uncorrected and corrected lift and drag from five lift-based, five drag-based, and five Hessian-based adaptive simulations. The fine-grid values are computed on $N = 2$ subdivisions of the finest adapted grids from the output-based simulations. The Swanson et al. values are obtained from [48].

Lift-based simulations are performed for five different error tolerances: $e_0 = 0.05$, 0.025, 0.0125, 0.00625, and 0.003125. The final adapted grids range in size from 392 nodes ($e_0 = 0.05$) to 4745 nodes ($e_0 = 0.003125$). Drag-based simulations are performed for $e_0 = 0.005$, 0.0025, 0.00125, 0.000625, and 0.0003125 with final grids ranging in size from 358 nodes ($e_0 = 0.005$) to 3479 nodes ($e_0 = 0.0003125$). Pure Hessian-based adaptation is performed for five multiplicative factors: $\kappa = 25$, 50, 100, 200, and 400. The final grids range in size from 263 nodes ($\kappa = 25$) to 6395 nodes ($\kappa = 400$).

The left plot in Fig. 5 shows the final base and corrected lift values from the lift-based and Hessian-based adaptive simulations. The fine-grid value is computed on a single fine-grid ($N = 2$) corresponding to the finest adapted grid from the lift-based simulations. This fine grid contains 18746 nodes. The Swanson et al. value is obtained from [48]. Swanson et al. use a finite volume discretization that employs a structured C-type grid with 512×128 cells and farfield boundary at 10 chords. The right plot in Fig. 5 shows analogous results for the drag. The fine-grid value is computed on an $N = 2$ subdivision of the finest adapted grid from the drag-based simulations. This fine grid contains 13 716 nodes. As is evident from Fig. 5, the output-based scheme offers a considerable improvement in accuracy over pure Hessian-based adaptation for grids of comparable size. Furthermore, the requested error tolerance is surpassed for each of the output-based adaptive simulations performed for this test case.

Fig. 6 compares the final adapted grids from the $e_0 = 0.00625$ lift-based simulation (top) and the $\kappa = 200$ Hessian-based simulation (bottom). The inaccurate lift prediction from the Hessian-based simulation is attributed to insufficient grid resolution in the inviscid regions of the flow, and to insufficient resolution of the separation zone on the suction side of the airfoil. The elements adjacent to the upper surface near the trailing edge are inappropriately large, resulting in unacceptable discretization errors in both the flow solution and in the geometric representation of the airfoil.

6.1.2. $Re = 5000$, $M_\infty = 0.5$, $\alpha = 3°$ two element airfoil

Adaptive simulations of laminar flow past two NACA 0012 airfoil elements are presented. The leading edge of the downstream element is positioned four chord-lengths aft of the trailing edge of the upstream element. The flow conditions are $Re = 5000$ (based on the chord of a single element), $M_\infty = 0.5$ and $\alpha = 3°$. At this angle of attack, the wake from the upstream element passes directly over the suction side of the downstream element.

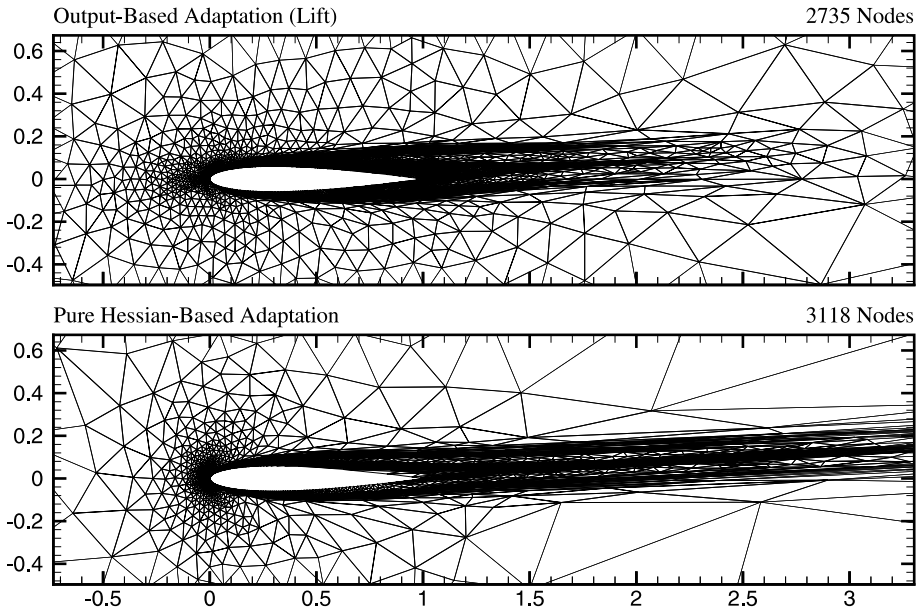

Fig. 6. NACA 0012 airfoil test case: $Re = 5000$, $M_\infty = 0.5$, $\alpha = 3°$. Top: final adapted grid using lift-based adaptation with $e_0 = 0.00625$. Bottom: final grid using pure Hessian-based adaptation with $\kappa = 200$.

Adaptive simulations are performed to demonstrate the ability of the output-based scheme to provide appropriate grid resolution for three different outputs. The output-based scheme is applied, independently, to the left-element drag, the right-element drag, and the total drag (both elements). In each case, the prescribed error tolerance on the output is $e_0 = 0.0005$. For comparison, an additional simulation is performed using pure Hessian-based adaptation. The farfield boundary is placed at 15 chords from the origin. Fig. 7 shows the final adapted grids for each case.

Output-based adaptation on the left-element drag alone produces a final grid with limited wake resolution beyond two chord-lengths downstream of the left element. The left-element drag is relatively insensitive to discretization errors downstream of that point. Correspondingly, the grid near the downstream element is only marginally refined. Output-based adaptation on the right-element drag produces a grid with significant wake resolution over the entire length of the inter-element gap, and beyond. The drag on the downstream element is more strongly effected by discretization errors in the vicinity of the upstream element. The adaptive algorithm responds accordingly by providing moderate grid resolution around the upstream element. Output-based adaptation applied to the total drag produces significant grid resolution near both elements. The wake from the upstream element is resolved well past the downstream element. Adaptation based on the Mach number Hessian yields a final grid with higher resolution in the leading edge and boundary layer regions relative to any of the output-based simulations. However, the inviscid regions further away from the elements are relatively underresolved. The multiplicative factor for this case is $\kappa = 325$.

The left-element, right-element, and total drag computed from each adaptive simulation is presented in Table 2 Numbers in brackets correspond to corrected values using the associated adjoint correction term (see Section 5.3). The drag values from the adaptive simulations are compared with the drag computed on a single fine grid obtained by uniformly refining ($N = 2$) the adapted grid corresponding to output-based adaptation on both airfoil elements. In each case, the output-based method is effective at ensuring that the computed

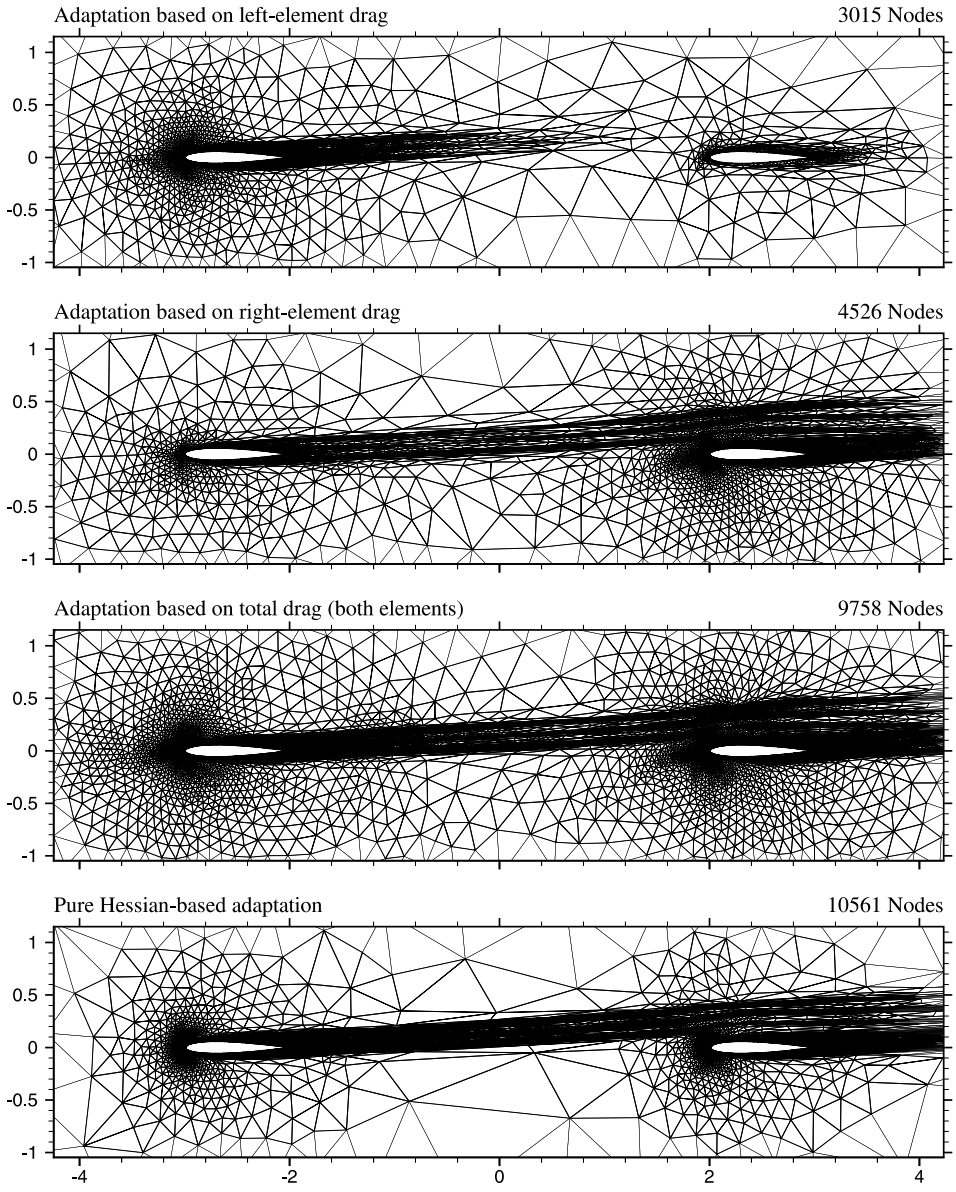

Fig. 7. Two element (NACA 0012) airfoil test case: $Re = 5000$, $M_\infty = 0.5$, $\alpha = 3°$. Final adapted grids from four different adaptive runs. The proposed output-based method is applied to the left-element drag (top), the right-element drag (second from top), and the total drag (third from top). The bottom plot corresponds to pure Hessian-based adaptation.

output (after correction) from the final adapted grid meets the imposed error tolerance. The adaptive algorithm resolves only those regions of the flow that are crucial for accurately computing the prescribed output. For the case where the output-based method is applied to the left-element drag alone, accuracy in the right-element drag is sacrificed in favor of a considerable reduction in grid size. This effect is not as severe in the opposite case, when the output-based method is applied to the right-element drag alone, for reasons

Table 2
Two element (NACA 0012) airfoil test case: $Re = 5000$, $M_\infty = 0.5$, $\alpha = 3°$

Grid		Drag (error)		
Adaptive criteria	Nodes	Left element	Right element	Total
Pure Hessian-based	10561	0.06049 (4.2%)	0.06148 (7.5%)	0.1220 (5.8%)
Output based				
Left-element adjoint	3015	0.05847 (0.7%) **0.05821 (0.3%)**	0.07265 (27.0%)	0.1311 (13.7%)
Right-element adjoint	4526	0.06018 (3.7%)	0.05800 (1.4%) **0.05741 (0.3%)**	0.1182 (2.6%)
Based on both elements	9758	0.05820 (0.3%)	0.05756 (0.6%)	0.1158 (0.5%) **0.1153 (0.0%)**
Fine grid	38 574	0.05804	0.05722	0.1153

Computed drag from four different adaptive runs: output-based adaptation on the left-element drag, right-element drag, and total drag (both elements), and pure Hessian-based adaptation. Numbers in bold face correspond to corrected drag values using the associated adjoint correction term. Numbers in brackets are the percentage error in the drag measured with respect to the fine grid value. The fine grid is obtained by uniformly refining ($N = 2$) the adapted grid corresponding to output-based adaptation on both airfoil elements.

outlined earlier. For grids of comparable size, output-based adaptation on both elements yields drag estimates that are dramatically more accurate than those obtained from pure Hessian-based adaptation.

6.2. Turbulent flow adaptive results

6.2.1. $Re = 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.466°$ RAE 2822 airfoil

Adaptive simulations are performed for turbulent flow over an RAE 2822 airfoil. The Reynolds number and free stream Mach number are $Re = 6.5 \times 10^6$ and $M_\infty = 0.725$, respectively, corresponding to Case 6 conditions in [14]. The angle of attack is fixed at $\alpha = 2.466°$, corresponding to the computation by Allmaras [3] in which the Case 6 lift is matched. Under these conditions, a shock appears on the suction side of the airfoil near the mid-chord position. The functionals considered for this test case are the lift and drag coefficients. The farfield boundary is placed at 100 chords. The initial grid for the adaptive simulations is a relatively coarse, inviscid-style grid analogous to the one shown in Fig. 4.

Lift-based simulations are performed for six different error tolerances ranging from $e_0 = 0.05$ to $e_0 = 0.0025$. The final adapted grids range in size from 668 nodes ($e_0 = 0.05$) to 22 159 nodes ($e_0 = 0.0025$). Drag-based simulations are performed for error tolerances ranging from $e_0 = 0.0025$ to $e_0 = 0.0001$ with final grids ranging in size from 741 nodes ($e_0 = 0.0025$) to 18 691 nodes ($e_0 = 0.0001$). Pure Hessian-based adaptation is performed for nine multiplicative factors ranging from $\kappa = 25$ to $\kappa = 300$. The final grids range in size from 2752 nodes ($\kappa = 25$) to 37 681 nodes ($\kappa = 300$).

Fig. 8 shows the base and corrected values of the computed lift and drag using the proposed output-based method and pure Hessian-based adaptation. The experimental values correspond to Case 6 in [14]. The Allmaras values [3] are obtained from an independent computation. Allmaras uses an upwind, second-order finite-volume discretization for the conservation equations and a first-order discretization for the Spalart-Allmaras turbulence model [45]. A structured C-type grid comprised of 768×192 cells is employed. The farfield boundary is placed at 20 chords and a point vortex correction is applied to the farfield boundary conditions.

The proposed output-based adaptive method offers a substantial improvement in accuracy over pure Hessian-based adaptation for both the lift and drag. Unlike the laminar test case presented in Section 6.1.1, the errors in the corrected outputs for this test case *appear* to be higher than the corresponding tolerances, at

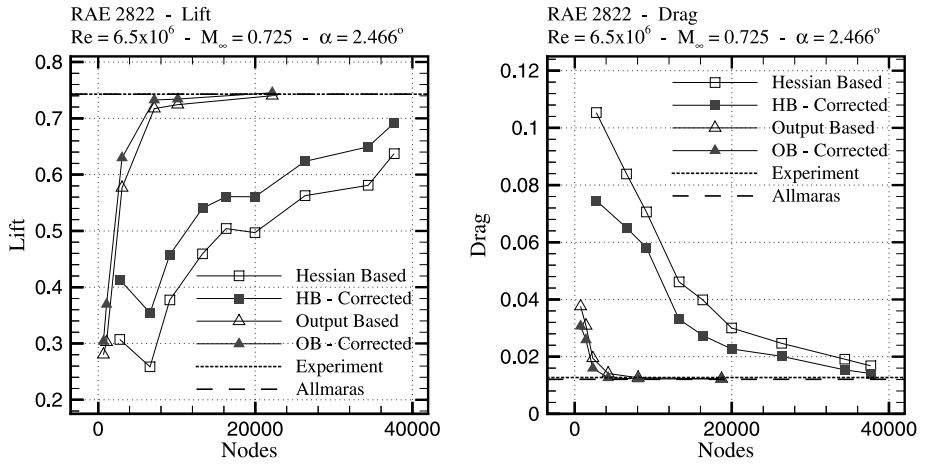

Fig. 8. RAE 2822 airfoil test case: $Re = 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.466°$. Plots comparing the uncorrected and corrected lift and drag from six lift-based, six drag-based, and nine Hessian-based adaptive simulations. The experimental values correspond to Case 6 in [14]. The Allmaras values are obtained from an independent computation [3] (see accompanying text).

least for the coarser grids in the adaptive series. Recall, however, that the prescribed error tolerance is measured with respect to the associated $N = 2$ fine-grid. It is likely that the fine grids associated with the three coarsest lift- and drag-based grids would themselves predict inaccurate output values for this problem.

Fig. 9 compares the final adapted grids from the $e_0 = 0.0025$ lift-based simulation (top), the $e_0 = 0.0001$ drag-based simulation (middle), and the $\kappa = 300$ Hessian-based simulation (bottom). The inaccurate lift and drag predictions from the Hessian-based simulation are primarily attributed to insufficient grid resolution in the inviscid portions of the flow, particularly near the leading edge and immediately adjacent to the boundary layer regions. Conversely, although it is not perceivable from the figure, the near-wall region in the boundary layer is significantly over-refined relative to the output-based grids, resulting in a larger overall grid size. Further, the wake is resolved farther downstream in the output-based grids relative to the Hessian-based grid where the wake appears to be truncated prematurely. In this region, wake refinement gets propagated downstream by the Hessian-based algorithm at a very slow rate. In comparison, the output-based algorithm is able to detect and resolve the wake far more rapidly over a comparable number of adaptive iterations.

6.2.2. $Re = 9 \times 10^6$, $M_\infty = 0.26$, $\alpha = 8°$ advanced EET three element airfoil

Adaptive simulations are performed for turbulent flow over the advanced energy efficient transport (EET) three element airfoil [28]. The Reynolds number (based on the chord of the EET with elements retracted), free stream Mach number, and angle of attack are $Re = 9 \times 10^6$, $M_\infty = 0.26$, and $\alpha = 8°$, respectively. The functional of interest for this test case is the total lift coefficient (all three elements). The farfield boundary is placed at 100 chords. The initial grid for the adaptive simulations is a relatively coarse, inviscid-style grid analogous to the one shown in Fig. 4.

Output-based simulations are performed for seven different error tolerances ranging from $e_0 = 0.1$ to $e_0 = 0.01$. The final adapted grids range in size from 1200 nodes ($e_0 = 0.1$) to 59 132 nodes ($e_0 = 0.01$). Pure Hessian-based adaptation is performed for six multiplicative factors ranging from $\kappa = 50$ to $\kappa = 300$. The final grids range in size from 8489 nodes ($\kappa = 50$) to 52 235 nodes ($\kappa = 300$).

Fig. 10 shows the base and corrected values of the computed lift using the proposed output-based method and pure Hessian-based adaptation. The experimental lift value is obtained from [28]. The

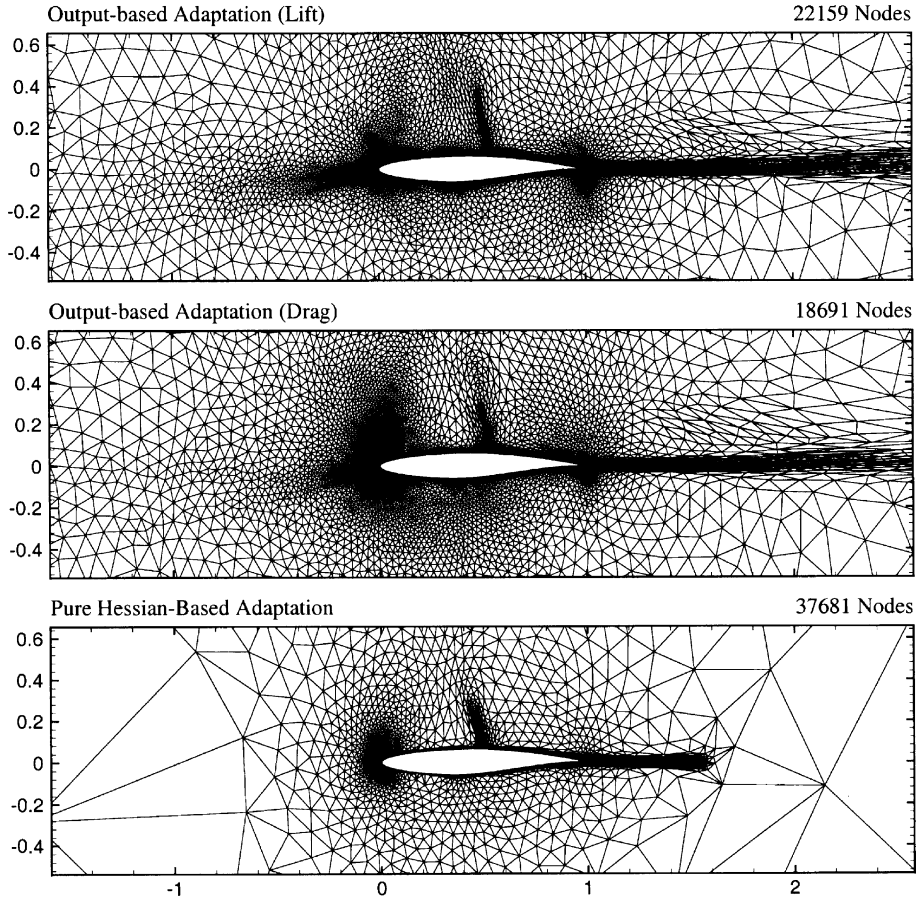

Fig. 9. RAE 2822 airfoil test case: $Re = 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.466°$. Comparison of final adapted grids using the proposed output-based method on the lift (top), and drag (middle), and using pure Hessian-based adaptation (bottom).

Anderson et al. [5] value is obtained from an independent computation. Anderson et al. use the FUN2D flow solver (see Section 4.1). An unstructured grid comprised of 70 686 nodes is employed for the computation. The grid has regular spacing near the airfoil boundaries with a minimum normal spacing of 2×10^{-6} chord units adjacent to the wall.

Lift predictions from the output-based grids are dramatically superior to those computed from the Hessian-based grids. Output-based grids with greater than 20 000 nodes are essentially converged with respect to the computed lift. Conversely, the uncorrected lift value from the finest of the Hessian-based grids is still in error by nearly 20%.

Fig. 11 compares the final adapted grids from a lift-based simulation corresponding to $e_0 = 0.02$ (top) and a Hessian-based simulation corresponding to $\kappa = 300$ (bottom). Fig. 12 shows near-field views of the same grids in the slat region (top) and flap region (bottom). As in previous test cases, the inaccurate lift prediction from the Hessian-based simulation is attributed, in part, to insufficient grid resolution in the inviscid regions of the flow. Furthermore, the near-wall boundary layer regions are generally overrefined relative to the output-based grid, resulting in unnecessary computing costs. Additional reasons for the poor lift prediction from the Hessian-based grid can be deduced from Fig. 12. In contrast to the output-based method, the Hessian-based method incorrectly predicts the location of the lower slat wake. As a result, the

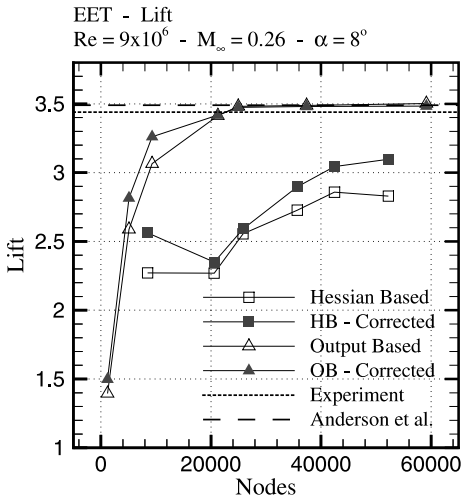

Fig. 10. Advanced EET three element airfoil test case: $Re = 9 \times 10^6$, $M_\infty = 0.26$, $\alpha = 8°$. Comparison of the uncorrected and corrected lift from seven output-based adaptive simulations and six Hessian-based simulations. The experimental lift value is obtained from [28]. The Anderson et al. value is obtained from an independent computation [5] (see accompanying text).

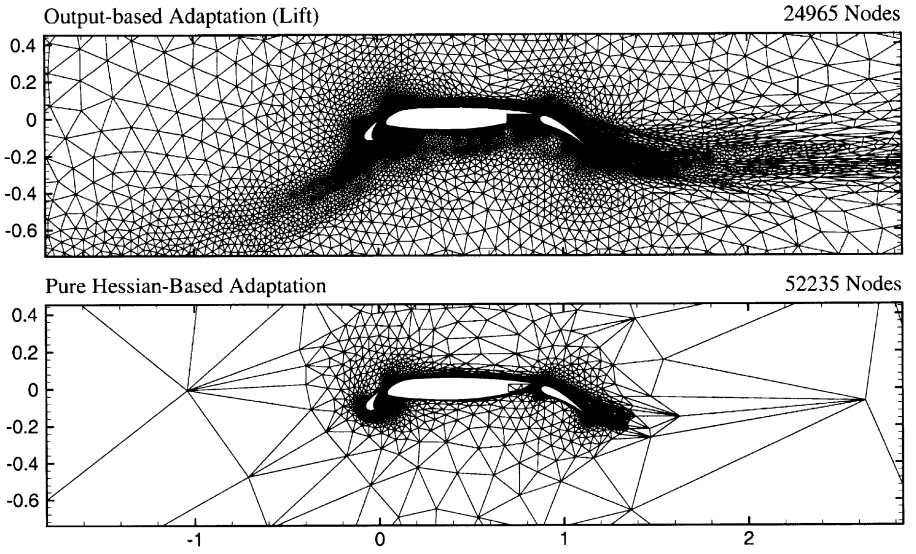

Fig. 11. Advanced EET three element airfoil test case: $Re = 9 \times 10^6$, $M_\infty = 0.26$, $\alpha = 8°$. Final adapted grids using the proposed output-based method (top), and pure Hessian-based adaptation (bottom).

flow at the leading-edge and upper surface of the main element has larger total pressure loss than the actual flow which would tend to lower the predicted lift generated by the airfoil. In fact, the results from Fig. 10 show that the lift is greatly under-predicted by the Hessian-based method. The severe lack of resolution in the cavity region at the rear of the main element is also a likely source of error in the Hessian-based grid.

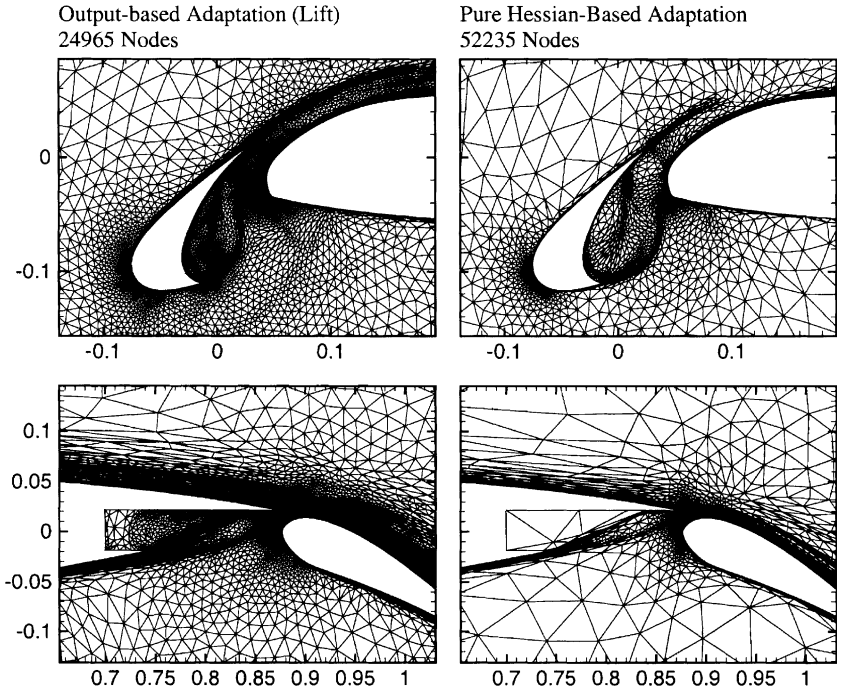

Fig. 12. Advanced EET three element airfoil test case: $Re = 9 \times 10^6$, $M_\infty = 0.26$, $\alpha = 8°$. Near-field views of final adapted grids using the proposed output-based method (left), and pure Hessian-based adaptation (right).

7. Conclusion

A novel anisotropic grid adaptive procedure was presented for improving the accuracy of functional outputs from numerical simulations of viscous, compressible flows. The method was demonstrated by application to two-dimensional laminar and turbulent airfoil test cases using a standard finite volume discretization. The proposed method was shown to be superior in terms of reliability, output accuracy and computational efficiency relative to pure Hessian-based adaptation.

The algorithms and procedures outlined in this paper are all extendible to three dimensions. Park [36] has implemented an extension of the proposed output-based procedure for isotropic adaptive refinement of three-dimensional Euler simulations. The proposed anisotropic adaptive method is expected to be particularly beneficial for three-dimensional Navier–Stokes applications, where the resolution requirements for the computational grid often become exceedingly unintuitive, and for which the current technology is often unreliable or inadequate [44].

Acknowledgements

The authors wish to thank Drs. S.R. Allmaras, M.B. Giles, E.J. Nielsen, M.A. Park, and J.T. Thomas for helpful comments and discussion pertaining to this work. The FUN2D flow solver was developed at NASA Langley by Dr. W.K. Anderson and Dr. E.J. Nielsen. The BL2D and BAMG grid generators were developed at INRIA-Rocquencourt, France, by Drs. P. Laug and H. Borouchaki, and by Dr. F. Hecht, respectively. The support of the National Science Foundation (ACI-9896341), The Boeing Company,

NASA Langley (NAG1-2275, NAG-1-02037), and FCAR (Fonds pour la Formation de Chercheurs et l'Aide à la Recherche) is gratefully acknowledged.

References

[1] M.J. Aftosmis, M.J. Berger, Multilevel error estimation and adaptive *h*-refinement for cartesian meshes with embedded boundaries, AIAA Paper 2002-0863, 2002.
[2] M. Ainsworth, J.T. Oden, A posteriori error estimation in finite element analysis, Comput. Meth. Appl. Mech. Eng. 142 (1997) 1–88.
[3] S.R. Allmaras, personal communication.
[4] W.K. Anderson, D.L. Bonhaus, An implicit upwind algorithm for computing turbulent flows on unstructured grids, Comput. Fluids 23 (1994) 1–21.
[5] W.K. Anderson, R.D. Rausch, D.L. Bonhaus, Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids, AIAA Paper 95-1740, 1995.
[6] W.K. Anderson, V. Venkatakrishnan, Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation, AIAA Paper 97-0643, 1997.
[7] T.J. Baker, Mesh adaptation strategies for problems in fluid dynamics, Finite Elements Anal. Design 25 (1997) 243–273.
[8] R. Becker, R. Rannacher, Weighted a posteriori error control in finite element methods, in: Proceedings of ENUMATH-97, Heidelberg, World Scientific, Singapore, 1998, pp. 621–637.
[9] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, in: A. Iserles (Ed.), Acta Numerica 2001, Cambridge University Press, Cambridge, 2001.
[10] H. Borouchaki, P.L. George, F. Hecht, P. Laug, E. Saltel, Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie I: Algorithmes. Research Report No. 2741, INRIA-Rocquencourt, France, 1995.
[11] M. Braack, R. Rannacher, Adaptive finite element methods for low-Mach-number flows with chemical reactions, VKI Lecture Series 1999-03, 1999.
[12] 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.
[13] M.J. Castro-Díaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructured mesh adaption for flow simulations, Int. J. Numer. Meth. Fluids 25 (1997) 475–491.
[14] P.H. Cook, M.A. McDonald, M.C.P. Firmin, Aerofoil RAE 2822 – pressure distributions, and boundary layer and wake measurements, AGARD-AR-138, 1979.
[15] A. Dervieux, D. Leservoisier, P.L. George, Y. Coudiere, About theoretical and practical impact of mesh adaptation on approximation of functions and of solutions of PDE, ECCOMAS CFD Conference, Swansea, UK, 2001.
[16] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaption with application to CFD problems, in: H.A. Mang et al., (Eds.) Fifth World Congress on Computational Mechanics, Vienna, Austria, 2002.
[17] M.B. Giles, M.C. Duta, J.D. Müller, Adjoint code developments using the exact discrete approach, AIAA Paper 2001-2596, 2001.
[18] M.B. Giles, M.G. Larson, M. Levenstam, E. Süli, Adaptive error control for finite element approximations of the lift and drag in a viscous flow, Technical Report NA 97/06, Oxford Computing Laboratory, Oxford, 1997.
[19] M.B. Giles, N.A. Pierce, Adjoint equations in CFD: duality, boundary conditions and solution behavior, AIAA Paper 97-1850, 1997.
[20] M.B. Giles, N.A. Pierce, Improved lift and drag estimates using adjoint Euler equations, AIAA Paper 99-3293, 1999.
[21] M.B. Giles, N.A. Pierce, Adjoint error correction for integral outputs, in: T. Barth, H. Deconinck (Eds.), Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics, Lecture Notes in Computer Science and Engineering, vol. 25, Springer-Verlag, Berlin, 2002, pp. 47–96.
[22] W.G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, 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 (2000) 725–744.
[23] F. Hecht, BAMG: bidimensional Anisotropic Mesh Generator, INRIA-Rocquencourt, France, 1998. Available from http://www-rocq.inria.fr/gamma/cdrom/www/bamg/eng.htm.
[24] P. Houston, E. Süli, *hp*-adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems, SIAM J. Sci. Comput. 23 (4) (2001) 1226–1252.
[25] M.G. Larson, T.J. Barth, A posteriori error estimation for discontinuous Galerkin approximations of hyperbolic systems, NAS Technical Report NAS-99-010, 1999.
[26] P. Laug, H. Borouchaki, The BL2D mesh generator: beginner's guide, user's and programmer's manual, Technical Report No. 0194, INRIA-Rocquencourt, France, 1996. Available from http://www-rocq.inria.fr/gamma/cdrom/www/bl2d/eng.htm.

[27] D.W. Levy, T. Zickuhr, J. Vassberg, S. Agrawal, R.A. Wahls, S. Pirzadeh, M.J. Hemsch, Summary of data from the first AIAA CFD drag prediction workshop, AIAA Paper 2002-0841, 2002.
[28] J.C. Lin, C.J. Dominick, Optimization of an advanced design three-element airfoil at high Reynolds numbers, AIAA Paper 95-1858, 1995.
[29] L. Machiels, J. Peraire, A.T. Patera, A posteriori finite element output bounds for the incompressible Navier–Stokes equations: application to a natural convection problem, J. Comput. Phys. 172 (2001) 401–425.
[30] D.J. Mavriplis, Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes, Int. J. Numer. Meth. Fluids 34 (2000) 93–111.
[31] J.D. Müller, M.B. Giles, Solution adaptive mesh refinement using adjoint error analysis, AIAA Paper 2001-2550, 2001.
[32] E. Nielsen, FUN2D/3D Fully Unstructured Navier–Stokes User Manual, NASA Langley Research Center, Computational Modeling and Simulation Branch, Virginia, 2002. Available from http://fun3d.larc.nasa.gov.
[33] E.J. Nielsen, W.K. Anderson, Aerodynamic design optimization on unstructured meshes using the Navier–Stokes equations, AIAA Paper 98-4809, 1998.
[34] E.J. Nielsen, W.K. Anderson, Recent improvements in aerodynamic optimization on unstructured meshes, AIAA Paper 2001-0596, 2001.
[35] E. Onate, G. Bugeda, Mesh optimality criteria for adaptive finite element computations, in: J.R. Whiteman (Ed.), The Mathematics of Finite Elements and Applications, Wiley, Chichester, England, 1994, pp. 121–135, Chapter 7.
[36] M.A. Park, Adjoint-based, three-dimensional error prediction and grid adaptation, AIAA Paper 2002-3286, 2002.
[37] J. Peraire, A.T. Patera, Bounds for linear-functional outputs of coercive partial differential equations: Local indicators and adaptive refinement, in: P. Ladevèze, J.T. Oden (Eds.), Advances in Adaptive Computational Methods in Mechanics, Elsevier, Amsterdam, 1998.
[38] J. Peraire, A.T. Patera, Asymptotic a posteriori finite element bounds for the outputs of non-coercive problems: the Helmholtz and Burgers equations, Comput. Meth. Appl. Mech. Eng. 171 (1999) 77–86.
[39] J. Peraire, M. Vahdati, K. Morgan, O.C. Zienkiewicz, Adaptive remeshing for compressible flow computations, J. Comput. Phys. 72 (1987) 449–466.
[40] N.A. Pierce, M.B. Giles, Adjoint recovery of superconvergent functionals from PDE approximations, SIAM Rev. 42 (2) (2000) 247–264.
[41] S.Z. Pirzadeh, An adaptive unstructured grid method by grid subdivision, local remeshing, and grid movement, AIAA Paper 99-3255, 1999.
[42] R. Rannacher, Adaptive Galerkin finite element methods for partial differential equations, J. Comput. Appl. Math. 128 (2001) 205–233.
[43] P. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[44] C.L. Rumsey, S.X. Ying, Prediction of high lift: review of present CFD capability, Progress in Aerospace Sciences 38 (2002) 145–180.
[45] P.R. Spalart, S.R. Allmaras, A one-equation turbulence model for aerodynamic flows, La Recherche Aérospatiale 1 (1994) 5–21.
[46] G. Strang, G.J. Fix, An Analysis of the Finite Element Method, Wellesley-Cambridge Press, Wellesley, MA, 1988.
[47] E. Süli, P. Houston, B. Senior, *hp*-discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity, in: M.J. Baines (Ed.), Numerical Methods for Fluid Dynamics VII, ICFD, Oxford, 2001.
[48] R.C. Swanson, E. Turkel, Multistage schemes with multigrid for Euler and Navier–Stokes equations – applications. Unpublished NASA Report, 2002.
[49] L.N. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, Philadelphia, PA, 1997.
[50] D.A. Venditti, Grid adaptation for functional outputs of compressible flow simulations. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2002.
[51] D.A. Venditti, D.L. Darmofal, Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow, J. Comput. Phys. 164 (2000) 204–227.
[52] D.A. Venditti, D.L. Darmofal, Grid adaptation for functional outputs: application to two-dimensional inviscid flows, J. Comput. Phys. 176 (2002) 40–69.
[53] G.P. Warren, W.K. Anderson, J.T. Thomas, S.L. Krist, Grid convergence for adaptive methods, AIAA Paper 91-1592, 1991.
[54] W.A. Wood, W.L. Kleb, On multi-dimensional unstructured mesh adaption, AIAA Paper 99-3254, 1999.
[55] G. Xia, D. Li, C.L. Merkle, Anisotropic grid adaptation on unstructured meshes, AIAA Paper 2001-0443, 2001.
[56] X.D. Zhang, M.-G. Vallet, J. Dompierre, P. Labbé, D. Pelletier, J.-Y. Trépanier, R. Camarero, J.V. Lassaline, L.M. Manzano, D.W. Zingg, Mesh adaptation using different error indicator for the Euler equations, AIAA Paper 2001-2549, 2001.
[57] O.C. Zienkiewicz, The Finite Element Method, third ed., McGraw-Hill, London, 1977.