

AN ENTROPY ADJOINT APPROACH TO MESH REFINEMENT*

KRZYSZTOF J. FIDKOWSKI[†] AND PHILIP L. ROE[‡]

Abstract. 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 adaptive indicator. The output associated with the entropy-variable adjoint is shown to be the entropy production in the computational domain, including physical viscous dissipation when present, minus entropy transport out of the domain. Adaptation using entropy variables, which is equivalent to adapting on the integrated residual of the entropy transport equation, thus targets areas of the domain responsible for numerical, or spurious, entropy production. Adaptive results for inviscid and viscous aerodynamic examples in two and three dimensions demonstrate performance efficiency on par with output-based adaptation, as measured by errors in various engineering quantities of interest, with the comparative advantage of the proposed approach that no adjoint equations need to be solved.

Key words. mesh adaptation, adjoint, entropy variables, discontinuous Galerkin

AMS subject classifications. 65N30, 65N50

DOI. 10.1137/090759057

1. 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 [6, 7, 10, 17, 23, 26, 38]. While adaptive methods can vary greatly in the mechanics by which degrees of freedom are added or removed from the computational mesh, they generally all rely on some form of indicator to drive the adaptation. 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.

Adaptive indicators may be described as heuristic if the theoretical link between the indicator and any measure of numerical error is tenuous or nonexistent. One popular class of heuristic indicators relies on solution features to dictate mesh refinement [2, 4, 39]. These features could be gradients, curvatures, or any other directly computable solution characteristics. While such indicators are cheap to evaluate and can yield visually pleasing results, they are not robust for controlling numerical error. In particular, feature-based adaptation schemes often perform poorly for hyperbolic problems when scalar outputs are of interest: they are liable to overrefine areas of the domain that do not affect the output and underrefine relatively featureless areas that are nevertheless important for the output calculation [38, 40].

*Received by the editors May 13, 2009; accepted for publication (in revised form) March 1, 2010; published electronically May 5, 2010.

<http://www.siam.org/journals/sisc/32-3/75905.html>

[†]Department of Aerospace Engineering, University of Michigan, 1320 Beal Avenue, 3029 FXB Building, Ann Arbor, MI 48109 (kfid@umich.edu).

[‡]Department of Aerospace Engineering, University of Michigan, 1320 Beal Avenue, 3021 FXB Building, Ann Arbor, MI 48109 (philroe@umich.edu). Current address: DAMTP, Centre for Mathematical Sciences, Wilberforce Road, Cambridge CB3 0WA, UK. This author’s research was supported by a William Penney Fellowship from the UK Ministry of Defence.

To address these shortcomings, output-based adaptation methods have been developed and demonstrated for practical CFD computations [5, 9, 13, 17, 26, 38]. The main idea behind these methods is to specifically target for refinement areas of the computational domain that are important for predicting the output quantity of interest. Output-based adaptation thus properly accounts for the propagation effects that are intrinsic to hyperbolic problems [19]. A critical requirement in these methods is an output-specific adjoint solution, which is used for its role as the sensitivity of the output to local residuals. Adaptive solution techniques based on output error estimates provide efficient meshes for prediction of the output in question. 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 state variables by a simple transformation, are shown to be equivalent to an 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 are not normally considered to be 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 can be thought of as defining “features” to be those areas where entropy production is difficult to compute.

The final form of the resulting adaptive indicator is not completely novel. Analysis in section 4.4 shows that the error indicator derived from the entropy variables is equivalent to an integrated residual of the entropy transport equation, with a prescribed procedure for calculating this residual from any given approximate solution. This entropy residual has been investigated for error estimation and adaptation in previous works. In [8], energy norm estimates are derived for the Navier–Stokes equations by using symmetrizing entropy variables. In [1], a cell-average approximation of the entropy residual, motivated in part by a desire to combine the residuals of the original conservation laws, is used in a cell vertex finite volume method to drive mesh movement. In [25], a weighted entropy residual is related to the truncation error of the conservative equations in a finite element discretization of the incompressible Euler equations. A general review of the use of entropy and the second law in CFD is presented in [24].

Novel aspects of the present work include the interpretation of the entropy variables as adjoint solutions; the formal calculation of the corresponding output quantities in inviscid and viscous flows; and a comparison of the performance of the resulting indicator to existing adjoint and feature-based indicators for flows of engineering interest in two and three dimensions. The outline of the remainder of this paper is as follows. Section 2 reviews properties of continuous adjoint solutions, and section 3 discusses the use of adjoint solutions for output error estimation and mesh adaptation. In section 4 the entropy variables are defined and shown to satisfy an adjoint equation. The corresponding output is derived for inviscid and viscous conservation laws. Section 5 discusses the implementation of the proposed method in a finite element adaptive strategy, which is then used to generate the results in section 6.

2. Continuous 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 intrinsic to hyperbolic problems, and they can be used to ascribe confidence levels to outputs or to drive adaptation. As mentioned in the introduction, a key component of output error estimation is the solution of an adjoint equation for the output of interest. In a continuous setting, an adjoint is a Green’s function that relates residual source perturbations in the original equation to an output of interest. This definition can be used to derive the adjoint equations and to formulate an output error estimation framework [11]. In this work, we take a more formal approach by employing a Lagrange multiplier definition of the adjoint. This approach allows for a more rigorous discussion of the adjoint equations and boundary conditions, which are crucial to the entropy variable exposition in section 4.

Consider a primal differential equation arising from a conservation law,

$$(2.1) \quad \mathbf{r}(\mathbf{u}) = \partial_i \mathcal{F}_i = 0 \quad \text{on } \Omega,$$

where Ω is the computational domain, $\mathbf{u} \in \mathcal{V}$ is a state vector, \mathcal{V} is an appropriate function space, i indexes the spatial dimension, and summation is implied on the repeated index. \mathcal{F}_i is a general flux that may consist of convective and diffusive components. Suitable boundary conditions are assumed to be specified on the domain boundary, $\partial\Omega$.

Given a scalar output $J(\mathbf{u})$, we define a Lagrangian as

$$(2.2) \quad \mathcal{L} = J(\mathbf{u}) - \int_{\Omega} \boldsymbol{\psi}^T \mathbf{r}(\mathbf{u}) d\Omega,$$

where the Lagrange multiplier, $\boldsymbol{\psi} \in \mathcal{V}$, is the adjoint solution [5, 22]. Enforcing stationarity of \mathcal{L} with respect to permissible variations in the state yields the adjoint equation

$$(2.3) \quad J'[\mathbf{u}](\delta\mathbf{u}) - \int_{\Omega} \boldsymbol{\psi}^T \mathbf{r}'[\mathbf{u}](\delta\mathbf{u}) d\Omega = 0 \quad \forall \delta\mathbf{u} \in \mathcal{V}^{\text{perm}},$$

where $\mathcal{V}^{\text{perm}}$ denotes the space of permissible state variations, that is, those allowed by the boundary conditions. The primes above denote Fréchet linearization with respect to the arguments in square brackets. Using the conservation form in (2.1), (2.2) can be integrated by parts and linearized to yield an interior differential equation and boundary conditions for the adjoint:

$$(2.4) \quad J'[\mathbf{u}](\delta\mathbf{u}) + \int_{\Omega} \partial_i \boldsymbol{\psi}^T \mathcal{F}'_i[\mathbf{u}](\delta\mathbf{u}) d\Omega - \int_{\partial\Omega} \boldsymbol{\psi}^T \mathcal{F}'_i[\mathbf{u}](\delta\mathbf{u}) n_i ds = 0.$$

For example, if $J(\mathbf{u})$ consists of an integral along a subset of $\partial\Omega$, then the domain integral term above yields the interior adjoint equation

$$(2.5) \quad (\mathcal{F}'_i[\mathbf{u}])^T \partial_i \boldsymbol{\psi} = 0,$$

while the remaining terms impose boundary conditions on $\boldsymbol{\psi}$:

$$J'[\mathbf{u}](\delta\mathbf{u}) - \int_{\partial\Omega} \boldsymbol{\psi}^T \mathcal{F}'_i[\mathbf{u}](\delta\mathbf{u}) n_i ds = 0 \quad \forall \delta\mathbf{u} \in \mathcal{V}^{\text{perm}}.$$

For an arbitrary output $J(\mathbf{u})$, variations in the state permitted by the boundary conditions constrain $\boldsymbol{\psi}$, whereas components of the state fixed by the boundary conditions do not constrain $\boldsymbol{\psi}$ [22]. This observation suggests a duality in the boundary conditions for the primal and adjoint solutions: where \mathbf{u} is unconstrained, $\boldsymbol{\psi}$ must be fully specified. However, such a duality is not necessary if $J(\mathbf{u})$ is chosen such that the two terms in the above equations cancel for all permissible $\delta\mathbf{u}$. This point is key to the entropy-variable analysis in section 4.

3. Output error estimation. An adjoint solution can be used to estimate the numerical error in the corresponding output of interest. The error estimation process, termed the adjoint-weighted residual method, relies on the following key observations:

- An approximate solution \mathbf{u}_H in a finite-dimensional space \mathcal{V}_H will generally not satisfy the original PDE: $\mathbf{r}(\mathbf{u}_H) \neq 0$. If $\delta\mathbf{u} \equiv \mathbf{u}_H - \mathbf{u}$ is small, we can write

$$\mathbf{r}(\mathbf{u}_H) = \mathbf{r}(\mathbf{u} + \delta\mathbf{u}) \approx \mathbf{r}'[\mathbf{u}](\delta\mathbf{u}),$$

where $\mathbf{r}(\mathbf{u}) = 0$ was used, and terms high-order in $\delta\mathbf{u}$ were dropped.

- The adjoint $\boldsymbol{\psi}$ translates the residual perturbation to an output perturbation via the Lagrangian stationarity property in (2.3):

$$(3.1) \quad \delta J \approx J'[\mathbf{u}](\delta\mathbf{u}) = \int_{\Omega} \boldsymbol{\psi}^T \mathbf{r}'[\mathbf{u}](\delta\mathbf{u}) \approx \int_{\Omega} \boldsymbol{\psi}^T \mathbf{r}(\mathbf{u}_H).$$

This expression quantifies the numerical error in the output via a weighted residual of the approximate solution. The approximation signs indicate that for noninfinitesimal perturbations the above expression is not exact and yields only an estimate of the numerical error. Note that if mean-value linearizations are used in the adjoint equation and in the output linearization, which is not the case in this work, then the approximation signs become equalities [3, 5, 17, 30].

The continuous adjoint $\boldsymbol{\psi}$ must be approximated to make the error estimate in (3.1) computable. In practice, a discrete version of the adjoint equations is solved approximately or exactly on a finer finite-dimensional space, $\mathcal{V}_h \supset \mathcal{V}_H$, to yield $\boldsymbol{\psi}_h \in \mathcal{V}_h$ [3, 31, 35]. This finer space can be obtained either through mesh subdivision or approximation order increase. The procedure for obtaining a consistent set of discrete adjoint equations is described in [16, 22, 27].

Working with $\boldsymbol{\psi}_h \in \mathcal{V}_h$ in a variational formulation, (3.1) becomes

$$(3.2) \quad \delta J \approx \mathcal{R}_h(\mathbf{u}_H, \boldsymbol{\psi}_h),$$

where $\mathcal{R}_h(\cdot, \cdot) : \mathcal{V}_h \times \mathcal{V}_h \rightarrow \mathbb{R}$ is a semilinear form associated with the weak form of the differential equation in (2.1). This adjoint-weighted residual evaluation can be localized to yield an adaptive indicator consisting of the relative contribution of each element to the total output error. The right-hand side of (3.2) can be written as a sum over elements,

$$(3.3) \quad \delta J \approx \sum_{\kappa_H \in T_H} \sum_{\kappa_h \in \kappa_H} \mathcal{R}_h(\mathbf{u}_H, \boldsymbol{\psi}_h|_{\kappa_h}),$$

where T_H is the mesh triangulation associated with \mathcal{V}_H , T_h is the triangulation associated with \mathcal{V}_h , κ_H and κ_h are elements of the coarse and fine triangulations, respectively, and $|_{\kappa_h}$ refers to restriction to element κ_h . Note that the coarse and fine

spaces can consist of the same triangulations, in which case $\kappa_H = \kappa_h$. Equation (3.3) 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 (3.3) [3, 5, 15, 17, 38]:

$$(3.4) \quad \eta_{\kappa_H} = \left| \sum_{\kappa_h \in \kappa_H} \mathcal{R}_h(\mathbf{u}_H, \psi_h|_{\kappa_h}) \right|.$$

For nonvariational discretizations, ψ_h in the above equation is replaced with $\psi_h - \psi_H$, where ψ_H is an adjoint solution computed with the coarse discretization, in order to target the remaining error in the output [11]. The two approaches are identical in a variational formulation with local Galerkin orthogonality, as is the case for discontinuous Galerkin methods. When the orthogonality is not local, the substitution of $\psi_h - \psi_H$ cannot be made in (3.4) but instead must be introduced in (3.2). For systems of equations the authors have found slightly improved adaptation robustness by additionally placing absolute values around contributions from each equation in the inner product in (3.4). Due to the absolute values in (3.4), the sum of the indicators, $\sum_{\kappa_H} \eta_{\kappa_H}$, is greater than 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.

4. Adaptation using entropy variables. The adaptive indicator derived in the previous section is specific to a user-defined scalar output of interest 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. In addition, the final form of the error indicator is shown to be equivalent to an integrated residual of the entropy transport equation.

4.1. First-order conservation laws. Consider a steady-state set of first-order conservation laws, in which $\mathcal{F}_i = \mathbf{f}_i(\mathbf{u})$, together with a scalar entropy conservation law

$$(4.1) \quad \mathbf{r}(\mathbf{u}) = \partial_i \mathbf{f}_i = 0, \quad \partial_i F_i = 0,$$

where $F_i(\mathbf{u})$ 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, where each $\mathbf{A}_i = \mathbf{f}_{i,\mathbf{u}}(\mathbf{u})$ is a flux Jacobian matrix. 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 [4, 20] the following hold:

- 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 conservation law in (4.1) can be linearized about \mathbf{u} and manipulated as follows:

$$(4.2) \quad \begin{aligned} 0 &= \mathbf{A}_i \partial_i \mathbf{u} = \mathbf{A}_i \mathbf{u}_{\mathbf{v}} \partial_i \mathbf{v} = (\mathbf{A}_i \mathbf{u}_{\mathbf{v}})^T \partial_i \mathbf{v} = \mathbf{u}_{\mathbf{v}}^T \mathbf{A}_i^T \partial_i \mathbf{v} = \mathbf{u}_{\mathbf{v}} \mathbf{A}_i^T \partial_i \mathbf{v} \\ &\Rightarrow \mathbf{A}_i^T \partial_i \mathbf{v} = 0. \end{aligned}$$

Comparing the above expression to (2.5), with $\mathbf{A}_i = \mathcal{F}'_i[\mathbf{u}]$, suggests that the entropy variables satisfy an adjoint equation for an output that has no domain integral contribution. To verify the validity of this assertion and to determine the associated output

functional, we substitute the entropy variables in place of ψ in the adjoint equation, (2.4), to obtain

$$J'[\mathbf{u}](\delta\mathbf{u}) + \int_{\Omega} \partial_i \mathbf{v}^T \mathbf{A}_i \delta\mathbf{u} d\Omega - \int_{\partial\Omega} \mathbf{v}^T \mathbf{A}_i \delta\mathbf{u} n_i ds = 0.$$

The domain integral drops out due to (4.2), leaving a relationship between the output and a domain boundary integral. Solving for the output perturbation and employing the definition of the entropy flux, we have

$$(4.3) \quad \delta J = J'[\mathbf{u}](\delta\mathbf{u}) = \int_{\partial\Omega} \underbrace{\mathbf{v}^T \mathbf{A}_i}_{(F_i)_{\mathbf{u}}} \delta\mathbf{u} n_i ds = \delta \left[\underbrace{\int_{\partial\Omega} F_i n_i ds}_J \right].$$

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 (4.2). By a suitable definition of the output as an entropy flux integral over the domain boundary, the adjoint boundary conditions are then automatically satisfied; that is, the Lagrangian in (2.2), with $\psi = \mathbf{v}$, is stationary with respect to permissible variations in the state.

Replacing the adjoint in (3.4) with the entropy variables yields an adaptive indicator that corresponds to the output $J = \int_{\partial\Omega} F_i n_i ds$. For an inviscid, shock-free flow, this integral should of course vanish. If it fails to vanish in a discrete solution, then the indicator will identify those flow regions responsible for the failure. Typically, these are found at the leading and trailing edges of an airfoil, and of course at shockwaves, although we examine the latter only briefly in the results.

4.2. Second-order conservation laws. A canonical set of viscous conservation laws reads as

$$(4.4) \quad \mathbf{r}(\mathbf{u}) = \partial_i \mathbf{f}_i - \partial_i (\mathbf{K}_{ij} \partial_j \mathbf{u}) = 0,$$

where $\mathbf{K}_{ij} \partial_j \mathbf{u}$ is the viscous flux, so that $\mathcal{F}_i = \mathbf{f}_i - \mathbf{K}_{ij} \partial_j \mathbf{u}$ is the total 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 $\tilde{\mathbf{K}}_{ij} = \tilde{\mathbf{K}}_{ji}^T$, where $\tilde{\mathbf{K}}_{ij} \equiv \mathbf{K}_{ij} \mathbf{u}_{\mathbf{v}}$ [20]. Substituting $\partial_i \mathbf{u} = \mathbf{u}_{\mathbf{v}} \partial_i \mathbf{v}$ into (4.4) and taking the transpose yields a useful relationship:

$$(4.5) \quad \partial_i \mathbf{v}^T \mathbf{A}_i \mathbf{u}_{\mathbf{v}} - \partial_i (\partial_j \mathbf{v}^T \tilde{\mathbf{K}}_{ji}) = 0.$$

By contrast to the inviscid case, this is no longer a mathematical adjoint to the primal equation, (4.4); 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 solely over the domain boundary. To demonstrate this statement, we again use \mathbf{v} in place of ψ in the adjoint equation, (2.4), to obtain

$$\begin{aligned} \delta J = J'[\mathbf{u}](\delta\mathbf{u}) &= - \int_{\Omega} \partial_i \mathbf{v}^T [\mathbf{A}_i \delta\mathbf{u} - \mathbf{K}_{ij,\mathbf{u}}(\delta\mathbf{u}) \partial_j \mathbf{u} - \mathbf{K}_{ij} \partial_j (\delta\mathbf{u})] d\Omega \\ &\quad + \int_{\partial\Omega} \mathbf{v}^T [\mathbf{A}_i \delta\mathbf{u} - \mathbf{K}_{ij,\mathbf{u}}(\delta\mathbf{u}) \partial_j \mathbf{u} - \mathbf{K}_{ij} \partial_j (\delta\mathbf{u})] n_i ds, \end{aligned}$$

where we have used

$$\mathcal{F}'_i[\mathbf{u}](\delta\mathbf{u}) = \mathbf{A}_i\delta\mathbf{u} - \mathbf{K}_{ij,\mathbf{u}}(\delta\mathbf{u})\partial_j\mathbf{u} - \mathbf{K}_{ij}\partial_j(\delta\mathbf{u}).$$

Using $\delta\mathbf{u} = \mathbf{u}_\nu\delta\mathbf{v}$, applying (4.5) to the first term in the domain integral, and applying $\mathbf{v}^T\mathbf{A}_i = (F_i)_\mathbf{u}$ to the first term in the boundary integral yields

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

The first term integrated by parts is

$$\begin{aligned} - \int_{\Omega} \partial_i(\partial_j\mathbf{v}^T\tilde{\mathbf{K}}_{ji})\delta\mathbf{v} d\Omega &= \int_{\Omega} \partial_j\mathbf{v}^T\tilde{\mathbf{K}}_{ji}\partial_i(\delta\mathbf{v})d\Omega - \int_{\partial\Omega} \partial_j\mathbf{v}^T\tilde{\mathbf{K}}_{ji}\delta\mathbf{v}n_i ds \\ &= \int_{\Omega} \partial_i(\delta\mathbf{v}^T)\mathbf{K}_{ij}\partial_j\mathbf{u}d\Omega - \int_{\partial\Omega} \delta\mathbf{v}^T\mathbf{K}_{ij}\partial_j\mathbf{u}n_i ds, \end{aligned}$$

where we have used $(\partial_j\mathbf{v}^T\tilde{\mathbf{K}}_{ji})^T = \tilde{\mathbf{K}}_{ij}\partial_j\mathbf{v} = \mathbf{K}_{ij}\mathbf{u}_\nu\partial_j\mathbf{v} = \mathbf{K}_{ij}\partial_j\mathbf{u}$. Substituting into the expression for δJ and grouping terms,

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

Thus, the entropy variables serve as an ‘‘adjoint’’ solution for the output:

$$(4.6) \quad J = \int_{\partial\Omega} F_i n_i ds + \int_{\Omega} \partial_i\mathbf{v}^T\mathbf{K}_{ij}\partial_j\mathbf{u} d\Omega - \int_{\partial\Omega} \mathbf{v}^T\mathbf{K}_{ij}\partial_j\mathbf{u} n_i ds.$$

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 (4.4) in a mathematical sense. The presence of the integral over the domain in the output accounts for this difference. With this output definition, the entropy variables fulfill their role as Lagrange multipliers that yield a stationary Lagrangian with respect to permissible variations in the state. Hence, they can be used in the same capacity as other adjoint solutions for output error estimation.

Furthermore, the terms in (4.6) 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 and vortices or across shocks, and the last term is the entropy diffusion across the boundary [20, 37]. 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 (4.6) 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 at 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.

4.3. 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 [20]. 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

$$(4.7) \quad \mathbf{v} = U_{\mathbf{u}}^T = \left[\frac{\gamma - S}{\gamma - 1} - \frac{1}{2} \frac{\rho V^2}{p}, \frac{\rho u_i}{p}, -\frac{\rho}{p} \right]^T,$$

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 $F_i = u_i U$.

4.4. Relation to the entropy residual. With the above choice of entropy variables, an entropy transport equation can be derived from the Navier–Stokes equations [20, 37]

$$(4.8) \quad \underbrace{\partial_i F_i + \partial_i \mathbf{v}^T \mathbf{K}_{ij} \partial_j \mathbf{u} - \partial_i (\mathbf{v}^T \mathbf{K}_{ij} \partial_j \mathbf{u})}_{\mathbf{v}^T \mathbf{r}(\mathbf{u})} = 0,$$

where the entropy flux, F_i , and the viscous coefficient matrix, \mathbf{K}_{ij} , are as defined earlier in this section. For an approximate solution, an entropy residual can be defined as the left-hand side of the above equation. This residual contains information on how well the entropy transport equation is satisfied. As indicated by the underbrace, taking the inner product of the primal residual in (4.4) with the entropy variables yields the left-hand side of (4.8). Therefore, the entropy-variable weighted primal residual used for the output error estimate in the previous section, for example in (4.6), is precisely the entropy residual. The adaptive indicator thus targets areas of the computational domain where the entropy residual is high, that is, where entropy transport is not predicted well.

We note that the calculation of the entropy residual for approximate finite-dimensional solutions and the resulting formation of an adaptive indicator are not unique. The nonlinear transformation from conservative variables, $\mathbf{u}_H \in \mathcal{V}_H$, to entropy variables, $\mathbf{v}_H \in \mathcal{V}_H$, can be done first, with the result substituted into (4.8) and evaluated on a finer space, \mathcal{V}_h ; alternately the conservative residuals evaluated on the finer space can be combined using the entropy variables calculated from either \mathbf{u}_H or a finer approximation, $\mathbf{u}_h \in \mathcal{V}_h$. The resulting residual can then be measured in a pointwise, average, or integrated sense in forming the adaptive indicator. The analogy between entropy variables and adjoint solutions presented in this work provides a formalism for these choices in the context of an output-based error estimation

framework, in particular dictating how the indicator should be formed to be consistent with the net entropy production output.

5. Numerical implementation. The adaptive indicator in (3.4) 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 [32] for the inviscid fluxes and an interior penalty formulation of the viscous fluxes [18, 34]. 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. This DG discretization and solution method are similar to those used in existing works [12, 29]. 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 the weak form of (2.3). The same line-Jacobi preconditioned GMRES solver was used for the adjoint solve in this work. Careful attention was given to the various discretization and output calculation terms to ensure adjoint consistency [16, 22, 27].

The fine approximation space, \mathcal{V}_h , required for the adjoint solution ψ_h in (3.3) and (3.4) is obtained by increasing the approximation 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 the two-dimensional results. For the three-dimensional runs, a full fine-space solve is prohibitively expensive, and the exact solves are replaced by ν_{fine} iterations of a block-Jacobi smoother on \mathcal{V}_h . Experiments have shown that several smoothing iterations on \mathcal{V}_h yield adaptive results very similar to an exact solve on \mathcal{V}_h , which is to be expected as the perturbation estimate in (3.2) is insensitive to adjoint errors lying in the coarse space, \mathcal{V}_H [11]. In this work $\nu_{\text{fine}} = 5$ is used for both the primal and the adjoint problems. The error indicator on each element of the mesh is then calculated using (3.4), which in a discrete setting reduces to an inner product between the discrete adjoint and residual vectors on the fine space, with absolute values as described in section 3.

When the entropy variables are used in place of ψ_h , they are calculated from the fine-space solution \mathbf{u}_h on each element by least-squares projection in \mathcal{V}_h . Alternatively, the entropy variable values derived from \mathbf{u}_h at element quadrature points can be used directly in the evaluation of the semilinear form in (3.4). This latter approach, which was not used in the present work, may prove more robust in the presence of solution discontinuities. We note that a nonzero indicator can also be obtained by calculating the fine-space entropy variables, $\mathbf{v}_h \in \mathcal{V}_h$, directly from $\mathbf{u}_H \in \mathcal{V}_H$, at significantly reduced cost. However, this indicator is nonzero only due to the nonlinear variable transformation, and experiments by the authors have shown that its performance is not robust.

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 is marked for refinement. Marked elements are adapted uniformly, creating hanging nodes as illustrated in Figure 5.1 for two dimensions. 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 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

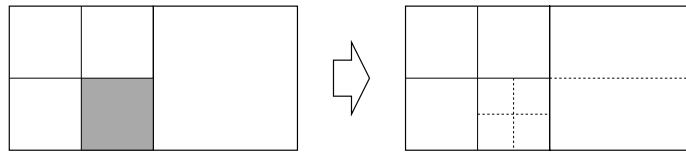


FIG. 5.1. *Hanging-node adaptation for a quadrilateral mesh, with a maximum of one level of refinement separating two elements. The shaded element on the left is marked for refinement, and the dashed lines on the right indicate the additional new edges formed.*

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 to obtain ψ_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 ψ_h . Instead, if adapting using entropy variables, compute $\mathbf{v}_h(\mathbf{u}_h)$ using (4.7).
4. Calculate the adaptive indicator, η_{κ_H} , for each element using (3.4) with either ψ_h or \mathbf{v}_h .
5. Refine a fraction f^{adapt} of the elements with the largest indicator.
6. Initialize the solution on the adapted mesh with a projection of \mathbf{u}_H 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.

6. Results.

6.1. Entropy variables in flow over a Gaussian bump. To demonstrate the equivalence between entropy variables and output adjoints, an entropy flux output is constructed for an inviscid flow calculation in a duct with a Gaussian bump perturbation. The duct is discretized with 320 quartic, $q = 4$, curved triangular elements, as shown in Figure 6.1(a). The freestream Mach number is 0.5, and the solution is approximated with cubic, $p = 3$, polynomials. To increase the complexity of the flow, a nonuniform inflow entropy is prescribed by a linearly varying total temperature field. The corresponding entropy contours, which are of course simply the streamlines, are shown in Figure 6.1(b).

An integral entropy flux output is defined as

$$(6.1) \quad J = \int_{\partial\Omega} F_i(\mathbf{u}_h^b) n_i ds,$$

where \mathbf{u}_h^b is the boundary state, $F_i(\mathbf{u}_h^b)$ is the entropy flux, and the integral is taken over the entire domain boundary. This output corresponds to the perturbation derived in (4.3). According to that derivation, the entropy variables, \mathbf{v}_h , serve as the adjoint solution to J . This statement is verified by linearizing J in (6.1) and computing the

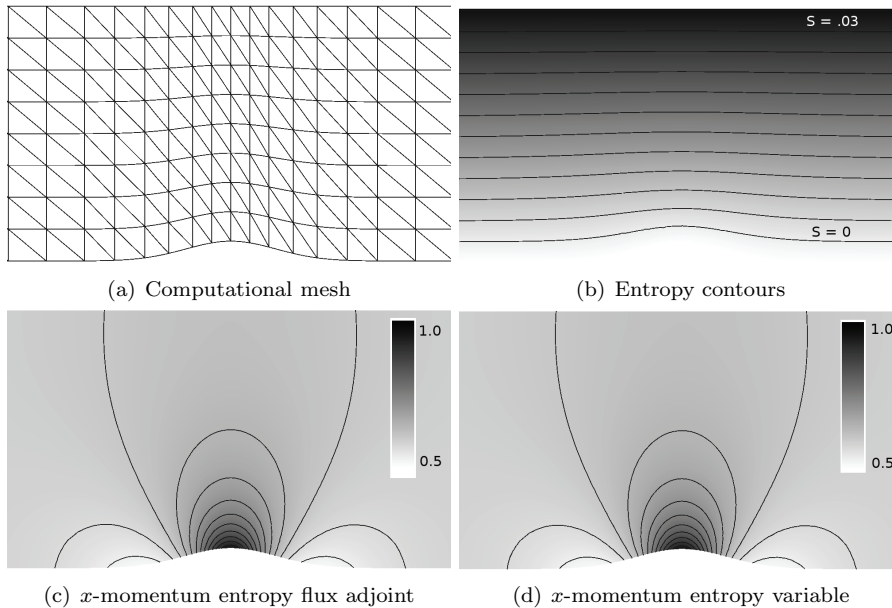


FIG. 6.1. *Gaussian bump, $M_\infty = 0.5$: Numerical illustration of the equivalence between entropy variables and entropy flux output adjoints.*

corresponding discrete adjoint solution. The x -momentum component of the adjoint solution is shown in Figure 6.1(c), and the x -momentum entropy variable is shown in Figure 6.1(d). Qualitatively, the two fields are nearly identical. The practical difference is that Figure 6.1(c) is obtained from a separate adjoint solve, whereas Figure 6.1(d) is almost free.

A quantitative comparison of the difference between entropy variables, \mathbf{v}_h , and the adjoint, ψ_h , corresponding to the entropy flux output in (6.1) is shown in Figure 6.2. This error is an L_2 measure of the difference between the two quantities, calculated in a continuous fashion over the domain, and in a discrete fashion over the state components:

$$(\text{Entropy variable adjoint error})^2 = \int_{\Omega} \|\psi_h - \mathbf{v}_h\|_2^2 d\Omega.$$

The error decreases with mesh refinement at an optimal rate of $O(h^{p+1})$, indicating that the adjoint variables for this problem are indeed equivalent to the discrete adjoint of the entropy integral output.

6.2. NACA 0012 in inviscid flow $M_\infty = 0.4$, $\alpha = 5^\circ$. In this example, the geometry is a NACA 0012 airfoil with a closed trailing edge and a farfield approximately 40 chord lengths away. The initial mesh is illustrated in Figure 6.5(a). This mesh consists of quadrilaterals, with cubic ($q = 3$) 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 the following results, quadratic solution approximation, $p = 2$, was used in the discretization, and the adaptation fixed fraction was set to $f^{\text{adapt}} = 0.1$, meaning that at each step of the adaptation those cells lying in the top 10% of the error criterion were chosen for refinement.

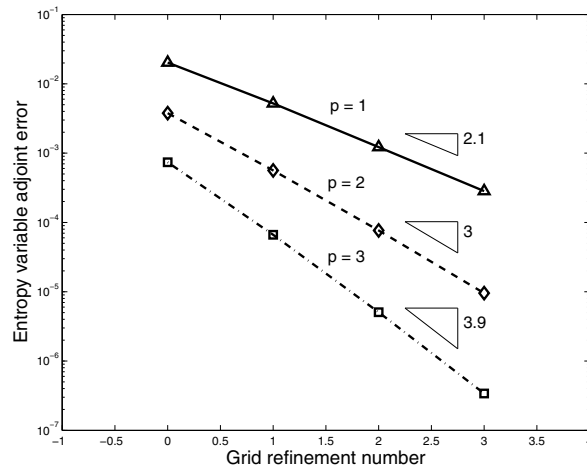


FIG. 6.2. Gaussian bump, $M_\infty = 0.5$: L_2 error between the entropy variables and the adjoint variables corresponding to the entropy flux output in (6.1). Convergence shown for uniform grid refinement, using approximation orders $p = 1, 2, 3$.

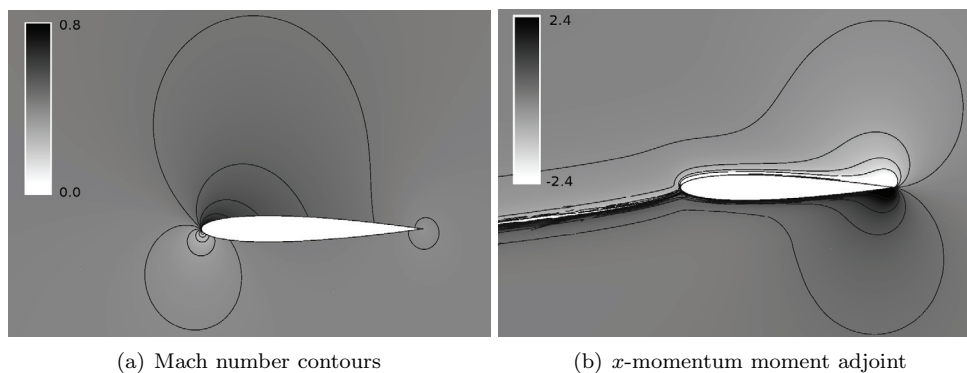


FIG. 6.3. NACA 0012, $M_\infty = 0.4$, $\alpha = 5^\circ$: Contours of the Mach number and the x -momentum component of the moment adjoint.

Mach number contours for the airfoil in inviscid flow at $M_\infty = 0.4$, $\alpha = 5^\circ$ are shown in Figure 6.3(a). Three different engineering outputs are considered: 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. One adaptation run was performed using the entropy variable 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 6.4 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 calculated from a $p = 3$ solution on a mesh obtained by uniformly refining the finest output-adapted mesh. For all three outputs of interest, the entropy and adjoint-based adaptive strategies perform similarly and are orders of magnitude better than uniform

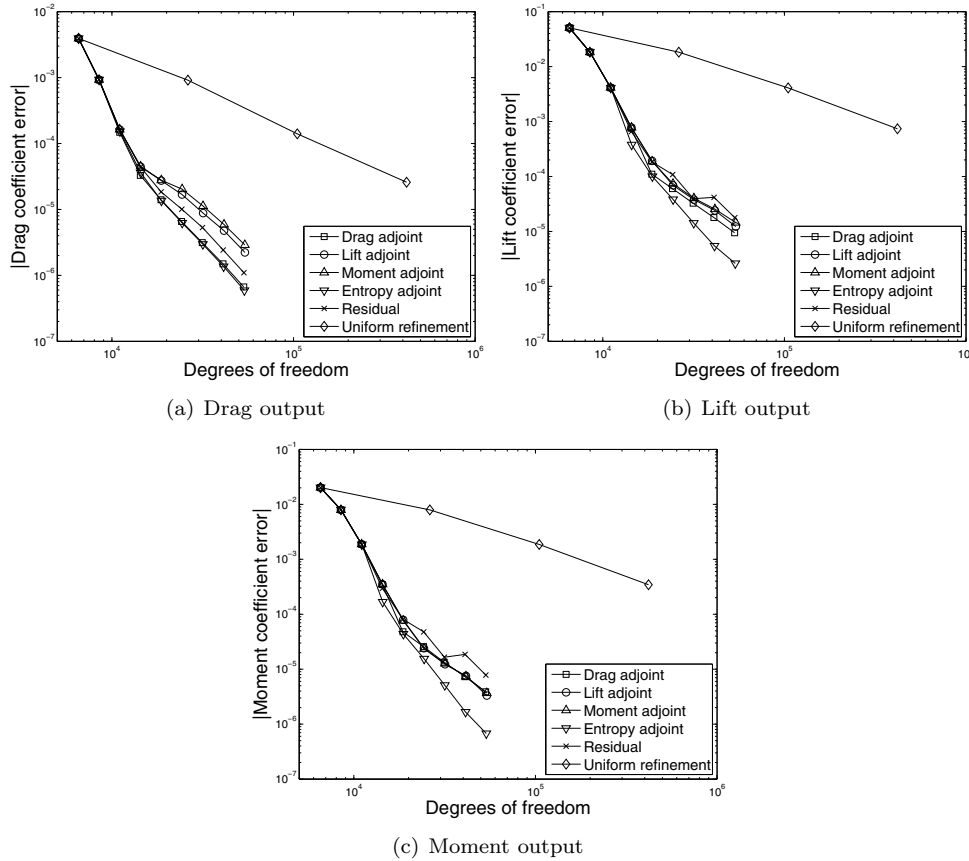


FIG. 6.4. *NACA 0012*, $M_\infty = 0.4$, $\alpha = 5^\circ$: Comparison of output convergence histories for various adaptation strategies.

mesh refinement. The unweighted residual indicator performs well for the drag output but somewhat worse for the lift and moment outputs compared to the entropy adjoint. Interestingly, the refinement based on the entropy adjoint actually gives better predictions for lift and moment than the refinements that specifically target those outputs. The prediction of drag is just as good as the specifically targeted prediction.

These results are certainly surprising, but not actually paradoxical, because the procedure does have empirical elements. One aspect worth mentioning is that the lift and moment adaptive indicators excessively target the stagnation streamline in front of the airfoil. As shown in Figure 6.3(b) for the moment output, the adjoint varies 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 [14]. Intuitively, a force output on the airfoil should respond differently to perturbations that affect the flow over the upper surface of the airfoil versus to those that affect the flow over the lower surface of the airfoil. The singularity is strongest for the lift and moment outputs, and for these cases the performance of the output adjoint adaptation deteriorates the most. The noise created by polynomial interpolation of the adjoint on discrete finite elements 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 6.5. The leading edge, trailing edge, and upper surface of the airfoil are consistently targeted for refinement by the output and entropy adjoint indicators. In the adjoint interpretation, these areas are important for the accurate prediction of the engineering outputs. In the entropy adjoint interpretation, these areas are locations of largest spurious entropy generation. The unweighted residual adaptation targets the vicinity of the leading edge and the trailing edge, but not the upper surface of the airfoil, leading to errors in the lift and moment outputs. Refinement of the stagnation streamline is evident in the adjoint-based runs, especially for the lift and moment adaptations.

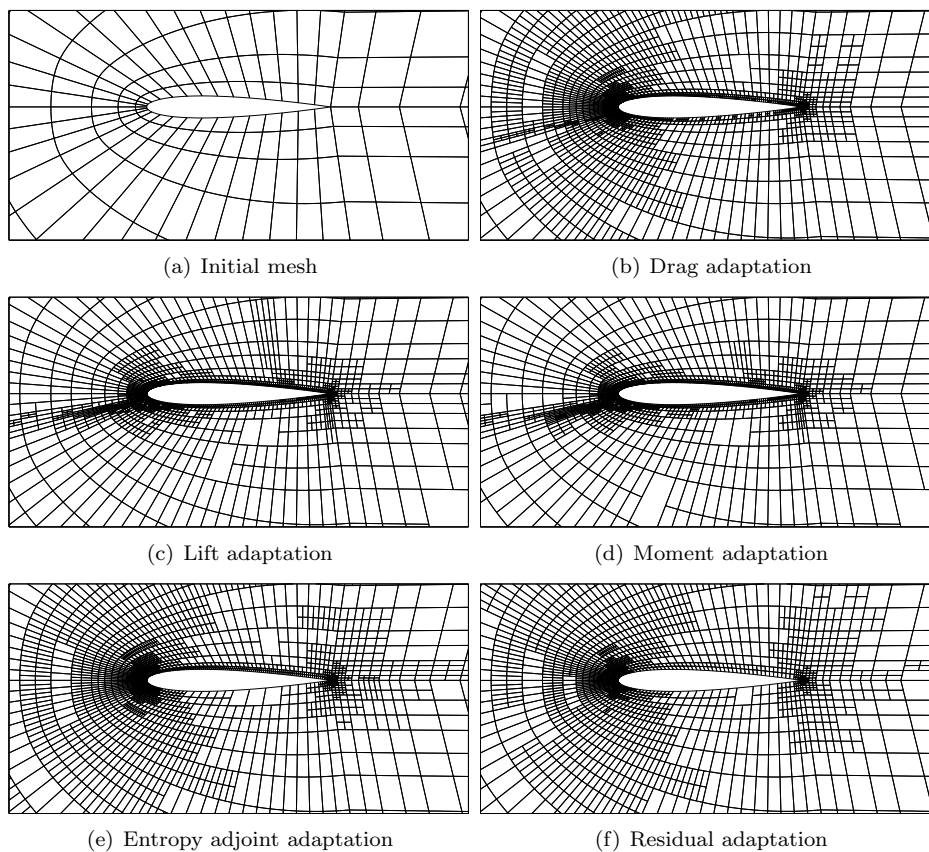


FIG. 6.5. *NACA 0012, $M_\infty = 0.4$, $\alpha = 5^\circ$: Meshes after eight adaptation iterations for the tested adaptation strategies.*

6.3. NACA 0012 in viscous flow, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$. The second example consists of a NACA 0012 airfoil in viscous flow. The Mach number distribution is illustrated in Figure 6.6. 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 50 chord lengths behind the airfoil.

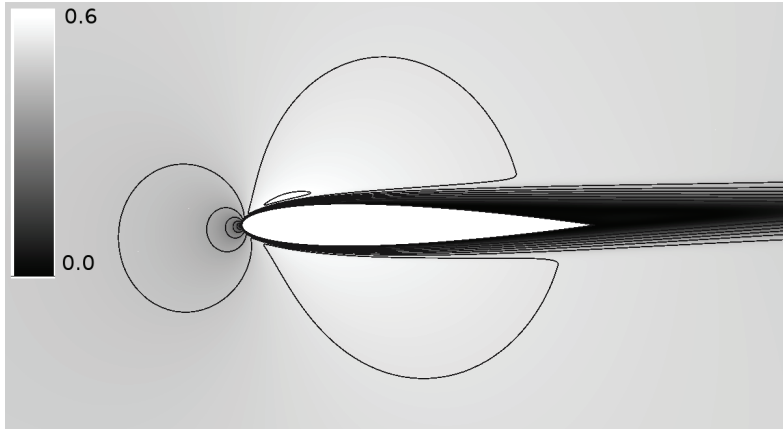


FIG. 6.6. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$: Mach number contours.

As in the first example, adaptation runs were performed using adjoints associated with each of these outputs, using the entropy adjoint, and using the unweighted residual.

Figure 6.7 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. Among the adjoint methods, the entropy adjoint is generally one of the two best, except in the case of lift, where the results are somewhat erratic.

The meshes after eight adaptation iterations are shown in Figure 6.8. 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 6.7, this indicator does not perform well at all. The corresponding mesh is shown in Figure 6.8(f): the entropy scalar indicator targets mainly the wake while leaving the vicinity of the airfoil too coarse.

6.4. NACA 0012 in three-dimensional flow $M_\infty = 0.4$, $\alpha = 3^\circ$. The third example consists of a NACA 0012, untapered, untwisted, aspect ratio 10 wing geometry with a closed trailing edge and a rounded body of revolution wing tip. In the computational domain, the farfield is approximately 40 chord lengths away from the wing. The domain is meshed with cubic, $q = 3$, hexahedral elements that are

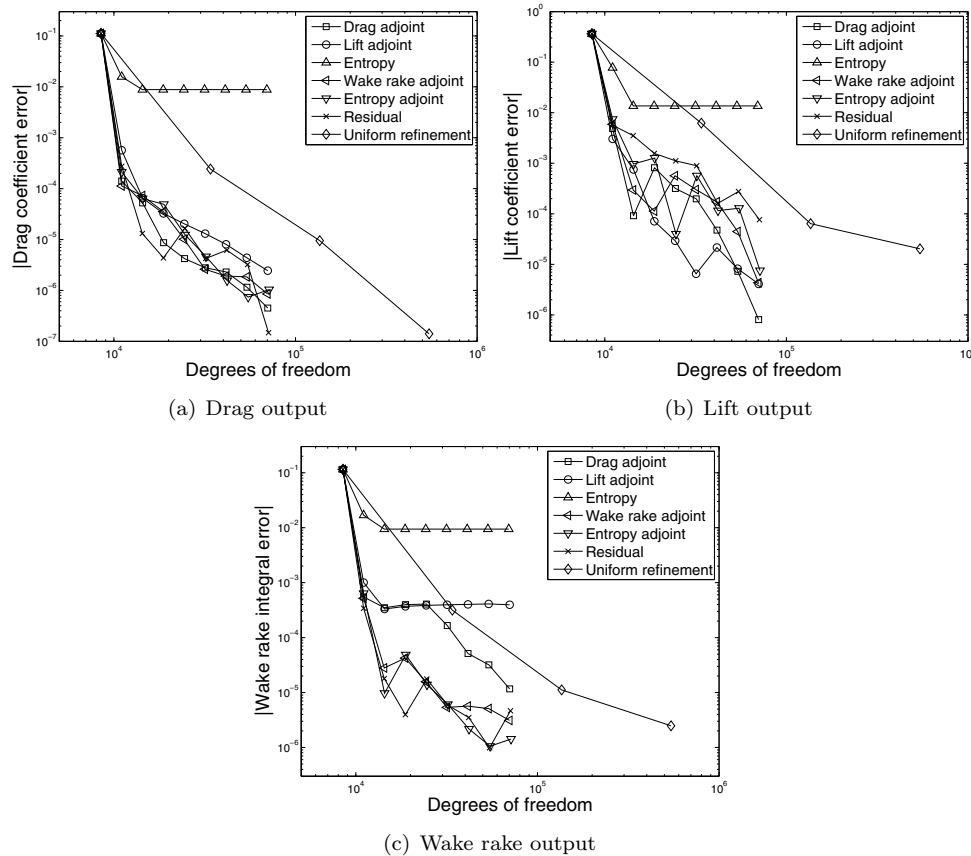
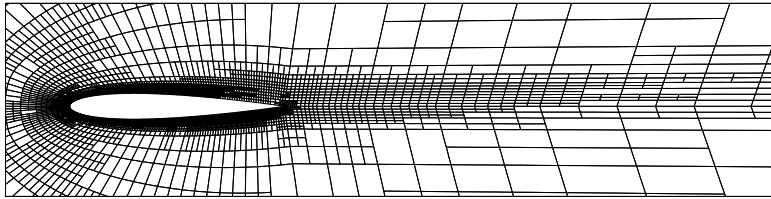


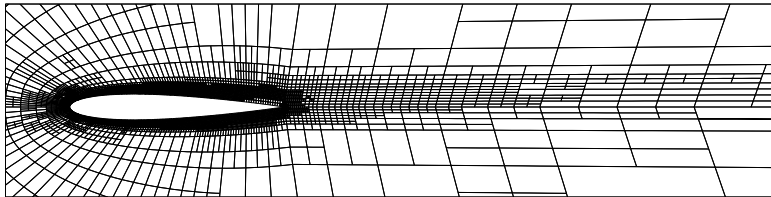
FIG. 6.7. *NACA 0012*, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$: Comparison of output convergence histories for the different adaptation strategies.

curved to conform to the geometry. The initial mesh of 4608 elements is shown in Figure 6.10(a). As in two dimensions, the apparent structure of the initial mesh disappears after the first adaptation iteration. Isotropic refinement is used in this example, with each flagged hexahedron divided into eight identical (in reference space) subhexahedra. To maintain at most one level of refinement between adjacent elements, additional elements are flagged for refinement, similarly to the two-dimensional case. This additional refinement need not be isotropic and is determined by an algorithm that minimizes the degrees of freedom introduced. Quadratic solution approximation, $p = 2$, was used for all adaptive runs, and the adaptation fixed fraction was set to $f^{\text{adapt}} = 0.1$. Finally, artificial viscosity stabilization, discussed in the next section, was included in the discretization to add stability to the solution, especially near the wing-tip trailing edge, on underresolved meshes.

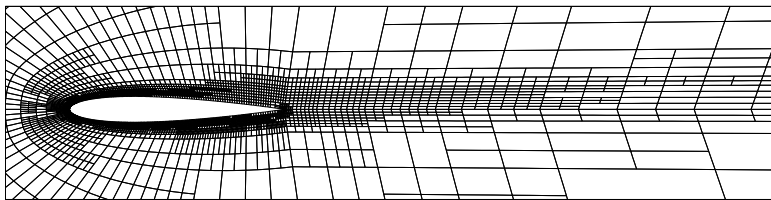
It is worth giving consideration to what we might expect from an inviscid calculation with trailing vorticity. We know from experiment [36] that a wake forms behind the trailing edge, partly as an extension of the surface boundary layer and partly in the form of streamwise vorticity that is shed due to variation in the lift distribution. As the viscosity goes to zero, the first of these vanishes, but the second does not. At the tip, vorticity is shed strongly, and the wake begins to roll up. The exact mechanism involved when the viscosity is vanishingly small is not clear. Even



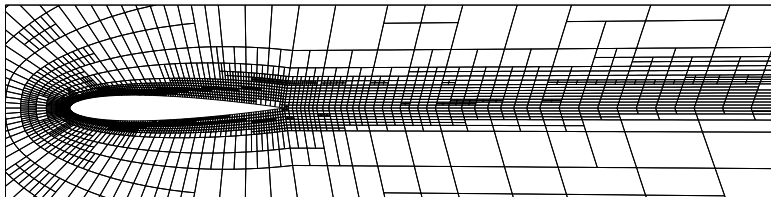
(a) Drag adaptation



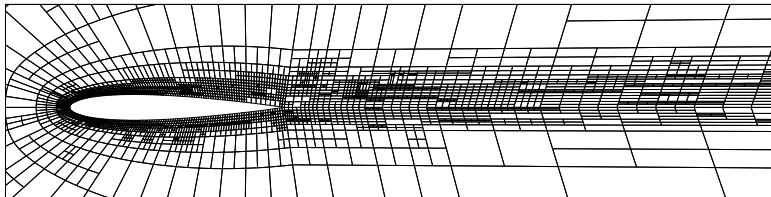
(b) Lift adaptation



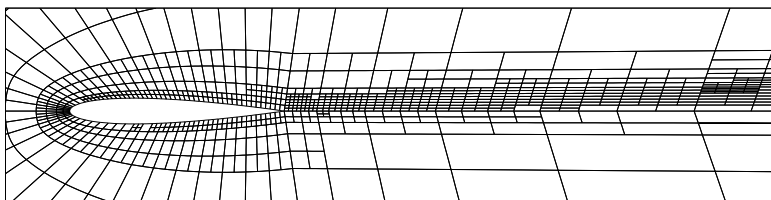
(c) Wake rake adaptation



(d) Entropy adjoint adaptation



(e) Residual adaptation



(f) Entropy adaptation

FIG. 6.8. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$: Meshes after eight adaptation iterations for various adaptation strategies. The entropy indicator flags many elements in the full, 50 chord extent of the wake, and hence the mesh in the vicinity of the airfoil appears sparser for the same number of adaptation iterations.

the careful study by Krasny [21] is of a very idealized case, and the structure close to the trailing edge does not appear to be known. Our numerical results seem to show a very concentrated vortex lying almost in a straight line, with extremely low pressures in its core, and carrying the great majority of the shed vorticity.

Two outputs are considered for this wing: drag and lift. The convergence of both of these outputs is compared for adaptive runs using five different indicators. These include two indicators based on the drag and lift adjoints, an indicator based on the entropy adjoint, an indicator based on the unweighted residual, and an indicator based on the entropy scalar. Figure 6.9 shows the results of the adaptation runs with each of the five indicators compared to uniform refinement. Six adaptive refinements were performed with each indicator. Exact values of the outputs were not computed due to limits on computational resources. Hence, the values plotted in Figure 6.9 are the lift and drag coefficients themselves, instead of their errors. For the drag output, adaptation based on the drag adjoint appears to be converging to a final value most rapidly. Assuming that this value is indeed the correct drag, the next most rapid convergences come almost equally from adaptation based on the lift adjoint and adaptation based on the entropy adjoint. Next is the residual-based indicator, with a performance that is still better than uniform refinement. Finally, adaptation on the entropy scalar performs the least well, stagnating at a drag value that appears worse than the drag at the first iteration in any of the other runs. For the lift output, the performance of the indicators is more clustered relative to uniform refinement, with the entropy adjoint close to the lift and drag adjoint. Adaptation on the entropy scalar performs least well, again apparently converging to an incorrect output value.

The final meshes after six adaptation iterations are shown in Figure 6.10. From the symmetry-plane and wing-surface meshes, it is clear that the indicators based on the output adjoints and the entropy adjoint target the leading and trailing edges the most. The entropy scalar adaptation instead mostly adds degrees of freedom to the wake and does not resolve the primary source of this entropy, which is the insufficient resolution near the wing. The symmetry-plane meshes show that, as in the two-dimensional case, the lift and drag adjoint indicators are targeting the stagnation streamline in front of the wing. This is again most likely due to the singularity in the adjoint solution across the stagnation streamline. It is not clear whether this level of refinement is necessary, as the lift adjoint indicator did not exhibit the best performance in the two-dimensional case.

The cut plane behind the wing tip in each of the mesh plots in Figure 6.10 shows the level of relative refinement of the wing-tip vortex. The entropy adjoint and residual indicators target the tip vortex most strongly, while the lift adaptation targets the tip vortex the least. Figure 6.11 illustrates the solution behavior near the tip vortex in more detail. It shows an entropy isosurface in the flow and entropy contours in a transverse cut of the wing-tip vortex. In an infinitely resolved vortex line, the entropy isosurface would disappear. However, with finite mesh resolution, the vortex line singularity cannot be fully resolved, leading to spurious entropy generation in a tube of diameter on the order of the mesh resolution. The entropy adjoint indicator targets the vortex line because of this spurious entropy generation. However, it does not target just the vortex, since spurious entropy is also generated in other parts of the flow, notably at the leading and trailing edges. The fact that all areas of spurious entropy generation are targeted leads to a very good performance of the entropy adjoint indicator for both the drag and the lift outputs.

For this three-dimensional case, some reasonable representation of the trailing vortex is necessary to give an account of induced drag and lost lift, but in this re-

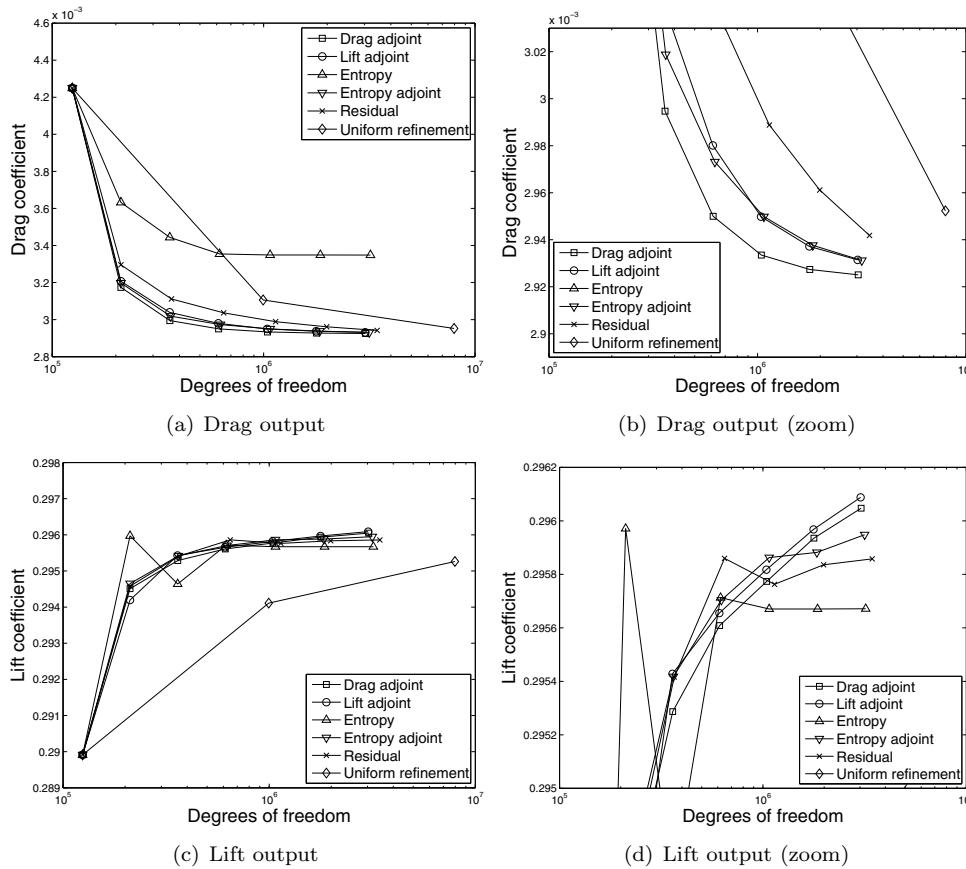


FIG. 6.9. NACA 0012 wing, $M_\infty = 0.4$, $\alpha = 3^\circ$: Comparison of output convergence histories for various adaptation strategies.

gard, the present simulations extending to 40 chord lengths downstream are probably overkill. This is demonstrated by the fact that those calculations driven by either the lift or drag adjoints do not heavily refine the vortex wake region but actually allow the vortex to disappear about 3–4 chord lengths behind the trailing edge. Bearing this in mind, it is interesting to see what happens to the vortex as it extends further downstream.

Figure 6.12 shows the vortices produced by refining on the drag adjoint, the unweighted residual, the entropy adjoint, and the entropy scalar. Respectively, these extend the existence of the vortex to 4 chord lengths, 12 chord lengths, and apparently indefinitely for both the entropy adjoint and the entropy scalar, as measured by the contiguity of the entropy isosurface chosen for visualization. The excessive refinement of the vortex region certainly handicaps the efficiency of the entropy adjoint in the comparisons of lift and drag, although it still comes out quite well. It would probably come out much better if the downstream extent of the domain were truncated. However, there are many circumstances in which vortices must be tracked much further, such as vortices shed from helicopter blades or from high-lift systems, and in these cases the entropy adjoint might show considerable advantage.

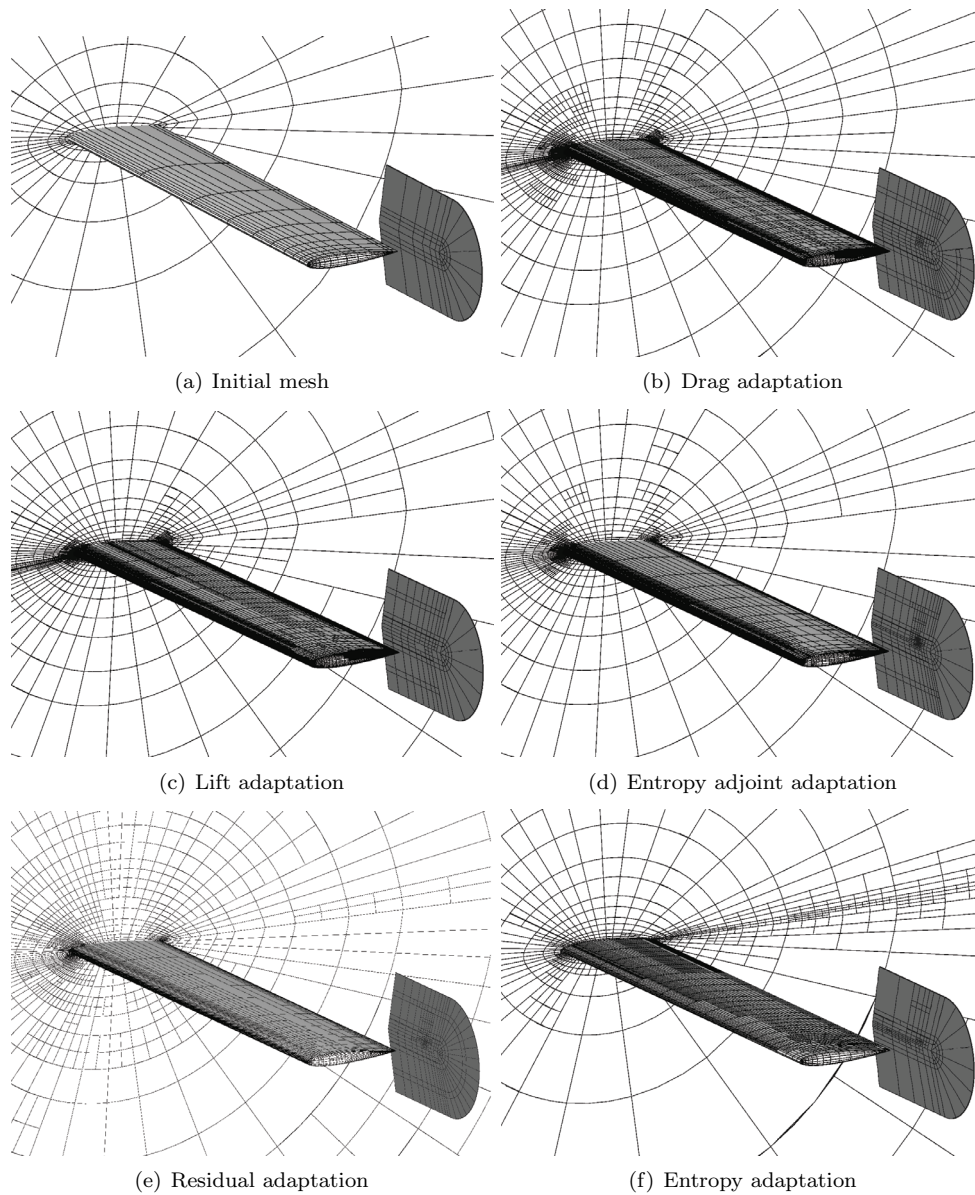


FIG. 6.10. *NACA 0012 wing, $M_\infty = 0.4$, $\alpha = 3^\circ$: Initial mesh and meshes after six adaptation iterations for various adaptation strategies.*

6.5. NACA 0012 in inviscid transonic flow: $M_\infty = 0.8$, $\alpha = 1.25^\circ$. The fourth example consists of a NACA 0012 airfoil in inviscid transonic flow. The Mach number contours are shown in Figure 6.13 for an adapted solution. A resolution-based artificial viscosity is used to stabilize the solution in the presence of shocks [28]. The initial mesh is the same as that used in the first example, and $p = 2$ solution approximation is used. Four adaptive indicators are compared: drag adjoint, lift adjoint, entropy adjoint, and the unweighted residual. Seven adaptation iterations are performed with $f^{\text{adapt}} = 0.1$.

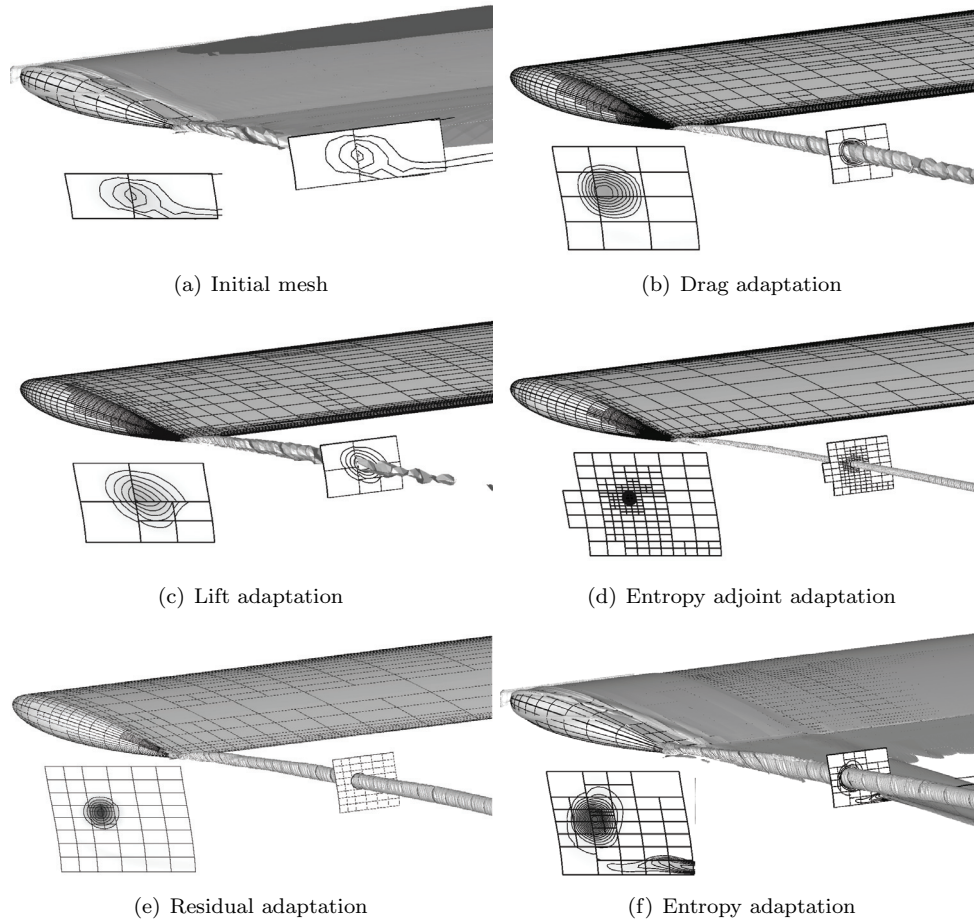


FIG. 6.11. *NACA 0012 wing, $M_\infty = 0.4$, $\alpha = 3^\circ$: Entropy isosurfaces and tip vortex entropy contours for each of the adaptation strategies.*

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 the shock will always be targeted for refinement. Such a problem is common in feature-based adaptation methods, where a single prominent feature draws into itself all of the computational resources. 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, which may require implementation changes to be realized, 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 will eventually terminate, but only when the element size is on the order of the shock thickness, which is impractical for realistic flows.

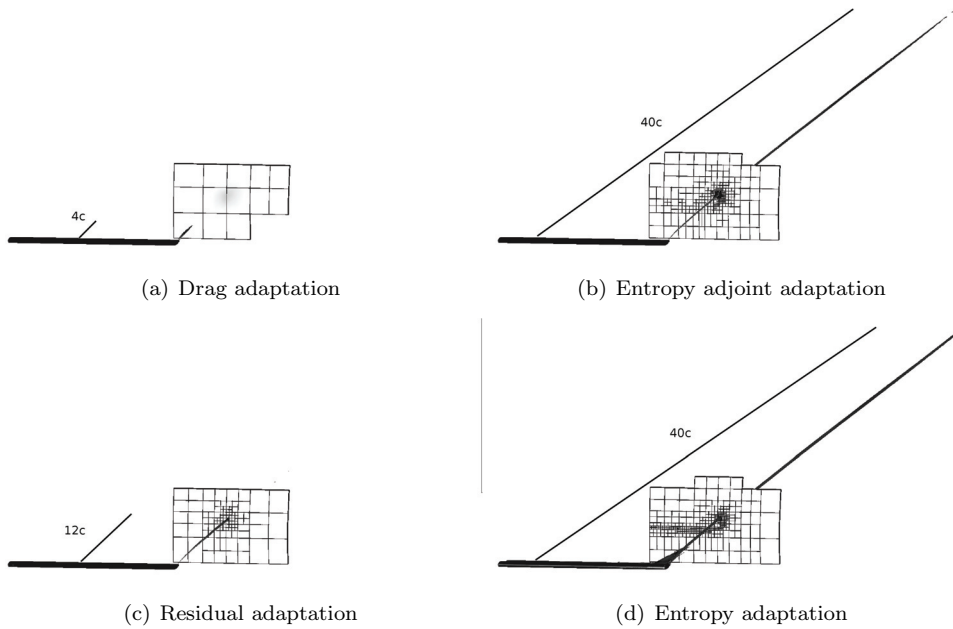


FIG. 6.12. *NACA 0012 wing, $M_\infty = 0.4$, $\alpha = 3^\circ$: Entropy isosurfaces of the tip vortex for several adaptation strategies, showing the extent downstream.*

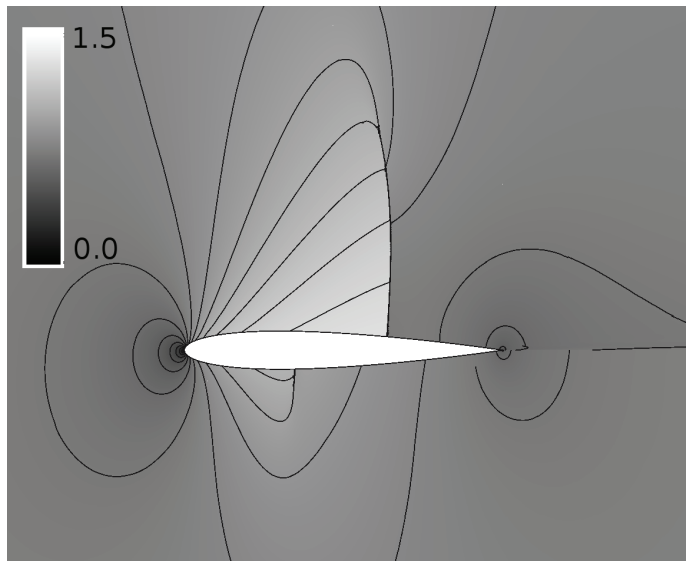


FIG. 6.13. *NACA 0012, $M_\infty = 0.8$, $\alpha = 1.25^\circ$: Mach number contours.*

The purpose of this example is to empirically test this trade-off for one flow. Figure 6.14 shows the convergence of the lift and drag errors using the various indicators and using uniform refinement. Truth values were computed using $p = 3$ approximation on uniformly refined versions of the final output-adapted meshes. The entropy adjoint performs very similarly to the output-based adjoints in both cases,

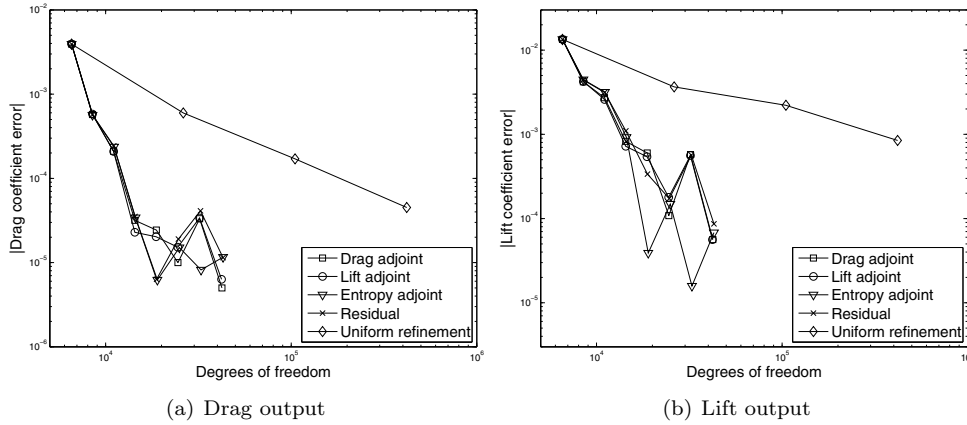


FIG. 6.14. *NACA 0012, $M_\infty = 0.8$, $\alpha = 1.25^\circ$: Comparison of output error convergence histories for various adaptation strategies.*

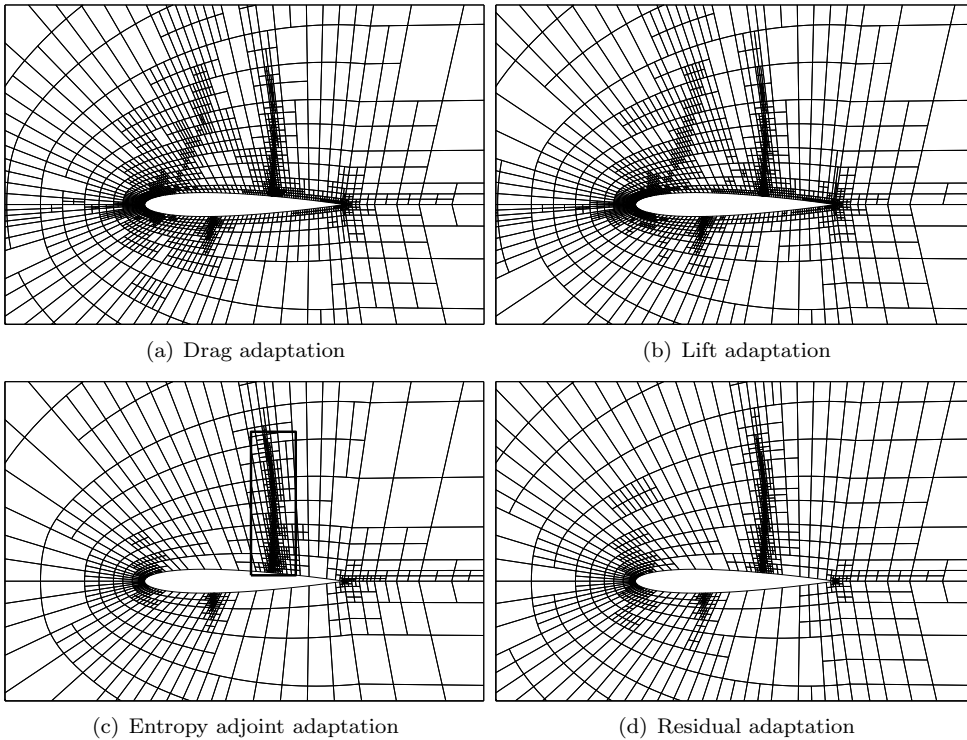


FIG. 6.15. *NACA 0012, $M_\infty = 0.8$, $\alpha = 1.25^\circ$: Meshes after seven adaptation iterations. The box in the entropy adjoint mesh indicates the region used for the element count.*

although the scatter in the convergence histories precludes a definitive ranking. The final meshes are shown in Figure 6.15. 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 overrefines the shock region, we count the number of cells with centroids lying inside the rectangular region outlined

in Figure 6.15(c). 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 that 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.

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 overrefinement can occur in the shock region. Accordingly, the analysis of the entropy adjoint indicator for flows with shocks is the subject of ongoing research.

7. 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 fulfill the adjoint role as Lagrange multipliers that can be used 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 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. In a three-dimensional wing test, the entropy adjoint indicator is shown to resolve both the tip vortex and the leading and trailing edges of the wing. Similarly, for an airfoil in transonic flow, the entropy adjoint indicator targets not only the shock but also the leading and trailing edges.

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. It is important to realize that the indicator does not simply target regions of large entropy or regions of large entropy production, nor does it attempt to conserve entropy. It targets only regions of large *spurious entropy production*. That has to be interpreted to include errors of entropy transport.

Future work can proceed in several directions. We will apply the method as presently developed to progressively more complicated situations, such as those involving rotorcraft with extensive vortex shedding. We will investigate the suitability of this indicator for unsteady flows, to which the entropy adjoint connection extends in a straightforward manner. We will also exploit the fact that our technique will apply to any set of governing equations with the same mathematical structure, that is, comprising conservation laws with an entropy extension. For example, the equations of ideal magnetohydrodynamics should yield to the same attack [4]. There are other systems of equations that do not possess a thermodynamic entropy but do possess

a mathematical entropy that is conserved in smooth dissipation-free flow and which symmetrizes the equations [33]. For incompressible flow the Bernoulli “constant” serves this purpose, and for the shallow-water equations it is the total energy. Since the properties of symmetry and conservation are the only ones involved in the development of our method, we expect to obtain equally satisfying results. Finally, there are systems of equations, including popular Reynolds-averaged Navier–Stokes (RANS) closures, for which symmetrizing entropy variables do not exist and hence to which the entropy adjoint approach does not apply directly. This observation motivates further research in this area, for example into symmetrizable RANS models.

We have not yet explored any variations of the numerical technique. The code used here was not written specifically for this purpose, although it had been used previously for adjoint-based error estimation. We believe that results will be essentially similar for structured and unstructured grids and for any type of regularization involving artificial viscosity or limiting procedures. Likewise, the choice between finite element and finite volume discretizations is likely not relevant. Nevertheless, at the level of detailed coding, there are some not unimportant decisions to be made. We are certainly prepared to find that some of these decisions will influence the success of the method in dealing with strong shockwaves.

Acknowledgments. The authors would like to acknowledge the constructive feedback from the manuscript referees. In addition, 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.

REFERENCES

- [1] J. ANDREWS AND K. MORTON, *Spurious entropy generation as a mesh quality indicator*, in Proceedings of the Fourteenth International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Phys. 453, Springer-Verlag, Berlin, Heidelberg, 1995, pp. 122–126.
- [2] T. J. BAKER, *Mesh adaptation strategies for problems in fluid dynamics*, Finite Elem. Anal. Des., 25 (1997), pp. 243–273.
- [3] T. BARTH AND M. LARSON, *A posteriori error estimates for higher order Godunov finite volume methods on unstructured meshes*, in Finite Volumes for Complex Applications III, R. Herban and D. Kröner, eds., Hermes Penton, London, 2002, pp. 27–49.
- [4] T. J. BARTH, *Numerical methods for gasdynamic systems on unstructured meshes*, in An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Lect. Notes Comput. Sci. Eng. 5, D. Kröner, M. Ohlberger, and C. Rhode, eds., Springer-Verlag, Berlin, 1999, pp. 195–285.
- [5] R. BECKER AND R. RANNACHER, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numer., 10 (2001), pp. 1–102.
- [6] M. J. CASTRO-DIAZ, F. HECHT, B. MOHAMMADI, AND O. PIRONNEAU, *Anisotropic unstructured mesh adaptation for flow simulations*, Internat. J. Numer. Methods Fluids, 25 (1997), pp. 475–491.
- [7] J. DOMPIERRE, M.-G. VALLET, Y. BOURGAULT, M. FORTIN, AND W. G. HABASHI, *Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part III: Unstructured meshes*, Internat. J. Numer. Methods Fluids, 39 (2002), pp. 675–702.
- [8] P. DUTT, *Stable boundary conditions and difference schemes for Navier–Stokes equations*, SIAM J. Numer. Anal., 25 (1988), pp. 245–267.
- [9] R. P. DWIGHT, *Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation*, J. Comput. Phys., 227 (2008), pp. 2845–2863.
- [10] K. J. FIDKOWSKI AND D. L. DARMOFAL, *A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier–Stokes equations*, J. Comput. Phys., 225 (2007), pp. 1653–1672.

- [11] K. J. FIDKOWSKI AND D. L. DARMOFAL, *Output-based error estimation and mesh adaptation: Overview and recent results*, AIAA J., to appear.
- [12] K. J. FIDKOWSKI, T. A. OLIVER, J. LU, AND D. L. DARMOFAL, *p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations*, J. Comput. Phys., 207 (2005), pp. 92–113.
- [13] M. GILES AND N. PIERCE, *Adjoint error correction for integral outputs*, in Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics, Lect. Notes Comput. Sci. Eng. 25, T. Barth and H. Deconinck, eds., Springer-Verlag, Berlin, 2003, pp. 47–95.
- [14] M. B. GILES AND N. A. PIERCE, *Adjoint Equations in CFD: Duality, Boundary Conditions, and Solution Behavior*, AIAA paper 97-1850, AIAA, Reston, VA, 1997.
- [15] M. B. GILES AND E. SÜLI, *Adjoint methods for PDEs: A posteriori error analysis and post-processing by duality*, Acta Numer., 11 (2002), pp. 145–236.
- [16] R. HARTMANN, *Adjoint consistency analysis of discontinuous Galerkin discretizations*, SIAM J. Numer. Anal., 45 (2007), pp. 2671–2696.
- [17] R. HARTMANN AND P. HOUSTON, *Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations*, J. Comput. Phys., 183 (2002), pp. 508–532.
- [18] R. HARTMANN AND P. HOUSTON, *An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier–Stokes equations*, J. Comput. Phys., 227 (2008), pp. 9670–9685.
- [19] P. HOUSTON AND E. SÜLI, *Error estimation and adaptive discretization methods in computational fluid dynamics*, in Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics, Lect. Notes Comput. Sci. Eng. 25, T. Barth and H. Deconinck, eds., Springer-Verlag, Berlin, 2003, pp. 269–344.
- [20] T. J. R. HUGHES, L. P. FRANCA, AND M. MALLETT, *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*, Comput. Methods Appl. Mech. Engrg., 54 (1986), pp. 223–234.
- [21] R. KRASNY, *Vortex sheet computations: Roll-up, wakes, separation*, in Vortex Dynamics and Vortex Methods, Lectures in Appl. Math. 28, AMS, Providence, RI, 1991, pp. 385–402.
- [22] J. LU, *An A Posteriori Error Control Framework for Adaptive Precision Optimization Using Discontinuous Galerkin Finite Element Method*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005.
- [23] D. J. MAVRIPLIS, J. C. VASSBERG, E. N. TINOCO, M. MANI, O. P. BRODERSEN, B. EISFELD, R. A. WAHLS, J. H. MORRISON, T. ZICKUHR, D. LEVY, AND M. MURAYAMA, *Grid Quality and Resolution Issues from the Drag Prediction Workshop Series*, AIAA paper 2008-930, AIAA, Reston, VA, 2008.
- [24] G. F. NATERER AND J. A. CAMBEROS, *Entropy and the second law fluid flow and heat transfer simulation*, J. Thermophysics Heat Transfer, 17 (2003), pp. 360–371.
- [25] G. F. NATERER AND D. RINN, *Towards entropy detection of anomalous mass and momentum exchange in incompressible fluid flow*, Internat. J. Numer. Methods Fluids, 39 (2002), pp. 1013–1036.
- [26] M. NEMEC, M. J. AFTOSMIS, AND M. WINTZER, *Adjoint-Based Adaptive Mesh Refinement for Complex Geometries*, AIAA paper 2008-0725, AIAA, Reston, VA, 2008.
- [27] T. A. OLIVER, *A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier–Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2008.
- [28] P.-O. PERSSON AND J. PERAIRE, *Sub-cell Shock Capturing for Discontinuous Galerkin Methods*, AIAA paper 2006-112, AIAA, Reston, VA, 2006.
- [29] P.-O. PERSSON AND J. PERAIRE, *Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier–Stokes equations*, SIAM J. Sci. Comput., 30 (2008), pp. 2709–2733.
- [30] N. A. PIERCE AND M. B. GILES, *Adjoint recovery of superconvergent functionals from PDE approximations*, SIAM Rev., 42 (2000), pp. 247–264.
- [31] R. RANNACHER, *Adaptive Galerkin finite element methods for partial differential equations*, J. Comput. Appl. Math., 128 (2001), pp. 205–233.
- [32] P. L. ROE, *Approximate Riemann solvers, parametric vectors, and difference schemes*, J. Comput. Phys., 43 (1981), pp. 357–372.
- [33] D. SERRE, *Systems of Conservation Laws 1: Hyperbolicity, Entropies, Shock Waves*, Cambridge University Press, Cambridge, UK, 1999.
- [34] K. SHAHBAZI, *A high-order discontinuous Galerkin method for the unsteady incompressible Navier–Stokes equations*, J. Comput. Phys., 222 (2007), pp. 391–407.
- [35] P. SOLÍN AND L. DEMKOWICZ, *Goal-oriented hp-adaptivity for elliptic problems*, Comput. Meth-

- ods Appl. Mech. Engrg., 193 (2004), pp. 449–468.
- [36] P. R. SPALART, *Airplane trailing vortices*, in Annual Review of Fluid Mechanics, Vol. 30, Annual Reviews, Palo Alto, CA, 1998, pp. 107–138.
- [37] E. TADMOR AND W. ZHONG, *Entropy stable approximations of Navier–Stokes equations with no artificial numerical viscosity*, J. Hyperbolic Differ. Equ., 3 (2006), pp. 529–559.
- [38] D. A. VENDITTI AND D. L. DARMOFAL, *Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows*, J. Comput. Phys., 187 (2003), pp. 22–46.
- [39] G. P. WARREN, W. K. ANDERSON, J. L. THOMAS, AND S. L. KRIST, *Grid Convergence for Adaptive Methods*, AIAA paper 1991-1592, AIAA, Reston, VA, 1991.
- [40] X. D. ZHANG, M.-G. VALLET, J. DOMPIERRE, P. LABBE, D. PELLETIER, J.-Y. TREPANIER, R. CAMARERO, J. V. LASSALINE, L. M. MANZANO, AND D. W. ZINGG, *Mesh Adaptation Using Different Error Indicators for the Euler Equations*, AIAA paper 2001-2549, AIAA, Reston, VA, 2001.