

Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics: Overview and Recent Results

Krzysztof J. Fidkowski *

University of Michigan, Ann Arbor, MI 48109

David L. Darmofal †

Massachusetts Institute of Technology, Cambridge, MA 02139

Error estimation and control are critical ingredients for improving the reliability of computational simulations. Adjoint-based techniques can be used to both estimate the error in chosen solution outputs and to provide local indicators for adaptive refinement. This article reviews recent work on these techniques for Computational Fluid Dynamics (CFD) applications in aerospace engineering. The definition of the adjoint as the sensitivity of an output to residual source perturbations is used to derive both the adjoint equation, in fully-discrete and variational formulations, and the adjoint-weighted residual method for error estimation. Assumptions and approximations made in the calculations are discussed. Presentation of the discrete and variational formulations enables a side-by-side comparison of recent work in output error estimation using the finite volume method and the finite element method. Recent adaptive results from a variety of applications show the power of output-based adaptive methods for improving the robustness of CFD computations. However, challenges and areas of additional future research remain, including computable error bounds and robust mesh adaptation mechanics.

I Introduction

The accessibility, fast turnaround time, and almost arbitrary test conditions offered by Computational Fluid Dynamics (CFD) make it an attractive tool in aerospace design. CFD simulations with sophisticated physical modeling are now used regularly to reduce design cycle costs and to improve final product design. This prevalence of CFD has been driven by increasing computational power and by improvements in numerical methods, to the extent that complex simulations on general three-dimensional geometries are now routine.

Such powerful simulation capability is a remarkable achievement for CFD, but it also comes with a new liability: ensuring that the computed solutions are sufficiently accurate. Typically, this liability is managed by practitioners knowledgeable about the assumptions and limitations of the models and discretization. However, even very experienced practitioners cannot quantify the error in a discrete approximation of a complex flowfield. In addition, reliance on best-practice guidelines for mesh generation and on previous experience is an open-loop solution that leaves the door open to large amounts of numerical error for computations on

*Assistant Professor, Department of Aerospace Engineering, 3029 François-Xavier Bagnoud Building, Member AIAA (corresponding author).

†Associate Department Head / Associate Professor, Department of Aerospace Engineering, Building 33, Room 207. Senior Member AIAA.

novel configurations.

Even for relatively standard simulations, questions arise regarding the robustness with which CFD methods can accurately compute outputs of interest. An example is the American Institute of Aeronautics and Astronautics Drag Prediction Workshop (DPW),¹⁻³ in which force and moment outputs for a representative set of wing-body geometries and flow conditions were compared across codes in industry and academia. Results from submissions have consistently shown a wide degree of scatter in computed outputs. For example, drag coefficient variations of .0025 observed in the third DPW on a DLR-F6 wing body (Figure 1) translate to a difference of over 100 passengers on a large transport aircraft.^{4,5} The results from this workshop constitute only a slight improvement over the results from the two previous workshops,^{1,2} even though computational power has increased substantially. Moreover, additional tests in the third DPW show that discrepancies persist even for simple wing-only geometries. For these geometries, results obtained using the same code but on independently-generated meshes appear to converge to different output values with uniform mesh refinement.⁷ The lack of grid convergence is illustrated in Figure 2 and indicates that the highly-disparate length scales in this three dimensional flow are not adequately resolved even with some of the finest *a priori* generated meshes.

The observations above suggest that gains in computational power coupled with best-practice expert mesh generation will be insufficient to reliably decrease numerical error to acceptable levels in the increasingly complex problems of the future. As such, error estimation and control are critical ingredients for improving the reliability of computational simulations. Control of errors is likely to be done most efficiently through adaptive methods in which the discretization is iteratively improved through local mesh refinement and/or increased order of accuracy in regions that most contribute to the solution error. A key feature of error estimation is the ability to identify such regions.

The general idea of error estimation is not a new concept, and a number of previous works have reviewed

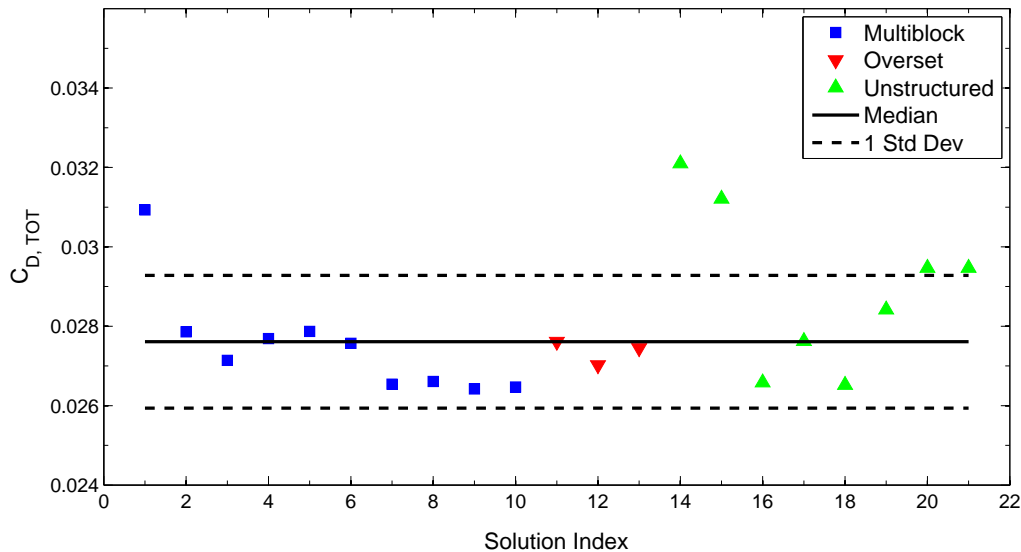


Fig. 1 DPW III results: total drag coefficient predictions for the DLR-F6 wing-body at $M = 0.75$, $C_L = 0.5$, $Re = 5 \times 10^6$. The solution index differentiates between different codes, turbulence models, and mesh types.^{3,6}

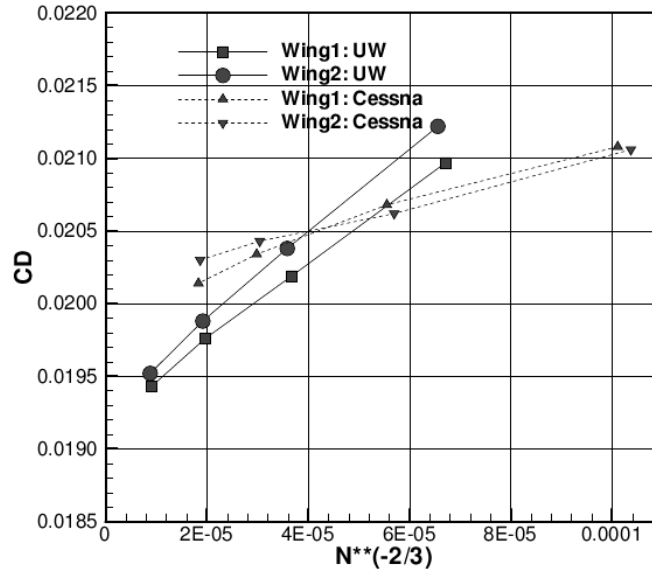


Fig. 2 DPW III wing-only results: total, pressure, and skin friction drag convergence for two families of grids of two wing geometries, at $M = 0.76$, $\alpha = 0.5^\circ$, $Re = 5 \times 10^6$. One set of grids was generated by Cessna Aircraft Co. and the other by the University of Wyoming. Reproduced with permission from.⁷

the subject. In the context of error estimates that also provide local indicators, Verfurth analyzes *a posteriori* error estimates for elliptic partial differential equations and shows an equivalence between estimates based on local residuals and on solutions of local problems.⁸ Ainsworth and Oden focus on mechanics and consider *a posteriori* energy norm error estimates for linear elliptic boundary value problems.⁹ Johnson *et al* note a marked gap between theory and practice in error estimation for CFD, and they derive quantitative discretization error estimates for laminar flows.^{10,11} In a recent work, Roy reviews various strategies for using discretization error estimates to drive mesh adaptation.¹²

As we describe in this review, adjoint-based techniques can be used to both estimate error in solution outputs (such as lift and drag) and provide local indicators for adaptive methods. Becker and Rannacher present a thorough review of the adjoint-weighted residual method for *a posteriori* error estimation in finite element discretizations of elliptic, parabolic, and hyperbolic equations.¹³ In addition, Giles and Pierce^{14,15} describe adjoint correction techniques and Giles and Süli¹⁶ review *a posteriori* output error estimation for finite element methods applied to linear and nonlinear partial differential equations relevant to CFD. Complementing these previous works, the purpose of this paper is to review output error estimation and mesh adaptation techniques in the context of aerospace CFD applications and to present a collection of recent results for aerospace problems including high lift, hypersonic heating, sonic boom, and launch abort vehicles. We address output error estimation techniques that are applicable to both finite element and general discretizations including finite difference and finite volume methods.

The structure of the paper is as follows. Section II introduces output adjoint solutions for both fully-discrete and variational problems. Section III then reviews the adjoint-weighted residual method for output-based error estimation. Error localization and mesh adaptation techniques are reviewed in Section IV. Section V presents recent implementations and results for aerospace engineering applications. Finally, challenges and ongoing research are discussed in Section VI.

II Outputs and Adjoints

Since the work of Aubin and Nitsche in *a priori* optimal order convergence proofs,¹⁷ adjoint solutions have been used in a variety of contexts, ranging from design optimization^{18–23} to output error estimation.^{24–36} Adjoint solutions are desirable in all of these contexts for the output sensitivity information that they provide. Starting from the output-sensitivity property, this section derives the adjoint equations in discrete and variational formulations.

II.1 Fully-Discrete Formulation

Consider a partial differential equation discretized into N_h , possibly nonlinear, algebraic equations,

$$\mathbf{R}_h(\mathbf{u}_h) = 0, \quad (1)$$

where $\mathbf{u}_h \in \mathbb{R}^{N_h}$ is the vector of unknowns and $\mathbf{R}_h \in \mathbb{R}^{N_h}$ is the vector of residuals that must be driven to zero. The subscript h denotes the fineness of the discretization and encompasses both mesh size and interpolation order. Given a scalar output $J_h(\mathbf{u}_h)$, the associated adjoint vector, $\boldsymbol{\psi}_h \in \mathbb{R}^{N_h}$, is the sensitivity of J_h to an infinitesimal residual perturbation, $\delta\mathbf{R}_h \in \mathbb{R}^{N_h}$, to the nonlinear system:

$$\delta J_h \equiv J_h(\mathbf{u}_h + \delta\mathbf{u}_h) - J_h(\mathbf{u}_h) \equiv \boldsymbol{\psi}_h^T \delta\mathbf{R}_h, \quad (2)$$

where $\delta\mathbf{u}_h$ is the infinitesimal solution perturbation satisfying

$$\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h} \delta\mathbf{u}_h + \delta\mathbf{R}_h = 0, \quad (3)$$

which is obtained by linearizing Eq. 1. The linearization assumes the discrete equations are differentiable. Further, assuming that the output is also differentiable,

$$\delta J_h = \frac{\partial J_h}{\partial \mathbf{u}_h} \delta\mathbf{u}_h = \boldsymbol{\psi}_h^T \delta\mathbf{R}_h = -\boldsymbol{\psi}_h^T \frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h} \delta\mathbf{u}_h, \quad (4)$$

where Eqs. 2 and 3 were used in the second and third equalities, respectively. In order for Eq. 4 to hold for all perturbations, we require

$$\frac{\partial J_h}{\partial \mathbf{u}_h} = -\boldsymbol{\psi}_h^T \frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}, \quad (5)$$

from which $\boldsymbol{\psi}_h$ must satisfy the *discrete adjoint equation*

$$\left(\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h} \right)^T \boldsymbol{\psi}_h + \left(\frac{\partial J_h}{\partial \mathbf{u}_h} \right)^T = 0. \quad (6)$$

II.2 Variational Formulation

In a variational setting, consider a general semilinear form arising from a Galerkin weighted residual statement: determine $\mathbf{u}_h \in \mathcal{V}_h$ such that

$$\mathcal{R}_h(\mathbf{u}_h, \mathbf{v}_h) = 0, \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \quad (7)$$

where \mathcal{V}_h is a finite-dimensional space of functions. The subscript h indicates a discretization of the computational domain, such as a triangulation in a finite element method. $\mathcal{R}_h(\cdot, \cdot) : \mathcal{V}_h \times \mathcal{V}_h \rightarrow \mathbb{R}$ is assumed to be a semilinear form, linear in the second argument. A scalar output of interest is denoted by $\mathcal{J}_h(\cdot) : \mathcal{V}_h \rightarrow \mathbb{R}$, where the subscript h is included because the output calculation may involve discretization-dependent terms.

Consider an infinitesimal perturbation, $\delta\mathcal{R}_h(\mathbf{v}_h)$, added to the weak statement in Eq. 7, where $\delta\mathcal{R}_h(\cdot) : \mathcal{V}_h \rightarrow \mathbb{R}$. An adjoint $\boldsymbol{\psi}_h \in \mathcal{V}_h$ can be defined as the sensitivity of the output to the residual perturbation by the following relationship:

$$\delta\mathcal{J}_h \equiv \mathcal{J}_h(\mathbf{u}_h + \delta\mathbf{u}_h) - \mathcal{J}_h(\mathbf{u}_h) = \delta\mathcal{R}_h(\boldsymbol{\psi}_h). \quad (8)$$

The infinitesimal state and residual perturbations are related via the statement:

$$\mathcal{R}'_h[\mathbf{u}_h](\delta\mathbf{u}_h, \mathbf{v}_h) + \delta\mathcal{R}_h(\mathbf{v}_h) = 0, \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \quad (9)$$

where the prime denotes a Fréchet linearization with respect to the arguments in the square brackets. Also linearizing the output,

$$\delta\mathcal{J}_h = \mathcal{J}'_h[\mathbf{u}_h](\delta\mathbf{u}_h) = \delta\mathcal{R}_h(\boldsymbol{\psi}_h) = -\mathcal{R}'_h[\mathbf{u}_h](\delta\mathbf{u}_h, \boldsymbol{\psi}_h), \quad (10)$$

where Eqs. 8 and 9 were used in the second and third equalities, respectively. For these linearizations to exist, both the semilinear form and the output are assumed to be differentiable. In order for Eq. 10 to be true for general perturbations, the adjoint must satisfy the statement: find $\boldsymbol{\psi}_h \in \mathcal{V}_h$ such that

$$\mathcal{R}'_h[\mathbf{u}_h](\mathbf{v}_h, \boldsymbol{\psi}_h) + \mathcal{J}'_h[\mathbf{u}_h](\mathbf{v}_h) = 0, \quad \forall \mathbf{v}_h \in \mathcal{V}_h. \quad (11)$$

Once a basis is chosen for the weighted residual statements, Eqs. 7 and 11 are equivalent to their discrete counterparts, Eqs. 1 and 6, respectively. While the fully-discrete formulation is simpler from an implementation perspective, the weighted-residual formulation offers more rigor for error estimation.

II.3 Adjoint Consistency

Eqs. 6 and 11 yield the discrete adjoint $\boldsymbol{\psi}_h$, either as a vector or as a function in a finite-dimensional space. Of interest is how this discrete solution compares to its infinite dimensional (“exact”) counterpart. Given the exact primal solution, $\mathbf{u} \in \mathcal{V}$, satisfying

$$\mathcal{R}(\mathbf{u}, \mathbf{v}) = 0, \quad \forall \mathbf{v} \in \mathcal{V}, \quad (12)$$

for an appropriately defined space \mathcal{V} , the exact adjoint $\boldsymbol{\psi} \in \mathcal{V}$ satisfies

$$\mathcal{R}'[\mathbf{u}](\mathbf{v}, \boldsymbol{\psi}) + \mathcal{J}'[\mathbf{u}](\mathbf{v}) = 0, \quad \forall \mathbf{v} \in \mathcal{V}. \quad (13)$$

For simplicity, we have assumed that both \mathbf{u} and $\boldsymbol{\psi}$ are in \mathcal{V} . However, the space for the adjoint solution does not have to be the same as the space for the primal solution.³⁷

The exact adjoint can be regarded as a Green’s function relating source perturbations in the original partial differential equation to perturbations in the output.^{38,39} To demonstrate this interpretation, a

sample adjoint solution is illustrated in Figure 3 for subsonic flow over a lifting airfoil. Upstream of the

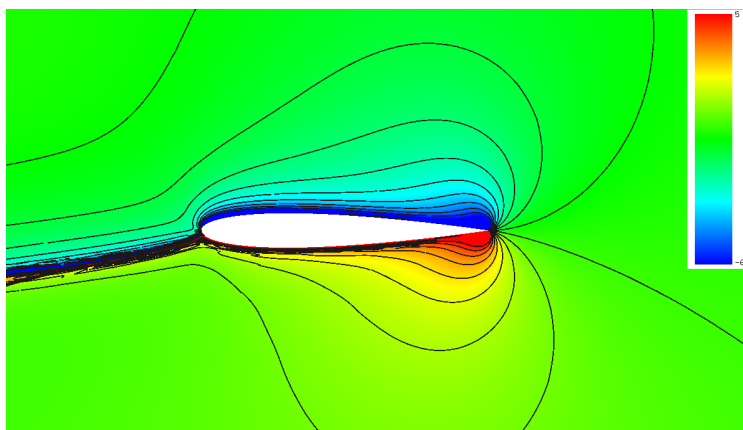


Fig. 3 x -momentum component of the lift adjoint for a NACA 0012 airfoil at $M = 0.4$, $\alpha = 5^\circ$. A positive residual perturbation to the x -momentum equation increases the lift where the adjoint is positive (red shading) and decreases the lift where the adjoint is negative (blue shading).

airfoil, the adjoint is seen to vary rapidly across the stagnation streamline. This behavior was suggested in the analysis of Giles and Pierce who found that a square root singularity with respect to distance from the stagnation streamline exists for sources that perturb the stagnation pressure.³⁸

To be precise, the adjoint field in Figure 3 is the discrete adjoint solution on a fine mesh. It can only be regarded as a faithful representation of the exact adjoint if the discretization is in some manner consistent with the exact adjoint problem. *Primal consistency* in the variational problem requires that the exact solution \mathbf{u} satisfies the discrete variational statement,

$$\mathcal{R}_h(\mathbf{u}, \mathbf{v}) = 0, \quad \forall \mathbf{v} \in \mathcal{W}_h, \quad (14)$$

where $\mathcal{W}_h = \mathcal{V}_h + \mathcal{V} = \{h = f + g : f \in \mathcal{V}_h, g \in \mathcal{V}\}$. Similarly, the combination of the discrete semi-linear form \mathcal{R}_h and the functional \mathcal{J}_h is said to be *adjoint consistent* if^{37,40,41}

$$\mathcal{R}'_h[\mathbf{u}](\mathbf{v}, \boldsymbol{\psi}) + \mathcal{J}'_h[\mathbf{u}](\mathbf{v}) = 0, \quad \forall \mathbf{v} \in \mathcal{W}_h. \quad (15)$$

Discretizations that are not adjoint consistent may still be *asymptotically adjoint consistent* if Eq. 15 holds in the limit $h \rightarrow 0$, by which we mean the limit of uniformly increasing resolution, over suitably normalized $\mathbf{v} \in \mathcal{W}_h$.

Adjoint consistency has an impact on the convergence of not only the adjoint approximation but also the primal approximation.^{16,37,40–44} In error estimation, an adjoint-inconsistent discretization can lead to irregular or oscillatory adjoint solutions that pollute the error estimate with noise and lead to adaptation in incorrect areas. Enforcing adjoint consistency imposes restrictions on the output definition and on the interior and boundary discretizations that enter into the semi-linear form. These restrictions have been studied by several authors in the context of the discontinuous Galerkin method.^{37,40,42} In general, discretizations that are found to be adjoint inconsistent can often be made adjoint consistent by adding terms to either the semi-linear form or the output functional.

II.4 Adjoint Implementation

The discrete adjoint is obtained as the solution to the linear system in Eq. 6. Since this system requires a linearization about the primal solution, the primal problem in Eq. 1 is generally solved before the adjoint problem. Concurrent primal-adjoint solutions, however, have been investigated with certain superconvergence properties in the output estimates.⁴⁵

The implementation of the adjoint solve varies depending on the structure of the code. When the full Jacobian matrix, $\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}$, and an associated linear solver are available from solution of the primal problem, the transpose linear solve can be implemented in a straightforward manner. For example, if a Krylov method is used for the linear solve, the adjoint solve will require transpose applications of the matrix and the preconditioner, which generally pose little difficulty when $\frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}$ is stored.

Conversely, when the Jacobian matrix is not stored, the discrete adjoint solve is more involved. The fact that the Jacobian matrix is transposed in Eq. 6 means that all operations in the primal solve must be linearized, transposed, and applied in reverse order for use in the adjoint solve. For example, if a finite volume code calculates residuals by reconstructing the flow state and then computing nonlinear fluxes, the adjoint residual must be obtained by first applying the transpose of the linearized fluxes and then the transpose of the reconstruction operator. The required linearizations are often calculated analytically by hand, although automatic code differentiation techniques are becoming increasingly sophisticated.^{46,47} Multistage and multigrid solution schemes have to be modified to ensure that the asymptotic stability of the adjoint solver is equal to that of the original primal flow solver.⁴⁸ Implicit schemes employing point or line relaxation also have to be modified to preserve discrete duality, as discussed by Nielsen *et al.*⁴⁹

Adjoint approximations can also be developed by deriving the adjoint partial differential equations and then discretizing them. This approach is referred to as the continuous adjoint technique (as opposed to the discrete adjoint techniques described above) and was pioneered for aerospace applications by Jameson.¹⁸ The continuous adjoint method has largely not been applied in the context of output error estimation and adaptation, though no barrier exists to prevent such an application.

III Error Estimation

III.1 Forms of Error Estimation

The error in a solution can be quantified by various means. Discretization error is the difference between the discrete solution and the exact solution. Its magnitude is governed by the size of the spatial and temporal mesh spacings, and it can be measured locally on individual elements or globally under a chosen norm. For general problems, the exact solution is unknown and the discretization error must be estimated, often using reconstructions based on smoothness assumptions. Another error estimate relies on the residual, which is obtained by substituting the approximate solution into the underlying partial differential equation.⁵⁰⁻⁵² Nonzero residuals, calculated point-wise or integrated on an enriched space, indicate regions where the governing equations are not strongly enforced. Residual error estimates can also be expressed locally or integrated globally under a norm, although care must be taken in the choice of norm for hyperbolic problems to prevent uncontrollable growth in the vicinity of a shock.⁵⁰

For simulations of predominantly elliptic equations, such as those of structural elasticity or low-speed flows, error estimates based on local discretization errors are often sufficient.⁸ However, many aerospace CFD applications deal with hyperbolic equations, for which such estimates lose their efficacy. Zhang *et*

al compare adaptive results using discretization error and residual indicators for the Euler equations.^{53,54} For one-dimensional, subsonic flows, Zhang *et al* find that a residual indicator is more efficient compared to a discretization-error indicator in driving the adaptation to reduce the total solution error. However, for transonic or multi-dimensional flows, neither indicator is adequately effective. In general, error estimates based on residual or discretization errors fail to capture propagation effects that are inherent to hyperbolic problems.⁵⁵ For hyperbolic problems, the residual and discretization error may not necessarily be large in certain crucial areas that significantly affect the solution downstream and the computed outputs. For example, for separated flow over an airfoil, small perturbations in certain upstream areas may have large effects on the location of the separation point, which in turn has a large effect on the calculated lift and drag. Stated another way, engineering outputs can be highly sensitive to discretization or residual errors in areas that may not be easily identifiable *a priori*.

Output error estimates based on adjoint analysis help to address these problems by quantifying how residual errors impact the output, accounting for propagation effects in the process. The resulting error estimate can be used to ascribe confidence levels to the engineering output and to drive an adaptive method with the goal of reducing the output error below a user-specified tolerance. This section reviews such existing output-error estimation techniques. We begin in Section III.2 by introducing the adjoint-weighted residual method that connects residuals to output error for variational discretizations. Then, we show how these techniques can be extended to general discretizations using an algebraic approach in Section III.3. Finally, Sections III.4-III.6 describe various aspects that impact the practical implementation of these error estimates including the approximation of fine mesh primal and adjoint states, the effectivity of error estimates, and the impact of shocks.

III.2 The Adjoint-Weighted Residual Method

Consider a variational solution on a “fine” discretization, $\mathbf{u}_h \in \mathcal{V}_h$, that satisfies $\mathcal{R}_h(\mathbf{u}_h, \mathbf{v}_h) = 0, \forall \mathbf{v}_h \in \mathcal{V}_h$, and a variational solution on a coarser discretization, $\mathcal{R}_H(\mathbf{u}_H, \mathbf{v}_H) = 0, \forall \mathbf{v}_H \in \mathcal{V}_H$. The discretization spaces are assumed to be nested, $\mathcal{V}_H \subset \mathcal{V}_h$, so that $\delta \mathbf{u}_h = \mathbf{u}_H - \mathbf{u}_h \in \mathcal{V}_h$. Such a situation is illustrated in Figure 4 for a one-dimensional finite element solution.

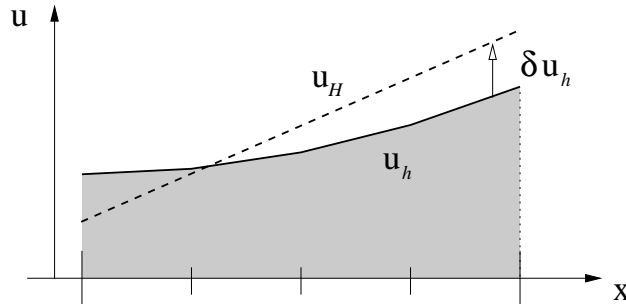


Fig. 4 Comparison in one dimension of a fine solution $\mathbf{u}_h \in \mathcal{V}_h$, a coarse solution $\mathbf{u}_H \in \mathcal{V}_H$, and the difference $\delta \mathbf{u}_h = \mathbf{u}_H - \mathbf{u}_h \in \mathcal{V}_h$. In this example, the solution spaces consist of piecewise linear functions on uniform elements, and \mathcal{V}_h is nested in \mathcal{V}_H with four times as many elements. One coarse element is shown.

To connect the output error to residuals for finite perturbations, the adjoint equation in Eq. 11 is generalized using a mean-value linearization.^{13,33,35,56} Specifically, the mean-value adjoint $\psi_h^{\text{mv}} \in \mathcal{V}_h$ is the

solution to

$$\bar{\mathcal{R}}_h[\mathbf{u}_h, \mathbf{u}_H](\mathbf{v}_h, \boldsymbol{\psi}_h^{\text{mv}}) + \bar{\mathcal{J}}_h[\mathbf{u}_h, \mathbf{u}_H](\mathbf{v}_h) = 0, \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \quad (16)$$

where $\bar{\mathcal{R}}_h : \mathcal{V}_h \times \mathcal{V}_h \rightarrow \mathbb{R}$ and $\bar{\mathcal{J}}_h : \mathcal{V}_h \rightarrow \mathbb{R}$ are defined by

$$\begin{aligned} \bar{\mathcal{R}}_h[\mathbf{u}_h, \mathbf{u}_h + \delta \mathbf{u}_h](\mathbf{v}_h, \mathbf{w}_h) &= \int_0^1 \mathcal{R}'_h[\mathbf{u}_h + \theta \delta \mathbf{u}_h](\mathbf{v}_h, \mathbf{w}_h) d\theta, \\ \bar{\mathcal{J}}_h[\mathbf{u}_h, \mathbf{u}_h + \delta \mathbf{u}_h](\mathbf{v}_h) &= \int_0^1 \mathcal{J}'_h[\mathbf{u}_h + \theta \delta \mathbf{u}_h](\mathbf{v}_h) d\theta, \end{aligned} \quad (17)$$

for $\mathbf{v}_h, \mathbf{w}_h \in \mathcal{V}_h$. Since

$$\begin{aligned} \bar{\mathcal{R}}_h[\mathbf{u}_h, \mathbf{u}_h + \delta \mathbf{u}_h](\delta \mathbf{u}_h, \mathbf{w}_h) &= \mathcal{R}_h(\mathbf{u}_h + \delta \mathbf{u}_h, \mathbf{w}_h) - \mathcal{R}_h(\mathbf{u}_h, \mathbf{w}_h), \\ \bar{\mathcal{J}}_h[\mathbf{u}_h, \mathbf{u}_h + \delta \mathbf{u}_h](\delta \mathbf{u}_h) &= \mathcal{J}_h(\mathbf{u}_h + \delta \mathbf{u}_h) - \mathcal{J}_h(\mathbf{u}_h), \end{aligned}$$

the output perturbation can be related to the residuals,

$$\begin{aligned} \mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}_h(\mathbf{u}_h) &= \mathcal{J}_h(\mathbf{u}_H) - \mathcal{J}_h(\mathbf{u}_h) \\ &= \bar{\mathcal{J}}_h[\mathbf{u}_h, \mathbf{u}_H](\delta \mathbf{u}_h) \\ &= -\bar{\mathcal{R}}_h[\mathbf{u}_h, \mathbf{u}_H](\delta \mathbf{u}_h, \boldsymbol{\psi}_h^{\text{mv}}) \\ &= -\mathcal{R}_h(\mathbf{u}_H, \boldsymbol{\psi}_h^{\text{mv}}) + \mathcal{R}_h(\mathbf{u}_h, \boldsymbol{\psi}_h^{\text{mv}}) \\ \Rightarrow \mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}_h(\mathbf{u}_h) &= -\mathcal{R}_h(\mathbf{u}_H, \boldsymbol{\psi}_h^{\text{mv}}). \end{aligned} \quad (18)$$

The assumption that $\mathcal{J}_H(\mathbf{u}_H) = \mathcal{J}_h(\mathbf{u}_H)$ was made above, which is generally true as long as any geometry used for the output calculation does not change between the two spaces. In the last step, the fact that \mathbf{u}_h is a solution to the original weighted residual statement is used, so that $\mathcal{R}_h(\mathbf{u}_h, \mathbf{w}_h) = 0, \forall \mathbf{w}_h \in \mathcal{V}_h$. The name ‘‘adjoint-weighted residual method’’ (also referred to as the dual-weighted residual method²⁹) describes Eq. 18: the adjoint solution weights the residual of the coarse solution to produce the output error. The method is also referred to as ‘‘goal-oriented’’^{34,57} or ‘‘output-based’’ error estimation.

By Galerkin orthogonality of the variational formulation, an arbitrary function in \mathcal{V}_H can be subtracted from $\boldsymbol{\psi}_h^{\text{mv}}$ in Eq. 18,

$$\mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}_h(\mathbf{u}_h) = -\mathcal{R}_h(\mathbf{u}_H, \boldsymbol{\psi}_h^{\text{mv}} - \mathbf{v}_H), \quad \forall \mathbf{v}_H \in \mathcal{V}_H. \quad (19)$$

This form of the adjoint weighted residual shows that when $\boldsymbol{\psi}_h^{\text{mv}}$ is well-approximated by the coarse space, the output error will be small. Further, applying this result locally, in regions where the fine-mesh adjoint is well-approximated by the coarse space, the contribution of local residual errors to the output will be small.

III.3 Fully-Discrete Adjoint-Weighted Residual Method

The adjoint-weighted residual method for error estimation can also be applied in a fully-discrete formulation. Again, two discretization levels are assumed: a fine one with N_h degrees of freedom, and a coarse one with N_H degrees of freedom. A representation of \mathbf{u}_H on the fine space is assumed to be given by $\mathbf{u}_h^H \equiv \mathbf{I}_h^H \mathbf{u}_H$, where $\mathbf{I}_h^H \in \mathbb{R}^{N_h \times N_H}$ is a suitably-chosen prolongation matrix. In a standard finite element discretization,

\mathbf{I}_h^H is the natural injection operator. In the general case, \mathbf{I}_h^H should be consistent with the discretization used. The perturbation in the fine solution relative to the prolonged coarse solution is $\delta \mathbf{u}_h = \mathbf{u}_h^H - \mathbf{u}_h$,

The discrete mean-value adjoint, $\boldsymbol{\psi}_h^{\text{mv}} \in \mathbb{R}^{N_h}$, satisfies

$$(\bar{\mathbf{R}}_h[\mathbf{u}_h, \mathbf{u}_h^H])^T \boldsymbol{\psi}_h^{\text{mv}} + (\bar{J}_h[\mathbf{u}_h, \mathbf{u}_h^H])^T = 0, \quad (20)$$

where $\bar{\mathbf{R}}_h \in \mathbb{R}^{N_h \times N_h}$ and $\bar{J}_h \in \mathbb{R}^{N_h}$ satisfy

$$\begin{aligned} \bar{\mathbf{R}}_h[\mathbf{u}_h, \mathbf{u}_h + \delta \mathbf{u}_h] &= \int_0^1 \frac{\partial \mathbf{R}_h}{\partial \mathbf{u}_h}[\mathbf{u}_h + \theta \delta \mathbf{u}_h] d\theta, \\ \bar{J}_h[\mathbf{u}_h, \mathbf{u}_h + \delta \mathbf{u}_h] &= \int_0^1 \frac{\partial J_h}{\partial \mathbf{u}_h}[\mathbf{u}_h + \theta \delta \mathbf{u}_h] d\theta. \end{aligned} \quad (21)$$

Since

$$\begin{aligned} \bar{\mathbf{R}}_h[\mathbf{u}_h, \mathbf{u}_h + \delta \mathbf{u}_h] \delta \mathbf{u}_h &= \mathbf{R}_h(\mathbf{u}_h + \delta \mathbf{u}_h) - \mathbf{R}_h(\mathbf{u}_h), \\ \bar{J}_h[\mathbf{u}_h, \mathbf{u}_h + \delta \mathbf{u}_h] \delta \mathbf{u}_h &= J_h(\mathbf{u}_h + \delta \mathbf{u}_h) - J_h(\mathbf{u}_h), \end{aligned}$$

the output perturbation can be related to the residuals,

$$\begin{aligned} J_H(\mathbf{u}_H) - J_h(\mathbf{u}_h) &= J_h(\mathbf{u}_h^H) - J_h(\mathbf{u}_h) \\ &= \bar{J}_h[\mathbf{u}_h, \mathbf{u}_h^H] \delta \mathbf{u}_h \\ &= -(\boldsymbol{\psi}_h^{\text{mv}})^T \bar{\mathbf{R}}_h[\mathbf{u}_h, \mathbf{u}_h^H] \delta \mathbf{u}_h \\ &= -(\boldsymbol{\psi}_h^{\text{mv}})^T \mathbf{R}_h(\mathbf{u}_h^H) + (\boldsymbol{\psi}_h^{\text{mv}})^T \mathbf{R}_h(\mathbf{u}_h) \\ \Rightarrow J_H(\mathbf{u}_H) - J_h(\mathbf{u}_h) &= -(\boldsymbol{\psi}_h^{\text{mv}})^T \mathbf{R}_h(\mathbf{u}_h^H). \end{aligned} \quad (22)$$

The output for the coarse discretization is assumed to be given by the evaluation of the output on the fine level discretization using the prolonged solution, i.e. $J_H(\mathbf{u}_H) = J_h(\mathbf{u}_h^H)$. Further, in the last step, $\mathbf{R}_h(\mathbf{u}_h) = 0$ is used.

This adjoint-weighted residual in Eq. 22 can be split into two parts by expressing the mean-value adjoint on the fine level as a correction from a prolonged coarse adjoint, $\boldsymbol{\psi}_h^{\text{mv}} = \boldsymbol{\psi}_h^{H, \text{mv}} - \delta \boldsymbol{\psi}_h^{\text{mv}}$, giving,

$$J_H(\mathbf{u}_H) - J_h(\mathbf{u}_h) = - \underbrace{\left(\boldsymbol{\psi}_h^{H, \text{mv}} \right)^T \mathbf{R}_h(\mathbf{u}_h^H)}_{\text{computable correction}} + \underbrace{(\delta \boldsymbol{\psi}_h^{\text{mv}})^T \mathbf{R}_h(\mathbf{u}_h^H)}_{\text{remaining error}}. \quad (23)$$

The first term, which would be zero due to Galerkin orthogonality for a variational formulation, is often called the *computable correction* since it can be computed without solving the primal or the adjoint on the fine level (only a residual evaluation on the fine level is required). In particular, it is nonzero for reconstruction-based finite volume schemes. While the computable correction could be used as an adaptive indicator, previous results indicate that adapting on the computable correction is not significantly better than heuristic indicators.^{58,59} Thus, in practice, the approach taken in finite volume applications has been to adapt on the *remaining error* while including the computable correction in the estimate of the output.⁶⁰

III.4 Approximations

Evaluating the output perturbation in Eq. 18 requires a residual evaluation on the fine space \mathcal{V}_h , weighted by the mean-value adjoint, ψ_h^{mv} (note: the same issues apply to the fully-discrete adjoint-weighted residual method, but for simplicity we refer only to the variational formulation in this section). A residual evaluation on \mathcal{V}_h is tractable, but solving Eq. 20 to calculate ψ_h^{mv} requires both a primal and an adjoint solve on \mathcal{V}_h . These calculations on \mathcal{V}_h are expensive and defeat the purpose of estimating the error since $\mathcal{J}_H - \mathcal{J}_h$ could be calculated directly if \mathbf{u}_h were available. Such an approach can still be useful for obtaining an accurate indicator for adaptation.⁵⁷ However, for error estimates and often for adaptive indicators, approximations are made to the above formulations.

One approximation is to avoid the mean-value linearization and estimate the output error with the following second-order method,

$$\mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}_h(\mathbf{u}_h) = \mathcal{R}_h(\mathbf{u}_H, \delta\psi_h) + R^{(2)}(\|\delta\mathbf{u}_h\|, \|\delta\psi_h\|) \quad (24)$$

where $\delta\psi_h \equiv \psi_H - \psi_h$, ψ_h is the solution to Eq. 11, and $\psi_H \in \mathcal{V}_H$ is the solution to,

$$\mathcal{R}'_H[\mathbf{u}_H](\mathbf{v}_H, \psi_H) + \mathcal{J}'_H[\mathbf{u}_H](\mathbf{v}_H) = 0, \quad \forall \mathbf{v}_H \in \mathcal{V}_H. \quad (25)$$

$R^{(2)}(\|\delta\mathbf{u}_h\|, \|\delta\psi_h\|)$ is a remainder that is second order in the primal and adjoint error. This approach has been used in both finite element^{13,61} and finite volume^{62,63} applications. In finite volume applications, the computable correction must also be include in the error estimate. Becker and Rannacher show that the error estimate can be improved to third order by including the residual of the adjoint problem,

$$\mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}_h(\mathbf{u}_h) = \frac{1}{2}\mathcal{R}_h(\mathbf{u}_H, \delta\psi_h) + \frac{1}{2}\mathcal{R}_h^\psi[\mathbf{u}_H](\delta\mathbf{u}_h, \psi_H) + R^{(3)}(\|\delta\mathbf{u}_h\|, \|\delta\psi_h\|) \quad (26)$$

where,

$$\mathcal{R}_h^\psi[\mathbf{u}_H](\delta\mathbf{v}_h, \mathbf{w}_h) \equiv \mathcal{R}'_h[\mathbf{u}_H](\delta\mathbf{v}_h, \mathbf{w}_h) + \mathcal{J}'_h[\mathbf{u}_H](\delta\mathbf{v}_h), \quad (27)$$

and $R^{(3)}(\|\delta\mathbf{u}_h\|, \|\delta\psi_h\|)$ is a remainder that is third order in the primal and adjoint error. As with Eq. 24, this form of error estimate has also been used with both finite element^{5,37,64–66} and finite volume^{36,59,60,67–73} discretizations.

While the error estimates in Eq. 24 and 26 remove the need for the mean-value linearization, \mathbf{u}_h and ψ_h are still required to determine $\delta\mathbf{u}_h$ and $\delta\psi_h$. One approach for approximating \mathbf{u}_h is to reconstruct \mathbf{u}_H on \mathcal{V}_h using a higher-accuracy stencil. In the finite element setting, this could be least squares patch reconstruction.^{5,35,37,61} Such reconstruction makes use of a superconvergence property,^{74,75} which requires a smoothness assumption that loses validity near discontinuities. In addition, without limiting, no guarantees exist that reconstructed solutions will remain physical for nonlinear problems. An alternate approach is therefore to project \mathbf{u}_H into \mathcal{V}_h and to apply several steps of an iterative solution scheme.^{65,66} In either case, the difference between the approximated \mathbf{u}_h and \mathbf{u}_H can be used directly in Eq. 26 to compute the error estimate.

ψ_h can be approximated in several ways. Just as \mathbf{u}_h , it can be reconstructed from ψ_H using a higher-accuracy stencil.^{35,37,60,61} In the standard finite volume setting, this reconstruction is typically performed

with quadratic interpolation on a uniformly-refined mesh. The least squares problem can be tailored to penalize first derivative differences, increasing the robustness to oscillations in the presence of under-resolved or discontinuous features.⁶⁰ In high-order finite element methods, the reconstruction can be simplified by using an interpolation order increment on the same mesh.³⁷ Reconstruction on a fine space obtained by both uniform mesh refinement and interpolation order increase has also been investigated.⁷⁶ A disadvantage of the reconstruction approach is that it does not incorporate physics of the problem, which can be important for convection-dominated equations. A more robust approach is therefore to solve the adjoint problem exactly on the chosen fine space,^{35,56} although this is a costly proposition. When an inexpensive iterative solver is available, a cheaper alternative is, as in the primal problem, to inject ψ_H into the fine space and to apply several steps of the iterative solver, with the linearization based on an approximation of \mathbf{u}_h .^{65,66}

III.5 Error Effectivity

In the limit of a very fine (and consistent) discretization, “ $h \rightarrow 0$ ” and $\mathbf{u}_h \rightarrow \mathbf{u}$, Eq. 18 yields the true output error in the solution: $\mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}(\mathbf{u})$. In practice, however, a finite dimensional \mathcal{V}_h is employed, obtained from \mathcal{V}_H by uniform refinement or interpolation order increase. Hence, the calculated output error is generally not equal to and not a bound for the true error. It is an estimate whose accuracy depends on the enrichment of \mathcal{V}_h relative to \mathcal{V}_H . Indeed, the choice of enrichment governs the behavior of the error effectivity,

$$\eta_H^e \equiv \frac{\mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}_h(\mathbf{u}_h)}{\mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}(\mathbf{u})}. \quad (28)$$

An effectivity close to 1 is desirable. If H denotes mesh size and the output error converges as $\mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}(\mathbf{u}) = C_H H^k$, a choice of $h = H/2$ for the enriched space yields an effectivity of $\eta_H^e = 1 - (1/2)^k$. Thus, even as $H \rightarrow 0$, the effectivity does not approach one. Potentially, this underprediction of the true error could be accounted for if the convergence rate k were known. Another option is to construct the error estimate using p -enrichment. In this case, the effectivity behaves as $\eta_H^e = 1 - C_k H^{\delta k}$ where δk is the increase in convergence rate of \mathbf{u}_h relative to \mathbf{u}_H . Under these assumptions, the effectivity approaches 1 as $H \rightarrow 0$.

III.6 Impact of Shocks and Artificial Stabilization

Shock waves (or other under-resolved phenomena) can present a variety of problems when estimating errors. For example, estimation of \mathbf{u}_h through reconstruction can introduce oscillations that contaminate error estimates. This contamination can be reduced by using monotonic reconstruction procedures.

Another issue is the use of shock-capturing stabilization terms in the discretization that are non-zero even when acting upon the exact solution. In these situations, the semilinear form is inconsistent since $\mathcal{R}_H(\mathbf{u}, \mathbf{v})$ is not necessarily zero for all $\mathbf{v} \in \mathcal{V}$. However, for the method to be convergent, the stabilization terms are assumed to asymptote to zero as $H \rightarrow 0$. In other words, the method has asymptotic primal consistency.

The error due to asymptotically consistent stabilization terms can be estimated by separating the weighted residual statement into consistent and asymptotically consistent parts,

$$\mathcal{R}_H(\mathbf{u}_H, \mathbf{v}_H) + \mathcal{R}_H^e(\mathbf{u}_H, \mathbf{v}_H) = 0, \quad \forall \mathbf{v}_H \in \mathcal{V}_H. \quad (29)$$

where $\mathcal{R}_H(\cdot, \cdot)$ is a consistent semilinear form, and $\mathcal{R}_H^e(\cdot, \cdot)$ is an asymptotically consistent form. Then, using

Eq. 8, the output error due to using asymptotically consistent stabilization is

$$\delta \mathcal{J}_H^\epsilon = \mathcal{R}_H^\epsilon(\mathbf{u}_H, \psi_H), \quad (30)$$

where ψ_H is the solution to Eq. 25, and where the residual perturbation is approximated as infinitesimally small. When performing error estimation, approximations to \mathbf{u}_H and ψ_H are available, and hence $\delta \mathcal{J}_H^\epsilon$ is computable without a residual calculation on a finer space. Dwight takes advantage of this observation to efficiently compute the sensitivity of an output to explicitly-added dissipation for finite volume discretizations of the Euler equations.⁷⁷ Dwight observes that in many test cases, the artificial dissipation accounts for the majority of the output error, so that the calculated sensitivity is a good approximation to the output error.

IV Mesh Adaptation

A typical adaptive solution process is illustrated in Figure 5. The input is an initial coarse mesh along

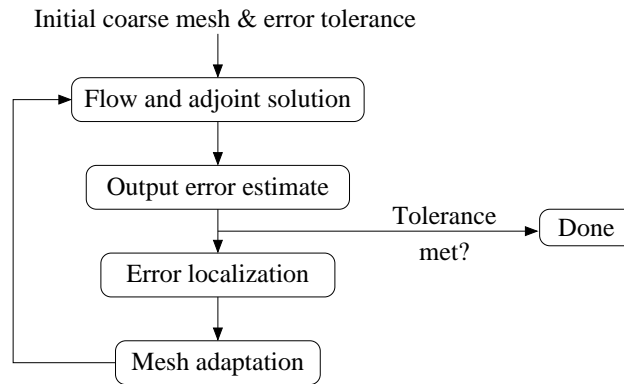


Fig. 5 Adaptive solution process flowchart. The input consists of an initial coarse mesh and a requested error tolerance. Adaptation stops when the error tolerance is met.

with a user-prescribed error tolerance for an output. The iterative process starts by solving the primal and adjoint problems on the initial coarse mesh. Next, the output error is estimated using the adjoint-weighted residual method described in Section III.2. If the global error tolerance criterion is met, the adaptive process terminates. Otherwise, the error estimate is localized to the elements, and the mesh is adapted. The process then repeats until the tolerance is met. This process is only valid for the case of one output, although multiple output functionals can sometimes be treated by defining a weighted-sum output.⁷⁸

In output-based error estimation, the error localization is fairly straightforward. However, numerous strategies exist for translating the error indicator into a modified mesh. In CFD, the most popular adaptation strategy is h -adaptation, in which only the triangulation forming the mesh is varied. In high-order methods, additional strategies include p -adaptation, in which the interpolation order is changed on a fixed triangulation,^{37, 79} and hp -adaptation in which both the order and the triangulation are varied.^{80–89} For CFD applications, in which solutions often possess localized, singular features, h -adaptation is key to an efficient adaptation strategy. In addition, most practical codes operate at one or a limited number of orders, making h -adaptation the only practical approach. With the growing popularity of high-order methods, however, hp -adaptation will be an important strategy for increased efficiency in the future.

This section reviews general aspects of h -adaptation for CFD. Many of the aspects, especially pertaining

to adaptation mechanics, incorporation of anisotropy, and general optimization strategy, are also relevant to non-output based adaptation. Additional information on these topics can be found in existing reviews.^{90–95} The discussion below will focus on aspects of mesh adaptation specifically relevant to output-based error estimation.

IV.1 Error Localization

The output error estimates in Eqs. 18, 24, and 26 consist of a residual evaluation on the refined space \mathcal{V}_h . In a finite element method, this residual evaluation is a sum over all elements in the fine space. Since the coarse and fine spaces are assumed nested, Eq. 24 (with an analogous expression for Eq. 18) can be written as,

$$\mathcal{J}_H(\mathbf{u}_H) - \mathcal{J}_h(\mathbf{u}_h) \approx \sum_{\kappa_H \in T_H} \sum_{\kappa_h \in \kappa_H} \mathcal{R}_h(\mathbf{u}_H, \delta\psi_h|_{\kappa_h}), \quad (31)$$

where T_H is the coarse triangulation, κ_H/κ_h is an element of the coarse/fine triangulation, and $|_{\kappa_h}$ refers to restriction to element κ_h . Note that the coarse/fine spaces can consist of the same triangulation, for example when only the interpolation order is increased, in which case $\kappa_H = \kappa_h$. Eq. 31 expresses the output error in terms of contributions from each coarse element. A common approach for obtaining an *error indicator* is to take the absolute value of the elemental contribution,^{13, 16, 35, 56, 60}

$$\epsilon_{\kappa_H} = \left| \sum_{\kappa_h \in \kappa_H} \mathcal{R}_h(\mathbf{u}_H, \delta\psi_h|_{\kappa_h}) \right|. \quad (32)$$

When the adjoint residual contribution is used as in Eq. 26, an adjoint error indicator can be defined as

$$\epsilon_{\kappa_H}^\psi = \left| \sum_{\kappa_h \in \kappa_H} \mathcal{R}_h^\psi[\mathbf{u}_H](\delta\mathbf{u}_h|_{\kappa_h}, \psi_H) \right|. \quad (33)$$

This indicator targets areas of nonzero adjoint residual, weighted by a primal interpolation error estimate. Numerical experiments have shown that the two error indicators, ϵ_{κ_H} and $\epsilon_{\kappa_H}^\psi$, yield similar mesh distributions when used to drive adaptation.

The above error localization is applicable to finite volume and discontinuous Galerkin discretizations, since weighted residuals vanish locally on each element for these discretizations. Thus, no systematic inter-element error cancellation is expected in the output error estimates and the absolute value signs in Eqs. 32 and 33 are justified. However, local residuals do not necessarily vanish for continuous finite element discretizations. Consider for example a continuous finite element discretization of Poisson’s equation, in which the elemental contributions to the residual contain terms of the form $\int_{\kappa_h} \nabla \mathbf{u}_H \cdot \nabla \psi_h$. Simply placing absolute value signs around these terms to obtain the elemental error indicator would lead to a systematic over-estimate of the output error via a sum of the indicators.⁹⁶ This over-estimate is due to a poor bookkeeping choice for the error and can be fixed by integrating the residual terms by parts on each element. The result is a set of element-interior terms containing the strong form of the residual, and a set of face flux jump terms, which are present because the gradient of \mathbf{u}_H is not continuous. Both of these terms are expected to go to zero with mesh refinement, and the flux jump terms will dominate for low orders.⁹⁷ The face flux residuals can be pushed back onto the elements by assigning half of the flux residual to each of the two elements adjacent to the face.⁶¹ For convection equations, the continuity of \mathbf{u}_H eliminates the need for interior flux residuals,

although inflow flux residuals are still required and the stabilization terms must be treated appropriately.³²

For systems of equations, indicators are typically computed separately for each equation and summed together. Due to the absolute values, the sum of the error indicators, $\epsilon = \sum_{\kappa_H} \epsilon_{\kappa_H}$, is greater or equal to the original output error estimate. However, it is not a bound on the actual error, or even on $J_H(\mathbf{u}_H) - J_h(\mathbf{u}_h)$, because of the approximations made in the derivation. In practice, the validity of the approximations improves with refinement, and the above estimate becomes an accurate measure of the true error.

IV.2 *h*-Adaptation Mechanics

Many approaches to adapting a mesh rely upon the application of local operators through which the mesh is modified incrementally. A simple example of a local operator is element sub-division in a setting that supports non-conforming, or hanging, nodes.^{61,85,90,98,99} For triangular and tetrahedral meshes, local mesh modification operators consist of node insertion, face/edge swapping, edge collapsing, and node movement. These operators have been studied extensively by various authors^{68,69,92,100–105} in different contexts. The primary advantage of local operators is their robustness: the entire mesh is not regenerated all at once, but rather each operator affects only a prescribed number of nodes, edges, or elements.

Another approach to adapting a mesh is global re-meshing, in which a new mesh is generated for the entire computational domain. The original, or background, mesh is used to store desired mesh characteristics during re-generation. For applications to adaptation, the desired mesh characteristics are often described using a Riemannian metric, the idea being that in an optimal mesh, all edge lengths will have unit measure under the metric.^{102,104} In a Cartesian coordinate system, an infinitesimal segment $\delta\mathbf{x}$ has length $\delta\Gamma$ under a Riemannian metric \mathbf{M} ,

$$\delta\Gamma^2 = \delta\mathbf{x}^T \mathbf{M} \delta\mathbf{x} = \delta x_i M_{ij} \delta x_j, \quad (34)$$

where δx_i are the components of $\delta\mathbf{x} \in \mathbb{R}^d$ and M_{ij} are the components of the symmetric, positive definite metric, $\mathbf{M} \in \mathbb{R}^{d \times d}$.

The metric \mathbf{M} contains information on the desired mesh edge lengths in physical space. As \mathbf{M} is symmetric and positive definite, the unit measure requirement,

$$\mathbf{x}^T \mathbf{M} \mathbf{x} = 1,$$

describes an ellipsoid in physical space that maps to a sphere under the action of the metric. The eigenvectors of \mathbf{M} form the orthogonal axes of the ellipsoid – i.e. the principal directions. The corresponding eigenvalues, λ_i , are related to the lengths of the axes, h_i , via

$$\lambda_i = \frac{1}{h_i^2} \quad \Rightarrow \quad \frac{h_i}{h_j} = \left(\frac{\lambda_j}{\lambda_i} \right)^{1/2}$$

Physically, the h_i are the principal stretching magnitudes. A diagram of a possible ellipse resulting from the unit-measure requirement in two dimensions is given in Figure 6. Thus, the ratio of eigenvalues of \mathbf{M} can be used to define a desired level of anisotropy.

A successful approach for generating simplex meshes based on a Riemannian metric is mapped Delaunay triangulation, in which a Delaunay mesh generation algorithm⁹⁵ is applied in the mapped space, allowing for the creation of stretched and variable-size triangles or tetrahedra.¹⁰⁶ This method is implemented in the Bi-

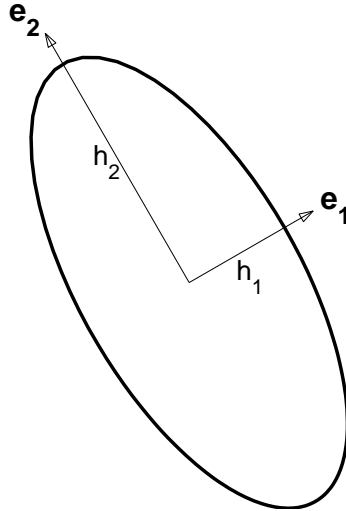


Fig. 6 Ellipse representing requested mesh sizes implied by equal measure under a Riemannian metric M . Also shown are the principal directions, e_i , and the associated principal stretching magnitudes, h_i .

dimensional Anisotropic Mesh Generator (BAMG),^{107, 108} which has been used in various finite volume^{36, 109} and discontinuous Galerkin^{65, 66, 110} applications requiring anisotropic meshes. Examples of output-adapted meshes obtained using BAMG are shown in Figures 10, 20, and 22.

IV.3 Overview of Adaptation Strategies

In h -adaptation, the determination of which elements to refine or coarsen has important implications for practical simulations: too little refinement at each adaptation iteration may result in an unnecessary number of iterations; too much refinement may ask for an expensive solve on an overly-refined mesh. Aftosmis and Berger discuss adaptation strategies in terms of error distribution histograms,¹¹¹ in which elements are binned according to the error indicator (Eq. 32 for output-based adaptation). The assumption made in virtually all adaptation strategies is that in an ideal mesh the user-prescribed error tolerance is satisfied and the error is equidistributed among the elements.⁹² This situation corresponds to a “delta” histogram, in which all elements lie in the same bin. In contrast, the initial coarse mesh will generally have some distribution of error indicators, as illustrated in Figure 7. The goal of an adaptation strategy is then to drive the histogram towards the ideal delta distribution using a prescribed adaptation strategy.

Nemec *et al*⁶³ discuss two adaptation strategies based on either a constant or a decreasing refinement threshold. In a constant threshold strategy, depicted in Figure 7, all elements with error above a certain fixed value are refined, in the same manner. This strategy is simple but potentially expensive: initial refinement targets virtually all of the elements and leads to a rapid growth in the mesh size in the first few iterations, while elements with the highest error (in the right tail of the histogram) are likely to be among the last elements to have their error reduced to the target level. In contrast, with a decreasing threshold, shown in Figure 8, elements with the highest error are targeted for refinement first, so that the mesh size grows more slowly and multiple expensive solves on the finest meshes are avoided. Note, general adaptive mechanics that

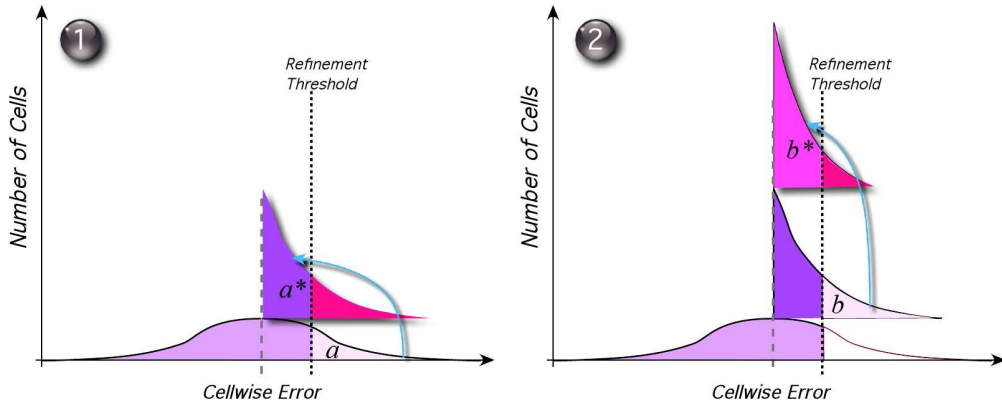


Fig. 7 Sample error indicator histogram and a constant-threshold refinement strategy. Reproduced with permission from.⁶³

do not employ the same level of refinement for each element are not limited to these strategies. For example, when using global re-meshing, all elements could be refined or coarsened based on their error indicators.

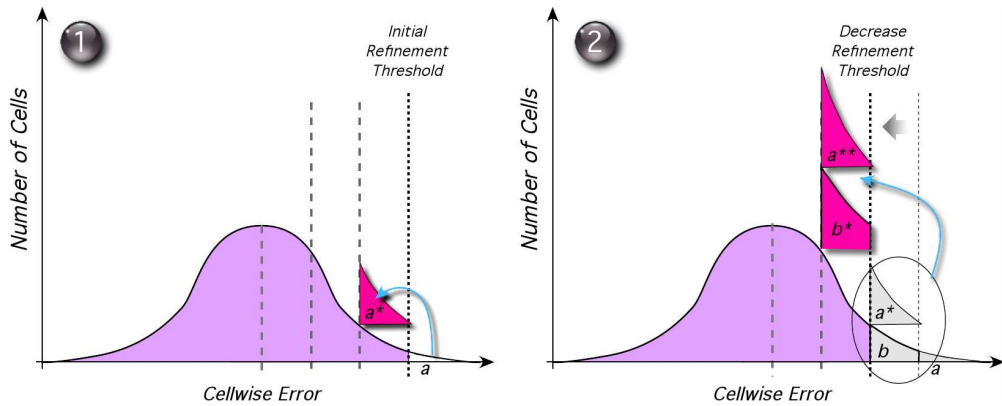


Fig. 8 Adaptation strategy using a decreasing threshold. Reproduced with permission from.⁶³

Most adaptation optimization methods follow some variation of a decreasing threshold strategy. For example, a fixed-fraction approach prescribes a fraction of elements with the highest error indicator to be refined at each adaptation iteration, such that the decreasing threshold is a function of the shape of the error histogram. Then, the elements targeted for adaptation are typically refined in a locally uniform manner, e.g. by splitting all edges in half. This simple approach has been applied to output-based adaptation in several studies^{29, 35, 56, 57, 62, 89} The fixed-fraction parameter is often chosen heuristically in a trade-off between an excessive number of iterations and a risk of over-refinement. Nevertheless, the method works quite well for practical problems.

IV.4 Incorporating *a priori* Analysis and Anisotropy

The fixed-fraction adaptive strategy with locally uniform refinement does not account for the rate at which the error decreases with mesh refinement in a given adaptive iteration. This disregard for the error convergence rate could lead to over-refinement of the mesh or to an excessive number of adaptive iterations to achieve the desired target error. Adaptation strategies have been developed that attempt to meet the

global tolerance while equidistributing the error among elements through the incorporation of *a priori* error analysis. In the context of isotropic, output-based adaptation, Venditti and Darmofal⁶⁰ developed such a method based on the previous work of Zienkiewicz and Zhu.¹¹² In this method a permissible element error $e_\kappa = e_0/N$ is defined at each adaptation iteration, where e_0 is the user tolerance, and N is the current number of elements. Coupled with an *a priori* error estimate that the error converges as $\mathcal{O}(h^r)$, where r is the *a priori* estimated convergence rate, element size requests can be made that equidistribute the error.

An important ingredient in h -adaptation for aerodynamic computations is the ability to generate stretched elements in areas such as boundary layers, wakes, and shocks, where the solution exhibits anisotropy, which refers to variations of disparate magnitudes in different directions. While stretched elements can be created to a limited extent with hanging-node refinement, by optimally choosing the refinement direction,^{113,114} unstructured triangular and tetrahedral grids offer the most flexibility in anisotropic refinement. The first output-based adaptive method to incorporate anisotropy was proposed by Venditti and Darmofal^{36,115} and applied to a nominally second-order accurate finite volume algorithm. Their approach was to combine the isotropic, output-based approach using *a priori* estimates with existing Hessian-based methods for anisotropic adaptation.

For spatially second-order methods, the dominant method for detecting anisotropy involves estimating the Hessian matrix, \mathbf{H} , of a scalar solution, u .^{102,104,106,116} The components of \mathbf{H} are given by

$$H_{ij} = \frac{\partial^2 u}{\partial x_i \partial x_j}, \quad i, j \in [1, \dots, d], \quad d = \text{dimension}.$$

The second derivatives can be estimated by, for example, a quadratic reconstruction of a linear solution. For the Euler or Navier-Stokes equations, the Hessian of the Mach number has been found to perform sufficiently well as the scalar u .

The metric is obtained from the Hessian by requiring that the interpolation error estimate of the scalar quantity u be the same in any chosen spatial direction. For linear interpolation of u along the segment $\delta\mathbf{x}$, the maximum interpolation error can be bounded by the second derivative of u along $\delta\mathbf{x}$. The Hessian matrix stores precisely this information, so that requiring the interpolation error bound to be approximately constant independent of the direction of $\delta\mathbf{x}$, leads to the metric choice

$$\mathbf{M} = C|\mathbf{H}|, \tag{35}$$

where C is a constant independent of direction, and $|\mathbf{H}|$ is the positive, semi-definite form of the Hessian: $|\mathbf{H}| = \mathbf{V}|\mathbf{\Lambda}|\mathbf{V}^{-1}$ for $\mathbf{H} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$. Two intervals, $\delta\mathbf{x}_1$ and $\delta\mathbf{x}_2$, having the same measure under this \mathbf{M} will have the same estimated interpolation error bounds.

To fully define the metric, the absolute mesh size, i.e. the constant C in Eq. 35, has to be fixed. While in pure Hessian-based adaptation a global value for C is used, the output-based method of Venditti and Darmofal sets C locally according to the output error indicator.³⁶ As a result, the smallest mesh length is controlled by the output error indicator while the anisotropy is controlled by the solution Hessian.

The definition of a metric tensor becomes difficult for high-order methods because the standard Hessian matrix approach assumes linear interpolation of the scalar quantity. For general order p interpolation, the interpolation error is governed by the order $p+1$ derivatives. One possible extension of the Hessian approach is based on constructing a metric around the direction of maximum $p+1$ st derivative.^{5,64} In two dimensions,

the anisotropy is set equal to the $p + 1$ root of the ratio between this maximum derivative and the derivative normal to this direction. This approach has the disadvantage of requiring a search over all directions to determine the maximum $p + 1$ derivative. In three dimensions, the approach would require two searches and seems impractical. More recently, Pagnutti and Ollivier-Gooch developed a method to calculate a metric for general p utilizing a Fourier series representation of $p + 1$ order terms.¹¹⁷ This approach appears to extend to three dimensions quite readily, though to date has not been implemented.

An additional problem with higher-order discretization is the need for curved mesh elements and high-fidelity geometry representations. Recent work by Oliver⁶⁶ explores a novel implementation approach for high-order metric-driven meshing, in which the adaptation is performed on a mapped linear-triangle mesh. An elasticity analogy is then used to transform the linear mesh to a curved, boundary-conforming mesh around the true geometry. The robustness of this approach relies on the success of the linear meshing, which may not be guaranteed for highly-anisotropic boundary-layer meshes. Recently, Persson and Peraire developed an approach to curved meshes based on a nonlinear elasticity analogy using Lagrangian solid mechanics.¹¹⁸ This approach appears quite robust, though involves solution of a nonlinear set of equations to perform the mesh motion.

The metric tensor may also be used to guide an adaptation procedure based on local operators. In the context of pure Hessian-based adaptation, Diaz *et al*¹⁰² present a two-dimensional algorithm that uses the metric-based edge length to decide which operation to apply. Specifically, edge splitting, edge collapsing, edge swapping, and node movement are applied to make all edges approximately the same length when measured using the metric tensor. Habashi *et al*^{104, 119, 120} and Xia *et al*¹⁰⁵ present similar algorithms, with slight modifications in Hessian definition and in the local operators. Park^{68, 69} extends these local mesh modification operators to output-based mesh adaptation, in both two and three dimensions.

IV.5 Direct Optimization Adaptation

The output-based adaptive approaches described in Section IV.4 rely upon *a priori* analysis to estimate desired grid characteristics. Further, the interpolation error assumptions are made without regard to the output of interest by using a single scalar, such as the Mach number, to control the anisotropy for a system of equations. While the Mach number choice has worked well so far, it is arbitrary. Diaz *et al*¹⁰² propose choosing an intersection of metrics derived from all variables in the system, although this choice relies on the variables used (e.g. conservative versus primitive), and using more variables can make the resulting intersected metric too restrictive. More generally, for output-based adaptation, the assumption that the directional interpolation error must be equidistributed for one or more scalar variables at each point in the domain may not be valid. Of interest are only the interpolation errors that create residuals that affect the output. This observation has motivated research into adaptation algorithms that more directly target the error indicator.

Formaggia *et al*^{121–123} combine Hessian-based interpolation error estimates with output-based *a posteriori* error analysis to arrive at an output-based error indicator that explicitly includes the anisotropy of each element. However, for the purpose of mesh adaptation, a metric is still defined using the resulting element modification requests. Schneider and Jimack¹²⁴ calculate the sensitivities of the output error estimate with respect to node positions and formulate an optimization problem to reduce the output error estimate by redistributing the nodes. The sensitivities with respect to node positions are calculated efficiently by solving an additional adjoint problem. This approach directly targets the output error estimate and automatically

leads to anisotropic meshes where appropriate. Schneider and Jimack then combine this node repositioning with isotropic local mesh refinement sequentially in a hybrid optimization/adaptation algorithm.

Park⁷³ introduces an algorithm that directly targets the output error through local mesh operators of element swapping, node movement, element collapse, and element splitting. Using the output error indicator to rank elements and nodes, these operations are performed in sequence and automatically result in mesh anisotropy. The details of the adaptation are also given in an earlier work, in the context of interpolation error.¹²⁵ While the grids produced by this technique lack the regularity of those produced using metric-based adaptation, their accuracy is comparable.

IV.6 Cut-Cell Methods

A successful adaptation algorithm relies on automation and robustness of the mesh generation or modification. Standard boundary-conforming meshers must ensure both geometry fidelity and mesh validity, a task that becomes difficult, for example, for anisotropic meshes around curved geometries. An alternate approach to mesh generation is the class of cut-cell methods, in which the computational domain is formed by intersecting the geometry of interest with a volume-filling background mesh. Without the boundary-conforming constraint, generation or adaptation of the volume-filling background mesh is straightforward. However, the burden of robustness is transferred to the computational geometry problem of intersecting the background mesh with the geometry.

The most common cut-cell technique is the *Cartesian method*, a name that refers to the rectangular or hexahedral cells on a regular lattice used for the background mesh (see Figure 9). The Cartesian method was

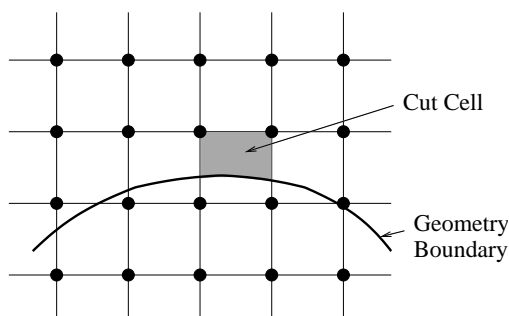


Fig. 9 Sample Cartesian mesh in two dimensions. The square lattice mesh does not conform to the geometry. Cut cells are portions of intersected elements that lie inside the computational domain (above the geometry boundary in this case).

pioneered in the early days of CFD,^{126–130} has been used in industry,^{131–133} and is the subject of ongoing research.⁶² Recently, the cut-cell technique has also been applied to simplex background meshes, which provide greater flexibility in directional resolution.^{64, 125, 134}

The advantage of cut-cell methods for mesh adaptation is the automation that results from removing the boundary-conforming constraint. In a Cartesian method, hanging-node refinement is the single practical option for adaptation, and has been implemented efficiently.¹³⁵ In simplex cut-cell methods, adaptation can also be performed through global re-meshing⁶⁴ or through local operators.¹²⁵ Cut-cell methods have been successfully applied to output-based adaptive simulations of Euler flows and to moderate Reynolds number viscous flows. For boundary-layer viscous flows, the Cartesian method quickly becomes inefficient at achieving the desired anisotropic resolution. While simplex cut-cell methods alleviate this problem, current

research has only been with linear background meshes, which eventually become inefficient for high Reynolds number flows around curved geometries.

V Implementations and Results

V.1 Finite Volume Methods

V.1.1 High-lift RANS (Venditti and Darmofal)

Venditti and Darmofal apply output-based error estimation and mesh adaptation to a range of inviscid and viscous aerodynamic cases in two dimensions.^{36,60} They use a node-based, unstructured finite volume solver and solve the linear adjoint equations by time marching, similarly to the forward problem. They adapt on the remaining error in Eq. 23 and use an average of the primal and adjoint residual localizations for the adaptive indicator. For anisotropic meshing, they use the Hessian of the Mach number to define a metric, and they remesh the domain using BAMG.

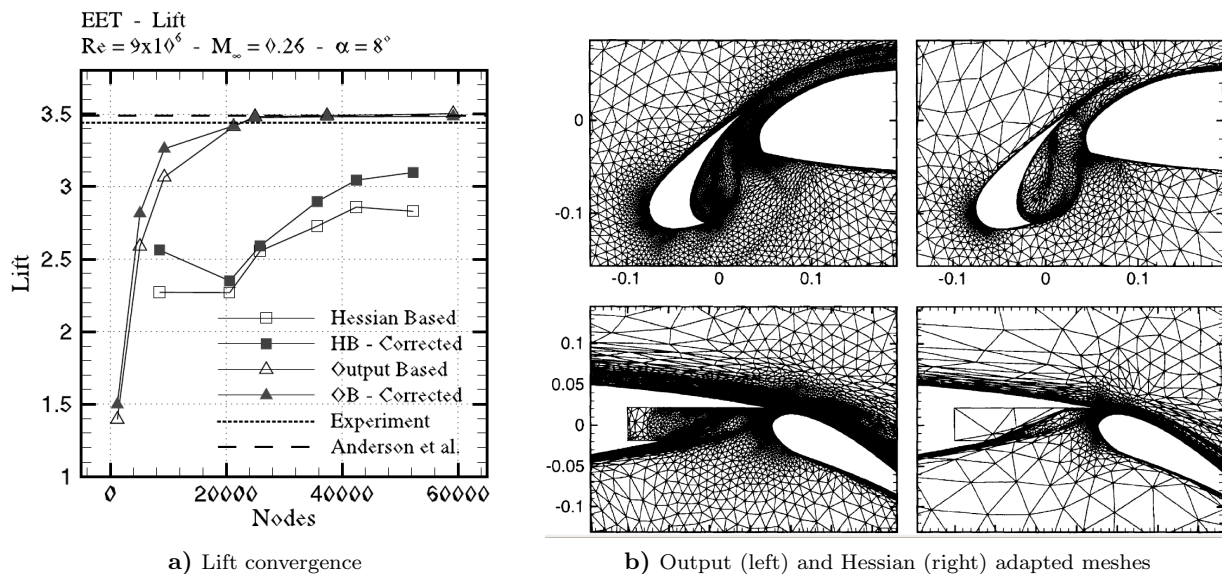


Fig. 10 Advanced energy-efficient-transport (EET) airfoil, $M_\infty = 0.26$, $\alpha = 8^\circ$, $Re = 9 \times 10^6$. Comparison of lift convergence for output-based and Hessian-based adaptation, and near-field views of the final adapted meshes. Reproduced with permission from.³⁶

A representative example from the work of Venditti and Darmofal is that of adaptive simulation for turbulent flow over an advanced energy-efficient-transport (EET) airfoil. In this example, a sequence of lift-adapted meshes is compared to meshes adapted using only the Hessian of the Mach number with no output-error information. The resulting convergence of the lift output is shown in Figure 10a. The corrected output in both runs was calculated using the computable correction in Eq. 23. The improved convergence of the runs adapted on the output error compared to those adapted on the Hessian is clear. The finest adapted meshes from both runs are shown in Figure 10b. Note the increased resolution of the output-adapted mesh near the main-element leading edge and over the upper surface of the main element. Also note that the Hessian-based mesh predicts the lower slat wake in a different location and does not resolve the flow in the cavity region of the main element.

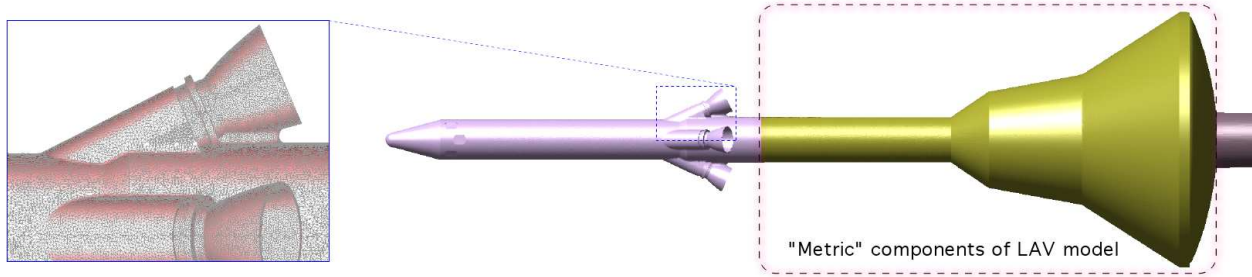


Fig. 11 Definition of metric components for the LAV model. Reproduced with permission from.⁶³

V.1.2 Launch Abort Vehicle (Nemec *et al*)

Nemec *et al* apply an output-based adaptive framework to a Cartesian, cut-cell, finite-volume code.^{62,63} They solve the discrete adjoint equations by marching to steady-state with the same Runge-Kutta scheme and multigrid solver used for the flow solution. The adjoint solve requires transpose linearizations of the residual evaluation applied in reverse order, and this process is simplified by freezing the limiter used for the spatial reconstruction. Details on the adjoint implementation are given in.¹³⁶

For the fine space \mathcal{V}_h in output error estimation, Nemec *et al* use an embedded grid obtained by uniformly refining each hexahedral cell in the Cartesian grid. They then obtain an error indicator by weighting the residual of the coarse, linearly-reconstructed solution on the embedded grid with an adjoint error that is the difference between piecewise linear and piecewise constant reconstructions of the coarse adjoint solution. Results in⁶² compare the performance of this error estimate versus one that employs a more rigorous quadratic reconstruction of the adjoint and show reduced accuracy of the constant/linear output error estimate but simpler implementation.

Nemec *et al* then define a refinement threshold error level for adaptation and at each iteration refine cells with error above this threshold, using the decreasing threshold strategy described in Section IV.3. The Cartesian hanging-node adaptation makes use of the robust cut-cell mesh generation capability in the code,¹³⁵ allowing for adaptive results with complex geometries. A representative example is that of aerodynamic analysis of a Launch Abort Vehicle (LAV), illustrated in Figure 11. The output of interest for this case consists of a linear combination of the normal (N) and axial (A) force coefficients,

$$J = C_N + 0.2C_A,$$

where the weight on the linear combination was determined empirically as one that yielded adequate results for both forces and moments. Note, the forces and moments are evaluated on the “metric” portion of the geometry, as specified in Figure 11.

The robustness and automation of the mesh generation process allowed Nemec *et al* to consider a range of Mach numbers and angles of attack. A representative case, at $M_\infty = 1.1$, $\alpha = -25^\circ$, is shown in Figure 12. Also shown in the figure is a contour plot of the adaptive indicator, where regions shown in gray-scale fall below the refinement threshold. Areas marked for refinement include the edges of the heat shield and the vicinity of the abort motors. Note that only moderate refinement is requested at the shocks, which often attract excessive refinement with heuristic feature-based indicators.

An example of a final mesh generated by the adaptive process is shown in Figure 13. As expected from

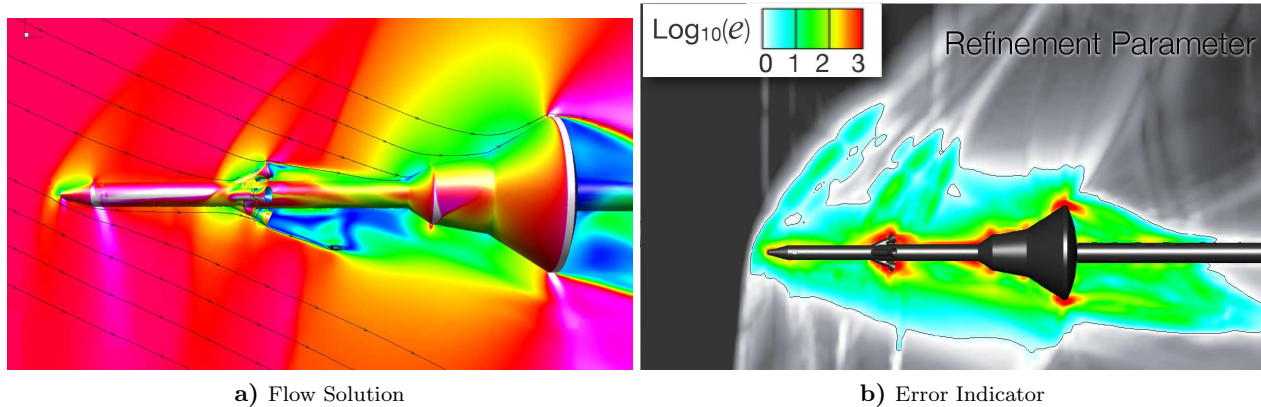


Fig. 12 Launch Abort Vehicle (LAV) Mach number contours, $M_\infty = 1.1$, $\alpha = -25^\circ$, and the localized error indicator. Reproduced with permission from.⁶³

the error indicator, the refinement concentrates on the edges of the heat shield and on the abort motors. The convergence of the output for this case is shown in Figure 14 on the left. Included on the same plot is the corrected output, calculated as described in Eq. 23. The right plot in Figure 14 shows the convergence of the output error estimate. The jump in the error estimate on the final mesh is due to an incompletely-converged adjoint solution caused by the appearance of small-scale unsteadiness in the primal problem. Nevertheless, unsteady simulations on the final mesh show that the time-averaged coefficients are in good agreement with the steady results for this case.⁶³

V.1.3 Sonic Boom (Park)

Park presents output-based, adaptive results for an unstructured, cut-cell finite volume method.⁷³ The method is node-based, and the cut-cell approach allows for automated mesh generation. Park solves the linear adjoint equation using a dual-consistent time-marching method^{48,49} and adapts on the remaining error (Eq. 23) using quadratic interpolation to obtain the fine-space solutions. He adapts on an indicator computed from the average of the localized primal and adjoint residuals. The tetrahedral grid adaptation is based on anisotropic local mesh modification operators combined with mesh movement, as described in Section IV.5.

An example case from Park's work is shown in Figure 15. The case consists of a delta wing body used in existing wind tunnel experiments,¹³⁷ at $M_\infty = 1.68$, $\alpha = 0^\circ$. Of interest is the pressure signature 3.6 body lengths away from the geometry. Specifically the output consists of an integral of the square of the pressure deviation from free-stream, taken over the measurement region. The triangular surface mesh in Figure 15a is the geometry representation that is used in the cut-cell method. The initial background mesh from which the geometry was cut contained 2,800 control volumes, while the final adapted background mesh in Figure 15b contained 4.9 million control volumes. Note the alignment of the cells in the final mesh with the propagated signal.

The adaptation history of the output is shown in Figure 16a, where the error bars denote the remaining error estimate. Note that the error is severely under-predicted on the very coarse initial meshes. As the shock is resolved, the error estimate becomes more accurate and begins to decrease in the latter stages of adaptation. The pressure signature 3.6 body lengths away is shown in Figure 16b. The dotted lines indicate

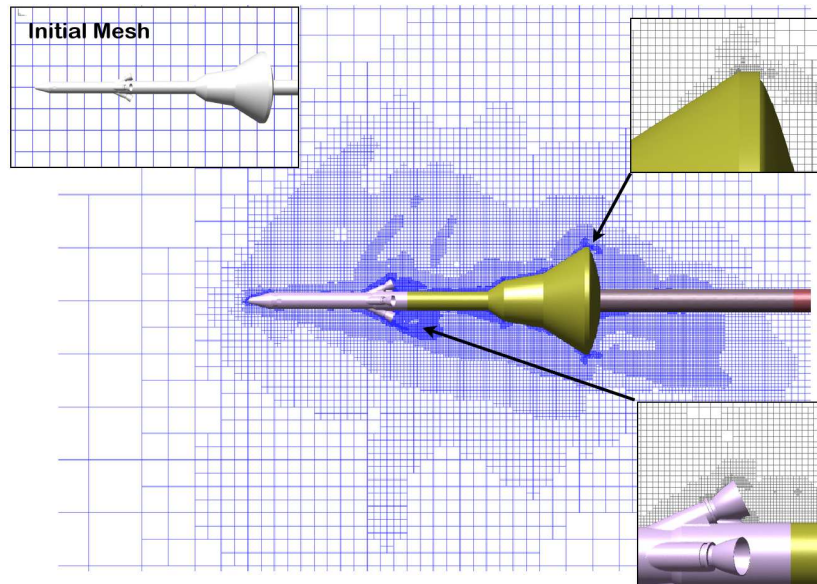


Fig. 13 Initial and adapted meshes for the LAV, at $M_\infty = 1.1, \alpha = -25^\circ$. The initial mesh contains 3,700 cells, while the final mesh after eight adaptation iteration contains almost two million cells. Reproduced with permission from.⁶³

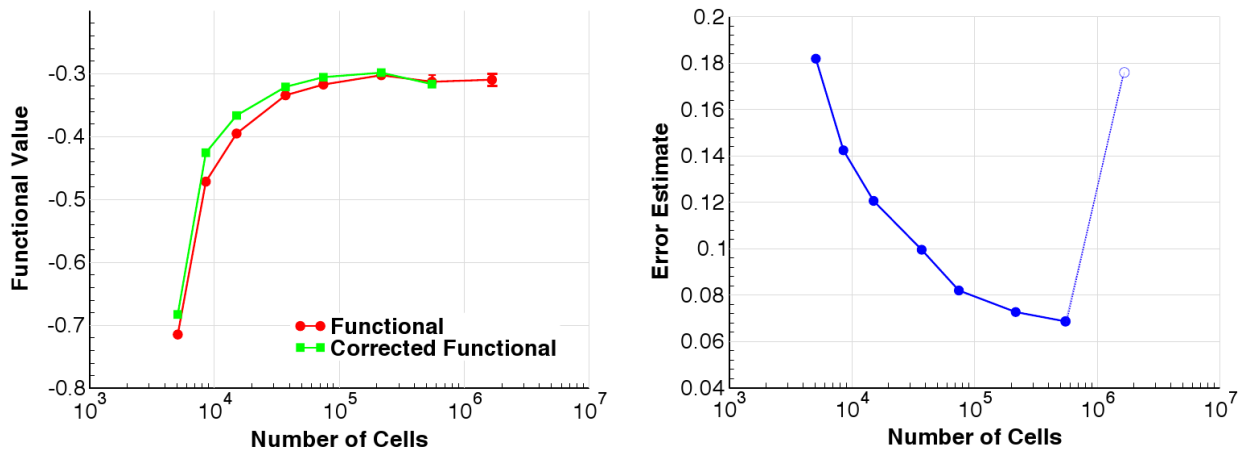


Fig. 14 Output functional convergence for the LAV, at $M_\infty = 1.1, \alpha = -25^\circ$. Reproduced with permission from.⁶³

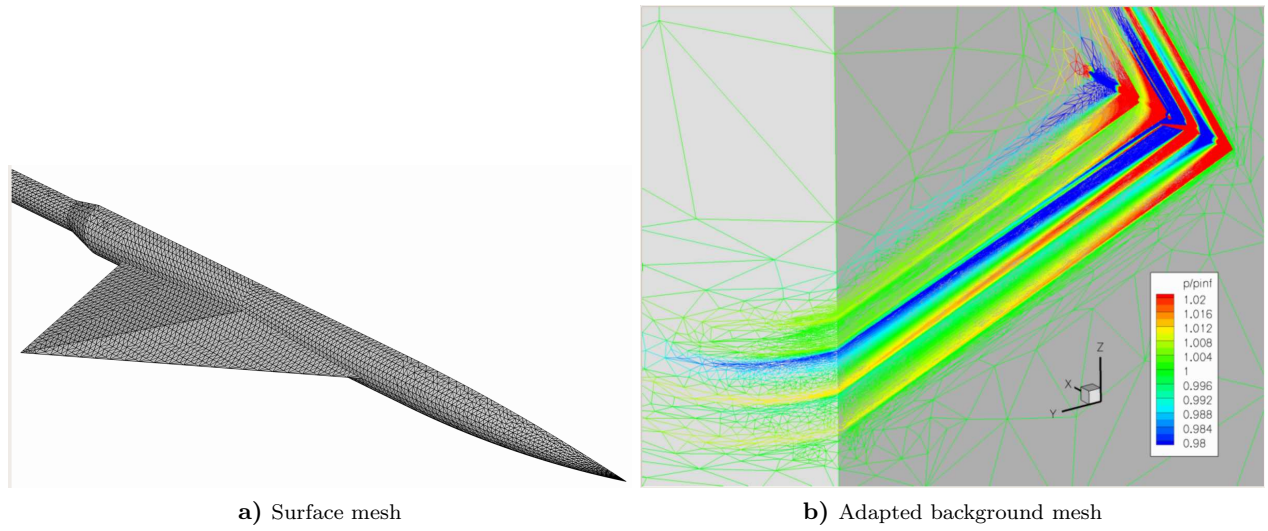


Fig. 15 Delta wing-body sonic boom prediction, $M_\infty = 1.68, \alpha = 0^\circ$. Surface geometry mesh and an output-adapted mesh colored by pressure. Reproduced with permission from.⁷³

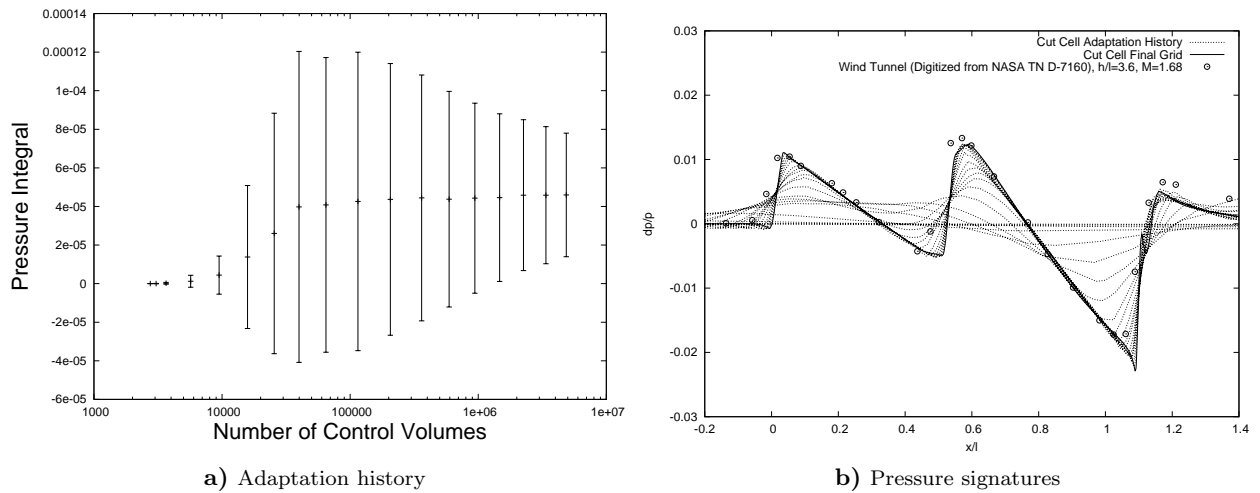


Fig. 16 Delta wing-body sonic boom prediction, $M_\infty = 1.68, \alpha = 0^\circ$. Pressure integral output history with error estimates and pressure signature convergence. Reproduced with permission from.⁷³

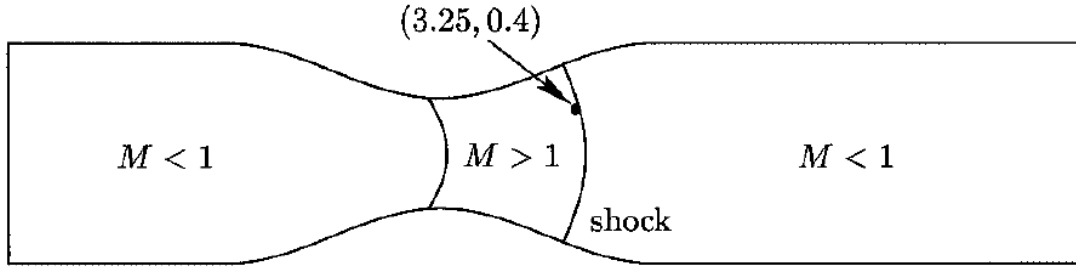


Fig. 17 Converging-diverging nozzle geometry. The output of interest is the density immediately before the shock. Reproduced with permission from.⁵⁶

signatures at intermediate grids during adaptation. The computed signature on the final adapted meshes agrees well with experimental data.

V.2 Discontinuous Galerkin Methods

V.2.1 Point Error in Transonic Flow (Hartmann and Houston)

Hartmann and Houston compare two types of error indicators for $p = 1$ discontinuous Galerkin solutions of the compressible Euler equations.⁵⁶ The first, type I, indicator is derived from the adjoint-weighted residual method described in this paper. The second, type II, indicator is an unweighted residual indicator, in which *a priori* bounds are placed on the adjoint values, eliminating the need for the adjoint solution. Adaptive results show the superior performance of using the adjoint-weighted, type I, indicator.

An example case considered by Hartmann and Houston is the converging-diverging nozzle problem shown in Figure 17. Hartmann and Houston use a damped Newton method to obtain $p = 1$ primal and adjoint solutions on a geometry represented by quadratic ($q = 2$) elements. As the fine space for error estimation, they use order $p = 2$ interpolation on the same mesh. They adapt the quadrilateral meshes based on the indicators with fixed-fraction, hanging-node refinement and coarsening. Figure 18 shows adapted meshes for the two types of error indicators. The type II error indicator, which does not use the adjoint solution, refines mainly the region near the shock. On the other hand, the type I error indicator leads to refinement along the characteristics upstream of the point of interest. This targeted refinement yields a lower output error with fewer degrees of freedom compared to the type II refinement.

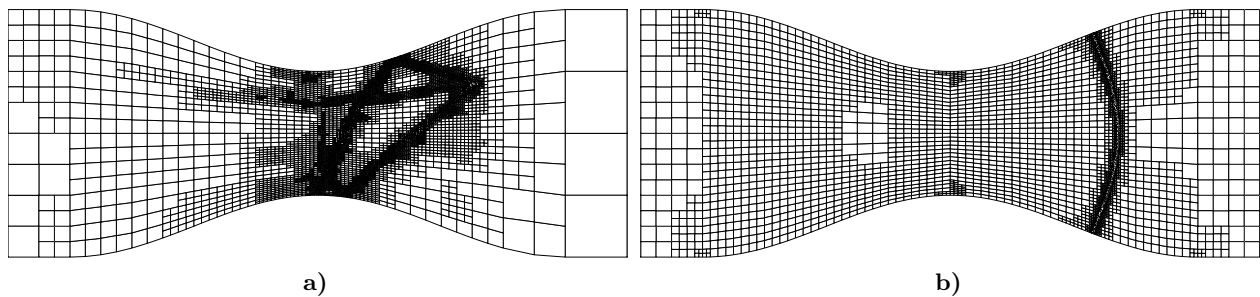


Fig. 18 Converging-diverging nozzle: (a) mesh adapted using the type I error indicator, based on the adjoint-weighted residual, with 172,880 degrees of freedom and an output error of 6.947×10^{-6} . (b) mesh adapted using the type II error indicator, with 341,648 degrees of freedom and an output error of 2.842×10^{-5} . Reproduced with permission from.⁵⁶

V.2.2 Hypersonic Heat Transfer (Barter and Darmofal)

Barter and Darmofal apply output-based error estimation and mesh adaptation to discontinuous Galerkin solutions containing shocks, using a discretization stabilized with smooth artificial-viscosity.^{65,138} In this discretization, the artificial viscosity on each element is interpolated with the same polynomial basis as the state, and the coefficients for these polynomials are agglomerated into the unknown state vector. The necessary additional equation is obtained by discretizing a diffusion partial differential equation for the artificial viscosity.

Barter and Darmofal use several sweeps of a block-Jacobi smoother in order to approximate ψ_h on a space enriched to order $p + 1$. As in related work, the adaptive indicator is formed by averaging localized primal and adjoint output error estimates.

The artificial viscosity stabilization enables Barter and Darmofal to adaptively solve transonic, supersonic, and hypersonic flow problems. A representative example is that of hypersonic flow at $M_\infty = 17.605$, $Re = 376,930$ over a cylinder geometry, also studied in previous work.¹³⁹ The problem setup and the initial mesh for adaptation are shown in Figure 19. The output used for adaptation is the integrated heat flux to the cylinder, non-dimensionalized to form the average Stanton number on the surface,

$$Q_{wall} = \int q_{wall} ds; \quad \overline{C_h} = \frac{\frac{1}{\pi R} Q_{wall}}{c_p \rho_\infty V_\infty (T_{t,\infty} - T_{wall})}.$$

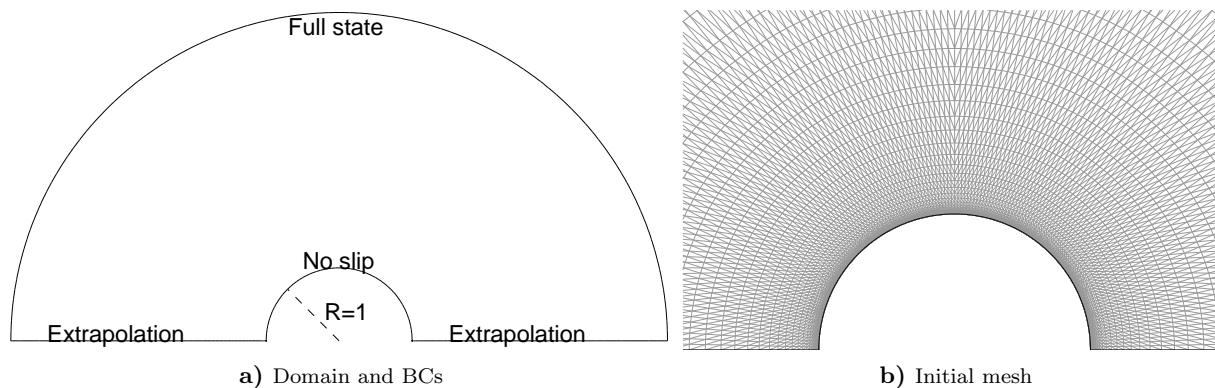


Fig. 19 Hypersonic flow over a 2D half-cylinder at $M_\infty = 17.605$, $Re = 376,930$: domain with boundary conditions and initial mesh for adaptation. Note, flow is from top to bottom. Reproduced with permission from.⁶⁵

BAMG is used as the mesh generator for the metric-driven adaptation, where an anisotropy metric is defined using the Mach number interpolation error. To create meshes with curved ($q = 3$) anisotropic elements for the thin boundary layer around the cylinder, Barter and Darmofal perform adaptation in a mapped, rectangular space, transforming the requested metric appropriately as described in.⁶⁶ Figure 20 shows the final adapted mesh for $p = 2$ and the output convergence history compared to uniform refinement. The final adapted mesh exhibits refinement in the bow shock, but only to the extent that it impacts the heat flux on the cylinder. Refinement is also seen in the stagnation streamline, an area to which the heat flux output is highly sensitive. The convergence history shows that the adapted run converges to an output

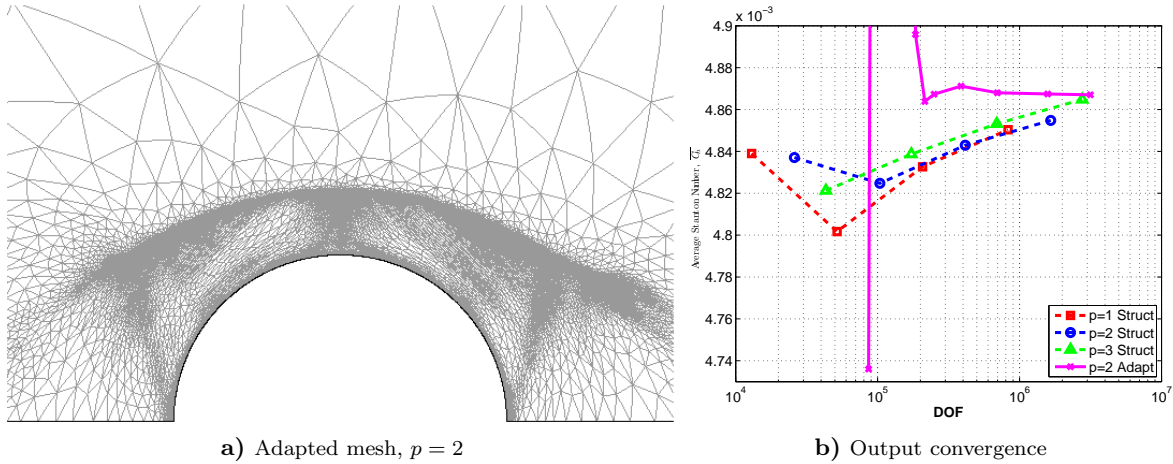


Fig. 20 Hypersonic flow over a 2D half-cylinder at $M_\infty = 17.605$, $Re = 376,930$: adaptation on average surface Stanton number. Final adapted mesh for $p = 2$ and output convergence for $p = 2$ adaptation (“Adapt”) compared to uniform refinement at orders $p = 1, 2, 3$ (“Struct”). Reproduced with permission from.⁶⁵

within 0.02 percent of the final value with approximately three million degrees of freedom.

V.2.3 Laminar Flows using Simplex Cut Cells (Fidkowski and Darmofal)

Fidkowski and Darmofal use an output-based adaptive indicator to drive cut-cell h -adaptation for a high-order discontinuous Galerkin discretization of the compressible Navier-Stokes equations.^{64,110} They use an implicit solution procedure for the primal problem that stores the full, compact-stencil, residual linearization, so that the discrete adjoint solution requires only one transpose application of the same linear solver. The fine-space adjoint solution ψ_h is approximated by reconstructing the coarse adjoint, ψ_H , on the same mesh with order enriched to $p + 1$. The adaptive indicator is formed by averaging localized primal and adjoint output error estimates.

The cut-cell method, illustrated in Figure 21, employs simplex elements and metric-driven global remeshing of the background domain to enable automated and anisotropic mesh adaptation. Meshing is performed using BAMG with anisotropy based on interpolation error in the Mach number. Fidkowski and Darmofal apply the cut-cell adaptive method to several inviscid and viscous flows. A representative example is that of viscous flow around a NACA 0012 airfoil at $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5,000$. Drag-adapted meshes for cubic, $p = 3$, solution interpolation are shown in Figure 22 for both the cut-cell method and a boundary-conforming method that requires curving of boundary elements and is consequently not as robust. In these meshes, areas of refinement include the boundary layer, a large extent of the wake, and, to a lesser extent, the flow in front of the airfoil.

Figure 23 compares adaptive convergence histories of the drag error for interpolation orders $p = 1, 2, 3$. The boundary-conforming and cut-cell runs converge to the same drag value, and the histories are similar. In both sets of runs, $p = 3$ requires only slightly fewer degrees of freedom than $p = 2$, while $p = 1$ remains the most expensive.

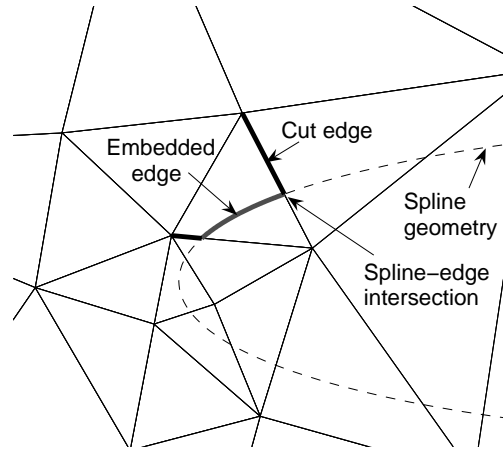


Fig. 21 Illustration of the simplex cut-cell method used in⁶⁴ for h -adaptation on high-order discontinuous Galerkin solutions. Curved cut cells are obtained by intersecting a cubic spline geometry representation with a triangular background mesh generated using BAMG. High order integration rules are derived on the resulting irregularly-shaped cut cells.

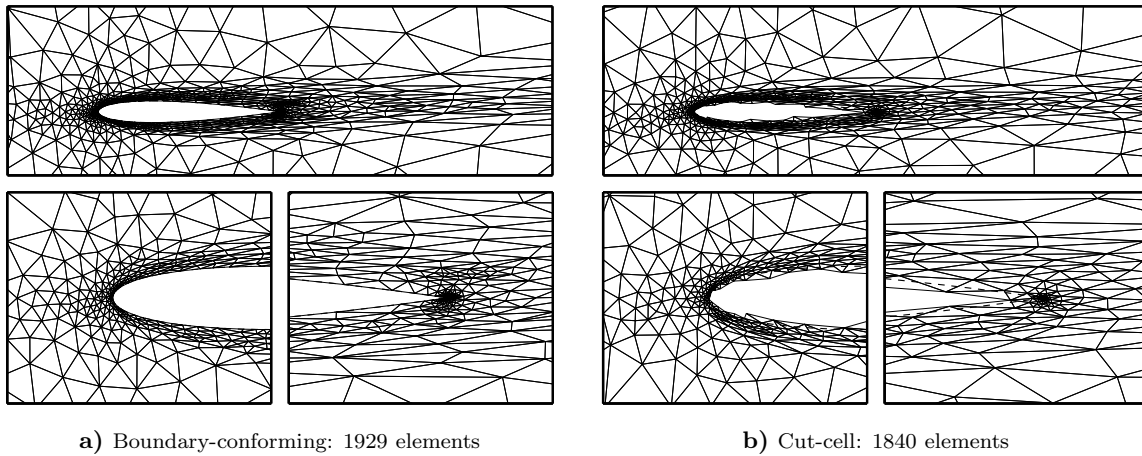


Fig. 22 NACA 0012 $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5,000$, $p = 3$ interpolation. Final boundary-conforming and cut-cell meshes adapted on drag.⁶⁴

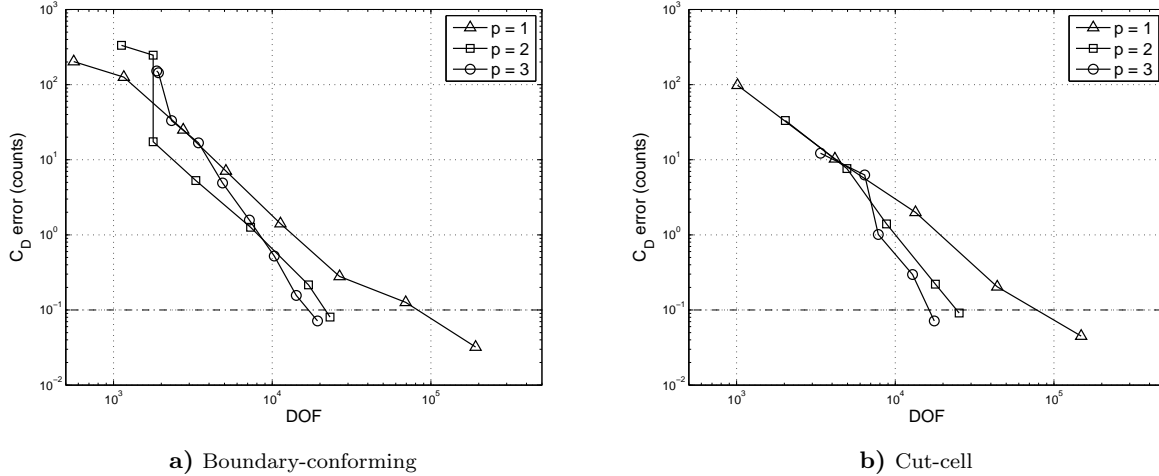


Fig. 23 NACA 0012 $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5,000$, Drag error versus degrees of freedom for interpolation orders $p = 1, 2, 3$. Dashed line indicates prescribed tolerance of 0.1 drag counts.⁶⁴

VI Challenges and Ongoing Research

VI.1 Robust Mesh Adaptation

Performing mesh adaptation robustly and efficiently for complex three-dimensional configurations is still a challenge and an area of ongoing research. The lack of robust and efficient mesh adaptation is probably the largest barrier limiting the application of output-based adaptation to simple geometries and/or simplified physics (i.e. inviscid flows as opposed to viscous flows).

The main robustness issue occurs during boundary point insertion on curved geometries: maintaining geometry fidelity can lead to invalid, negative-volume, elements on the interior, especially for highly-anisotropic meshes.⁶⁸ Currently, resolution of such situations is often attempted with iterative application of local operators and local re-meshing.¹⁴⁰ However, guaranteed geometric fidelity and element validity in highly anisotropic meshes around curved geometries has yet to be demonstrated.

Efficient adaptation refers to using available degrees of freedom as effectively as possible. A notable example of a situation in which efficient adaptation is important is in the resolution of curved, anisotropic, solution features, as illustrated in Figure 24. Simple isotropic mesh adaptation in these areas, Figure 24a, wastes resolution in directions where the solution is not changing significantly. In three dimensions, the additional degrees of freedom in the two directions where the solution is not changing quickly make resolution of thin layers impractical. Stretching elements along these directions helps significantly, especially when the anisotropic layer is approximately straight. When the layer is curved, however, the maximum feasible aspect ratio of linear stretched elements is bounded by the curvature of the layer since the geometry must be resolved, as illustrated in Figure 24b. Efficient resolution of very thin and curved features, such as those encountered in Reynolds-Averaged Navier-Stokes boundary layers, must therefore employ curved elements, as shown in Figure 24c.

Robust boundary-conforming mesh generation and adaptation techniques currently exist for many applications requiring only isotropic meshes, e.g.,¹⁴¹ and such techniques have been successfully applied to drive output-based adaptation on practical geometries, e.g.⁷⁰ An exception here is mesh generation for high-order

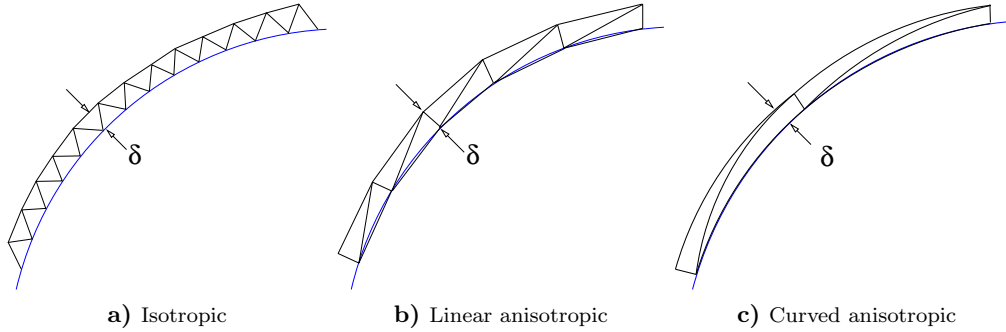


Fig. 24 Resolution of a thin, curved, anisotropic, two-dimensional layer using various mesh adaptation strategies. A resolution length of δ is required normal to the feature, whereas the solution does not change much tangentially to the feature. Stretching elements to increase their aspect ratio, defined as the ratio of the largest to smallest element diameters, improves efficiency of the mesh, measured in terms of degrees of freedom required to resolve the feature. Additionally curving the elements further improves efficiency by alleviating the curvature-imposed geometry resolution constraint.

methods that require curved elements on geometry boundaries, although in the isotropic case, node movement and certain heuristics are generally sufficient. However, robust boundary-conforming meshing and adaptation for three-dimensional anisotropic solution features are still areas of ongoing research. Curving elements for improved efficiency adds another layer of complexity, especially when curved anisotropic features occur away from the geometry.

Cut-cell techniques eliminate the requirement that the mesh needs to respect the geometry boundary, and can be robust for complex geometries. However, existing Cartesian methods only allow for isotropic refinement, except in special cases when the features are aligned with the mesh.^{63,135} Simplex cut-cell methods allow for element stretching, and hence resolution with improved efficiency.⁶⁴ However, additional research is required to extend these methods to stretch and curve elements so as to handle curved anisotropic features in three dimensions.

VI.2 Computable Error Bounds

The output error estimate in Eqs. 18 is not a bound for the true output error in the discrete solution because of the use of a finite-dimensional fine space, \mathcal{V}_h . If the computational mesh is very coarse, the fine space obtained by uniform mesh refinement or interpolation order increase may still be too coarse to faithfully resolve the output of interest. In such a case, the output error estimate may be severely unreliable.

An example of this effect is the pressure signature adaptation on the delta wing-body sonic boom case considered by Park, as described in Section V.1.3. The vertical bars in Figure 16a show the pressure integral error estimate at each adaptation iteration. The relatively small size of the error bars in the first few iterations indicates that the output error is severely under-predicted on the first meshes. On these meshes and the fine spaces derived from them, the sonic boom signature is not at all resolved, and the output error estimate is meaningless. The estimate only becomes accurate after six or seven adaptation iterations, which corresponds to a substantial increase in the number of degrees of freedom. It does not start dropping until about ten adaptation iterations. The risk of an inaccurate error estimate on coarse meshes is that an automated adaptive process may terminate early, without sufficiently resolving the output of interest.

There exists a body of research that addresses this risk through the computation of error *bounds* on

the outputs of interest.^{30,142–146} The goal of this research has been to derive strict, constant-free, lower and upper bounds for outputs of interest. The bounds calculations are based on a reformulation of the output calculation into a constrained minimization problem with a convex objective function, with the model equations entering the problem as equality constraints. Initially, the bounds calculations were strict with respect to a conservatively refined computational mesh;^{30,142,143} more recently the calculations have been extended so that the bounds are strict with respect to the exact weak solution of the partial differential equation.^{144–146} These calculations rely on the solution of a local adjoint problem that transforms an infinite-dimensional minimization problem into a finite-dimensional feasibility problem.¹⁴⁴

This strategy has been applied to symmetric and non-symmetric coercive problems,^{142,144} certain constrained and non-coercive problems,³⁰ and also to problems with nonlinear outputs and equations.^{143,147,148} For nonlinear problems, the present procedure yields bounds only for sufficiently resolved meshes, where the required mesh resolution is not known *a priori*.¹⁴³

The bound gap, which is the difference between the upper and lower bounds on the output, can be separated into positive contributions from each element, yielding an indicator for mesh adaptation. A strategy that refines elements with a large contribution to the bound gap will efficiently yield a tighter estimate of the output. The result of applying such a strategy is a solution with fully certifiable precision of integrated outputs. Additional research in this area is necessary to extend the bounds computations to additional equation sets, especially for aerospace Computational Fluid Dynamics applications.

VI.3 Unsteady Applications

As output-based adaptation is applied to increasingly complex flows, problems that include unsteadiness will naturally arise. Even for applications targeting nominally steady solutions, unsteadiness is likely to occur as wakes are resolved with adaptation. The work of Nemec *et al* described in Section V.1.2 is an example of this occurring.

Existing work in the application of adjoint sensitivity analysis to unsteady problems comes largely from shape optimization research, and ranges from frequency domain methods for periodic unsteady flows^{149,150} to time-accurate continuous and discrete adjoint methods.^{151–155} Incorporation of time dependence in adjoint analysis is theoretically a well understood problem. However, the algorithmic issues involved in solving an unsteady adjoint are substantial as the unsteady adjoint must be marched backward in time from the final to the initial state. For nonlinear problems, the solution time history must be stored or reconstructed to build the required Jacobians and output linearizations at each time level.

Mani and Mavriplis apply the adjoint-weighted residual technique to drive time step adaptation in unsteady flow simulations with deforming meshes.¹⁵⁴ For time adaptation, the time step is allowed to vary, but the spatial mesh distribution is not changed (except for prescribed deformation). The output error is thus calculated by evaluating the residual of the unsteady primal solution on a finer temporal discretization and weighting it with the adjoint solution on that discretization. The results in Mani and Mavriplis's work indicate a computational savings over uniform temporal refinement for smooth unsteady problems. An area of future research is combined spatial and temporal adaptation for problems exhibiting non-smooth spatial and temporal features, where the computational savings of an output-based adaptive method could be very significant.

VII Acknowledgments

The authors would like to acknowledge M. Aftosmis, G. Barter, R. Hartmann, P. Houston, D. Mavriplis, M. Nemec, T. Oliver, and M. Park, for permission to present their results and to use their figures. In addition the authors gratefully acknowledge the support of the Massachusetts Institute of Technology and the University of Michigan.

References

- ¹Levy, D. W., Zickuhr, T., Vassberg, J., Agrawal, S., Wahls, R. A., Pirzadeh, S., and Hensch, M. J., “Data summary from the First AIAA Computational Fluid Dynamics Drag Prediction Workshop,” *Journal of Aircraft*, Vol. 40, No. 5, 2003, pp. 875–882.
- ²Laffin, K. R., Vassberg, J. C., Wahls, R. A., Morrison, J. H., Brodersen, O., Rakowitz, M., Tinoco, E. N., and Godard, J.-L., “Summary of data from the second AIAA CFD drag prediction workshop,” AIAA Paper 2004-0555, 2004.
- ³Frink, N. T., “Test case results from the 3rd AIAA drag prediction workshop,” NASA Langley, 2007, http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3/final_results_jm.tar.gz.
- ⁴Vassberg, J. C., DeHaan, M. A., and Sclafani, T. J., “Grid generation requirements for accurate drag predictions based on OVERFLOW calculations,” AIAA Paper 2003-4124, 2003.
- ⁵Fidkowski, K. J., *A Simplex Cut-Cell Adaptive Method for High-order Discretizations of the Compressible Navier-Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2007.
- ⁶Morrison, J. H. and Hensch, M. J., “Statistical analysis of CFD solutions from the third AIAA drag prediction workshop,” AIAA Paper 2007-254, 2007.
- ⁷Mavriplis, D. J., “Results from the 3rd drag prediction workshop using the NSU3D unstructured mesh solver,” AIAA Paper 2007-256, 2007.
- ⁸Verfürth, R., “A posteriori error estimation and adaptive mesh-refinement techniques,” *Journal of Computational and Applied Mathematics*, Vol. 50, 1994, pp. 67–83.
- ⁹Ainsworth, M. and Oden, J. T., “A posteriori error estimation in finite element analysis,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 142, 1997, pp. 1–88.
- ¹⁰Johnson, C., “On computability and error control in CFD,” *International Journal for Numerical Methods in Fluids*, Vol. 20, 1995, pp. 777–788.
- ¹¹Johnson, C., Rannacher, R., and Boman, M., “Numerics and hydrodynamic stability: Toward error control in computational fluid dynamics,” *SIAM Journal on Numerical Analysis*, Vol. 32, No. 4, 1995, pp. 1058–1079.
- ¹²Roy, C. J., “Strategies for driving mesh adaptation in CFD,” AIAA Paper 2009-1302, 2009.
- ¹³Becker, R. and Rannacher, R., “An optimal control approach to a posteriori error estimation in finite element methods,” *Acta Numerica*, edited by A. Iserles, Cambridge University Press, 2001, pp. 1–102.
- ¹⁴Giles, M. and Pierce, N., “Adjoint error correction for integral outputs,” *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, Vol. 25, Springer, Berlin, 2002.
- ¹⁵Pierce, N. A. and Giles, M. B., “Adjoint and defect error bounding and correction for functional estimates,” *Journal of Computational Physics*, Vol. 200, No. 2, 2004, pp. 769–794.

- ¹⁶Giles, M. B. and Süli, E., “Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality,” *Acta Numerica*, Vol. 11, 2002, pp. 145–236.
- ¹⁷Strang, G. and Fix, G. J., *An Analysis of the Finite Element Method*, Wellesley-Cambridge Press, 1988.
- ¹⁸Jameson, A., “Aerodynamic design via control theory,” *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233–260.
- ¹⁹Anderson, W. K. and Venkatakrishnan, V., “Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation,” Tech. Rep. Report 97-9, ICASE, 1997.
- ²⁰Elliott, J. and Peraire, J., “Practical three-dimensional aerodynamic design and optimization using unstructured meshes,” *AIAA Journal*, Vol. 35, No. 9, 1997, pp. 1479–1485.
- ²¹Nielsen, E. and Anderson, W., “Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations,” *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419.
- ²²Mohammadi, B. and Pironneau, O., “Mesh adaption and automatic differentiation in a CAD-free framework for optimal shape design,” *International Journal for Numerical Methods in Fluids*, Vol. 30, 1999, pp. 127–136.
- ²³Martins, J. R., Alonso, J. J., and Reuther, J. J., “A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design,” *Optimization and Engineering*, Vol. 6, 2005, pp. 33–62.
- ²⁴Babuška, I. and Rheinboldt, W., “A posteriori error estimates for the finite element method,” *International Journal for Numerical Methods in Engineering*, Vol. 12, 1978, pp. 1597–1615.
- ²⁵Babuška, I. and Miller, A., “The post-processing approach in the finite element method – part 1: calculation of displacements, stresses and other higher derivatives of the displacements,” *International Journal for Numerical Methods in Engineering*, Vol. 20, 1984, pp. 1085–1109.
- ²⁶Babuška, I. and Miller, A., “The post-processing approach in the finite element method – part 2: the calculation of stress intensity factors,” *International Journal for Numerical Methods in Engineering*, Vol. 20, 1984, pp. 1111–1129.
- ²⁷Giles, M., Larson, M. G., Levenstam, J. M., and Süli, E., “Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow,” Tech. Rep. Technical Report NA 76/06, Oxford University Computing Lab Numerical Analysis Group, 1997.
- ²⁸Monk, P. and Süli, E., “The adaptive computation of far-field patterns by a posteriori error estimation of linear functional,” *SIAM Journal on Numerical Analysis*, Vol. 36, No. 1, 1998, pp. 251–274.
- ²⁹Becker, R. and Rannacher, R., “A feed-back approach to error control in finite element methods: Basic analysis and examples,” *East-West Journal of Numerical Mathematics*, Vol. 4, 1996, pp. 237–264.
- ³⁰Peraire, J. and Patera, A., “Asymptotic a posteriori finite element bounds for the outputs of non-coercive problems: the Helmholtz and Burgers equation,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 171, 1999, pp. 77–86.
- ³¹Giles, M. B. and Pierce, N. A., “Improved lift and drag estimates using adjoint Euler equations,” , No. 99-3293, 1999.
- ³²Houston, P., Rannacher, R., and Süli, E., “A posteriori error analysis for stabilised finite element approximations of transport problems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, 2000, pp. 1483–1508.

- ³³Pierce, N. A. and Giles, M. B., “Adjoint recovery of superconvergent functionals from PDE approximations,” *SIAM Review*, Vol. 42, No. 2, 2000, pp. 247–264.
- ³⁴Oden, J. and Prudhomme, S., “Goal-oriented error estimation and adaptivity for the finite element method,” *Computers and Mathematics with Applications*, Vol. 41, 2001, pp. 735–756.
- ³⁵Barth, T. and Larson, M., “A posteriori error estimates for higher order Godunov finite volume methods on unstructured meshes,” *Finite Volumes for Complex Applications III*, edited by R. Herban and D. Kröner, Hermes Penton, London, 2002, pp. 41–63.
- ³⁶Venditti, D. A. and Darmofal, D. L., “Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows,” *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46.
- ³⁷Lu, J., *An a Posteriori Error Control Framework for Adaptive Precision Optimization Using Discontinuous Galerkin Finite Element Method*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.
- ³⁸Giles, M. B. and Pierce, N. A., “Adjoint equations in CFD: duality, boundary conditions and solution behavior,” AIAA Paper 97-1850, 1997.
- ³⁹Giles, M. and Pierce, N., “Analytic adjoint solutions for the quasi-one-dimensional Euler equations,” *Journal of Fluid Mechanics*, Vol. 426, 2001, pp. 327–345.
- ⁴⁰Hartmann, R., “Adjoint consistency analysis of discontinuous Galerkin discretizations,” *SIAM Journal on Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696.
- ⁴¹Oliver, T. A. and Darmofal, D. L., “Impact of turbulence model irregularity on high-order discretizations,” AIAA Paper 2009-953, 2009.
- ⁴²Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., “Unified analysis of discontinuous Galerkin methods for elliptic problems,” *SIAM Journal on Numerical Analysis*, Vol. 39, No. 5, 2002, pp. 1749–1779.
- ⁴³Harriman, K., Houston, P., Senior, B., and Süli, E., “hp-version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form.” Tech. Rep. Technical Report NA 02/21, Oxford University Computing Lab Numerical Analysis Group, 2002.
- ⁴⁴Harriman, K., Gavaghan, D., and Süli, E., “The importance of adjoint consistency in the approximation of linear functionals using the discontinuous Galerkin finite element method,” Tech. Rep. Technical Report NA 04/18, Oxford University Computing Lab Numerical Analysis Group, 2004.
- ⁴⁵Lu, J. and Darmofal, D. L., “A quasi-minimal residual method for simultaneous primal-dual solutions and superconvergent functional estimates,” *SIAM Journal on Scientific Computing*, Vol. 24, No. 5, 2003, pp. 1693–1709.
- ⁴⁶Müller, J. and Cusdin, P., “On the performance of discrete adjoint CFD codes using automatic differentiation,” *International Journal for Numerical Methods in Fluids*, Vol. 47, No. 8–9, 2005, pp. 939–945.
- ⁴⁷Mader, C. A., Martins, J. R., Alonso, J. J., and van der Weide, E., “ADjoint: An approach for the rapid development of discrete adjoint solvers,” *AIAA Journal*, Vol. 46, No. 4, 2008, pp. 863–873.
- ⁴⁸Giles, M. B., Duta, M. C., Müller, J.-D., and Pierce, N. A., “Algorithm developments for discrete adjoint methods,” *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 198–205.
- ⁴⁹Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., “An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids,” *Computers and Fluids*, Vol. 33, 2004, pp. 1131–1155.
- ⁵⁰Sonar, T., “Strong and weak norm refinement indicators based on the finite element residual for

compressible flow computation,” *Impact of Computing in Science and Engineering*, Vol. 5, 1993, pp. 111–127.

⁵¹Sonar, T., Hanneman, V., and Hempel, D., “Dynamic adaptivity and residual control in unsteady compressible flow computation,” *Mathematical and Computer Modelling*, Vol. 20, No. 10, 1994, pp. 201–213.

⁵²Thomas, I. and Sonar, T., “On a second order residual estimator for numerical schemes for nonlinear hyperbolic conservation laws,” *Journal of Computational Physics*, Vol. 171, No. 1, 2001, pp. 227–242.

⁵³Zhang, X. D., Pelletier, D., Trepanier, J.-Y., and Camarero, R., “Error control and mesh adaptation for the Euler equations,” AIAA Paper 2001-0441, 2001.

⁵⁴Zhang, X. D., Vallet, M.-G., Dompierre, J., Labbe, P., Pelletier, D., Trepanier, J.-Y., Camarero, R., Lassaline, J. V., Manzano, L. M., and Zingg, D. W., “Mesh adaptation using different error indicators for the Euler equations,” AIAA Paper 2001-2549, 2001.

⁵⁵Houston, P. and Süli, E., “Error estimation and adaptive discretization methods in computational fluid dynamics,” *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, edited by T. Barth and H. Deconinck, Springer-Verlag, Heidelberg, Lecture Notes in Computational Science and Engineering Vol 25, 2002, pp. 269–344.

⁵⁶Hartmann, R. and Houston, P., “Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations,” *Journal of Computational Physics*, Vol. 183, No. 2, 2002, pp. 508–532.

⁵⁷Solín, P. and Demkowicz, L., “Goal-oriented hp-adaptivity for elliptic problems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 193, 2004, pp. 449–468.

⁵⁸Müller, J. D. and Giles, M. B., “Solution adaptive mesh refinement using adjoint error analysis,” AIAA Paper 2001-2550, 2001.

⁵⁹Balasubramanian, R. and Newman III, J., “Adjoint-based error estimation and grid adaptation for functional outputs: Application to two-dimensional, inviscid, incompressible flows,” *Computers and Fluids*, Vol. 38, 2009, pp. 320–332.

⁶⁰Venditti, D. A. and Darmofal, D. L., “Grid adaptation for functional outputs: application to two-dimensional inviscid flows,” *Journal of Computational Physics*, Vol. 176, No. 1, 2002, pp. 40–39.

⁶¹Rannacher, R., “Adaptive Galerkin finite element methods for partial differential equations,” *Journal of Computational and Applied Mathematics*, Vol. 128, 2001, pp. 205–233.

⁶²Nemec, M. and Aftosmis, M. J., “Error estimation and adaptive refinement for embedded-boundary Cartesian meshes,” AIAA Paper 2007-4187, 2007.

⁶³Nemec, M., Aftosmis, M. J., and Wintzer, M., “Adjoint-based adaptive mesh refinement for complex geometries,” AIAA Paper 2008-0725, 2008.

⁶⁴Fidkowski, K. J. and Darmofal, D. L., “A triangular cut-cell adaptive method for high-order discretizations of the compressible navier-stokes equations,” *Journal of Computational Physics*, Vol. 225, 2007, pp. 1653–1672.

⁶⁵Barter, G. E., *Shock Capturing with PDE-Based Artificial Viscosity for an Adaptive Higher-Order Discontinuous Galerkin Finite Element Method*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.

⁶⁶Oliver, T. A., *A High-order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.

⁶⁷Venditti, D. A. and Darmofal, D. L., “Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow,” *Journal of Computational Physics*, Vol. 164, No. 1, 2000, pp. 204–227.

⁶⁸Park, M. A., “Adjoint-based, three-dimensional error prediction and grid adaptation,” AIAA Paper 2002-3286, 2002.

⁶⁹Park, M. A., “Three-dimensional turbulent RANS adjoint-based error correction,” AIAA Paper 2003-3849, 2003.

⁷⁰Lee-Rausch, E. M., Park, M. A., Jones, W. T., Hammond, D. P., and Nielsen, E. J., “Application of parallel adjoint-based error estimation and anisotropic grid adaptation for three-dimensional aerospace configurations,” AIAA Paper 2005-4842, 2005.

⁷¹Jones, W. T., Nielsen, E. J., and Park, M. A., “Validation of 3d adjoint based error estimation and mesh adaptation for sonic boom predictions,” AIAA Paper 2006-1150, 2006.

⁷²Balasubramanian, R. and Newman III, J., “Comparison of adjoint-based and feature-based grid adaptation for functional outputs,” *International Journal for Numerical Methods in Fluids*, Vol. 53, 2007, pp. 1541–1569.

⁷³Park, M. A., *Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.

⁷⁴Zienkiewicz, O. C. and Zhu, J. Z., “The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique,” *International Journal for Numerical Methods in Engineering*, Vol. 33, 1992, pp. 1331–1364.

⁷⁵Zienkiewicz, O. C. and Zhu, J. Z., “The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity,” *International Journal for Numerical Methods in Engineering*, Vol. 33, 1992, pp. 1365–1382.

⁷⁶Fidkowski, K. J. and Darmofal, D. L., “Output-based adaptive meshing using triangular cut cells.” M.I.T. Aerospace Computational Design Laboratory Report. ACDL TR-06-2, 2006.

⁷⁷Dwight, R. P., “Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation,” *Journal of Computational Physics*, Vol. 227, 2008, pp. 2845–2863.

⁷⁸Hartmann, R. and Houston, P., “Goal-oriented a posteriori error estimation for multiple target functionals,” *Hyperbolic Problems: Theory, Numerics, Applications*, edited by T. Hou and E. Tadmor, Springer-Verlag, 2003, pp. 579–588.

⁷⁹Szabo, B. A., “Estimation and control of error based on p convergence,” *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, edited by I. Babuska, O. C. Zienkiewicz, J. Gago, and E. R. de Oliveira, John Wiley & Sons Ltd., 1986, pp. 61–78.

⁸⁰Gui, W. and Babuska, I., “The h , p , and $h - p$ versions of the finite element method in 1 dimension. Part III. the adaptive $h - p$ version,” *Numerische Mathematik*, Vol. 49, 1986, pp. 105–158.

⁸¹Bey, K. S. and Oden, J. T., “ hp -version discontinuous Galerkin methods for hyperbolic conservation laws,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 133, 1996, pp. 259–286.

⁸²Rachowicz, W., Demkowicz, L., and Oden, J., “Toward a universal $h - p$ adaptive finite element strategy, part 3. design of $h - p$ meshes.” *Computer Methods in Applied Mechanics and Engineering*, Vol. 77, 1989, pp. 181–212.

- ⁸³Ainsworth, M. and Senior, B., “An adaptive refinement strategy for *hp*-finite element computations,” *Applied Numerical Mathematics*, Vol. 26, 1998, pp. 165–178.
- ⁸⁴Houston, P. and Süli, E., “*hp*-adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems,” *SIAM Journal on Scientific Computing*, Vol. 23, No. 4, 2001, pp. 1226–1252.
- ⁸⁵Demkowicz, L., Rachowicz, W., and Devloo, P., “A fully automatic *hp*-adaptivity,” *Journal of Scientific Computing*, Vol. 17, 2002, pp. 117–142.
- ⁸⁶Heuveline, V. and Rannacher, R., “Duality-based adaptivity in the *hp*-finite element method,” *Journal of Numerical Mathematics*, Vol. 11, No. 2, 2003, pp. 95–113.
- ⁸⁷Moore, P. K., “Applications of lobatto polynomials to an adaptive finite element method: A posteriori error estimates for *hp*-adaptivity and grid-to-grid interpolation,” *Numerische Mathematik*, Vol. 94, 2003, pp. 367–401.
- ⁸⁸Houston, P., Senior, B., and Süli, E., “Sobolev regularity estimation for *hp*-adaptive finite element methods,” *Numerical Mathematics and Advanced Applications*, edited by F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, Springer-Verlag, 2003, pp. 619–644.
- ⁸⁹Houston, P. and Süli, E., “A note on the design of *hp*-adaptive finite element methods for elliptic partial differential equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, 2005, pp. 229–243.
- ⁹⁰Berger, M. and Jameson, A., “Automatic adaptive grid refinement for the Euler equations.” *AIAA Journal*, Vol. 23, 1985, pp. 561–568.
- ⁹¹Mavriplis, D. J., “Unstructured mesh generation and adaptivity,” *26th Computational Fluid Dynamics Lecture Series. VKI Lecture Series*, Vol. 1995-2, 1995.
- ⁹²Baker, T. J., “Mesh adaptation strategies for problems in fluid dynamics,” *Finite Elements in Analysis and Design*, Vol. 25, 1997, pp. 243–273.
- ⁹³McRae, D. S., “*r*-Refinement grid adaptation algorithms and issues,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 2000, No. 4, 189, pp. 1161–1182.
- ⁹⁴Bern, M., “Adaptive mesh generation,” *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, edited by T. Barth and H. Deconinck, Springer-Verlag, Heidelberg, Lecture Notes in Computational Science and Engineering Vol 25, 2002, pp. 1–46.
- ⁹⁵Shewchuk, J., “Delaunay refinement algorithms for triangular mesh generation,” *Computational Geometry: Theory and Applications*, Vol. 22, 2002, pp. 21–74.
- ⁹⁶Oden, J. and Prudhomme, S., “New approaches to error estimation and adaptivity for the Stokes and Oseen equations,” *International Journal for Numerical Methods in Fluids*, Vol. 31, 1999, pp. 3–15.
- ⁹⁷Carstensen, C. and Verfürth, R., “Edge residuals dominate a posteriori error estimates for low order finite element methods,” *SIAM Journal on Numerical Analysis*, Vol. 36, No. 5, 1999, pp. 1571–1587.
- ⁹⁸Johansen, H. and Colella, P., “A Cartesian grid embedded boundary method for Poisson’s equation on irregular domains,” *Journal of Computational Physics*, Vol. 147, 1998, pp. 60–85.
- ⁹⁹Sun, S. and Wheeler, M., “Mesh adaptation strategies for discontinuous Galerkin methods applied to reactive transport problems,” *International Conference on Computing, Communication and Control Techniques*, H.W. Chu, M. Savoie, and B. Sanchez, Eds., 2004, pp. 223–228.
- ¹⁰⁰Freitag, L. A. and Ollivier-Gooch, C., “Tetrahedral mesh improvement using swapping and smoothing,” *International Journal for Numerical Methods in Engineering*, Vol. 40, 1997, pp. 3979–4002.

- ¹⁰¹Buscaglia, G. C. and Dari, E. A., “Anisotropic mesh optimization and its application in adaptivity,” *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 22, November 1997, pp. 4119–4136.
- ¹⁰²Castro-Diaz, M. J., Hecht, F., Mohammadi, B., and Pironneau, O., “Anisotropic unstructured mesh adaptation for flow simulations,” *International Journal for Numerical Methods in Fluids*, Vol. 25, 1997, pp. 475–491.
- ¹⁰³Wood, W. A. and Kleb, W. L., “On multi-dimensional unstructured mesh adaptation,” AIAA Paper 99-3254, 1999.
- ¹⁰⁴Habashi, W. G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M., and Vallet, M.-G., “Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles,” *International Journal for Numerical Methods in Fluids*, Vol. 32, 2000, pp. 725–744.
- ¹⁰⁵Xia, G., Li, D., and Merkle, C. L., “Anisotropic grid adaptation on unstructured meshes,” AIAA Paper 2001-0443, 2001.
- ¹⁰⁶Mavriplis, D. J., “Adaptive mesh generation for viscous flows using Delaunay triangulation,” *Journal of Computational Physics*, Vol. 90, 1990, pp. 271–291.
- ¹⁰⁷Borouchaki, H., George, P., Hecht, F., Laug, P., and Saltel, E., “Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie I: Algorithmes,” INRIA-Rocquencourt, France. Tech Report No. 2741, 1995.
- ¹⁰⁸Hecht, F., “BAMG: Bidimensional anisotropic mesh generator,” INRIA-Rocquencourt, France, 1998, <http://pauillac.inria.fr/cdrom/www/bamg/bamg/eng.htm>.
- ¹⁰⁹Schall, E., Leservoisier, D., Dervieux, A., and Koobus, B., “Mesh adaptation as a tool for certified computational aerodynamics,” *International Journal for Numerical Methods in Fluids*, Vol. 45, 2004, pp. 175–196.
- ¹¹⁰Fidkowski, K. J. and Darmofal, D. L., “An adaptive simplex cut-cell method for discontinuous Galerkin discretizations of the Navier-Stokes equations,” AIAA Paper 2007-3941, 2007.
- ¹¹¹Aftosmis, M. and Berger, M., “Multilevel error estimation and adaptive h-refinement for Cartesian meshes with embedded boundaries,” AIAA Paper 2002-14322, 2002.
- ¹¹²Zienkiewicz, O. C. and Zhu, J. Z., “Adaptivity and mesh generation,” *International Journal for Numerical Methods in Engineering*, Vol. 32, 1991, pp. 783–810.
- ¹¹³Siebert, K. G., “An a posteriori error estimator for anisotropic refinement,” *Numerische Mathematik*, Vol. 73, 1996, pp. 373–398.
- ¹¹⁴Sun, S. and Wheeler, M. F., “Anisotropic and dynamic mesh adaptation for discontinuous Galerkin methods applied to reactive transport,” Tech. Rep. 05-15, ICES, 2005.
- ¹¹⁵Venditti, D. A., *Grid Adaptation for Functional Outputs of Compressible Flow Simulations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2002.
- ¹¹⁶Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., “Adaptive remeshing for compressible flow computations,” *Journal of Computational Physics*, Vol. 72, 1987, pp. 449–466.
- ¹¹⁷Pagnutti, D. and Ollivier-Gooch, C., “A generalized framework for high order anisotropic mesh adaptation,” *Computers and Structures*, Vol. 87, No. 11-12, 2009, pp. 670 – 679.
- ¹¹⁸Persson, P.-O. and Peraire, J., “Curved mesh generation and mesh refinement using Lagrangian solid mechanics,” AIAA Paper 2009-0949, 2009.

- ¹¹⁹Ait-Ali-Yahia, D., Baruzzi, G., Habashi, W. G., Fortin, M., Dompierre, J., and Vallet, M.-G., “Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part II: Structured grids,” *International Journal for Numerical Methods in Fluids*, Vol. 39, 2002, pp. 657–673.
- ¹²⁰Dompierre, J., Vallet, M.-G., Bourgault, Y., Fortin, M., and Habashi, W. G., “Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part III: Unstructured meshes,” *International Journal for Numerical Methods in Fluids*, Vol. 39, 2002, pp. 675–702.
- ¹²¹Formaggia, L., Perotto, S., and Zunino, P., “An anisotropic a posteriori error estimate for a convection-diffusion problem,” *Computing and Visualization in Science*, Vol. 4, 2001, pp. 99–104.
- ¹²²Formaggia, L. and Perotto, S., “New anisotropic a priori error estimates,” *Numerische Mathematik*, Vol. 89, 2001, pp. 641–667.
- ¹²³Formaggia, L., Micheletti, S., and Perotto, S., “Anisotropic mesh adaptation with applications to CFD problems,” *Fifth World Congress on Computational Mechanics*, edited by H. A. Mang, F. G. Rammerstorfer, and J. Eberhardsteiner, Vienna, Austria, July 7-12 2002.
- ¹²⁴Schneider, R. and Jimack, P. K., “Toward anisotropic mesh adaptation based upon sensitivity of a posteriori estimates,” Tech. Rep. 2005.03, University of Leeds, School of Computing, 2005.
- ¹²⁵Park, M. A. and Darmofal, D. L., “Parallel anisotropic tetrahedral adaptation,” AIAA Paper 2008-917, 2008.
- ¹²⁶Purvis, J. W. and Burkhalter, J. E., “Prediction of critical Mach number for store configurations,” *AIAA Journal*, Vol. 17, No. 11, 1979, pp. 1170–1177.
- ¹²⁷Wedan, B. and South, J. C., “A method for solving the transonic full-potential equation for general configurations,” AIAA Paper 1983-1889, 1983.
- ¹²⁸Clarke, D. K., Salas, M. D., and Hassan, H. A., “Euler calculations for multielement airfoils using Cartesian grids,” *AIAA Journal*, Vol. 24, No. 3, 1986, pp. 353.
- ¹²⁹De Zeeuw, D. and Powell, K. G., “An adaptively refined Cartesian mesh solver for the Euler equations,” *Journal of Computational Physics*, Vol. 104, 1993, pp. 56–68.
- ¹³⁰Coirier, W. J. and Powell, K. G., “Solution-adaptive cut-cell approach for viscous and inviscid flows,” *AIAA Journal*, Vol. 34, No. 5, 1996, pp. 938–945.
- ¹³¹Rubbert, P. E., Bussoletti, J. E., Johnson, F. T., Sidwell, K. W., Rowe, W. S., Samant, S. S., SenGupta, G., Weatherill, W. H., Burkhart, R. H., Everson, B. L., Young, D. P., and Woo, A. C., “A new approach to the solution of boundary value problems involving complex configurations,” *Computational Mechanics – Advances and Trends*, edited by A. K. Noor, 1986, pp. 49–84.
- ¹³²Young, D. P., Melvin, R. G., Bieterman, M. B., Johnson, F. T., Samant, S. S., and Bussoletti, J. E., “A higher-order boundary treatment for Cartesian-grid methods,” *Journal of Computational Physics*, Vol. 92, 1991, pp. 1–66.
- ¹³³Karman, S. L., “SPLITFLOW: A 3d unstructured Cartesian/prismatic grid CFD code for complex geometries,” AIAA Paper 1995-0343, 1995.
- ¹³⁴Löhner, R., Appanaboyina, S., and Cebal, J. R., “Comparison of body-fitted, embedded, and immersed solutions of low-Reynolds number 3-d incompressible flows,” AIAA Paper 2007-1296, 2007.
- ¹³⁵Aftosmis, M. J., “Solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries,” *von Karman Institute for Fluid Dynamics, Lecture Series 1997-02*, Rhode-Saint-Genése, Belgium, Mar. 3-7, 1997.

- ¹³⁶Nemec, M., Aftosmis, M., Murman, S., and Pulliam, T., “Adjoint formulation for an embedded-boundary Cartesian method,” AIAA Paper 2005-0877, 2005.
- ¹³⁷Hunton, L., Hicks, R. M., and Mendoza, J. P., “Some effects of wing planform on sonic boom,” NASA TN D-7160, 1973.
- ¹³⁸Barter, G. and Darmofal, D., “Shock capturing with higher-order, PDE-based artificial viscosity,” AIAA Paper 2007-3823, 2007.
- ¹³⁹Gnoffo, P. A. and White, J. A., “Computational aerothermodynamic simulation issues for unstructured grids,” AIAA Paper 2004-2371, 2004.
- ¹⁴⁰Li, X., Shephard, M. S., and Beall, M. W., “Accounting for curved domains in mesh adaptation,” *International Journal for Numerical Methods in Engineering*, Vol. 58, 2003, pp. 247–276.
- ¹⁴¹Si, H., “Tetgen: A quality tetrahedral mesh generator and three-dimensional Delaunay triangulator,” Weierstrass Institute for Applied Analysis and Stochastics, 2005, <http://tetgen.berlios.de>.
- ¹⁴²Paraschivoiu, M., Peraire, J., and Patera, A., “A posteriori finite element bounds for linear functional outputs of elliptic partial differential equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 150, 1997, pp. 289–312.
- ¹⁴³Machiels, L., Peraire, J., and Patera, A., “A posteriori finite-element output bounds for the incompressible Navier-Stokes equations: Application to a natural convection problem,” *Journal of Computational Physics*, Vol. 172, 2001, pp. 401–425.
- ¹⁴⁴Sauer-Budge, A. and Peraire, J., “Computing bounds for linear functionals of exact weak solutions to the advection-diffusion-reaction equation,” *Journal on Scientific Computing*, Vol. 26, No. 2, 2004, pp. 636–652.
- ¹⁴⁵Xuan, Z., Pares, N., and Peraire, J., “Computing upper and lower bounds for the J-integral in two-dimensional linear elasticity,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, 2006, pp. 430–443.
- ¹⁴⁶Parés, N., Bonet, J., Huerta, A., and Peraire, J., “The computation of bounds for linear-functional outputs of weak solutions to the two-dimensional elasticity equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, 2006, pp. 406–429.
- ¹⁴⁷Sarrate, J., Peraire, J., and Patera, A., “A posteriori finite element error bounds for nonlinear outputs of the Helmholtz equation,” *International Journal for Numerical Methods in Fluids*, Vol. 31, 1999, pp. 17–36.
- ¹⁴⁸Wong, J. and Peraire, J., “Implicit a-posteriori bounds for discontinuous Galerkin methods,” AIAA Paper 2003-3844, 2003.
- ¹⁴⁹Lee, K.-H., Alonso, J. J., and van der Weide, E., “Mesh adaptation criteria for unsteady periodic flows using a discrete adjoint time-spectral formulation,” AIAA Paper 2006-692, 2006.
- ¹⁵⁰Nadarajah, S. K. and Jameson, A., “Optimum shape design for unsteady three-dimensional viscous flows using a nonlinear frequency-domain method,” *AIAA Journal of Aircraft*, Vol. 44, No. 5, 2007, pp. 1513–1527.
- ¹⁵¹Nadarajah, S. K. and Jameson, A., “Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint methods,” *AIAA Journal*, Vol. 45, No. 7, 2007, pp. 1478–1491.
- ¹⁵²Nadarajah, S. K. and Jameson, A., “Optimal control of unsteady flows using a time accurate method,” AIAA Paper 2002-5436, 2002.

¹⁵³Rumpfkeil, M. P. and Zingg, D. W., “A general framework for the optimal control of unsteady flows with applications,” AIAA Paper 2007-1128, 2007.

¹⁵⁴Mani, K. and Mavriplis, D. J., “Discrete adjoint based time-step adaptation and error reduction in unsteady flow problems,” AIAA Paper 2007-3944, 2007.

¹⁵⁵Mani, K. and Mavriplis, D. J., “An unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes,” AIAA Paper 2007-60, 2007.