Anisotropic Grid Adaptation for Multiple Aerodynamic Outputs

David A. Venditti [*] and David L. Darmofal [†]

Massachusetts Institute of Technology

Department of Aeronautics and Astronautics

77 Massachusetts Avenue, Room 37–427

Cambridge, MA 02139

---

[*] Research Consultant

[†] Associate Professor

Abstract

Anisotropic grid–adaptive strategies are presented for viscous flow simulations in which the accurate prediction of multiple aerodynamic outputs (such as the lift, drag, and moment coefficients) is required from a single adaptive solution. The underlying adaptive procedure is based on a merging of adjoint error estimation and Hessian-based anisotropic grid adaptation. Airfoil test cases are presented to demonstrate the various adaptive strategies including a single element airfoil at cruise conditions and a multi-element airfoil in high-lift configuration with flow separation. Numerical results indicate that the lift, drag and moment coefficients are accurately predicted by all of the output–based strategies considered, although slightly better accuracy is obtained in the output(s) for which a particular strategy is specifically designed. Furthermore, the output-based strategies are all shown to be significantly more efficient than pure Hessian-based adaptation in terms of output accuracy for a given grid size.

Key words: anisotropic grid adaptation; adjoint error correction/estimation; multiple functional outputs; aerodynamics; finite volume; finite element; computational fluid dynamics.

1 Introduction

Proper grid resolution is critical for industrial computational fluid dynamics (CFD) applications due to the conflicting and exacting requirements of adequate solution accuracy and limited computer execution time. Solution–based, automatic grid adaptation has proven to be an effective means of iteratively improving the local grid resolution during the solution procedure [1–10]. The basic premise is to assess the solution error at intermediate stages of the overall solution process and modify the local grid density accordingly in order to improve the solution accuracy and/or reduce the nominal grid size.

A limiting factor in the application of grid adaptation for Navier–Stokes computations has been the lack of reliable error indicators for driving the adaptive process. Many methods use indicators based on large flow gradients or undivided differences [2,8,9]. These methods tend to increase the grid density near certain flow features such as shocks, wakes, and boundary layers. Unfortunately, indicators of this type may lead to incorrect results. For example, continuously refining the grid near a shock does not necessarily lead to an improvement in overall solution accuracy. Predicting the proper shock strength and location, as well as other derived quantities, may depend more on the quality of the grid well upstream of the shock rather than in its immediate vicinity [9].

Adaptive indicators for nonlinear flow problems have also been derived from interpolation error estimates for linear finite elements [3,4,7,9]. These methods attempt to equidistribute the estimated interpolation error in one or more scalars throughout the computational domain. Essentially, this amounts to adapting on the local second–derivatives or Hessian of the solution and, therefore, shares some of the potential deficiencies associated with feature–based indicators. Traditional feature– and Hessian–based indicators are local in nature and do not provide a reliable indication of how the discretization error is distributed or transported throughout the domain. The equidistribution of interpolation error does not generally translate into the equidistribution of discretization error for nonlinear flow problems. Furthermore, even the equidistribution of discretization error may not be optimal within an engineering context.

An alternate approach to making error estimation more relevant for engineering applications is to assess the error made in predicting integral quantities representing basic engineering outputs. Examples include the lift and drag forces acting on an aircraft wing, the mass-averaged total–pressure drop across a high–pressure turbine stage, or the average noise levels generated by an aircraft at takeoff condition. There has been a significant volume of research into *a posteriori* error analysis and grid adaptation for functional outputs within the context of finite element methods for fluid dynamics [11–22]. For general discretizations, Pierce and Giles [23–26] 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 functional outputs. The error estimation procedure is based on the functional correction technique of Pierce and Giles. 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 and error estimation terms used in this work have already been incorporated into isotropic adaptive schemes for finite volume discretizations of quasi-one-dimensional [27] and two-dimensional [28] inviscid flows, and for a Galerkin finite element discretization for low-Peclet-number convection-diffusion [29]. The approach has been generalized for anisotropic adaptation [29, 30] by merging the functional correction and error estimation procedure with elements of traditional Hessian–based adaptation.

To the authors' knowledge, this was the first reported output–based, anisotropic, adaptive method for Navier–Stokes flows; it has been successfully applied to practical two-dimensional laminar [29] and turbulent [30] test cases. Park has applied and extended the methods in references [28–30] to three-dimensional inviscid [31] and turbulent flows [32]. Formaggia, Micheletti and Perotto [33, 34] have independently developed output–based, anisotropic adaptive procedures for advection–diffusion–reaction and Stokes flows within a finite–element framework that also combine adjoint or duality arguments with anisotropic interpolation error estimates.

In this paper we propose several adaptive strategies in which the accurate prediction of multiple integral outputs is required from a single simulation. The underlying adaptive methodology is presented in references [29] and [30] and is summarized in Section 2. Section 3 outlines the various adaptive strategies, and Section 4 presents numerical results in which the strategies are applied to two airfoil test cases: a single element airfoil at cruise conditions and a multi-element airfoil in high-lift configuration with flow separation. The effectiveness of each strategy is gauged in part by the accuracy of the predicted lift, drag, and moment coefficients relative to the corresponding grid size. Numerical results indicate that the aerodynamic coefficients are accurately predicted by all of the output–based strategies considered, although slightly better accuracy is obtained in the output(s) for which a particular strategy is specifically designed. Furthermore, the output-based strategies are all shown to be significantly more efficient than pure Hessian-based adaptation in terms of output accuracy for a given grid size.

2 Adaptive Methodology

This section presents a succinct summary of the output–based adaptive methodology outlined in references [29] and [30]. The methodology borrows concepts from adjoint error estimation/correction for functional outputs [24, 26], and from Hessian–based anisotropic grid adaptation [3, 7].

The adaptive procedure is designed to improve the accuracy of an integral output $f_H(U_H)$ that can be written as a nonlinear functional of the discrete flow solution U_H on the current grid Ω_H. The system of residual equations associated with the flow discretization on Ω_H is denoted $R_H(U_H) = 0$.

Each iteration in a single–output adaptive simulation requires approximate solutions of the discrete flow (primal) and adjoint (dual) equations. The discrete system of adjoint equations is given by

$$\frac{\partial R_H}{\partial U_H}^T \Psi_H = \frac{\partial f_H}{\partial U_H}^T , \tag{1}$$

where $\partial R_H / \partial U_H$ is the Jacobian of the primal residual equations, $\partial f_H / \partial U_H$ is a row vector containing the functional sensitivities, and Ψ_H is the corresponding adjoint solution.

Once approximate flow and adjoint solutions have been obtained on the current grid a correction term [24, 26] is computed and used to improve the accuracy of the output. This is done prior to the grid–adaptive step for the current iteration. The corrected output is given by [30]

$$f_h^H = f_h(Q_h^H U_H) - (\bar{Q}_h^H \Psi_H)^T R_h(Q_h^H U_H). \tag{2}$$

In this last expression $f_h(\cdot)$ and $R_h(\cdot)$ are the functional and residual operators, respectively, defined on a fine grid Ω_h. The fine grid is constructed by uniformly subdividing the elements of the current grid. The rectangular matrices Q_h^H and $\bar{Q}_h^H$ represent quadratic prolongation operators that map the flow and adjoint solutions, respectively, onto the fine grid. Detailed descriptions of these terms are provided in reference [30].

The adaptive algorithm attempts to further enhance the accuracy of the corrected output by working to reduce the magnitude of the remaining error after correction. A crude approximation of the remaining error in the functional ε is obtained by summing positive contributions ε_k from each element k in the current grid [30],

$$\varepsilon = \sum_k \varepsilon_k, \tag{3}$$

where

$$\begin{aligned} \varepsilon_k = \frac{1}{2}\sum_{l(k)} \{ \quad & \left| \, [R_h^{\Psi}(\bar{L}_h^H \Psi_H)]^T_{l(k)} \; [Q_h^H U_H - L_h^H U_H]_{l(k)} \, \right| \\ + \quad & \left| \, [\bar{Q}_h^H \Psi_H - \bar{L}_h^H \Psi_H]^T_{l(k)} \; [R_h(L_h^H U_H)]_{l(k)} \, \right| \quad \}. \end{aligned} \tag{4}$$

The index $l(k)$ enumerates each fine–grid node associated with element k. The rectangular matrices L_h^H and $\bar{L}_h^H$ represent linear prolongation operators analogous to Q_h^H and $\bar{Q}_h^H$, respectively. The adjoint residual operator R_h^{Ψ} on the fine grid is defined as

$$R_h^{\Psi}(\cdot) \equiv \frac{\partial R_h}{\partial U_h}^T (\cdot) - \frac{\partial f_h}{\partial U_h}^T. \tag{5}$$

Note that the evaluation of (2)–(4) does not require solutions to be performed on the fine grid. Only functional and residual evaluations are needed on the fine grid, preceded by the prolongation of the flow and adjoint solutions from the current grid.

For each element k a desired new element size H'_k is computed from the old one, H_k, according to

$$H'_k = H_k \left(\frac{1}{\eta_g \eta_k} \right)^{\omega}, \tag{6}$$

where η_k and η_g are local and global adaptation parameters, respectively, and ω is an under-relaxation parameter that controls how aggressively the subsequent grid is refined. In particular, $\eta_k = \varepsilon_k / \bar{e}_o$ and $\eta_g = \varepsilon / e_o$ where e_o is a user-specified desired error level, $\bar{e}_o = e_o / N_e$

is the target error for each element, and N_e is the total number of elements in the current grid. A value of $\omega = 1/8$ is used for the test cases in this paper.

Equation (6) only provides isotropic information about the desired element size in the subsequent grid. Anisotropic criteria are incorporated into the adaptive procedure by extracting stretching and orientation information from interpolation–error estimates for linear finite elements [3, 7]. This merging of adjoint error estimation and Hessian–based anisotropic grid adaptation was originally proposed in references [29, 30] and is briefly outlined in the following.

The Hessian (matrix of second derivatives) of a specified scalar field, such as the Mach number distribution, is computed using a quadratic recovery algorithm [28] resulting in piecewise constant values over each element in the current grid. A Hessian–based metric $\mathcal{M}_{\mathcal{H}}$ is then computed by diagonalizing the Hessian matrix and taking the absolute value of its eigenvalues [3, 7] yielding

$$\mathcal{M}_{\mathcal{H}} = R \, |\Lambda| \, R^T. \tag{7}$$

In two dimensions, the orthonormal eigenvector matrix R can be written as

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{8}$$

and the corresponding eigenvalue matrix Λ as

$$\Lambda = \begin{bmatrix} 1/h_1^2 & 0 \\ 0 & 1/h_2^2 \end{bmatrix}. \tag{9}$$

The Hessian–based metric can be interpreted as a transformation matrix that maps physical (Euclidean) sets onto a metric space [3]. For example, the unit circle centered at the origin in the metric space is mapped onto an ellipse in the physical space with major and minor principal lengths h_1 and h_2, respectively, and with its major axis rotated by an angle θ to the x–axis.

In the proposed output–based adaptive procedure, stretching and orientation parameters, $\beta = h_1/h_2$ and θ, respectively, are obtained directly from $\mathcal{M}_{\mathcal{H}}$, and an element–size parameter H' is obtained from Equation (6). An output–based metric $\mathcal{M}'$ is then constructed for each element in the current grid as

$$\mathcal{M}' = R\,|\Lambda'|\,R^T, \tag{10}$$

where

$$\Lambda' = \begin{bmatrix} 1/(\beta H')^2 & 0 \\ 0 & 1/(H')^2 \end{bmatrix}. \tag{11}$$

Note from the last expression that the element–size parameter H' has been attributed to the minor principal length associated with the output–based metric. Accordingly, the current element–size parameter H in Equation (6) is computed as the minor principal length associated with the local grid–implied metric on the current grid. The grid–implied metric [29,30] is the metric distribution that yields a constant metric length for all edges in the current grid. An average value for the grid–implied metric is calculated for each element in the current grid by solving a local 3 × 3 system of equations for the metric components. The equations are formed by specifying a unit metric length for each edge in the element. The square of the metric length $l_{\mathcal{M}}$ of a straight edge with respect to the constant metric $\mathcal{M}$ is given by

$$\left(l_{\mathcal{M}}\right)^2 = \hat{s}^T \mathcal{M} \hat{s}\ \left(l_I\right)^2, \tag{12}$$

where $\hat{s}$ is a unit vector tangent to the edge and l_I is the physical (Euclidean) length of the edge.

The output–based metric components are transferred to the nodes of the current grid using area–weighted averages of the piecewise–constant values over the elements surrounding each node. The nodal values are then provided as input to an anisotropic grid generator and the new grid is created.

For the test cases presented in this paper, the adaptive algorithm is run for at least 16 iterations, at which point the change in the total number of nodes from one grid to the next is typically on the order of 1%. The intent is to ensure convergence of the grid–evolution process; no attempt is made to optimize the rate of convergence in this work.

If the desired error tolerance e_o 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_o is prescribed initially and then gradually ramped down to the desired error level over the course of several iterations.

3 Adaptive Strategies

Several output–based adaptive strategies are implemented using the adaptive methodology outlined in Section 2. Airfoil test cases are presented in Section 4 to demonstrate the various adaptive strategies including a single element airfoil at cruise conditions and a multi-element airfoil in high-lift configuration with flow separation. The effectiveness of each strategy is gauged in part by the accuracy of the predicted lift, drag, and moment coefficients computed on the final adapted grids versus the total number of nodes in the respective grid.

Lift, Drag, and Moment Coefficients

In the first three strategies, the lift, drag, and moment coefficients are prescribed, individually, as the output for driving the adaptive simulations. A goal of this study is to assess the effectiveness of each single–coefficient strategy to accurately predict the two other coefficients for which the strategy is not explicitly designed.

Weighted Average

One strategy for incorporating the lift, drag, and moment coefficients, C_L, C_D, and C_M, respectively, into a single adaptive simulation is to define the output f as a weighted average of the square of the coefficients,

$$f = W_L C_L^2 + W_D C_D^2 + W_M C_M^2. \tag{13}$$

The weights, W_L, W_D, and W_M, are chosen according to specified relative error levels, e_L, e_D, and e_M, respectively, as

$$W_L = \left(\frac{1}{e_L}\right)^2 ;\ W_D = \left(\frac{1}{e_D}\right)^2 ;\ W_M = \left(\frac{1}{e_M}\right)^2 . \tag{14}$$

For the test cases in this paper, the relative error levels are arbitrarily chosen as $e_L = 1.0$, $e_D = 0.1$, and $e_M = 0.2$.

Multi–Output Adaptation

In the previous strategy, the output is linked to all three coefficients simultaneously, while requiring only one adjoint calculation per adaptive iteration. The adaptive algorithm attempts to satisfy the specified error levels in an average sense. In contrast, the current strategy attempts ensure that all three error levels are satisfied simultaneously at the cost of performing three adjoint calculations per adaptive iteration.

The following is a general description of the proposed multi–output strategy in which there are N_f outputs f^i and corresponding error tolerances e_o^i, $i = 1, N_f$. A single iteration in the adaptive procedure is comprised of the following steps:

1. On the current grid, compute the flow solution and the N_f adjoint solutions associated with the specified outputs.

2. For each element, compute piecewise–constant estimates of the local Mach number Hessian.

3. For each element, compute the Hessian–based metric $\mathcal{M}_{\mathcal{H}}$ given by (7)–(9) and extract the stretching and orientation parameters, $\beta = h_1/h_2$ and θ, respectively.

4. For each element, estimate the grid–implied metric and compute the current element–size parameter H.

5. For each output, compute the adjoint–based adaptation parameters, η^i_g and η^i_k, $i = 1, N_f$. Specifically, $\eta^i_g = \varepsilon^i/e^i_o$ and $\eta^i_k = \varepsilon^i_k/\bar{e}^i_o$ for each element k, where $\bar{e}^i_o = e^i_o/N_e$ is the element target error, N_e is the total number of elements in the current grid, $\varepsilon^i = \sum_k \varepsilon^i_k$, and ε^i_k is obtained from (4) using the corresponding adjoint solution.

6. For each element, the element–size parameter H'_i, associated with the ith functional, is obtained in terms of the current element size parameter H using (6).

7. For each element, the final element size parameter H', used for generating the next grid, is chosen as the minimum of the element–size parameters associated with each functional. That is, $H' = \min_i (H'_i)$.

8. For each element, the new output–based metric $\mathcal{M}'$ is constructed using the parameters β', θ', and H', and Equations (8), (10), and (11). The elemental metric components are transfered to the nodes using area–weighted averages of the piecewise–constant values over the elements surrounding each node.

9. The nodal metrics are specified as inputs to the anisotropic grid generator and the grid is regenerated.

In the present case, there are $N_f = 3$ outputs corresponding to the lift, drag, and moment coefficients: $f^1 = C_L$, $f^2 = C_D$, and $f^3 = C_M$, respectively. The corresponding error tolerances are chosen as $e_o^1 = \epsilon$, $e_o^2 = 0.1\epsilon$, and $e_o^3 = 0.2\epsilon$, where ϵ is varied in order to define several different overall adaptive tolerances while keeping the relative error levels for the coefficients constant.

Domain Integral

The aforementioned strategies all involve aerodynamic outputs expressible as integrals of the flow solution over portions of the domain boundary (i.e. airfoil surfaces). The current strategy attempts to provide overall solution accuracy by targeting an integral over the entire domain. In particular, the output f is defined as

$$f = \int_{\Omega} (M - M_\infty)^2 \, \mathrm{d}\Omega \tag{15}$$

That is, the square–integral of the Mach number relative to the free–stream value M_∞. This specific choice is made to further illustrate the differences between pure Hessian–based adaptation (based on the Mach number Hessian) and the current output–based approach.

Pure Hessian–Based Adaptation

Adaptive simulations based solely on the local Mach number Hessian are performed for comparison with the proposed output–based strategies. Pure Hessian–based adaptation is essentially what is used in references [3, 4, 7] although their methodologies and implementations differ to varying degrees. In the present implementation, the algorithm attempts to equidistribute the estimated interpolation error in the Mach number distribution by equidistributing the metric lengths (see Equation (12)) of all the edges in the grid using the Hessian–based metric $\mathcal{M}_{\mathcal{H}}$ given by (7)–(9). Different error tolerances are established 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. 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 $h_2 \leq h_1 \leq h_{\max}$, where $h_{\max}$ is a specified maximum element length for the domain.

4 Numerical Results

Two airfoil test cases are presented to demonstrate the various adaptive strategies outlined in Section 3: turbulent flow over the RAE 2822 airfoil at cruise conditions; and turbulent flow over the Advanced Energy Efficient Transport (EET) three element airfoil [35] in high-lift configuration with flow separation.

The flow and adjoint solvers used in this work are part of the FUN2D suite of codes [36–38]. The governing flow equations are the compressible, Reynolds–averaged Navier–Stokes (RANS) equations. The eddy viscosity is obtained using the one–equation turbulence model of Spalart and Allmaras [39]. The BAMG [40] anisotropic grid generator is used to regenerate the grids in this work. BAMG and FUN2D are incorporated into the overall adaptive procedure using shell scripts.

4.1 RAE 2822 Airfoil

Adaptive simulations are performed for turbulent flow over the RAE 2822 airfoil. The Reynolds number and free stream Mach number are $\mathrm{Re} = 6.5 \times 10^6$ and $M_\infty = 0.725$, respectively, corresponding to Case 6 conditions in reference [41]. The angle of attack is fixed at $\alpha = 2.466^o$, corresponding to the computation by Allmaras [42] 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 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 Figure 4 of reference [30].

Figure 1 shows computed lift values obtained using the output–based strategies and pure Hessian–based strategy outlined in Section 3. Each marker in the figure corresponds to a fully converged adaptive simulation. Markers of the same shape correspond to the same adaptive strategy but with different error tolerances e_o specified. The experimental value corresponds to Case 6 in reference [41]. The Allmaras value [42] is 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 [39]. 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 colored markers in Figure 1 correspond to corrected values of the lift. The corrected values are computed using Equation (2). Note that corrections for the lift are only computed in the lift–based and multi–output–based strategies since these are the only two strategies for which the lift adjoint is readily available.

Figures 2 and 3 present analogous results for the drag and moment coefficients, respectively. The moment coefficient is defined with respect to the quarter–chord location. The proposed output-based adaptive strategies offer substantial improvements in accuracy over pure Hessian–based adaptation, particularly for the lift and drag coefficients. The output-based strategies also predict the moment coefficient accurately and with less variability than does the Hessian–based method. In light of the poor lift and drag predictions by the Hessian–based method, the relatively accurate moment predictions are likely the result of a fortuitous cancellation of errors. As expected, the corrected outputs are consistently more accurate than the corresponding uncorrected values. Furthermore, the most successful output–based strategy for any given coefficient is generally the strategy for which the coefficient is specifically designed, although the corresponding improvement is minor relative to the accuracy gains obtained over the Hessian–based results. For a similar level of output accuracy, the output–based strategies yield grids with an order of magnitude less nodes than the Hessian–based grids.

Figure 4 compares final adaptive grids from the domain–integral–based, drag–based, and Hessian–based strategies. Perhaps the most distinguishing feature in these plots is the resolution of the wake downstream of the airfoil. The wake appears to be truncated prematurely in the Hessian–based grid. In this region, wake refinement gets propagated downstream by the Hessian–based algorithm at a very slow rate. In comparison, the output–based algorithms are able to detect and resolve the wake far more rapidly over a comparable number of adaptive iterations. The drag–based algorithm only resolves the wake to approximately one chord–length downstream of the trailing edge. Evidently, this is all that is needed in order to accurately predict the drag. In the case of the domain–integral–based strategy, the algorithm refines the wake all the way to the farfield boundary in order to accurately capture the Mach number deficit in the wake. The inaccurate force 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 much larger overall grid size.

Figure 5 shows intermediate grids during the first few iterations in a domain–integral–based adaptive simulation. After 7 iterations, wake resolution has been propagated by more than 40 chords downstream of the airfoil. This compares with less than 1 chord of wake resolution by the Hessian–based algorithm after 16 iterations.

4.2 Advanced EET 3-Element Airfoil

Adaptive simulations are performed for turbulent flow over the Advanced Energy Efficient Transport (EET) three element airfoil [35]. The Reynolds number (based on the chord of the EET with elements retracted), free stream Mach number, and angle of attack are $\mathrm{Re} = 9 \times 10^6$, $M_\infty = 0.26$, and $\alpha = 8^o$, respectively. The farfield boundary is placed at 100 chords.

Figure 6 shows predicted lift and drag values using the proposed output–based strategies and pure Hessian–based adaptation. The experimental lift value is obtained from reference [35]. The Anderson *et al.* value is obtained from an independent computation [43]. Anderson *et al.* use the FUN2D flow solver. An unstructured grid comprised of 70686 nodes is employed for their 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.

The predicted coefficients from the output–based grids are substantially more accurate than those computed from the Hessian–based grids. Output–based grids with greater than 20000 nodes are essentially converged with respect to the computed lift. Conversely, the lift and drag values from the finest of the Hessian–based grids are still in error by over 20% and 600%, respectively.

Figure 7 shows the final adapted grid from the domain–integral–based adaptive strategy. The plot illustrates the detailed resolution of the wakes emanating from the slat, main element, and flap.

5 Conclusion

This paper presented applications of an anisotropic grid–adaptive method for functional outputs. The method combines aspects of adjoint error estimation/correction for functional outputs and Hessian–based anisotropic grid adaptation. Several adaptive strategies were proposed for aerodynamic simulations in which the accurate prediction of multiple outputs is required from a single adaptive simulation.

Airfoil test cases were presented to demonstrate the various adaptive strategies. Turbulent flow was simulated over a single element airfoil at cruise conditions and over a multi-element airfoil in high-lift configuration with flow separation. Numerical results indicate that the lift, drag and moment coefficients are accurately predicted by all of the output–based strategies considered. Slightly better accuracy is obtained in the output(s) for which a particular strategy is specifically designed. Furthermore, the output-based strategies were all shown to be significantly more efficient than pure Hessian-based adaptation, generally yielding grids with an order of magnitude less nodes for a similar level of output accuracy.

One of the proposed strategies attempts to provide overall solution accuracy by targeting an integral of the square of the relative Mach number over the entire domain. This strategy was shown to be far more efficient than pure Hessian–based adaptation (based on the Mach number Hessian) at rapidly resolving the wake of a single–element airfoil at cruise conditions to large distances downstream of the airfoil. This characteristic could be beneficial in applications where shear layers emanating from certain upstream components have significant impact on the aerodynamics of downstream components.

Some of the most important quantities needed from a CFD simulation are often integral outputs, such as lift and drag. It seems natural, therefore, that the accuracy of a simulation should be measured by the accuracy of these key outputs. This work and previous studies have shown that some of the most commonly used feature–based and Hessian–based adaptive methods 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. Developments in output–based grid adaptation are also expected to enhance the applicability of CFD for gradient–based design optimization, particularly those implementations that already employ adjoint solvers for the calculation of design sensitivities [38, 44, 45].

Acknowledgements

The authors wish to thank Steven Allmaras, Forrester Johnson, and James Thomas for valued comments, suggestions, and support. This work was funded under NASA contracts NAG1-2275, NAG-1-02037, NAG1-03035.

References

[1] M. J. Aftosmis and M. J. Berger. Multilevel error estimation and adaptive h-refinement for cartesian meshes with embedded boundaries. AIAA paper 2002-0863, 2002.

[2] T. J. Baker. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements Anal. Design*, 25:243–273, 1997.

[3] M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *Int. J. Numer. Meth. Fluids*, 25:475–491, 1997.

[4] 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*, 32:725–744, 2000.

[5] R. Löhner. Finite element methods in CFD: Grid generation, adaptivity and parallelization. In *von Karman Institute Lecture Series*, AGARD Publications R-787, 1992.

[6] D. J. Mavriplis. Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes. *Int. J. Numer. Meth. Fluids*, 34:93–111, 2000.

[7] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comput. Phys.*, 72:449–466, 1987.

[8] S. Z. Pirzadeh. An adaptive unstructured grid method by grid subdivision, local remeshing, and grid movement. AIAA paper 99-3255, 1999.

[9] G. P. Warren, W. K. Anderson, J. T. Thomas, and S. L. Krist. Grid convergence for adaptive methods. AIAA paper 91-1592, 1991.

[10] W. A. Wood and W. L. Kleb. On multi–dimensional unstructured mesh adaption. AIAA paper 99-3254, 1999.

[11] R. Becker and R. Rannacher. Weighted *a posteriori* error control in finite element methods, in *Proceedings of ENUMATH-97, Heidelberg*, (World Scientific, Singapore). pages 621–637, 1998.

[12] R. Becker and R. Rannacher. An optimal control approach to *a posteriori* error estimation in finite element methods, in *Acta Numerica 2001* (Ed. A. Iserles), Cambridge University Press. 2001.

[13] M. Braack and R. Rannacher. Adaptive finite element methods for low–Mach–number flows with chemical reactions. VKI Lecture Series 1999–03, 1999.

[14] M. B. Giles and E. Süli. Adjoint methods for PDEs: *a posteriori* error analysis and postprocessing by duality, *Acta Numerica 2002*, Cambridge University Press. 2002.

[15] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183:508–532, 2002.

[16] P. Houston and E. Süli. *hp*–adaptive discontinuous Galerkin finite element methods for first–order hyperbolic problems. *SIAM J. Sci. Comput.*, 23(4):1226–1252, 2001.

[17] 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.

[18] 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. *J. Comput. Phys.*, 172:401–425, 2001.

[19] J. Peraire and A. T. Patera. Bounds for linear–functional outputs of coercive partial differential equations: Local indicators and adaptive refinement. In *Advances in Adaptive Computational Methods in Mechanics* (Ed. P. Ladevèze and J. T. Oden), Elsevier Science Ltd., 1998.

[20] J. Peraire and A. T. Patera. Asymptotic *a posteriori* finite element bounds for the outputs of non–coercive problems: The Helmholtz and Burgers equations. *Comp. Meth. Appl. Mech. Engrg.*, 171:77–86, 1999.

[21] R. Rannacher. Adaptive Galerkin finite element methods for partial differential equations. *Journal of Computational and Applied Mathematics*, 128:205–233, 2001.

[22] E. Süli, P. Houston, and B. Senior. *hp*–discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity. In *Numerical Methods for Fluid Dynamics VII* (Ed. M. J. Baines), ICFD Oxford, 2001.

[23] M. B. Giles and N. A. Pierce. Adjoint equations in CFD: Duality, boundary conditions and solution behavior. AIAA paper 97-1850, 1997.

[24] M. B. Giles and N. A. Pierce. Adjoint error correction for integral outputs. in *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, editors T. Barth and H. Deconinck, volume 25, in *Lecture Notes in Computational Science and Engineering*, Springer–Verlag, 2002.

[25] M. B. Giles and N. A. Pierce. Improved lift and drag estimates using adjoint Euler equations. AIAA paper 99-3293, 1999.

[26] N. A. Pierce and M. B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Rev.*, 42(2):247–264, 2000.

[27] 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.*, 164:204–227, 2000.

[28] D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *J. Comput. Phys.*, 176:40–69, 2002.

[29] D. A. Venditti. *Grid Adaptation for Functional Outputs of Compressible Flow Simulations.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2002.

[30] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. *J. Comput. Phys.*, 187:22–46, 2003.

[31] M. A. Park. Adjoint–based, three–dimensional error prediction and grid adaptation. AIAA paper 2002-3286, 2002.

[32] M. A. Park. Three–dimensional turbulent RANS adjoint–based error correction. AIAA paper 2003-3849, 2003.

[33] L. Formaggia, S. Micheletti, and S. Perotto. Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection–diffusion–reaction and the Stokes problems. MOX Report No. 15, Politecnico of Milano, Italy. Submitted for publication in *Mathematics and Computers in Simulation*, 2003.

[34] S. Micheletti and S. Perotto. Anisotropic mesh adaptivity in CFD. MOX Report No. 29, Politecnico of Milano, Italy. Submitted for publication in "Adaptive Mesh Refinement–Theory and Applications", *Lecture Notes in Computational Sciences and Engineering*, Proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods, Eds.: T. Plewa, T. Linde, V. G. Weirs, Springer–Verlag, 2003.

[35] J. C. Lin and C. J. Dominick. Optimization of an advanced design three-element airfoil at high Reynolds numbers. AIAA paper 95-1858, 1995.

[36] W. K. Anderson and D. L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers Fluids*, 23:1–21, 1994.

[37] E. Nielsen. FUN2D/3D Fully Unstructured Navier–Stokes User Manual. NASA Langley Research Center, Computational Modeling and Simulation Branch, Virginia, 2002, `http://fun3d.larc.nasa.gov`.

[38] E. J. Nielsen and W. K. Anderson. Aerodynamic design optimization on unstructured meshes using the Navier–Stokes equations. *AIAA J.*, 37(11):1411–1419, 1999.

[39] P. R. Spalart and S. R. Allmaras. A one–equation turbulence model for aerodynamic flows. *La Recherche Aérospatiale*, 1:5–21, 1994.

[40] F. Hecht. BAMG: Bidimensional Anisotropic Mesh Generator, INRIA–Rocquencourt, France, 1998, `http://www-rocq.inria.fr/gamma/cdrom/www/bamg/eng.htm`.

[41] P. H. Cook, M. A. McDonald, and M. C. P. Firmin. Aerofoil RAE 2822 - pressure distributions, and boundary layer and wake measurements. AGARD-AR-138, 1979.

[42] S. R. Allmaras. personal communication.

[43] W. K. Anderson, R. D. Rausch, and D. L. Bonhaus. Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids. AIAA paper 95-1740, 1995.

[44] J. Elliott and J. Peraire. Practical 3D aerodynamic design and optimization using unstructured meshes. *AIAA J.*, 35:1479–1486, 1997.

[45] A. Jameson. Aerodynamic design via control theory. *J. Sci. Comput.*, 3:233–260, 1988.

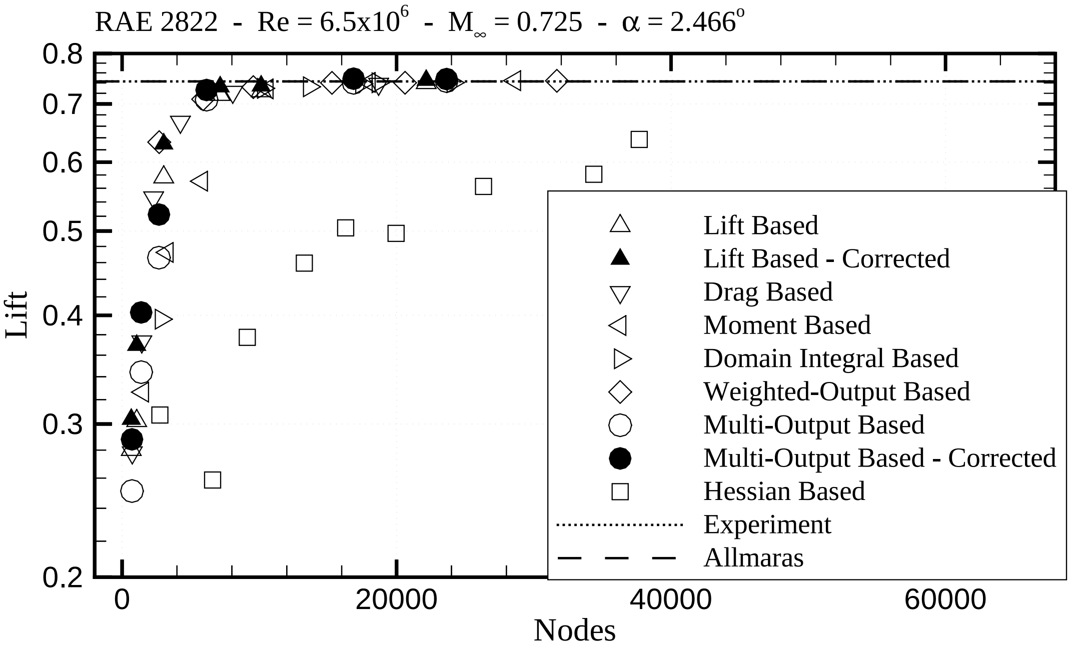

Figure 1: RAE 2822 Airfoil test case: Re $= 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.466^o$. Lift coefficient on final adapted grid for various adaptive strategies and error tolerances.

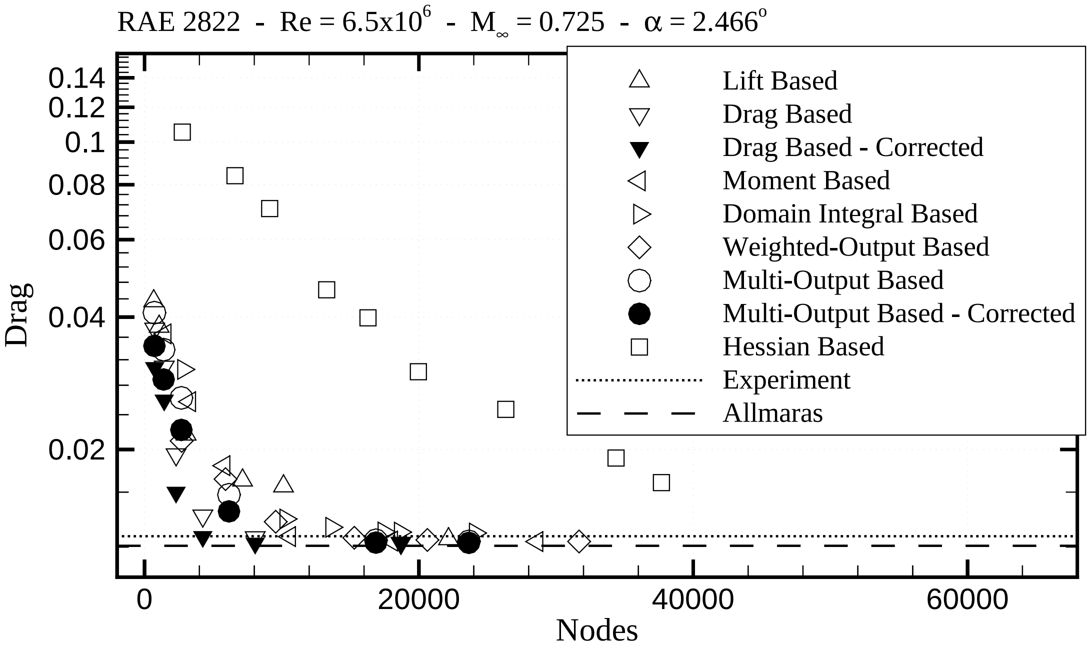

Figure 2: RAE 2822 Airfoil test case: Re $= 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.466^o$. Drag coefficient on final adapted grid for various adaptive strategies and error tolerances.

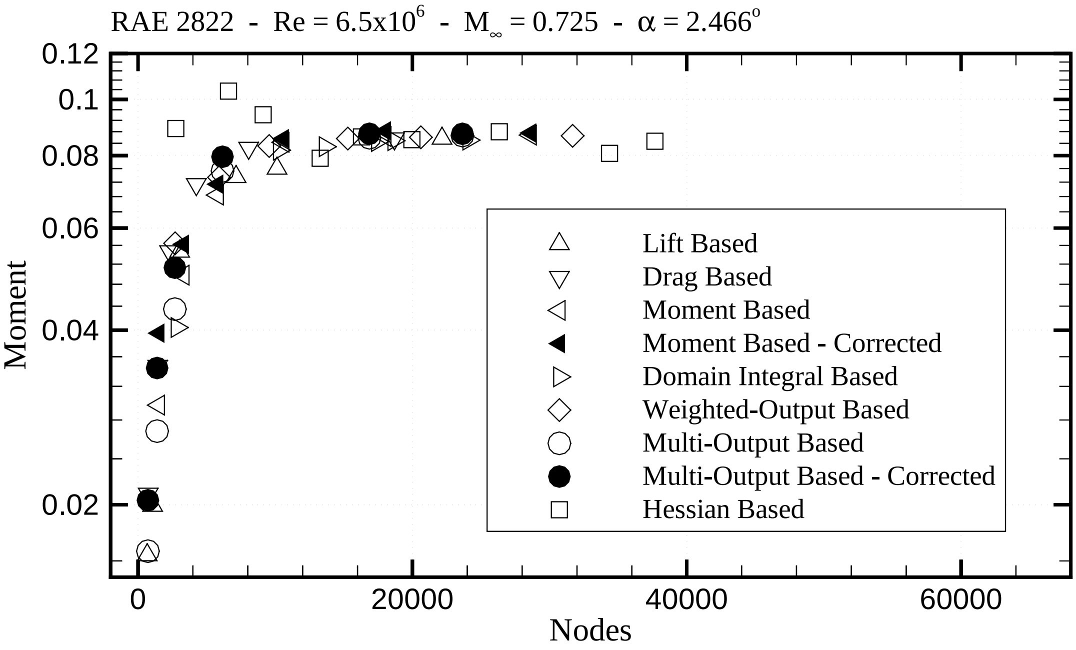

Figure 3: RAE 2822 Airfoil test case: Re $= 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.466^o$. Moment coefficient on final adapted grid for various adaptive strategies and error tolerances.

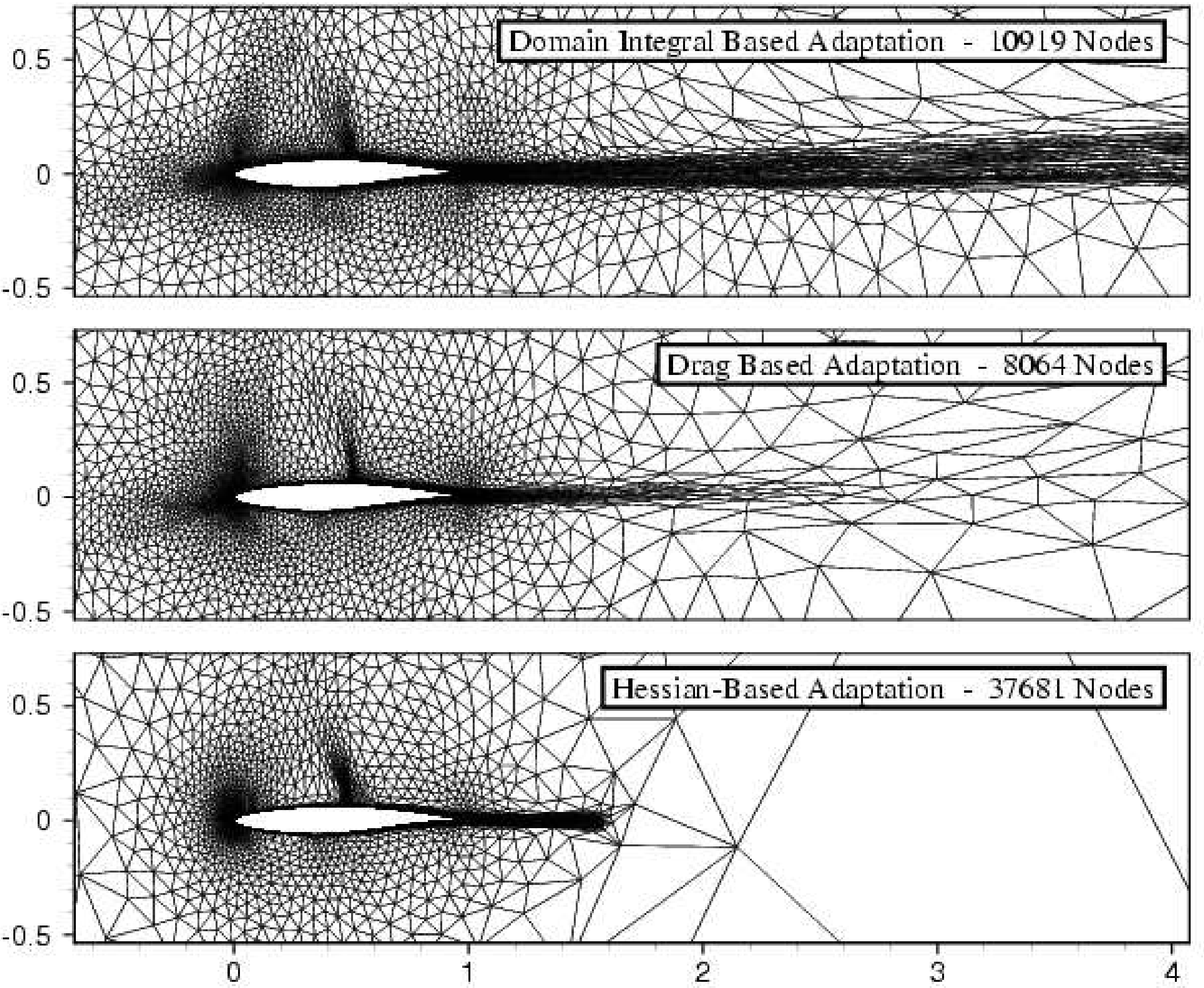

Figure 4: RAE 2822 Airfoil test case: Re $= 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.466^o$. Final adapted grids. Top: domain–integral–based adaptation; middle: drag–based adaptation; Hessian–based adaptation.

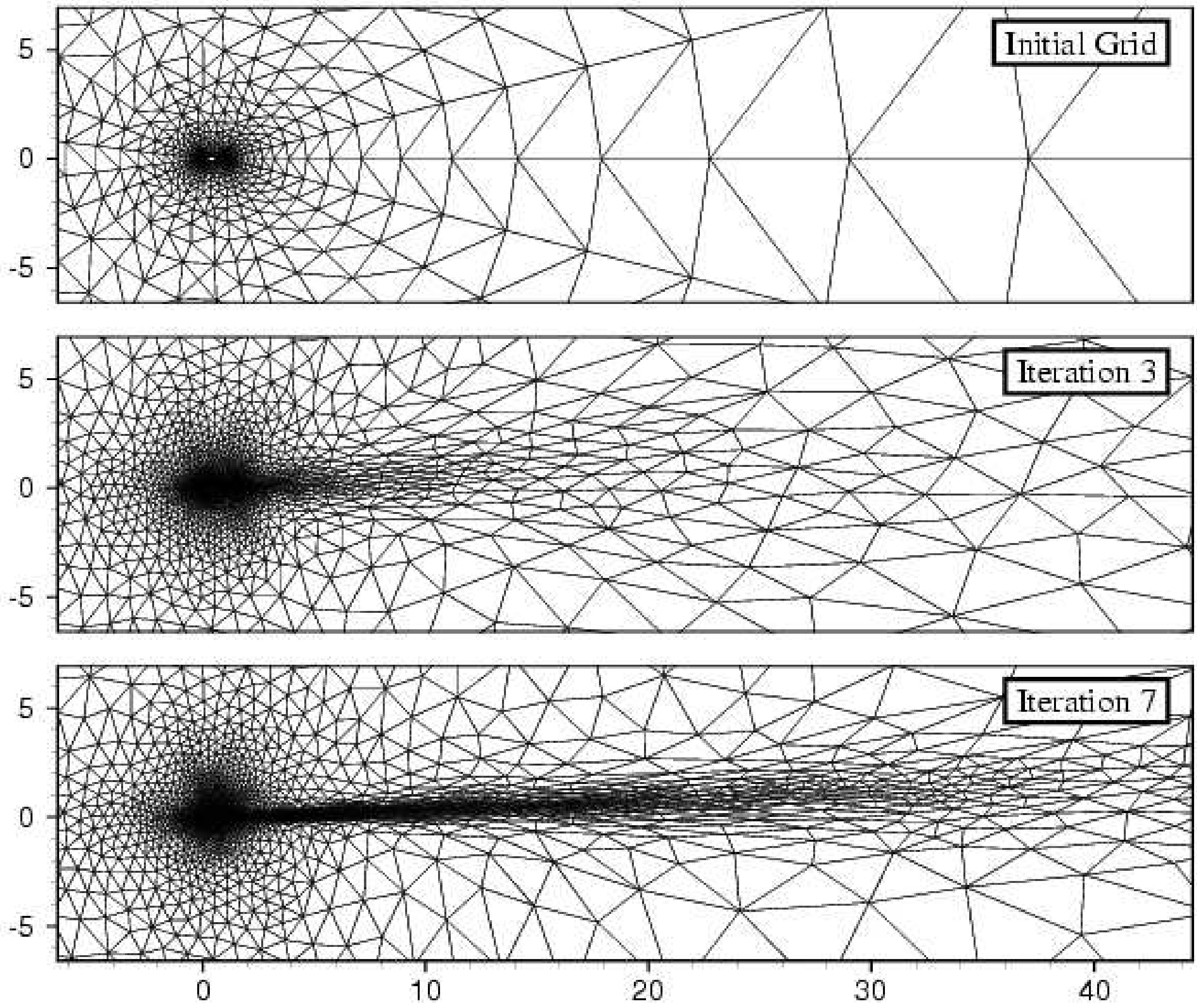

Figure 5: RAE 2822 Airfoil test case: Re $= 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.466^o$. Intermediate grids during a domain-integral-based adaptive run. Top: initial grid; middle: after 3 iterations; after 7 iterations.

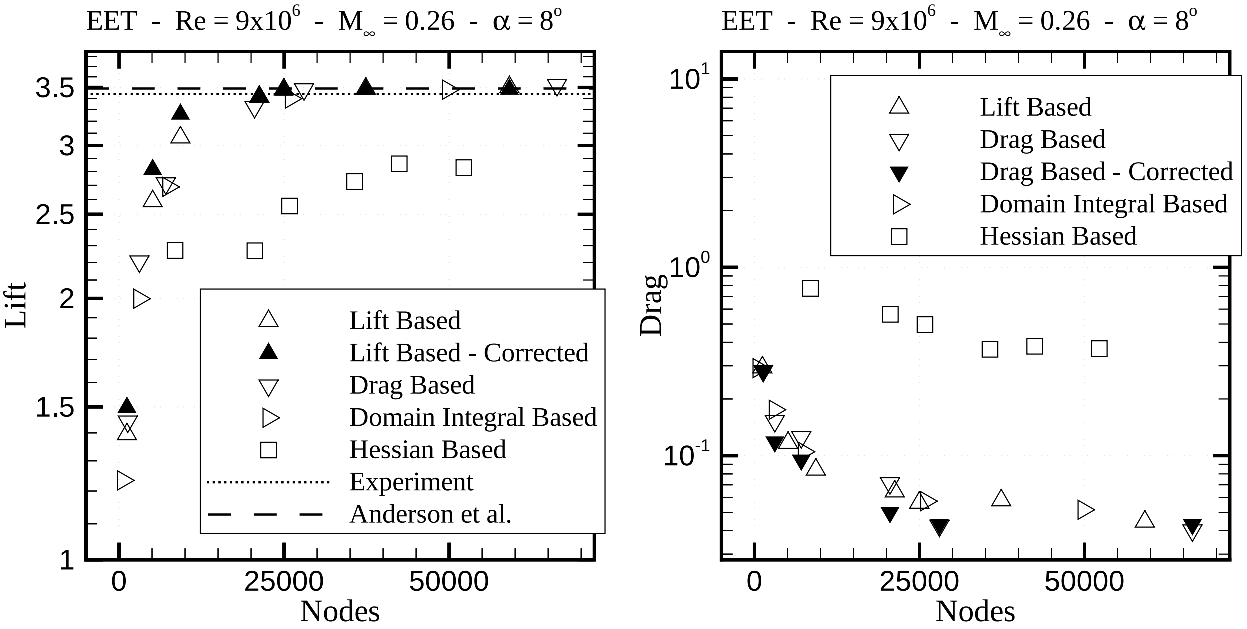

Figure 6: Advanced EET Three Element Airfoil test case: Re $= 9\times 10^6$, $M_\infty = 0.26$, $\alpha = 8^o$. Lift (left) and drag (right) coefficients on final adapted grid for various adaptive strategies and error tolerances.

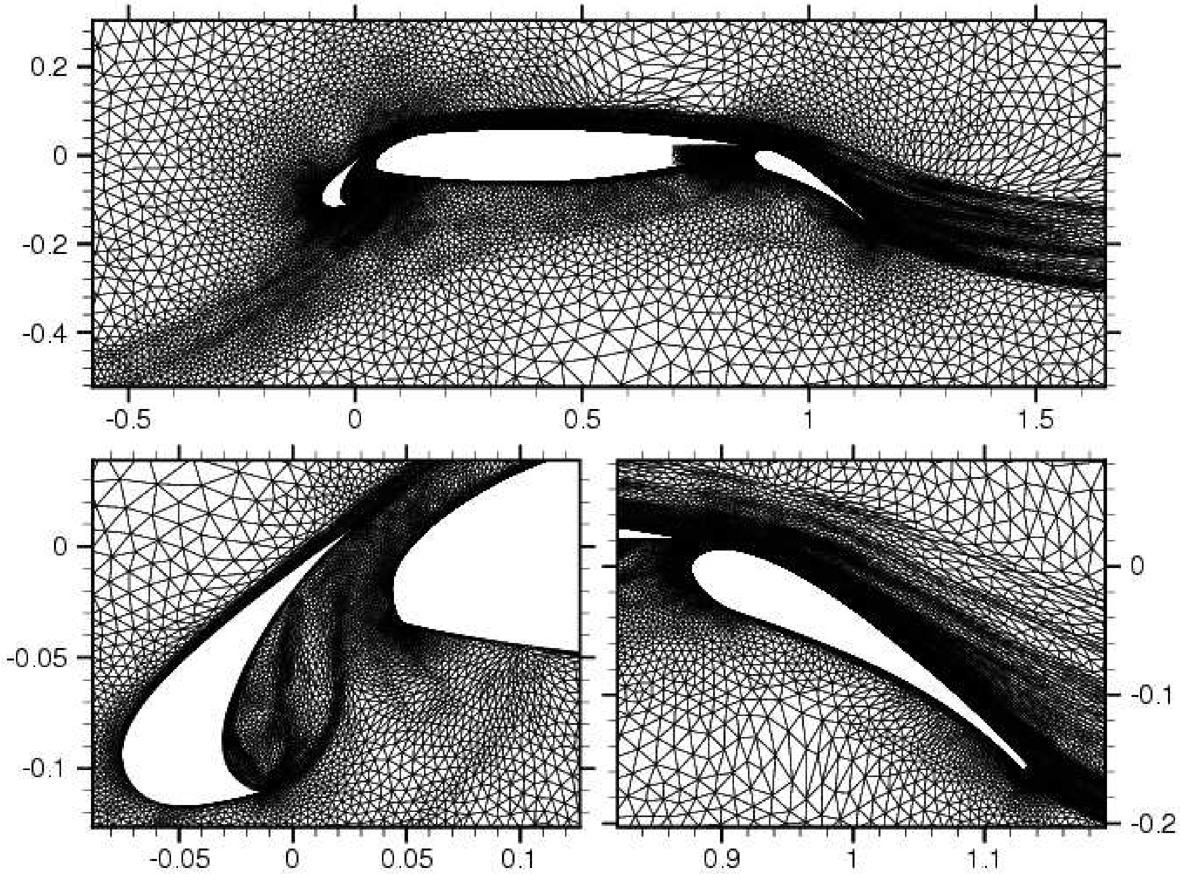

Figure 7: Advanced EET Three Element Airfoil test case: Re $= 9 \times 10^6$, $M_\infty = 0.26$, $\alpha = 8^o$. Top: final domain–integral–based adapted grid; lower left: blowup near slat; lower right: blowup near flap.