19th AIAA Computational Fluid Dynamics
22 - 25 June 2009, San Antonio, Texas

AIAA 2009-3790

Entropy-based Mesh Refinement, I: The Entropy Adjoint Approach

Krzysztof J. Fidkowski * and Philip L. Roe †

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109

This work presents a mesh refinement indicator based on entropy-variables, with an application to the compressible Navier-Stokes equations. The entropy variables are shown to satisfy an adjoint equation, an observation that allows recent work in adjoint-based error estimation to be leveraged in constructing a relatively cheap but effective adaptation indicator. The output associated with the entropy-variable adjoint is shown to be the net entropy generation in the computational domain, including physical viscous dissipation when present. Adaptation using entropy variables thus targets areas of the domain responsible for numerical, or spurious, entropy generation. Adaptive results for inviscid and viscous aerodynamic examples demonstrate performance efficiency on par with output-based adaptation, as measured by errors in various engineering quantities of interest, with the comparative advantage that no adjoint equations need to be solved.

I. Introduction

Solution-based adaptive methods are becoming increasingly popular in Computational Fluid Dynamics (CFD), where they are used to obtain accurate solutions to problems that exhibit a wide range of spatial length scales whose distribution is generally not known *a priori*.[1–7] These methods all rely on some form of indicator to drive adaptation of the computational mesh. Various indicators have been studied in the literature, ranging from ones that are cheap but lacking in robustness, to ones that are theoretically sound but expensive to compute. In this work, we take advantage of a connection between entropy variables and adjoints to devise an inexpensive adaptive indicator that is something of a compromise.

Previously developed heuristic indicators are often based on the idea that the mesh should be refined in "interesting" areas, such as those where solution gradients, curvatures, or other feature measures are active.[8–10] While such heuristics are cheap and can yield visually pleasing results, they often perform poorly in terms of computed output quantities of interest, due to over-refinement of certain features and under-refinement of relatively feature-less but nevertheless important areas.[4,11]

Output-based adaptive methods address these shortcomings by specifically targeting for refinement areas of the computational domain that are important for predicting the output of interest.[3,4,12,13] These methods properly account for the propagation effects that are intrinsic to

*Assistant Professor, AIAA Member

†Professor, AIAA Fellow, presently at the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK

hyperbolic problems,[14] through the use of an adjoint solution as the sensitivity of the output to local residuals. A common assumption is that adaptation on several key outputs yields an adequate, general-purpose solution.

Choosing representative outputs involves a certain degree of arbitrariness, and in some cases no clear outputs present themselves. More practically, a code may not possess adjoint capability, or the required multiple adjoint solutions may be too expensive. A natural question is whether it is possible to obtain a "good" general-purpose solution without choosing specific outputs. In this work, a cheap and general adaptive indicator is analyzed that targets areas of spurious entropy generation. Specifically, the entropy variables, which derive from the conservative state variables by a simple transformation, are shown to be the adjoint solution corresponding to an output that measures the net entropy production in the computational domain. The entropy variables are thus used as adjoints in an output-based adaptation framework even though the flow and generation of entropy is not normally considered to be something of direct interest. The method can also be regarded as a form of "feature-based" adaptation, without any requirement for a user to define the features. Instead, the error indicator defines "features" to be those areas where entropy production is difficult to compute.

The outline of the remainder of the paper is as follows. Section II reviews the primary concepts behind error estimation and mesh adaptation with output adjoints. In Section III the entropy variables are defined and shown to satisfy an adjoint equation. The corresponding output is derived for inviscid and viscous conservation laws. Section IV discusses the implementation of the proposed method in a simple adaptive strategy, which is then used to generate the results in Section V.

II. Adaptation Using Output Adjoints

Output-based error estimation techniques identify all areas of the domain that are important for the accurate prediction of an engineering output. The resulting estimates properly account for error propagation effects that are inherent to hyperbolic problems, and they can be used to ascribe confidence levels to outputs and to drive adaptation. Output error estimation relies on the solution of an adjoint problem for the output of interest. In a continuous setting, an adjoint, $\boldsymbol{\psi}(x)$, is a Green's function relating residual source perturbations to an output of interest, $J(\mathbf{u})$, where $\mathbf{u}$ denotes the state variable vector. Specifically, consider a partial differential equation (PDE), $\mathbf{r}(\mathbf{u}) = 0$, in a variational formulation: determine $\mathbf{u} \in \mathcal{V}$ such that

$$\mathcal{R}(\mathbf{u}, \mathbf{w}) = 0, \qquad \forall \mathbf{w} \in \mathcal{V}, \tag{1}$$

where $\mathcal{V}$ is an appropriate function space and $\mathcal{R}(\cdot,\cdot) : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is a semilinear form. The adjoint $\boldsymbol{\psi} \in \mathcal{V}$ is the sensitivity of J to an infinitesimal source term $\delta\mathbf{r}$ added to the PDE,

$$\delta J = (\delta\mathbf{r}, \boldsymbol{\psi}), \tag{2}$$

where $(\cdot,\cdot)$ is an inner product over the computational domain, Ω. The output adjoint equation can be derived by linearizing Eqn. 1 to relate state perturbations to residual perturbations, and by requiring the sensitivity property in Eqn. 2 to hold.[15] The result is the variational formulation: determine $\boldsymbol{\psi} \in \mathcal{V}$ such that

$$\mathcal{R}'[\mathbf{u}](\mathbf{w}, \boldsymbol{\psi}) + J'[\mathbf{u}](\mathbf{w}) = 0, \qquad \forall \mathbf{w} \in \mathcal{V}, \tag{3}$$

where the primes denote Fréchét linearization with respect to the arguments in square brackets. An adjoint solution is useful for estimating the numerical error in an output in the following manner:

- An approximate solution $\mathbf{u}_H$ in a finite-dimensional approximation space $\mathcal{V}_H$ will generally not satisfy the original PDE. Instead it will satisfy a perturbed PDE whose weak form reads: find $\mathbf{u}' \in \mathcal{V}$ such that

$$\mathcal{R}(\mathbf{u}', \mathbf{w}) + (\delta\mathbf{r}, \mathbf{w}) = 0, \quad \forall \mathbf{w} \in \mathcal{V}, \qquad \text{where } (\delta\mathbf{r}, \mathbf{w}) = -\mathcal{R}(\mathbf{u}_H, \mathbf{w}).$$

- The adjoint $\boldsymbol{\psi} \in \mathcal{V}$ translates the residual perturbation to an output perturbation via Eqn. 2:

$$\delta J = (\delta\mathbf{r}, \boldsymbol{\psi}) = -\mathcal{R}(\mathbf{u}_H, \boldsymbol{\psi}). \tag{4}$$

  This expression quantifies the numerical error in the output via a weighted residual of the approximate solution. The approximation signs indicate that for non-infinitesimal perturbations, the above expression is not exact and yields only an estimate of the numerical error.

The continuous adjoint $\boldsymbol{\psi}$ must be approximated to make the error estimate in Eqn. 4 computable. In practice, a discrete version of the adjoint equation is solved approximately or exactly on a finer finite-dimensional space, $\mathcal{V}_h \supset \mathcal{V}_H$, to yield $\boldsymbol{\psi}_h \in \mathcal{V}_h$.[16–18] The adjoint-weighted residual evaluation in Eqn. 4 can be localized to yield an adaptive indicator consisting of the relative contribution of each element to the total output error. Working with $\boldsymbol{\psi}_h \in \mathcal{V}_h$, the output perturbation in Eqn. 4 is approximated as

$$\delta J \approx -\sum_{\kappa_H \in T_H} \mathcal{R}_h(\mathbf{u}_H, \boldsymbol{\psi}_h|_{\kappa_H}), \tag{5}$$

where $|_{\kappa_H}$ refers to restriction to element κ_H of the triangulation T_H, and where for simplicity we assume that $\mathcal{V}_h$ is obtained from $\mathcal{V}_H$ solely by an order increase, keeping the triangulation fixed. Eqn. 5 expresses the output error in terms of contributions from each coarse element. A common approach for obtaining an adaptive indicator is to take the absolute value of the elemental contribution in Eqn. 5,[3, 12, 19, 20]

$$\eta_{\kappa_H} = \left| \mathcal{R}_h(\mathbf{u}_H, \boldsymbol{\psi}_h|_{\kappa_h}) \right|. \tag{6}$$

For systems of equations, indicators are typically computed separately for each equation and summed together. Due to the absolute values, the sum of the indicators, $\sum_{\kappa_H} \eta_{\kappa_H}$, is greater or equal to the original output error estimate. However, it is not a bound on the actual error because of the approximations made in the derivation.

III. Adaptation Using Entropy Variables

The adaptive indicator derived in the previous section is specific to a user-defined scalar output and requires the solution of an adjoint problem. In this section, a cheaper indicator is presented based on the entropy variables. This indicator is motivated by the observation that the entropy variables serve the role of an adjoint solution for an output that measures the net entropy production in the computational domain.

A. Inviscid Conservation Laws

Consider a steady-state set of conservation laws in quasi-linear form together with a scalar entropy conservation law,

$$\mathbf{A}_i \partial_i \mathbf{u} = 0, \quad \partial_i F_i = 0,$$

where i indexes the spatial dimension, $\mathbf{u}$ is the state vector, $\mathbf{A}_i$ is the inviscid flux Jacobian, and $F_i(\mathbf{u})$ is the entropy flux associated with an entropy function $U(\mathbf{u})$. The entropy conservation law holds only if the compatibility relation $U_{\mathbf{u}}\mathbf{A}_i = (F_i)_{\mathbf{u}}$ is satisfied. For a convex entropy function U, the set of corresponding entropy variables is defined by $\mathbf{v} \equiv U_{\mathbf{u}}^T$. The entropy variables symmetrize the conservation laws in the sense that[9,21]

- The transformation Jacobian matrix, $\mathbf{u}_{\mathbf{v}}$, is symmetric, positive definite,
- $\mathbf{A}_i\mathbf{u}_{\mathbf{v}}$ is symmetric.

Using these symmetry properties, the quasi-linear form of the conservation law can be manipulated as follows:

$$0 = \mathbf{A}_i\partial_i\mathbf{u} = \mathbf{A}_i\mathbf{u}_{\mathbf{v}}\partial_i\mathbf{v} = \mathbf{u}_{\mathbf{v}}\mathbf{A}_i^T\partial_i\mathbf{v} \quad \Rightarrow \quad \mathbf{A}_i^T\partial_i\mathbf{v} = 0. \tag{7}$$

The appearance of the transpose on $\mathbf{A}_i$ indicates that the entropy variables satisfy an adjoint equation. A Lagrange multiplier analysis, not included here, shows that the entropy variables also satisfy the required adjoint boundary conditions. The associated output can be derived by regarding the adjoint as a Green's function sensitivity to residual source perturbations. Using the quasi-linear form, state and residual perturbations are related via

$$\delta\mathbf{r} = \mathbf{A}_i\partial_i\delta\mathbf{u}.$$

The output perturbation is obtained by an inner product of the adjoint with the residual perturbation over the domain Ω,

$$\begin{aligned}\delta J = \int_\Omega \mathbf{v}^T\delta\mathbf{r}\,d\Omega = \int_\Omega \mathbf{v}^T\mathbf{A}_i\partial_i\delta\mathbf{u}\,d\Omega &= \underbrace{-\int_\Omega \partial_i\mathbf{v}^T\mathbf{A}_i\delta\mathbf{u}\,d\Omega}_{=0 \text{ by Eqn. 7}} + \int_{\partial\Omega}\underbrace{\mathbf{v}^T\mathbf{A}_i}_{(F_i)_{\mathbf{u}}}\delta\mathbf{u}\,n_i ds \\ &= \int_{\partial\Omega}(F_i)_{\mathbf{u}}\delta\mathbf{u}\,n_i ds = \delta\left[\underbrace{\int_{\partial\Omega}F_i n_i ds}_{J}\right].\end{aligned}$$

Therefore the entropy variables serve as the adjoint solution to an output that, up to an additive constant, measures the net entropy flow out of the domain. The enabling property of the entropy variables that allows this derivation is the symmetrizing property that produced Eqn. 7. Replacing the adjoint in Eqn. 6 with the entropy variables yields an adaptive indicator that identifies all areas of the domain important for the correct prediction of the net entropy outflow. Specifically, areas of spurious entropy generation will be targeted for refinement.

B. Viscous Conservation Laws

In quasi-linear form, a set of viscous conservation laws reads

$$\mathbf{A}_i\partial_i\mathbf{u} - \partial_i(\mathbf{K}_{ij}\partial_j\mathbf{u}) = 0, \tag{8}$$

where $\mathbf{K}_{ij}\partial_j\mathbf{u}$ is the viscous flux. The entropy variable definitions from the previous section still hold, with an additional requirement on the entropy function $U(\mathbf{u})$: the entropy variable choice

$\mathbf{v} = U_{\mathbf{u}}^T$ must now also symmetrize $\mathbf{K}_{ij}$, in the sense that $\widetilde{\mathbf{K}}_{ij} = \widetilde{\mathbf{K}}_{ji}^T$, where $\widetilde{\mathbf{K}}_{ij} \equiv \mathbf{K}_{ij}\mathbf{u}_{\mathbf{v}}$.[21] Substituting $\partial_i \mathbf{u} = \mathbf{u}_{\mathbf{v}} \partial_i \mathbf{v}$ into Eqn. 8 and taking the transpose yields a useful relationship,

$$\partial_i \mathbf{v}^T \mathbf{A}_i \mathbf{u}_{\mathbf{v}} - \partial_i (\partial_j \mathbf{v}^T \widetilde{\mathbf{K}}_{ji}) = 0. \tag{9}$$

In contrast to the inviscid case, this is longer a mathematical adjoint to the primal equation, Eqn. 8; the sign of the second term should be reversed. However, the entropy variables still represent the sensitivity to residual perturbations of a specific output, although that output is no longer expressible as an integral over the domain boundary. Specifically, the corresponding output now includes entropy generation due to viscous dissipation.

Using Eqn. 8, the state and residual perturbations are related via

$$\delta \mathbf{r} = \mathbf{A}_i \partial_i \delta \mathbf{u} - \partial_i (\mathbf{K}_{i,j} \partial_j \delta \mathbf{u}),$$

so that the output perturbation is

$$\begin{aligned}
\delta J &= \int_\Omega \mathbf{v}^T \delta \mathbf{r} \, d\Omega \\
&= -\int_\Omega \partial_i \mathbf{v}^T \mathbf{A}_i \delta \mathbf{u} \, d\Omega + \int_{\partial\Omega} \mathbf{v}^T \mathbf{A}_i \delta \mathbf{u} \, n_i ds + \int_\Omega \partial_i \mathbf{v}^T \mathbf{K}_{ij} \partial_j \delta \mathbf{u} \, d\Omega - \int_{\partial\Omega} \mathbf{v}^T \mathbf{K}_{ij} \partial_j \delta \mathbf{u} \, n_i ds \\
&= -\int_\Omega \partial_i \mathbf{v}^T \mathbf{A}_i \mathbf{u}_{\mathbf{v}} \delta \mathbf{v} \, d\Omega + \delta \left[\int_{\partial\Omega} F_i n_i ds \right] + \int_\Omega \partial_i \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \delta \mathbf{v} \, d\Omega - \int_{\partial\Omega} \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \delta \mathbf{v} \, n_i ds \\
&= -\int_\Omega \partial_i (\partial_j \mathbf{v}^T \widetilde{\mathbf{K}}_{ji}) \delta \mathbf{v} \, d\Omega + \delta \left[\int_{\partial\Omega} F_i n_i ds \right] + \int_\Omega \partial_i \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \delta \mathbf{v} \, d\Omega - \int_{\partial\Omega} \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \delta \mathbf{v} \, n_i ds \\
&= \delta \left[\int_{\partial\Omega} F_i n_i ds \right] + \int_\Omega (\partial_i \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \delta \mathbf{v} + \partial_i \delta \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \mathbf{v}) \, d\Omega - \int_{\partial\Omega} (\mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \delta \mathbf{v} + \delta \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \mathbf{v}) \, n_i ds \\
&= \delta \left[\int_{\partial\Omega} F_i n_i ds + \int_\Omega \partial_i \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \mathbf{v} \, d\Omega - \int_{\partial\Omega} \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \mathbf{v} \, n_i ds \right].
\end{aligned}$$

In the above derivation, the relationship in Eqn. 9 was used to convert the domain integral involving $\mathbf{A}_i$ to one involving $\widetilde{\mathbf{K}}_{ij}$. Thus, the entropy variables serve as an "adjoint" solution for the output

$$J = \int_{\partial\Omega} F_i n_i ds + \int_\Omega \partial_i \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \mathbf{v} \, d\Omega - \int_{\partial\Omega} \mathbf{v}^T \widetilde{\mathbf{K}}_{ij} \partial_j \mathbf{v} \, n_i ds. \tag{10}$$

We put quotes around the word adjoint because, as mentioned above, the entropy variables do not satisfy a differential equation that is strictly adjoint to Eqn. 8 in a mathematical sense. The presence of the integral over the domain in the output accounts for this difference.

The terms in Eqn. 10 have a physical meaning. The first term is the convective outflow of entropy across the domain boundary, the second term is the generation of entropy due to viscous dissipation within shear layers, vortices, or across shocks, and the last term is the entropy diffusion across the boundary.[21,22] In the next section, the entropy flux F_i will be defined for an entropy function U that is the negative of physical entropy. This means that, for the outward pointing normal n_i, the first term in Eqn. 10 is the net convective inflow of physical entropy into the computational domain, and the last term with the minus sign is the net diffusive inflow of physical entropy into the computational domain. Including the middle generation term means that J is zero for the analytical solution: the net outflow of physical entropy equals the physical entropy generated in the domain, in steady state. The terms in J balance each other in the analytical solution but not necessarily in an approximate numerical solution. Adapting on J using the entropy variables as adjoints therefore targets areas of spurious entropy production.

C. Choice of Entropy Function

Up to additive and multiplicative constants, only one choice of entropy function will yield entropy variables that symmetrize both the inviscid and viscous terms in the Navier-Stokes equations with heat-conduction included.[21] This choice corresponds to taking

$$U = -\rho S/(\gamma - 1), \quad S = \ln p - \gamma \ln \rho,$$

where p is the pressure, ρ is the density, S is the physical entropy, and γ is the ratio of specific heats. Differentiating with respect to $\mathbf{u}$ yields the entropy variables

$$\mathbf{v} = U_{\mathbf{u}}^T = \left[\frac{\gamma - S}{\gamma - 1} - \frac{1}{2}\frac{\rho V^2}{p},\ \frac{\rho u_i}{p},\ -\frac{\rho}{p}\right]^T, \tag{11}$$

where u_i are the velocity components, $V^2 = u_i u_i$, and $p = (\gamma - 1)(\rho E - \rho V^2/2)$, where E is the total energy per unit mass. Note that the entropy variables are obtained via a nonlinear transformation of the conservative variables. The corresponding entropy flux is $F_i = u_i U$.

IV. Numerical Implementation

The adaptive indicator in Eqn. 6 was implemented in a discontinuous Galerkin (DG) finite element code and was used to drive a fixed-fraction hanging-node mesh adaptation strategy. The DG discretization of the compressible Navier-Stokes equations employs the Roe approximate Riemann solver[23] for the inviscid fluxes and an interior penalty formulation[24,25] of the viscous fluxes. The steady-state solution is obtained via a Newton-GMRES implicit solver with line-Jacobi preconditioning and backward Euler with local time stepping for improved robustness during initial iterations. The DG discretization and solution method are similar to those used in existing works.[26,27] While a DG finite element method was used in this work, the idea of treating the entropy variables as adjoint solutions is applicable to general finite element and finite volume formulations.

A discrete adjoint solution for an engineering output of interest is obtained by solving Eqn. 3 with the same line-Jacobi preconditioned GMRES solver used for the primal solve. Careful attention was given to the various discretization and output calculation terms to ensure adjoint consistency.[28–30] The fine approximation space, $\mathcal{V}_h$, required for the adjoint solution $\boldsymbol{\psi}_h$ in Eqns. 5 and 6 is obtained by increasing the interpolation order from p to $p+1$ on the same mesh. To minimize sources of error in the method comparisons, both the primal and the adjoint problems are solved exactly on $\mathcal{V}_h$. For large problems, a full fine-space solve will be prohibitively expensive, and reconstruction or iterative approximation of the primal and adjoint solutions should be used. Experiments have shown that several smoothing iterations on $\mathcal{V}_h$ yield very similar adaptive results to an exact solve on $\mathcal{V}_h$, The error indicator on each element of the mesh is then calculated using Eqn. 6, which in a discrete setting reduces to an inner product between the discrete adjoint and residual vectors on the fine space, with absolute values around the contribution from each equation in the system. When the entropy-variables are used in place of $\boldsymbol{\psi}_h$, they are calculated from the fine-space solution $\mathbf{u}_h$ on each element by least-squares projection onto $\mathcal{V}_h$.

The elemental adaptive indicator, η_{κ_H}, drives a fixed-fraction hanging-node adaptation strategy. In this strategy, a certain fraction, f^{adapt}, of the elements with the largest adaptive indicators are marked for refinement. Marked elements are adapted uniformly, creating hanging nodes as illustrated in Figure 1. Note that additional elements may be flagged for refinement, uniform or in one direction, such that two neighboring elements differ by at most one level of refinement. No

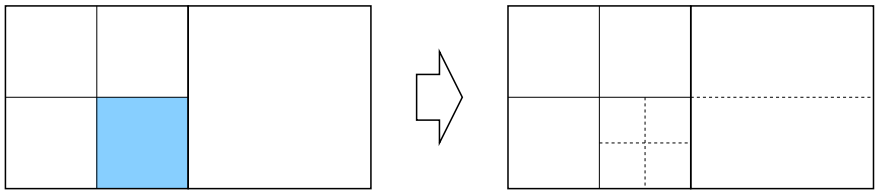


Figure 1. Hanging-node adaptation for a quadrilateral mesh, with a maximum of one level of refinement separating two elements. The shaded element on the left is marked for refinement, and the dashed lines on the right indicate the additional new edges formed.

element coarsening is performed in this study. The choices of a fixed fraction adaptation strategy and of hanging-node refinement were made to simplify the adaptive indicator comparisons. The steps in each adaptation iteration can be summarized as follows:

Adaptive Solution Steps

1. Solve the primal problem on the current mesh at order p to obtain $\mathbf{u}_H$. If adapting on an engineering output, solve the adjoint problem as well to obtain $\boldsymbol{\psi}_H$.
2. Inject $\mathbf{u}_H$ into an order $p+1$ space and either solve the primal problem exactly or iteratively smooth ν_{fine} times to obtain $\mathbf{u}_h$.
3. If adapting on an engineering output, solve or iterate the fine-space adjoint problem to obtain $\boldsymbol{\psi}_h$. Instead, if adapting using entropy variables, compute $\mathbf{v}_h(\mathbf{u}_h)$ using Eqn. 11.
4. Calculate the adaptive indicator, η_{κ_H}, for each element using Eqn. 6 with either $\boldsymbol{\psi}_h$ or $\mathbf{v}_h$.
5. Refine a fraction f^{adapt} of the elements with the largest indicator.
6. Inject the solution to the adapted mesh and return to step 1.

Note that no termination criterion is imposed in the adaptive solution, but rather a fixed number of iterations is used. Whereas an engineering output error can be driven below a certain user-prescribed tolerance, an allowable amount of spurious entropy generation is not straightforward to quantify. The formulation of a reasonable termination criterion for entropy-based adaptation is a topic for future work.

V. Results

This section presents results comparing adaptation using output adjoints versus adaptation using the entropy adjoint for three examples of aerodynamic interest. In all examples the geometry is a NACA 0012 airfoil with a closed trailing edge and a farfield at least 40 chord-lengths away. The initial mesh is illustrated in Figure 2. This mesh consists of quadrilaterals, with quartic, $q = 4$, curved elements representing the geometry. While the initial mesh appears structured, this structure disappears with the first adaptation iteration and the mesh storage is always fully unstructured. In all of the results, quadratic solution interpolation $p = 2$ was used in the discretization, and the adaptation fixed fraction was set to $f^{\text{adapt}} = 0.1$.

A. NACA 0012 $M_\infty = 0.4$, $\alpha = 5^o$

The first example consists of a NACA 0012 airfoil in inviscid flow at $M_\infty = 0.4$, $\alpha = 5^o$. The Mach number contours are illustrated in Figure 3a. Three different engineering outputs are considered:

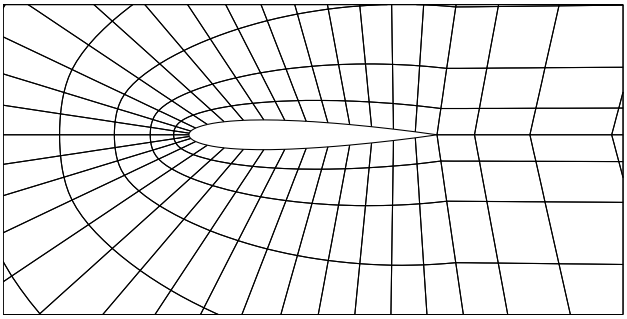
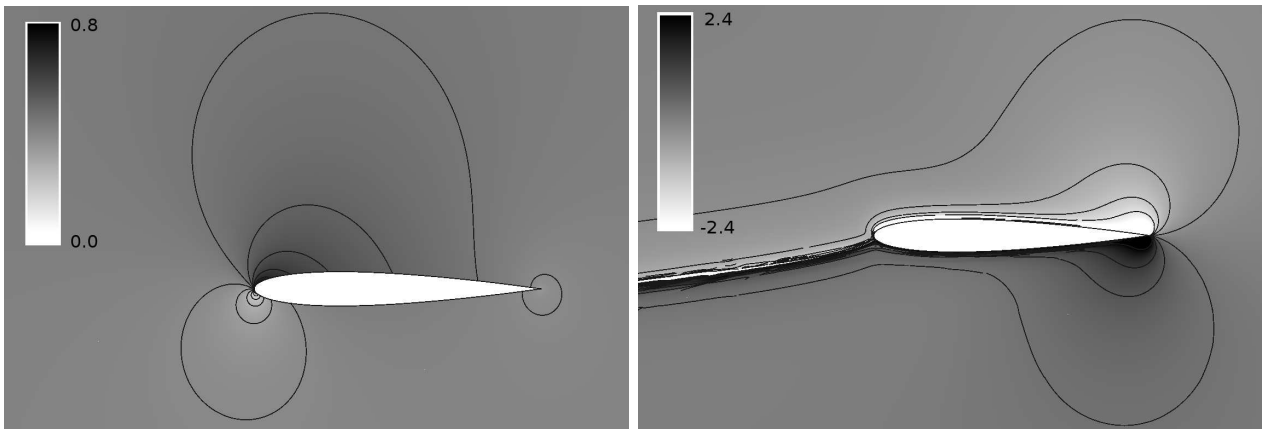


Figure 2. Initial 728-element mesh for a NACA 0012 airfoil, with quartic quadrilateral elements for geometry representation.



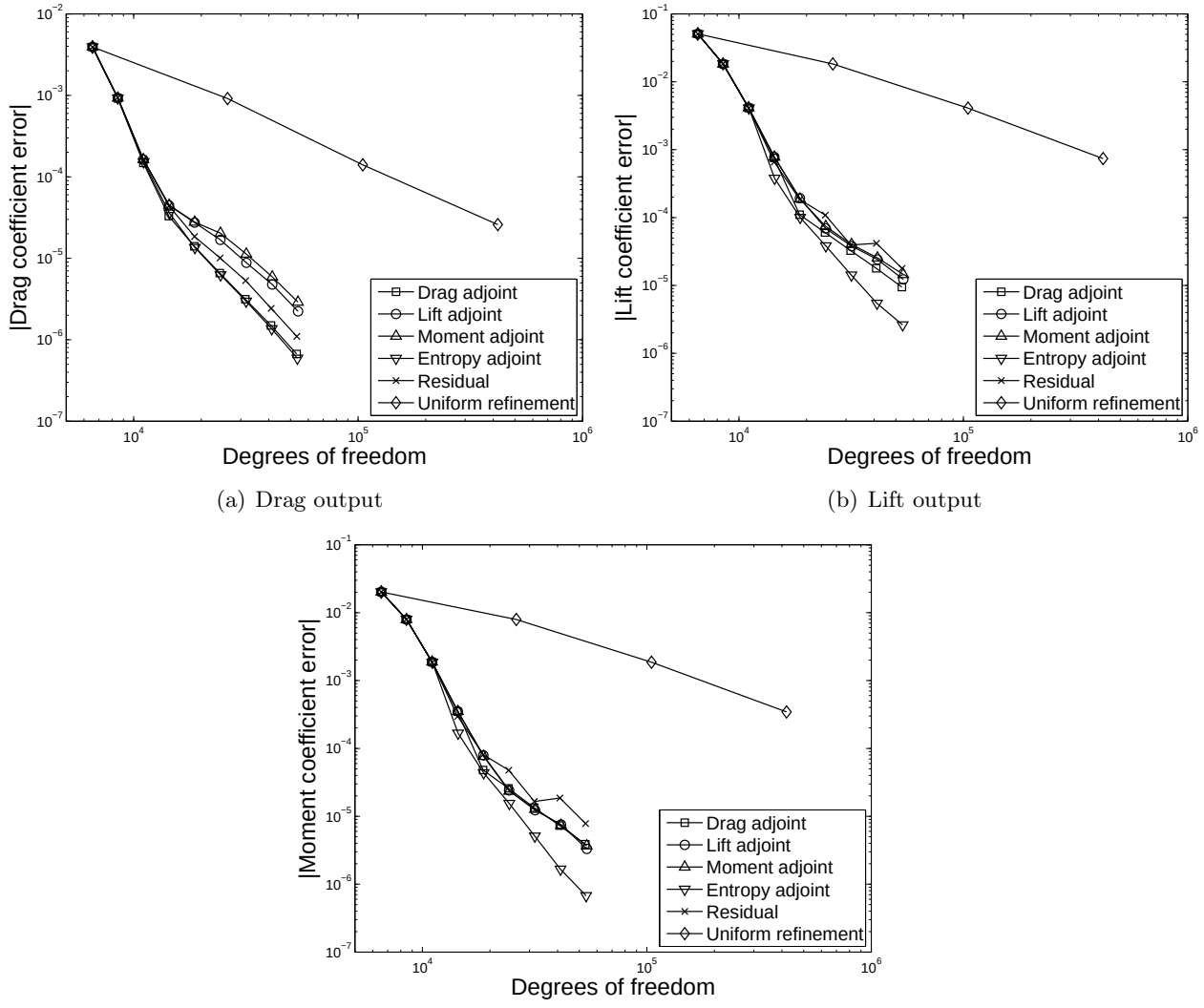
(a) Mach number contours

(b) x-momentum moment adjoint

Figure 3. NACA 0012, $M_\infty = 0.4$, $\alpha = 5^o$: Contours of the Mach number and the x-momentum component of the moment adjoint.

drag coefficient, lift coefficient, and leading-edge moment coefficient. All of these outputs were computed using integrals of the inviscid momentum flux, that is, the pressure, on the airfoil surface. Adjoint solutions associated with these outputs were used to drive three different adaptation runs. An adaptation run was also performed using the entropy adjoint indicator. For comparison, an unweighted residual indicator, equivalent to summing the absolute values of the discrete residuals on $\mathcal{V}_h$, was also tested.

Figure 4 shows the results of adaptation runs driven by the different indicators. Uniform mesh refinement results are given for comparison. The plots show the error in the engineering outputs versus degrees of freedom. Each "truth" output was obtained from a run at $p = 3$ on a mesh obtained by uniformly-refining the finest output-adapted mesh. For the three outputs of interest,



(a) Drag output

(b) Lift output

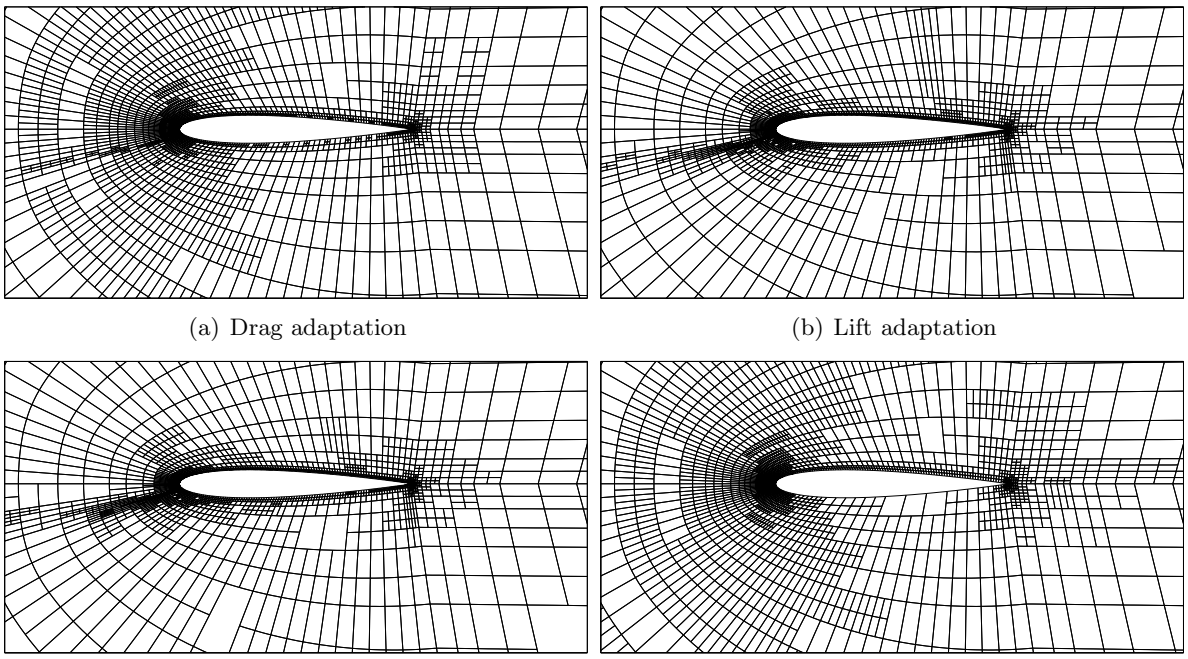
(c) Moment output

Figure 4. NACA 0012, $M_\infty = 0.4$, $\alpha = 5^o$: Comparison of output convergence histories for several adaptation strategies.

the entropy and adjoint-based adaptive strategies perform similarly and orders of magnitude better

than uniform mesh refinement. The entropy-variable based adaptation even shows improved performance compared to the targeted output-based adaptation in the lift and moment output cases. This result is surprising, but not actually paradoxical, because the procedure does have empirical elements. One possible explanation is that the lift and moment adaptive indicators excessively target the stagnation streamline in front of the airfoil, along which the adjoint solution possesses a singularity,[31] as shown in Figure 3b. The noise created by polynomial interpolation of the adjoint in this area may be responsible for the excessive refinement. On the other hand, the entropy variables remain smooth throughout the domain for this problem. In fact, they will always have the same smoothness as the primal solution.

The meshes after eight adaptation iterations of each strategy are shown in Figure 5. The leading edge, trailing edge, and upper surface of the airfoil are consistently targeted for refinement by all of the adaptation strategies. In the adjoint interpretation, these areas are important for the accurate prediction of the engineering outputs. In the entropy-variable interpretation, these areas are locations of largest spurious entropy generation. Refinement of the stagnation streamline is evident in the adjoint-based runs, especially for the lift and moment adaptations. In this high-lift case, the adjoint singularity is strongest for these outputs.



(a) Drag adaptation

(b) Lift adaptation

(c) Moment adaptation

(d) Entropy-variable adaptation

Figure 5. NACA 0012, $M_\infty = 0.4$, $\alpha = 5^o$: Meshes after eight adaptation iterations for several adaptation strategies.

B. NACA 0012 $M_\infty = 0.5$, $\alpha = 2^o$, $Re = 5000$

The second example consists of a NACA 0012 airfoil in viscous flow. The Mach number distribution is illustrated in Figure 6. The three engineering outputs of interest considered in this case are: drag coefficient, lift coefficient, and a wake "rake", taken as the integral of the x-momentum through the wake at the outflow boundary. As in the first example, adaptation runs were performed using

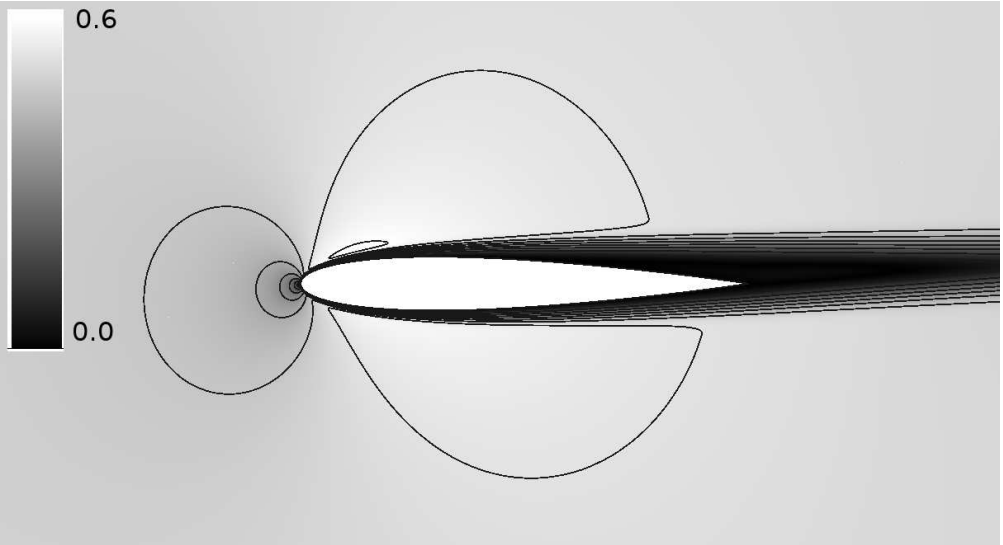


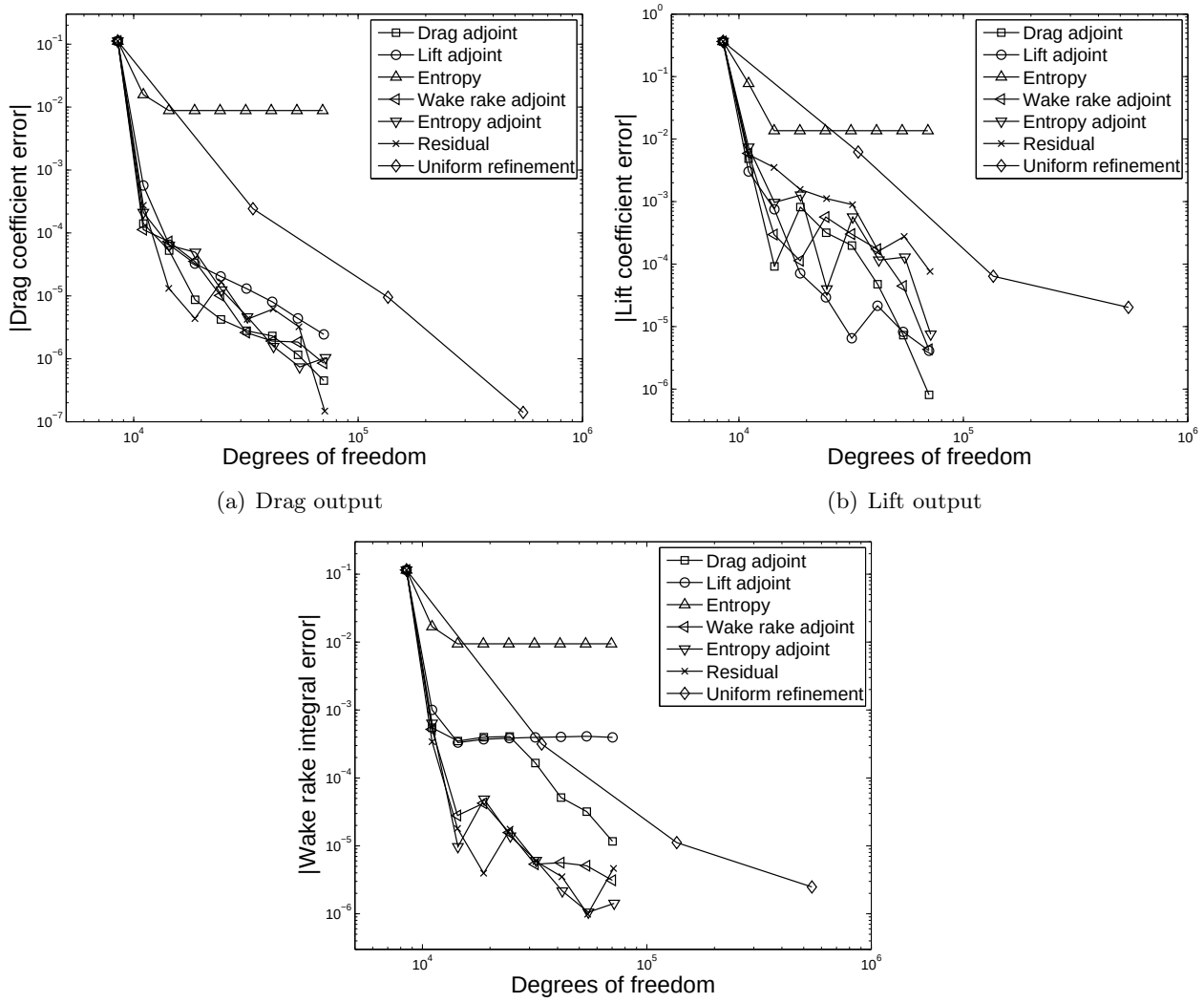
Figure 6. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^o$, $Re = 5000$: Mach number contours.

adjoints associated with each of these outputs, using the entropy adjoint, and using the unweighted residual.

Figure 7 shows the results of the adaptation runs for the different strategies, as well as for uniform mesh refinement. Each truth output was computed with a $p = 3$ solution on a uniform-refinement of the finest output-adapted mesh. The various adjoint strategies perform similarly for the drag and lift outputs. For the wake rake output, the drag and lift adjoint indicators do not sufficiently resolve the wake and hence do not perform very well. The unweighted residual performs well for the drag and wake rake outputs, but somewhat worse for the lift output. In general, among the adjoint methods, the entropy adjoint is generally one of the best two, except in the case of lift, where the results are somewhat erratic.

The meshes after eight adaptation iterations are shown in Figure 8. All strategies target the leading edge and sections of the upper and lower boundary layers. Lift-based adaptation targets the vicinity of the airfoil, leaving the wake relatively coarse, especially further downstream. On the other hand, entropy-based adaptation strikes a balance between adaptation near the airfoil surface and in the wake. Note that while entropy is created by viscous dissipation in the boundary layer and wake, this generation is already taken into account in the output corresponding to the entropy adjoint. The entropy adjoint adaptation introduced in this work targets areas of *spurious* entropy generation, not just areas of large entropy. If some region creates a lot of entropy, but does so correctly, then the output is not actually sensitive to this region.

To emphasize this point, we have made a study of refinement driven by an indicator based on the entropy scalar. This indicator is obtained by integrating the physical entropy minus the freestream entropy, over each element. As shown in Figure 7, this indicator does not perform well at all. The corresponding mesh is shown in Figure 8e: the entropy scalar indicator targets mainly the wake, while leaving the vicinity of the airfoil too coarse.

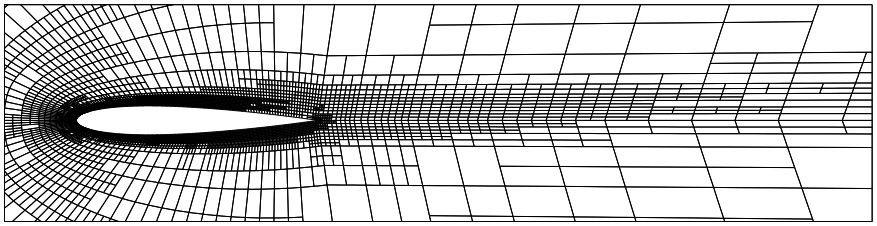


(a) Drag output

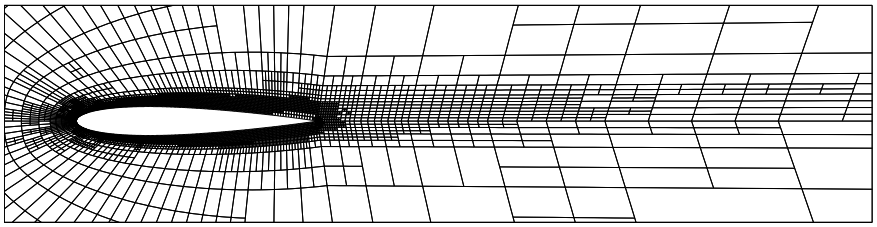
(b) Lift output

(c) Wake rake output

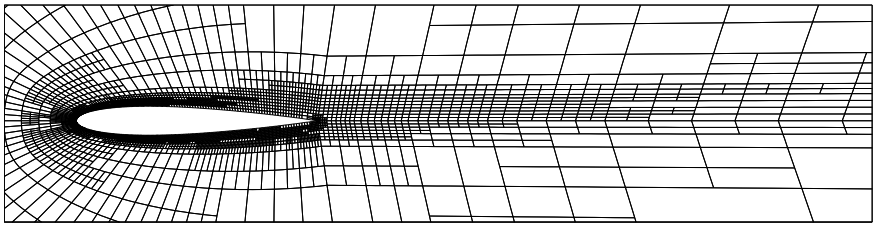
Figure 7. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^o$, $Re = 5000$: Comparison of output convergence histories for several adaptation strategies.



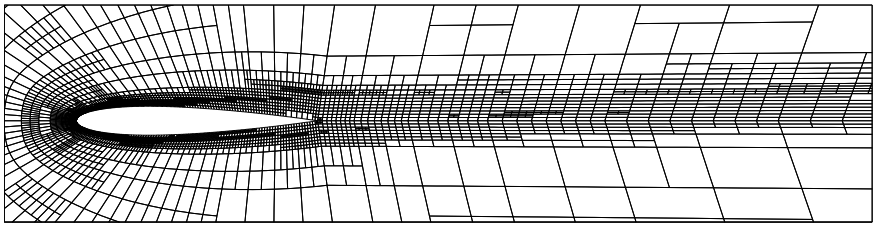
(a) Drag adaptation



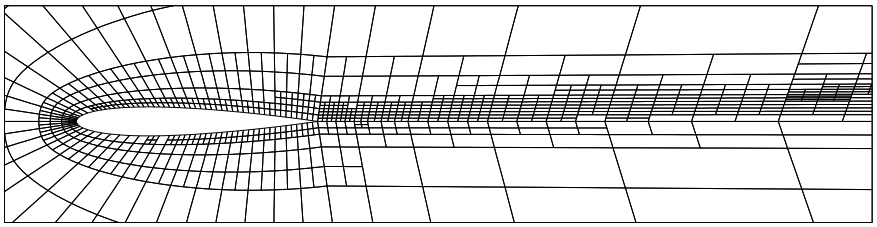
(b) Lift adaptation



(c) Wake rake adaptation



(d) Entropy adjoint adaptation



(e) Entropy adaptation

Figure 8. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^o$, $Re = 5000$: Meshes after eight adaptation iterations for several adaptation strategies. The entropy indicator flags many elements in the full extent of the wake, and hence the mesh in the vicinity of the airfoil appears sparser for the same number of adaptation iterations.

C. NACA 0012 in Inviscid Transonic Flow: $M_\infty = 0.8$, $\alpha = 1.25^o$

The third example consists of a NACA 0012 airfoil in inviscid transonic flow. The Mach number contours are shown in Figure 9 for an adapted solution. A resolution-based artificial viscosity is used to stabilize the solution in the presence of shocks.[32] Four adaptive indicators are compared: drag adjoint, lift adjoint, entropy adjoint, and the unweighted residual. Seven adaptation iterations are performed with $f^{\text{adapt}} = 0.1$.

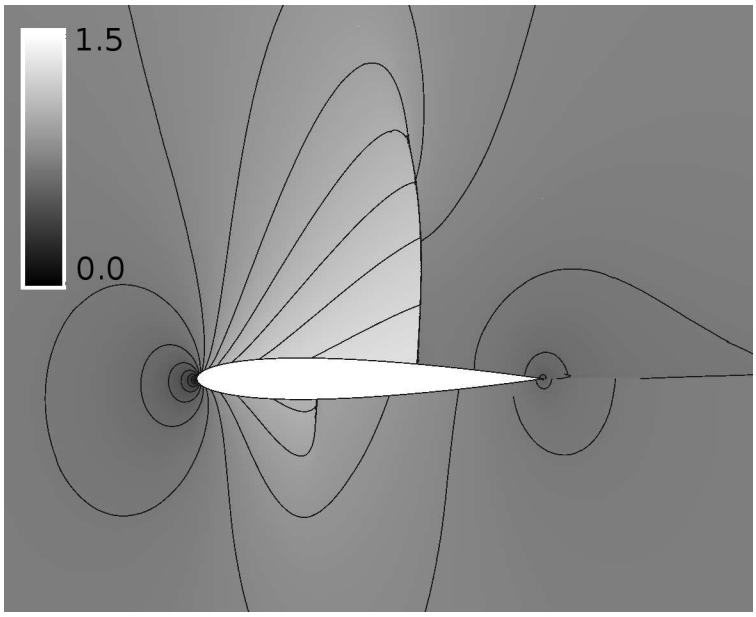
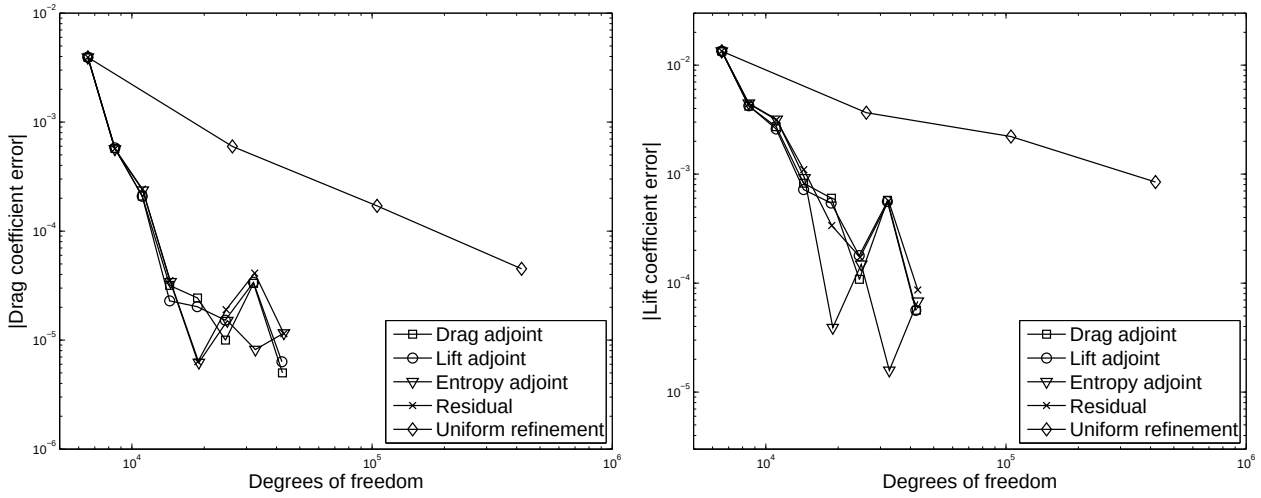


Figure 9. NACA 0012, $M_\infty = 0.8$, $\alpha = 1.25^o$: Mach number contours.

In an inviscid flow with shocks, entropy is no longer conserved, but is created abruptly at the shocks. Apparently, this does not bode well for the entropy adjoint indicator, which could target shocks for their seemingly spurious entropy production. That is, the residual of the inviscid entropy transport equation is nonzero at a shock, so that the entropy adjoint indicator will never be zero on elements straddling shocks, and therefore these elements will always be targeted for refinement. However, it can also be argued that the output function $J(\mathbf{u})$ will no longer be sensitive to mesh refinement once the shock is sufficiently refined, such that the jumps across it (including the jump in entropy) are sufficiently accurate. According to this interpretation the refinement at the shock will not be seen as more urgent than at other entropy-producing areas, such as under-resolved geometric features. Note that with the addition of physical viscosity in the Navier-Stokes equations, the refinement is certain to terminate, but only when the element size is on the order of the shock thickness, which is impractical for realistic flows.

The purpose of this example is to empirically test this trade-off for one flow, and one numerical implementation. Figure 10 shows the convergence of the lift and drag error using the various indicators and using uniform refinement. Truth values were computed using $p = 3$ interpolation on uniformly-refined versions of the final output-adapted meshes. The entropy adjoint performs very similarly to the output-based adjoints in both cases, although the scatter in the convergence histories precludes a definitive ranking. The final meshes are shown in Figure 11. The entropy adjoint indicator targets not only the shocks, but also the leading and trailing edges, although to

a lesser extent than the output adjoints. To test whether the entropy adjoint does over-refine the shock region, we count the number of cells with centroids lying inside the rectangular region outlined in Figure 11c. The resulting element counts are 2990 for the drag adjoint, 2997 for the lift adjoint, 2814 for the entropy adjoint, and 2372 for the unweighted residual. Note, the entropy adjoint targets the wake contact discontinuity more than the other indicators, which accounts for the fewer degrees of freedom in the vicinity of the airfoil compared to the output adjoints. The unweighted residual indicator also targets the contact discontinuity, and areas of the domain extending 10-12 chord lengths above and below the airfoil.



(a) Drag output

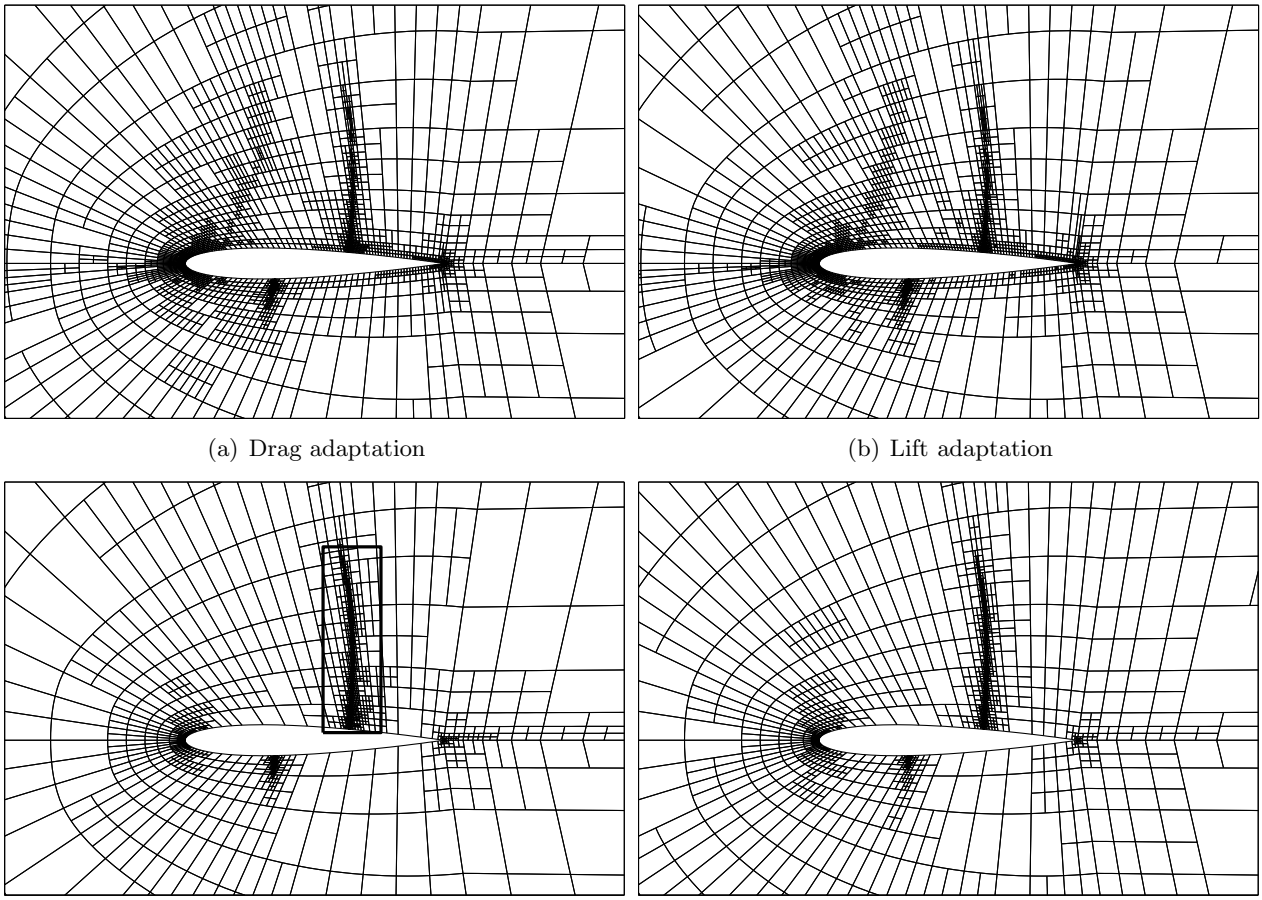
(b) Lift output

Figure 10. NACA 0012, $M_\infty = 0.8$, $\alpha = 1.25^o$: Comparison of output error convergence histories for various adaptation strategies.

At least for this particular transonic case, the performance of the entropy adjoint is very satisfactory. In other cases, involving stronger or more numerous shocks, we have found that over-refinement can occur in the shock region, especially when the output of interest lies outside the domain of influence of a significant portion of the shock. Accordingly, the analysis of the entropy adjoint indicator for flows with shocks is the subject of ongoing research.

VI. Conclusions

This paper introduces an adaptive indicator based on entropy variables. In the spirit of recent success with output-based adaptation, the entropy variables are related to adjoint solutions for an output that measures the net entropy production in the computational domain. In the inviscid case, the output consists of the entropy flux integral across the domain boundary, and the entropy variables satisfy the differential adjoint equation and the required boundary conditions. With the addition of viscosity, a domain-interior source term is included in the output, which accounts for the entropy generation present due to viscous dissipation. In this case, the entropy variables no longer satisfy the differential adjoint equation in a mathematical sense; however, the presence of a domain integral in the output accounts for this difference, so that the entropy variables can be used as adjoints in output error estimation. Hence, the same adaptive indicator developed for output-based error estimation applies when using entropy variables as the adjoints. Adaptive results for



(a) Drag adaptation

(b) Lift adaptation

(c) Entropy adjoint adaptation

(d) Residual adaptation

Figure 11. NACA 0012, $M_\infty = 0.8$, $\alpha = 1.25^o$: Meshes after seven adaptation iterations. Box in the entropy adjoint mesh indicates the region used for the element count.

the compressible Euler and Navier-Stokes equations demonstrate comparable performance between adaptation based on the entropy adjoint and that based on selected engineering output adjoints.

An advantage of the entropy adjoint adaptation is the simplicity due to the fact that an independent adjoint solution is not required: the entropy variables are obtained by a simple change of state variables. Solutions driven by the entropy adjoint can therefore be obtained at a computational savings of up to a factor of two over solutions driven by output adjoints, depending on the adjoint solver implementation. Future work will investigate applicability of this indicator to progressively more complicated simulations, such as those involving rotorcraft with extensive vortex shedding, to other equation sets with the same mathematical structure, such as ideal magnetohydrodynamics, as well as to unsteady flows, to which the entropy adjoint connection extends in a straightforward manner. In addition, the numerical implementation will be subject to further research, since at the level of detailed coding, there are some not unimportant decisions to be made that can have a significant effect on the adaptive indicator.

VII. Acknowledgements

The second author is grateful for hospitality provided by Dr. Nikos Nikiforakis at the Department of Applied Mathematics and Theoretical Physics, University of Cambridge UK, and the financial support of a William Penney Fellowship from the UK Ministry of Defence.

References

[1]Castro-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.

[2]Dompierre, 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.

[3]Hartmann, 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.

[4]Venditti, 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.

[5]Fidkowski, 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.

[6]Mavriplis, D. J., Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Levy, D., and Murayama, M., "Grid quality and resolution issues from the drag prediction workshop series," AIAA Paper 2008-930, 2008.

[7]Nemec, M., Aftosmis, M. J., and Wintzer, M., "Adjoint-based adaptive mesh refinement for complex geometries," AIAA Paper 2008-0725, 2008.

[8]Baker, T. J., "Mesh adaptation strategies for problems in fluid dynamics," *Finite Elements in Analysis and Design*, Vol. 25, 1997, pp. 243–273.

[9]Barth, T. J., "Numerical methods for gasdynamic systems on unstructured meshes," *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws, Berlin, Lecture Notes in Computational Science and Engineering*, edited by D. Kroner, M. Ohlberger, and C. Rhode, Springer-Verlag, 1999.

[10]Warren, G. P., Anderson, W. K., Thomas, J. L., and Krist, S. L., "Grid convergence for adaptive methods," AIAA Paper 1991-1592, 1991.

[11]Zhang, 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.

[12]Becker, 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.

[13]Giles, 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.
[14]Houston, 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.
[15]Fidkowski, K. J. and Darmofal, D. L., "Output-based error estimation and mesh adaptation: Overview and recent results," AIAA Paper 2009-1303, 2009.
[16]Rannacher, R., "Adaptive Galerkin finite element methods for partial differential equations," *Journal of Computational and Applied Mathematics*, Vol. 128, 2001, pp. 205–233.
[17]Barth, 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.
[18]Solín, P. and Demkowicz, L., "Goal-oriented hp-adaptivity for elliptic problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 193, 2004, pp. 449–468.
[19]Venditti, 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.
[20]Giles, 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.
[21]Hughes, T. J. R., Franca, L. P., and Mallet, M., "A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics." *Computer Methods in Applied Mechanics and Engineering*, Vol. 54, 1986, pp. 223–234.
[22]Tadmor, E. and Zhong, W., "Entropy stable approximations of Navier-Stokes equations with no artificial numerical viscosity," *Journal of Hyperbolic Differential Equations*, Vol. 3, No. 3, 2006, pp. 529–559.
[23]Roe, P. L., "Approximate Riemann solvers, parametric vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.
[24]Hartmann, R. and Houston, P., "An optimal order interior penalty discontinuous galerkin discretization of the compressible navier-stokes equations," *Journal of Computational Physics*, Vol. 227, 2008, pp. 9670–9685.
[25]Shahbazi, K., "A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 222, 2007, pp. 391–407.
[26]Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-multigrid solution of high–order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 207, 2005, pp. 92–113.
[27]Persson, P.-O. and Peraire, J., "Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations," *SIAM Journal for Scientific Computing*, Vol. 30, No. 6, 2008, pp. 2709–2733.
[28]Lu, 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.
[29]Hartmann, R., "Adjoint consistency analysis of discontinuous Galerkin discretizations," *SIAM Journal on Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696.
[30]Oliver, T. A., *A High–order, Adaptive, Discontinuous Galerkin Finite Elemenet Method for the Reynolds-Averaged Navier-Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.
[31]Giles, M. B. and Pierce, N. A., "Adjoint equations in CFD: duality, boundary conditions and solution behavior," AIAA Paper 97-1850, 1997.
[32]Persson, P.-O. and Peraire., J., "Sub-cell shock capturing for discontinuous Galerkin methods," AIAA Paper 2006-112, 2006.