A Combined Entropy and Output-based Adjoint Approach for Mesh Refinement and Error Estimation

Kevin T. Doetsch [1] and Krzysztof J. Fidkowski[2]
Department of Aerospace Engineering,

University of Michigan, Ann Arbor, MI 48109, USA

This paper presents a strategy for mesh refinement driven by a new indicator that combines two previously-investigated indicators: one based on a user-specified engineering output such as drag or lift coefficient, and the other based on entropy variables. Using the entropy-variable indicator to adapt a mesh is computationally advantageous since it does not require the solution of an auxiliary adjoint equation, which for unsteady problems is particularly costly. However, the entropy-variable indicator targets any region of the domain where spurious entropy is generated, regardless of whether or not this region affects an engineering output of interest. On the other hand, an indicator computed from an engineering output generally targets only those regions important for the chosen output, though it is more computationally taxing because of the required adjoint solution. Approximations in the adjoint calculation reduce this cost, at the expense of indicator accuracy. In combining these indicators, our objective is to maintain the low cost of approximate adjoint solutions while achieving improved indicator accuracy from the entropy variables. We demonstrate the potential for this method through several simulations governed by the compressible Navier-Stokes equations using both hanging-node refinement and MOESS. In general, MOESS will provide more optimal meshes, but often it is not a viable option. This demonstrates the need to

---

[1] Graduate Research Assistant, AIAA Member, kevintd@umich.edu
[2] Associate Professor, AIAA Senior Member, kfid@umich.edu

show that the combined approach benefits are not dependent on the mesh adaptation strategy.

Nomenclature

α = angle of attack

$\mathbf{A}$ = inviscid flux Jacobian

c_d = drag coefficient

δJ = output error

$\delta \mathbf{r}$ = infinitesimal source term

$\Delta \mathcal{E}_{ej}$ = error indicator

η_{κ_H} = elemental adaptive indicator

E = total energy per unit mass

F = entropy flux

f^{adapt} = fraction of elements to adapt

f^{error} = fraction of total error to refine

γ = ratio of specific heats

J = scalar output of interest

κ_H = element of computational mesh

$\mathbf{K}$ = diffusivity tensor

M = Mach number

n = normal pointing out of the computational domain

μ_{κ_H} = mask on the entropy-based indicator

ν_{fine} = number of smoothing operations

Ω = computational domain

P = pressure

p = solution polynomial order

$\boldsymbol{\psi}$ = adjoint state vector

$\boldsymbol{\psi}_h$ = fine-space element-wise polynomial representation of the adjoint

q = geometric polynomial order

$\mathbf{R}$ = residual operator

$\mathbf{r}(\cdot)$ = differential operator

$\mathcal{R}(\cdot,\cdot)$ = semilinear weak form

ρ = density

S = physical entropy

T_H = triangular computational mesh

U = entropy function

$\mathbf{u}$ = conservative state variables

$\mathbf{u}_H$ = element-wise polynomial representation of the state

u_i = velocity vector

$\mathcal{V}$ = trial and test function space

V = velocity magnitude

$\mathbf{v}$ = entropy variables

$\mathbf{w}$ = vector of weight function

Subscripts

H = current approximation space

h = fine approximation space

i, j = spatial indices

∞ = freestream conditions

I. Introduction

A popular approach in Computational Fluid Dynamics (CFD) to obtain accurate solutions for problems that exhibit a wide range of spatial length scales, whose distributions are generally not a known ahead of time, is solution-based adaptation [1–7]. Adaptive methods use an indicator computed a posteriori from the solution to drive the adaptation of the computational mesh. In the literature, extensive analysis has been done studying various indicators with regard to their accuracy, expense, and robustness. Two previously-studied indicators relevant to our present work are one based on a user-specified engineering scalar output [4, 8, 9] and one based on entropy variables [10, 11]. Both of these indicators have their strengths and weaknesses depending on the nature of a given problem. In this work, a new indicator is investigated that combines both the output-based adjoint and entropy-variable indicators. The main approach investigated in this paper

is a combination that consists of a simple product of the two indicators. In addition, various modifications to this approach are also considered.

Output-based adaptive methods are advantageous since they specifically target areas of the mesh that are critical to the prediction of the output of interest [3, 4, 12, 13]. Prior to using output-based indicators, a common practice in solution-based adaptation was to focus on using heuristic indicators that refined the mesh in "interesting" areas, such as those where solution gradients, curvatures, or other features were active [14–16]. These indicators are cheap to compute but not necessarily suitable for improving accuracy in the prediction of an engineering output since they fail to refine areas of the domain important to the prediction of that desired output [17]. Instead, output-based indicators link the desired engineering outputs and the computation mesh.

Output-based adaptive methods are particularly effective in accounting for propagation effects intrinsic to hyperbolic problems [18], through the use of output-specific adjoint solutions, which provide the sensitivity of the output to local residuals. The downside of this approach is that the generation of output-based adjoint solutions is often computationally expensive. This is because of the need to solve the adjoint problem on a refined-space computational domain. In addition, the output-based approach is not always effective due to a suboptimal prediction of the adjoint, especially in the presence of adjoint singularities such as the leading-edge stagnation streamlines and the trailing edges[19]. These singularities lead to noise in the error estimate, which leads to an inaccurate prediction of the output-based adjoint that results in over-refinement of the computational mesh.

The entropy-variable adaptive approach alleviates the issue of cost since the entropy variables are far less expensive to obtain: they are computed from a direct variable transformation of the conservative state. Unfortunately, because the entropy-variable indicator targets all regions of the domain where spurious entropy is produced, e.g. shocks or 3D vortices, this strategy is not as globally discriminating as the output-based approach. The objective of this paper is to more robustly extend the entropy-variable adaptive approach for these cases by combining it with indicators obtained from output-based adjoints. The goal is the calculation of a combined approach that encompasses the strengths of both the output-based adjoint and the entropy-variable approach while limiting the

downsides of each.

The outline of this paper is as follows. Section II reviews the concepts of using output-based adjoints and entropy variables separately for mesh refinement and error estimation. Section III introduces the combined output-based adjoint and entropy-variable approach to mesh refinement and error estimation. This section also reviews various modifications to obtaining the combined indicator that were investigated in this research. Section IV compares the two mesh adaptation strategies used in this work, as well as a sequential breakdown of the various steps for each adaptive iteration. Section V discusses the numerical implementation of these adaptive strategies and includes a discussion regarding the relative cost of each. Finally, Section VI presents the results generated using these methods.

II. Output Error Estimation

This section reviews previously-documented error estimation techniques that rely on mesh refinement indicators obtained either from an output-based adjoint or from entropy variables. The derivation of these two approaches, particularly regarding how the indicators are obtained, is the primary focus since both of these indicators are required for the combined approach that is the main focus of this work.

A. Adaptation Using Output Adjoints

Given a desired engineering output, output-based error estimation methods refine areas of the mesh important for an accurate prediction of the output. They account for error propagation effects that are inherent to convection-dominated flow simulations by targeting residuals to which the output is most sensitive. The resulting error estimates provide confidence levels to the output calculations and can be localized to elements to drive adaptation. These error estimations rely on the solution of an adjoint problem, which yields, in continuous form, the adjoint $\boldsymbol{\psi}(\vec{x})$. The adjoint is a Green's function that relates residual source perturbations to the output of interest, $J(\mathbf{u})$, where $\mathbf{u}$ denotes the state vector.

Consider a partial differential equation $\mathbf{r}(\mathbf{u}) = \mathbf{0}$, where $\mathbf{r}(\cdot)$ is a differential operator. The goal

of a variational solution method is to 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 the trial and test function space and $\mathcal{R}(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is the semilinear operator that represents the weak form of the differential equation. The adjoint $\boldsymbol{\psi} \in \mathcal{V}$ is the sensitivity of J to an infinitesimal source term, $\delta\mathbf{r}$, added to the governing equation,

$$\delta J =< \delta\mathbf{r}, \boldsymbol{\psi} >, \tag{2}$$

where $< \cdot, \cdot >$ is an inner product over the computational domain, Ω. The adjoint satisfies the following weak statement[8]: 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 about $\mathbf{u}$. Using the solution of the adjoint equation, the output error can be estimated by the adjoint-weighted residual method, which is based on the following two observations:

- In general the approximate solution $\mathbf{u}_H$ in a finite-dimensional approximation space $\mathcal{V}_H$ will not satisfy the governing PDE. However, it will satisfy a PDE with a small perturbation, 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 translates the residual perturbation to an output perturbation via Eqn. 2:

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

  Using a weighted residual approximation of the solution, the above expression quantifies the output's numerical error. For non-infinitesimal perturbations and nonlinear outputs or PDEs, the numerical error calculation is not exact.

Eqn. 4 is only computable if the continuous adjoint is approximated. This is achieved by solving the adjoint equation on a finer finite-dimensional space, $\mathcal{V}_h \supset \mathcal{V}_H$, either directly or iteratively[20–22]. Depending on the scope of the problem, this step can be computationally expensive. Solving the

adjoint equation yields a fine-space adjoint $\boldsymbol{\psi}_h$ that can be used to obtain an adaptive indicator whose purpose is to relate the relative contribution of each element to the total output error. Eqn. 4 can be approximated as

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

where $|_{\kappa_H}$ denotes the restriction of an interpolated function to element κ_H of the triangulation T_H. For the sake of simplicity, in this work, $\mathcal{V}_h$ is obtained from $\mathcal{V}_H$ by increasing the approximation order, while keeping the triangulation between the fine space and the coarse space the same. Note that in Eqn. 5 we use the difference between the fine and coarse space adjoints, $\delta\boldsymbol{\psi}_h \equiv \boldsymbol{\psi}_h - \boldsymbol{\psi}_H$, in order to minimize the error due to possible p-dependence of the residual. When Galerkin orthogonality holds, the subtraction of $\boldsymbol{\psi}_H$ does not impact the error estimate or the error indicator. However, DG discretizations often exhibit p-dependence, which refers to the presence of terms (e.g. in the BR2 stabilization) that explicitly depend on the solution approximation order, p. In these cases, a coarse-space solution injected to the fine space does not necessarily yield zero residuals when tested with coarse-space functions, and these residuals can pollute the error estimates. By subtracting the coarse-space adjoint from $\boldsymbol{\psi}_h$, the effect of these residuals on the error estimates is minimized [23]. The adaptive indicator is obtained by taking the absolute value of the elemental contributions in Eqn. 5,[3, 12, 24, 25]

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

This indicator is computed separately, with absolute values, for each equation in a system of equations and then summed together, possibly yielding a magnitude greater than the actual output error estimate. However, such an indicator is still not a bound for the actual error due to the previously made approximations.

B. Adaptation Using Entropy Variables

The previous section presented output-based error estimation, in which a user prescribes an engineering scalar output to drive mesh adaptation. This section describes the entropy-based adjoint indicator, which instead uses entropy variables to drive the adaptation. The areas of the mesh that

are targeted by the entropy indicator are those regions that exhibit high net production of *spurious* entropy. The subsequent review of the formulation of the entropy-variable approach follows our previous work [11].

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

$$\mathbf{A}_i \partial_i \mathbf{u} - \partial_i \left(\mathbf{K}_{ij} \partial_j \mathbf{u} \right) = \mathbf{0}, \tag{7}$$

where i is the spatial index, $\mathbf{A}_i \partial_i \mathbf{u}$ is the inviscid flux and $-\mathbf{K}_{i,j} \partial_j \mathbf{u}$ is the viscous flux. We assume the existence of a scalar entropy conservation law that has the form

$$\partial_i F_i = 0, \tag{8}$$

where $F_i(\mathbf{u})$ is the entropy flux associated with the entropy function $U(\mathbf{u})$. Both the entropy flux and entropy function satisfy the compatibility relation $U_{\mathbf{u}} \mathbf{A}_i = (F_i)_{\mathbf{u}}$. The entropy variables are defined as $\mathbf{v} = U_{\mathbf{u}}^{\mathrm{T}}$, and they symmetrize the unsteady set of conservation laws in the sense that[15, 26]: the transformation Jacobian matrix, $\mathbf{u}_{\mathbf{v}}$, is symmetric, positive definite; and $\mathbf{A}_i \mathbf{u}_{\mathbf{v}}$ is symmetric. Additionally, the entropy variables $\mathbf{v}$ must also symmetrize $\mathbf{K}_{ij}$, in the sense that $\widetilde{\mathbf{K}}_{ij} = \widetilde{\mathbf{K}}_{ji}^{\mathrm{T}}$, where $\widetilde{\mathbf{K}}_{ij} = \mathbf{K}_{ij} \mathbf{u}_{\mathbf{v}}$[26]. Substituting $\partial_i \mathbf{u} = \mathbf{u}_{\mathbf{v}} \partial_i \mathbf{v}$ into Eqn. 7, along with taking the transpose, yields the following equation for the entropy variables,

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

Due to the sign of the second term being negative, this is no longer a strict mathematical adjoint to the primal equation. However, the entropy variables still act as an adjoint when the output includes a domain-integral contribution [11],

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

Each of the terms in Eqn. 10 has its own physical meaning outlined in previous works[10, 11]. 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 third term is the entropy diffusion across the boundary.

The entropy function that yields entropy variables that symmetrize both the inviscid and viscous terms in the compressible Navier-Stokes equations, with heat-conduction included, is unique up to additive and multiplicative constants[26],

$$U = -\rho S/\left(\gamma - 1\right), \qquad S = \ln P - \gamma \ln \rho, \tag{11}$$

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

$$\mathbf{v} = U_{\mathbf{u}}^T = \left[\frac{\gamma - S}{\gamma - 1} - \frac{1}{2}\frac{\rho V^2}{P},\ \frac{\rho u_i}{P},\ -\frac{\rho}{P}\right]^T, \tag{12}$$

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

Adapting on J using the entropy variables as adjoints constitutes output-based adaptation. Without the need for a separate system solve to obtain the entropy variables, adapting using entropy variables instead of the output-based adjoint is far less computationally expensive. However, this indicator does not disregard areas of spurious entropy generation that have no effect on a particular engineering output. This may lead to over-refinement, particularly for cases with flow discontinuities.

III. Combined Approach

The previous section illustrated the methodology for obtaining the mesh refinement indicator using either the output-based adjoint or the entropy variables. This section will introduce the combined approach in which both the output-based and entropy-based indicators are obtained separately and then combined through direct, elemental multiplication. We can express that mathematically using the elemental indicators obtained through Eqn. 6 via

$$\eta_{\kappa_H,\text{combined}} = \eta_{\kappa_H,\text{output}} \cdot \eta_{\kappa_H,\text{entropy}}, \tag{13}$$

where $\eta_{\kappa_H,\text{output}}$ is the elemental indicator obtained using the output-based adjoint and $\eta_{\kappa_H,\text{entropy}}$ is the elemental indicator obtained using the entropy variables. It is important to note that the indicators are obtained using the fine-space adjoint, $\boldsymbol{\psi}_h$, and the fine-space entropy variables, $\mathbf{v}_h$.

The decision to combined the indicators together through multiplication is not the only choice as a certain amount of heuristics is involved. The two indicators could have simply been combined using addition. However, the reason elemental multiplication was chosen over addition is the desire to only target for adaptation regions of the domain important to both indicators. This way purely spurious regions of one indicator do not dominate. Addition of the indicators would not do this well: e.g. in the limit of one indicator going to zero, an element could still get refined if the other indicator were high. While multiplication was always performed to generate the final combined indicator, various modifications for obtaining the two indicators were tested. These modifications were made in the interest of improving adaptive efficiency and reducing the overall computational cost.

A. Combined Indicator using a Coarse-Space Adjoint

In Subsection II A, Eqn. 4 requires the computation of the adjoint on a finer finite-dimensional space, $\mathcal{V}_h$. This is an expensive step, especially for large or unsteady problems. To minimize the cost of this step, a new approach to combining the indicators was implemented in which no fine-space output-based adjoint is computed. Instead, both the output-based adjoint and state at the current approximation order are projected down one order from $\mathcal{V}_H$ to $\mathcal{V}_{\tilde{H}}$ prior to obtaining the output-based indicator. Therefore, no new fine-space adjoint needs to be obtained since the fine-space is now at the original approximation order, i.e. in $\mathcal{V}_H$. We still subtract off the output-based adjoint at the coarser space, $\mathcal{V}_{\tilde{H}}$, from output-based adjoint at the current approximation space, $\mathcal{V}_H$, to eliminate pollution from p-dependence of the weak form and non-converged residuals. This leads to a modified output-based adjoint from which the output-based indicator is obtained.

The entropy variables on the fine-space, $\mathbf{v}_h$, are still used to obtain the entropy-based indicator. These two indicators can be combined as in Eqn. 13 despite the fact they are not computed using adjoints at the same approximation space. This is possible because Eqn. 6 includes the restriction of the adjoints to the elements in the current approximation space, $\mathcal{V}_H$. Since the output-based indicator should still target similar general regions of the domain where more refinement is necessary, this less-expensive approach should ideally not lead to significant deterioration of the adaptive

performance.

B. Combined Indicator with a Mask

Building off of the previous approach, an additional approach was implemented with the addition of a mask on the entropy-based adjoint. The purpose of the mask is to limit the weight of the entropy indicator on the mesh adaptation so that the output-based indicator carries more weight. This is accomplished by creating an element-based refinement indicator, i.e. a mask, based on the output adaptation indicator. This mask, μ_{κ_H}, is a vector that is exactly the same size as the output-based indicator $\eta_{\kappa_H,\text{output}}$. The mask is created by examining the relative magnitude of the indicator for each element. Using a user-specified percentage, a subset of the output-based indicator is created. For example, if the user wishes to apply a 25% mask on the entropy-based indicator, μ_{κ_H} is created from the output indicator: it consists of 1 for the 25% elements with the largest $\eta_{\kappa_H,\text{output}}$, and 0 for all other elements. The mask is then multiplied by the entropy-based indicator in an element-wise fashion to obtain the final indicator that ultimately governs the mesh adaptation,

$$\eta_{\kappa_H,\text{combined}} = \mu_{\kappa_H} \cdot \eta_{\kappa_H,\text{entropy}}. \tag{14}$$

The reason for using a masking approach is the mask should completely eliminate elements targeted by the entropy-based indicator which have little influence on the desired engineering output, regardless of the magnitude of the entropy-based indicator for these elements. This will lead to an entirely new set of elements chosen for refinement compared to the standard approach in Eqn. 13.

IV. Adaptive Mechanics

For this research, two different strategies are used to drive the mesh adaptation for each adaptive iteration. The first is a fixed-fraction hanging-node adaptation strategy. The driver of this approach is the elemental adaptive indicator, η_{κ_H}. The term fixed-fraction means a certain fraction, f^{adapt}, of the elements with the largest adaptive indicators are marked for refinement. Marked elements are adapted uniformly, creating a series of hanging nodes. If two neighboring elements vary by more than one level of refinement, additional elements are flagged for refinement. Note that there is no element coarsening performed while using hanging-node adaptation in this research.

The second mesh adaptation approach is a variation of mesh optimization via error sampling and synthesis (MOESS)[23, 27]. The distinguishing feature of MOESS is an error sampling approach for determining the convergence rate tensor of the error on a single element. This local sampling approach determines the optimal local refinement for that element by computing an error indicator $\Delta\mathcal{E}_{ej}$ for a finite number of refinement options j of the element. These indicators estimate the error between the coarse-space solution and that on a given refinement option j. Calculating $\Delta\mathcal{E}_{ej}$ requires projecting the fine-space adjoint, $\boldsymbol{\psi}_{hj}^{p+1}|_{\Omega_e}$ or $\mathbf{v}_{hj}^{p+1}|_{\Omega_e}$ depending on whether the adjoint is output-based or entropy-based, down to the space of the refinement option j and order p, and then back up to order $p+1$ on the original element. Adjoint-weighted residuals are then evaluated on the original element. For the output-based adjoint, mathematically this can be expressed as

$$\Delta\mathcal{E}_{ej} \equiv \left| \mathcal{R}_h^{p+1}\left(\mathbf{u}_h^p, \delta\tilde{\boldsymbol{\psi}}_{hj}^p|_{\Omega_e}\right)\right|, \tag{15}$$

where $\tilde{\boldsymbol{\psi}}_{hj}^p$ is the fine-space adjoint $\boldsymbol{\psi}_h^{p+1}$ projected from $p+1$ to order p on refinement option j, and then back to order $p+1$ on the original element. $\delta\tilde{\boldsymbol{\psi}}_{hj}^p$ is then defined as $\delta\tilde{\boldsymbol{\psi}}_{hj}^p \equiv \tilde{\boldsymbol{\psi}}_{hj}^p - \boldsymbol{\psi}_h^p$. Note, all projections are done in a least-squares sense in element-reference space. More details on this error sampling approach, which does not require modification of the mesh, can be found in previous work [27]. The form of Eqn. 15 that corresponds to the entropy variables is identical except that $\mathbf{v}$ replaces $\boldsymbol{\psi}$. The refinement of each element based on local sampling creates anisotropy and leads to better mesh resolution, and thus better error estimation.

Just as the local elemental error is determined by both the output-based and the entropy-based indicators when using the combined approach, so is the local anisotropy. In this way, the method for detecting anisotropy is still linked to the chosen output. However, this is not necessary since many works on anisotropic adaptation preceding MOESS have used the Hessian of a scalar, e.g. Mach number, instead of the output error itself. Though heuristic, this worked well for many problems. Since the choice of combining the elemental error indicators is already heuristic in this work, so is the choice of detecting anisotropy. Therefore, for some cases using the combined approach, the local anisotropy was determined based only on the output-based indicator. Since the sampling of elements within the boundary layer is especially important for correctly assessing the need for anisotropy, this approach should lead to more output-tailored anisotropy, especially for elements in the boundary

layer. This strategy of using solely the output-based fine-space adjoint to calculate error samples for MOESS was determined empirically to produce better results than using the combined error indicator.

The following is a sequential breakdown of the various steps implemented for each adaptive iteration:

1. Solve the primal problem on the current mesh at order p to obtain $\mathbf{u}_H$. If the adaptation depends on an engineering output, solve the adjoint problem to obtain $\boldsymbol{\psi}_H$.
2. Inject $\mathbf{u}_H$ into a fine space of order $p+1$.
3. If the adaptation depends on an engineering output, solve or iterate the fine-space adjoint problem to obtain $\boldsymbol{\psi}_h$. If the adaptation depends on entropy variables, compute $\mathbf{v}_h(\mathbf{u}_h)$ using Eqn. 12 by either solving the fine-space problem exactly or iteratively smoothing ν_{fine} times to obtain $\mathbf{u}_h$.
4. On each element calculate the adaptive indicator, η_{κ_H}, using Eqn. 6 with $\boldsymbol{\psi}_h$ if the adaptation depends on an engineering output or $\mathbf{v}_h$ if the adaptation depends on entropy variables. For the combined approach, compute indicators based on both the output adjoint and the entropy variables, and then combine them, e.g. via an element-wise multiplication, to obtain a new indicator.
5. Refine the mesh on the elements with the largest indicator using either fixed-fraction hanging node adaptation or MOESS. If using MOESS, calculate local sampling errors for each element using Eqn. 15.
6. Initialize the solution on the adapted mesh with the solution from the previous mesh and return to step 1.

V. Numerical Implementation

A discontinuous Galerkin (DG) finite-element code was used to drive both mesh adaptation strategies. The DG discretization of the compressible Navier-Stokes equations uses the Roe approximate Riemann solver[28] for the inviscid fluxes and form of Bassi and Rebay for the viscous flux[29]. A Newton-GMRES implicit solver with element-line-Jacobi preconditioning is used to obtain the

steady-state solution. While a DG finite element method was used in this work, the practice of using output-based and entropy-variable adjoints to refine and adapt meshes is applicable to general finite element [12**?**] and finite volume methods [4, 30, 31].

For cases that implement the output-based approach alone, the discrete output-based adjoint is obtained using the same element-line-Jacobi preconditioned GMRES solver used for the primal solve. Adjoint consistency is achieved by paying close attention to various discretization and output calculation terms[32–34]. The fine approximation space, $\mathcal{V}_h$, required for the adjoint solution, $\boldsymbol{\psi}_h$, is obtained by reusing the same mesh while increasing the approximation order from p to $p+1$. The number of adjoint iterations on the fine-space is set to a sufficiently large number, typically 50, so that the specified tolerance is met. However, for cases in three dimensions where computations are quite expensive, an iterative method was chosen for the fine-space primal and adjoint problems in lieu of an exact solve due to the excessive expense and previous experimental work illustrating that several smoothing iterations on $\mathcal{V}_h$ yield similar results. For this work $\nu_{\text{fine}} = 5$ element-block-Jacobi smoothing iterations were used for both the primal and the adjoint problems. Finally, the error indicator based on an engineering output is found 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.

The entropy-variable approach differs from the output-based approach in that the fine-space adjoint is not solved for separately since it is unnecessary. Whereas the solution to the adjoint equation requires a fundamentally-different solve, the entropy variables can be calculated directly from the fine-space solution, $\mathbf{u}_h$, by using a change of variables. The number of nonlinear iterations for the fine-space solve is set to a sufficiently large value, 50 iterations, such that the tolerance used for the calculation of the primal problem is met. This leads to an accurate fine-space primal solution. However, just as with the output-based approach, for the three-dimensional cases, an iterative approach was used in lieu of an exact fine-space primal solution. This approach could be used for all of the cases to save a significant amount of computational expense. However, since the focus of this work was primarily on the performance of the combined approach, there was a desire to show the optimal error estimates obtained using the entropy-variable approach.

	Case 1	Case 2	Case 3	Case 4	Case 5
Uniform	X				
Output	X	X	X	X	X
Entropy	X	X	X	X	X
Combined	X	X	X	X	X
Coarse Combined			X		
Combined w/ Mask			X		

Table 1 Types of Adaptation Approaches for Each Case

The standard combined approach in this work implements the methodology used in the output-based approach and the entropy-variable approach. That is, the fine-space output-based adjoint is solved using GMRES, while the entropy variables are obtained using an exact primal solution from the fine-space. While this approach should be the most accurate, it also comes with a computational penalty that makes it more expensive than either of the other approaches. This explains the reason for the modified approaches described in Section III. In both of these modified approaches, no fine-space output-based adjoint is obtained. Instead, the output-based adjoint on the original approximation space, $\mathcal{V}_H$, is recycled. In addition, the fine-space primal solution, used in the generation of the entropy variables, was found using the iterative method with $\nu_{\text{fine}} = 5$ element-block-Jacobi iterations. This further reduces the computational cost of this approach, making it a much more attractive option, as long as the performance relative to the standard combined approach does not significantly deteriorate.

VI. Results

The results in this section compare adaptation using output-based adjoints, the entropy variables, and various applications of the combined approach for both two-dimensional and three-dimensional cases using various geometries. Some of the geometries, initial meshes, and solution parameters are similar to those used in previous works[10, 11, 27], while other cases were designed especially for this research. In total there are five different cases, each using multiple adaptive approaches, shown in Table 1.

A. NACA 0012 in Two-dimensional Inviscid Flow:

The first case is two-dimensional, inviscid flow over the NACA 0012 airfoil with a closed trailing edge and a farfield approximately 40 chord lengths away. The flow is transonic and produces a fishtail shock at the trailing edge, as shown in Figure 1. This shock makes it necessary to add artificial viscosity to stabilize the flow in the shock's immediate vicinity. The presence of the strong shock makes this case an instructive example of where the entropy-variable approach will overly target regions of the domain consumed by the shock. Since shocks are prevalent in many flow simulations in an engineering setting, the tendency of the entropy-based approach to overly target these regions for refinement is a serious issue. However, this issue can be mitigated significantly by combining the indicator obtained from the output-based adjoint with that obtained from the entropy variables.

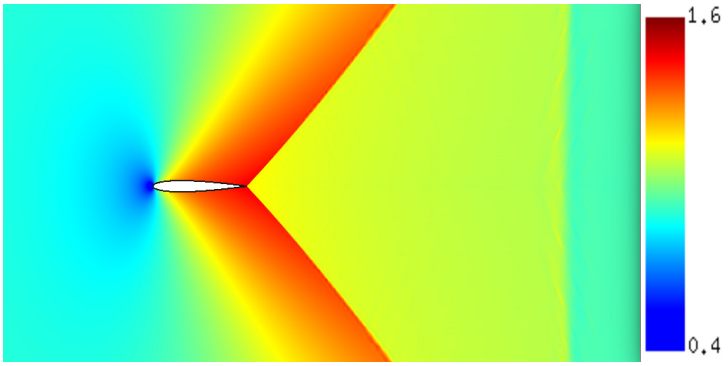


Fig. 1 NACA 0012 $M_\infty = 0.95$, $\alpha = 0°$: Mach number contours (range: 0.4-1.6).

The initial mesh of 572 elements is shown in Figure 2. It is an unstructured (initially structured but not after adaptation), quadrilateral mesh with quartic, $q = 4$, curved elements representing the airfoil geometry. The quadratic solution interpolation order for all of the cases is $p = 2$ and the adaptation fraction is set to $f^{\text{adapt}} = 0.1$ for each of the seven adaptation iterations. This means that for each adaptive iteration, only 10% of the elements are refined. To illustrate the effect this term has on the error convergence for the various methods, an additional set of runs were done with $f^{\text{adapt}} = 0.075$.

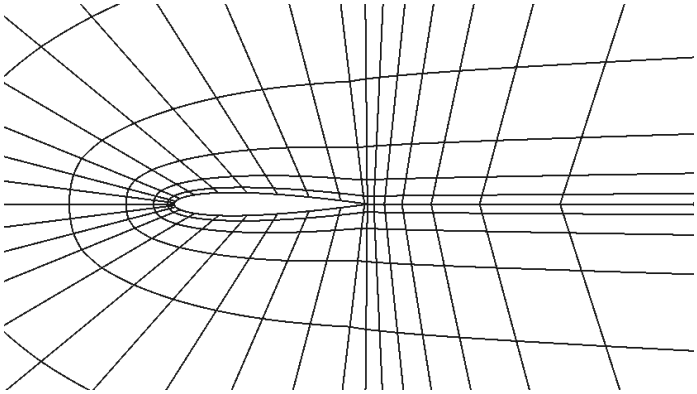


Fig. 2 Initial NACA 0012 airfoil mesh.

For this case, only one engineering output, the drag coefficient, was analyzed. Integrals of the inviscid momentum flux, i.e. the pressure on the airfoil surface, were used to obtain the aforementioned output. Adaptation was driven using adjoints associated with the drag coefficient, lift coefficient, and the entropy variables. In addition, combined output-based and entropy-based adjoint adaptive solutions were generated.

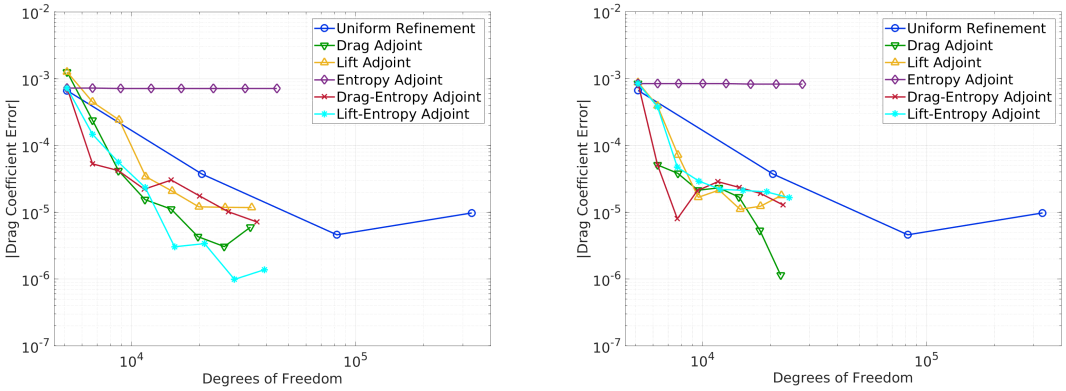
The presence of shocks in inviscid flows leads to the creation of entropy in the regions where shocks are present. This creates a problem for the entropy-variable indicator since it will target regions of the mesh where spurious entropy is generated. The shocks may cause the entropy indicator to continue to refine regions of the mesh to which an engineering output function $J(\mathbf{u})$ may be insensitive. This not only leads to unnecessary mesh refinement in these regions but also prevents regions of the mesh near the geometry to which $J(\mathbf{u})$ is sensitive from being refined since only a certain percentage of cells are refined in each adaptive iteration. By combining the output-based adjoint indicator with the entropy-variable indicator through direct, elemental multiplication, this issue can be alleviated because the output-based adjoint does not unnecessarily target regions of spurious entropy generation that do not affect $J(\mathbf{u})$.

Figure 3 shows the convergence of the drag coefficient for various adaptive indicators and uniform refinement using both $f^{\text{adapt}} = 0.1$ and $f^{\text{adapt}} = 0.075$. The truth drag coefficient was obtained from a case using $f^{\text{adapt}} = 0.1$ with a solution approximation order of $p = 5$ on a mesh obtained by uniformly refining the finest drag coefficient adapted mesh twice. A second truth solution was

obtained using the same uniformly refined mesh with $p = 4$. The difference in the drag coefficient between these two solutions is 6×10^{-9}. Clearly, the creation of entropy due to the fishtail shocks prevents the entropy-based adaptation from properly refining the mesh to produce a more accurate drag coefficient. However, the combined adjoint approaches converge as the mesh is refined since they are not as susceptible to unnecessary refinement in regions that do not affect the drag coefficient. When $f^{\text{adapt}} = 0.1$, the combined approach that incorporates the lift-based adjoint actually produces a better drag coefficient, while the combined approach that used the drag-based adjoint does not perform as well as the drag-based approach. However, this is not a universal result for this case, since the errors are sensitive to solver parameters, including coefficients in the artificial-viscosity-based shock-capturing strategy [35]. That is, adjustments to the shock capturing smoothness indicator parameters can make the combined approach perform better. Despite changes to the shock capturing methods, however, the drag coefficient error convergence using the entropy-based indicator never improves. In addition, when $f^{\text{adapt}} = 0.075$, the combined lift-entropy case does not perform nearly as well. This is the only method that changes significantly when f^{adapt} is changed. It is reasonable to expect the error convergence to change since f^{adapt} will effectively change the relative meshes at each adaptive iteration. Thus, the significant change in the error convergence for the combined lift-entropy case is noteworthy, but not entirely surprising.

Figure 4 compares the meshes after seven adaptation iterations for the error indicator strategies in Figure 3 that use $f^{\text{adapt}} = 0.1$. The output-based adaptation methods do not target the fishtail shock, outside of the region directly near the trailing edge, whereas the entropy adaptation focuses solely on the fishtail shock and does not refine the mesh anywhere else. This explains why in Figure 3 there is no change in drag coefficient error with each adapted mesh for the entropy-based indicator case. Since the mesh is barely refined near the airfoil surface, particularly the leading edge, the drag coefficient error never improves. The combined drag and entropy-variable approach yields a much-improved mesh compared to just using the entropy variables alone. However, there is still some unnecessary refinement due to the fishtail shock, because of how strong the entropy-based indicator is in that region. This is likely the reason why this combined approach does not perform as well as the drag-based alone approach. The combined lift and entropy-variable approach does

not have as much refinement in the fishtail shock, which might explain why it does much better than the combined approach that uses the drag-based adjoint. One option to remedy the issue with the drag-based combined approach is to use a mask on the entropy indicator so that its effect is not as pronounced. This option will be investigated for a later result.



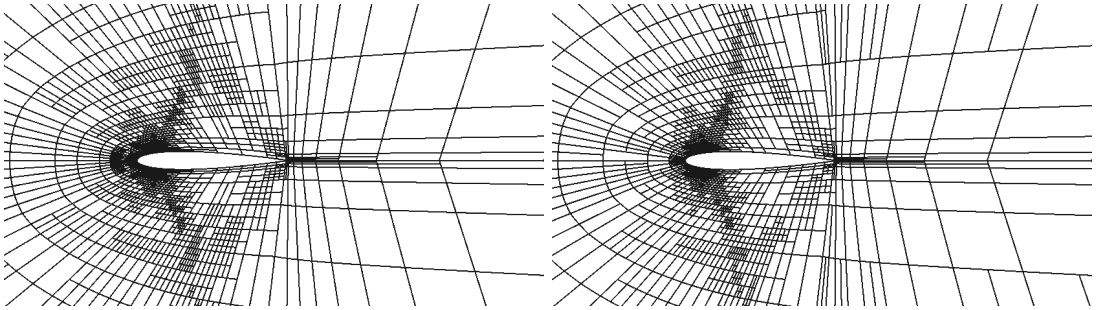
(a) $f^{\text{adapt}} = 0.1$ (b) $f^{\text{adapt}} = 0.075$

Fig. 3 NACA 0012 $M_\infty = 0.95$, $\alpha = 0°$: Comparison of drag coefficient convergence histories for various adaptation methods.

B. NACA 0012 Wing in Inviscid Flow: $M_\infty = 0.4$, $\alpha = 3°$

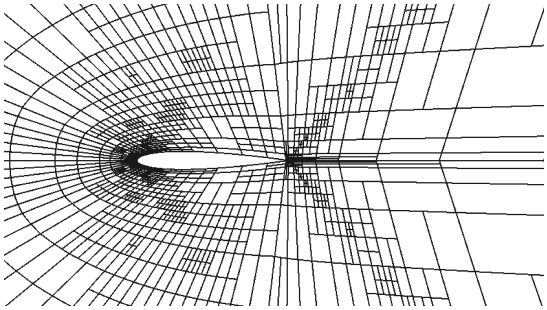
To demonstrate whether the combined approach is effective at higher dimensions, the next case is a NACA 0012, untapered, untwisted wing with a closed trailing edge and a rounded wing tip. The wing has an aspect ratio of 10 and is inside a domain with a farfield 40 chord lengths away. The unstructured mesh is composed of cubic, $q = 3$, hexahedral elements that curve to match the wing geometry. Quadratic solution interpolation, $p = 2$, was used in the discretization, but for these runs the adaptation fixed fraction was lowered to $f^{\text{adapt}} = 0.05$. Artificial viscosity stabilization [35] was added to the solution to provide stability in the vicinity of the trailing-edge vortices. The cores of the trailing vortices experience very low pressures and become singularities in the inviscid limit, which is more closely approximated as the mesh is adapted in these regions. Without the addition of artificial viscosity stabilization, these regions then cause convergence problems for the high-order solver [11].

The two engineering outputs that are considered for this case are drag and lift coefficients.

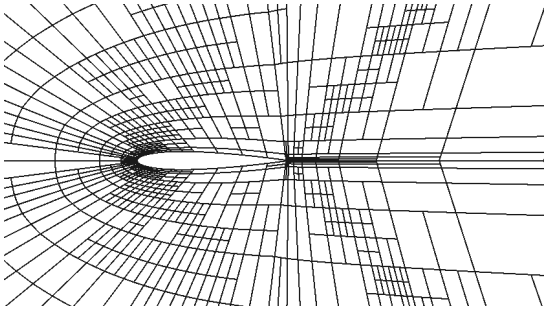


(a) Drag Adaptation

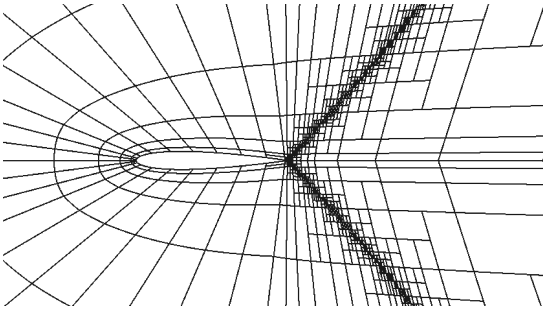
(b) Lift Adaptation



(c) Drag-Entropy Adaptation



(d) Lift-Entropy Adaptation



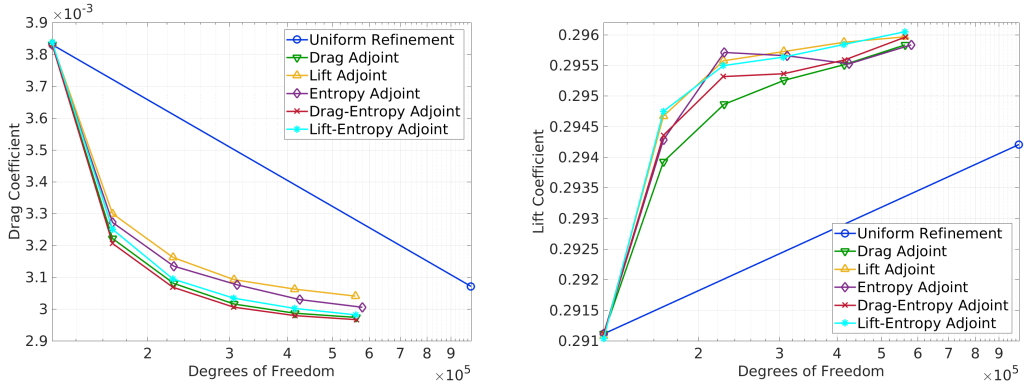
(e) Entropy Adaptation

Fig. 4 NACA 0012 $M_\infty = 0.95$, $\alpha = 0°$: Meshes after seven adaptation iterations for various error indicators that use $f^{\text{adapt}} = 0.1$

Adjoint solutions associated with these two outputs were used to drive separate adaptation runs. Additionally, the entropy-variable indicator, as well as the combined entropy and output-based adjoint indicator runs were tested. Figure 5 shows the results of these adaptation runs, along with a uniform refinement run for comparison. No truth solutions were obtained due to the computational expense of such runs. Consequently, what is shown in these figures is not the error but the actual lift and drag coefficients.

For the drag coefficient, the drag adjoint and combined drag and entropy variables behave

comparably and converge to a very similar value, assuming this value is close to the exact solution. It is possible that the combined approach might actually be slightly outperforming the output-based approach. However, as previously discussed, output-based adaptation is not always optimal because of adjoint singularities at the leading and trailing edges that can distract the adaptation. The entropy-variable approach does not converge to the same degree and particularly underperforms for the coarser meshes. Similarly to what was observed for the drag coefficient, the lift-based adjoint approach and combined lift and entropy-variable approach perform comparably regarding their ability to predict the lift coefficient output. The entropy-variable indicator by itself is much more erratic and does not perform nearly as well. Not unexpected is that the lift-based adjoint approach does not produce as accurate of a drag coefficient as the drag-based adjoint and vice versa. Since the drag-based adjoint specifically refines the mesh to produce the most accurate drag coefficient, it is reasonable to assume that the lift-based adjoint approach, while it produces the better lift coefficient, does not produce as accurate a drag coefficient. This same logic is the reason the combined approach that uses the drag-based adjoint does not produce as accurate of a lift coefficient compared to the combined approach that uses the lift-based adjoint.



(a) Drag Output

(b) Lift Output

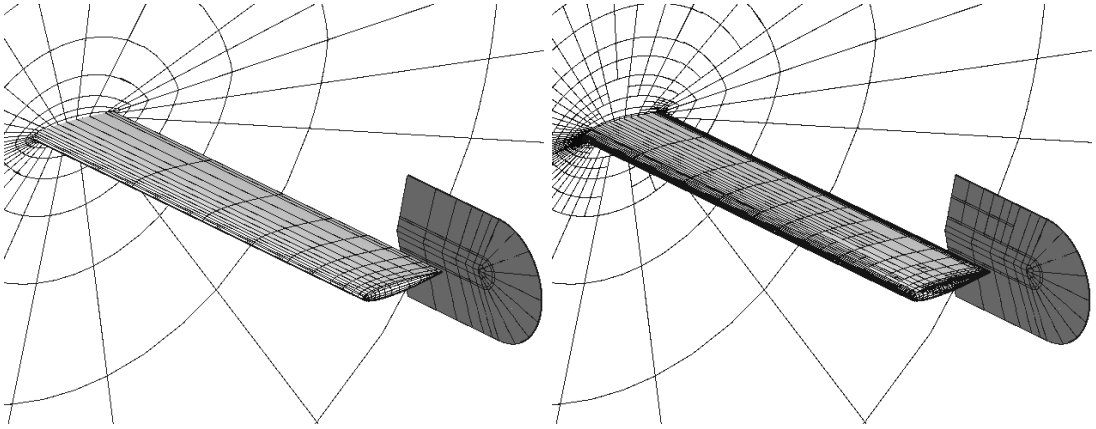
Fig. 5 NACA 0012 Wing $M_\infty = 0.4$, $\alpha = 3°$: Comparison of output convergence histories for various adaptation methods.

An explanation for why the entropy-variable indicator does not perform as well is its propensity to refine the mesh in areas consumed by the wing tip vortex. From theory and experimental work [36] it is known that wakes form behind wings. This occurs due to the natural extension of

the boundary layer from the wing surface, as well as streamwise vorticity that is shed due to lift distribution variation. This case is inviscid so there is no boundary layer. However, there still is shedding of streamwise vorticity, particularly at the wing tip. The numerical results presented in this paper show a very concentrated vortex emanating from the wing tip. Since the under-resolved wing tip vortex leads to spurious entropy generation, the entropy-variable indicator targets areas of the mesh through which the vortex travels.

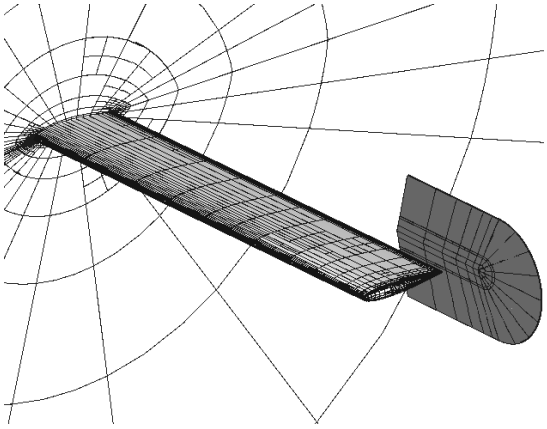
Figure 6 shows the initial mesh, as well as the final meshes, after five adaptive iterations for the various adaptive strategies. The cut plane in these images is not far behind the wing trailing edge. The output-based indicators refine the mesh near the wing surface and do not significantly refine the mesh downstream of the wing trailing edge. However, the entropy-variable indicator refines the regions of the mesh engulfed by the wake, especially where the wing tip vortex is. While areas of the mesh, particularly those near the leading edge, are also refined, the entropy-variable indicator sacrifices refinement on the upper surface of the wing due to the excessive refinement in regions of the mesh where the wing tip vortex is. The combined output-based and entropy-based adjoint indicators have some refinement due to the wing tip vortex, but it is not nearly excessive enough to lead to poor adaptation performance, as illustrated by the results in Figure 5.

The impact of the wing tip vortex can be further examined via the results in Figure 7, which show cuts of the mesh at the wing tip. Again, the output-based adjoint indicators focus all of the refinement near the wing tip surface, while the entropy-variable indicator focuses refinement downstream of the wing trailing edge due to the wing tip vortex. The combined drag and entropy-variable indicator refines the region of the mesh consumed by the wing tip vortex near the wing tip surface but does not excessively refine the mesh further downstream. It is reasonable to surmise that some refinement due to the wing tip vortex is necessary to accurately predict the drag coefficient, but not nearly to the extent the entropy-variable indicator indicates. The combined drag and entropy-variable indicator does a much better job balancing refinement between regions in the wake and near the airfoil surface. This is even more evident for the combined lift and entropy-variable indicator case, where the refinement due to the wing tip vortex is even less pronounced since the wing tip vortex under-resolution does not contribute to the lift coefficient error to the same degree

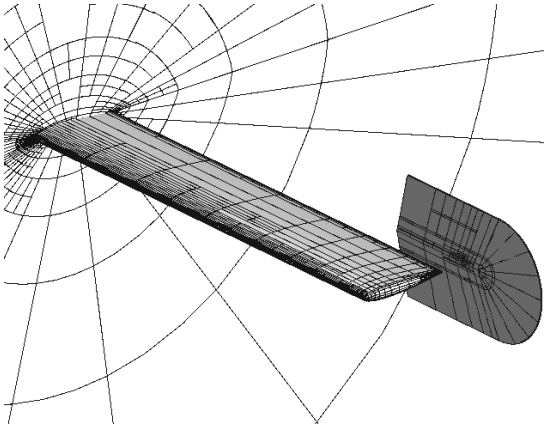


(a) Initial Mesh

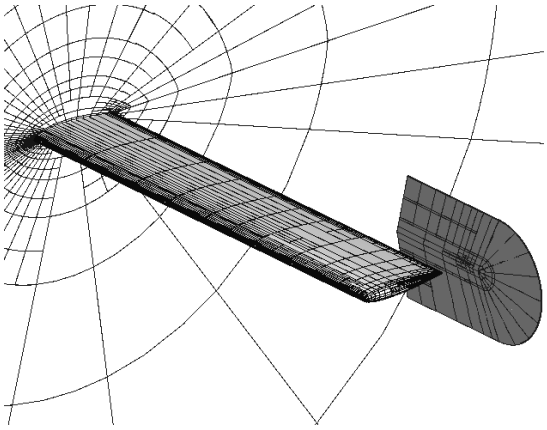
(b) Drag Adaptation



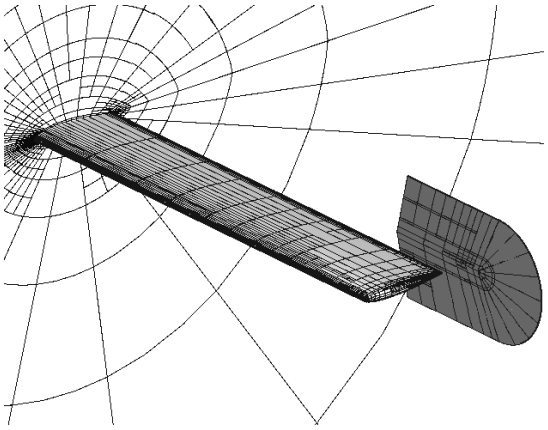
(c) Lift Adaptation



(d) Entropy Adaptation



(e) Drag-Entropy Adaptation



(f) Lift-Entropy Adaptation

Fig. 6 NACA 0012 Wing $M_\infty = 0.4$, $\alpha = 3°$: Initial mesh and meshes after five adaptation iterations for various adaptive strategies.

as to the drag coefficient error.

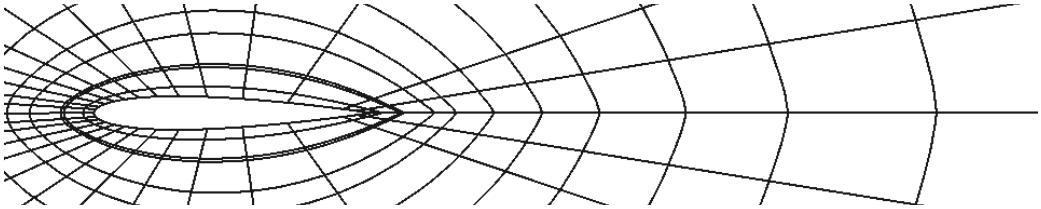
C. Diamond Airfoil in Two-Dimensional Inviscid Supersonic Flow: $M_\infty = 1.5$, $\alpha = 2^\circ$

The next case is two-dimensional, inviscid flow over a thin diamond airfoil with a thickness ratio of 0.05. The purpose of this example is to test the various modifications to the combined approach, particularly the masking on the entropy-based indicator. This is also a relatively challenging case for the adaptive indicator since there are numerous flow discontinuities present. The grid is a square with a side length of 10 chord lengths. The freestream flow is supersonic, which produces shocks emanating from both the top and bottom surfaces of the airfoil, as visible in Figure 8. However, because the airfoil is thin enough to produce weak shocks, no artificial viscosity was added to the solution. This is ideal since it is desirable to not have artificial viscosity present in the simulation since it might adversely affect the entropy-based indicator.

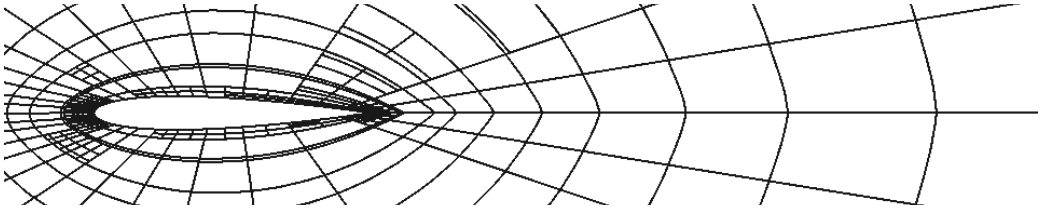
The airfoil is situated slightly off from the center of the domain so that the shocks emanating from the airfoil do not directly impact the corner of the mesh domain. Figure 9 shows the initial, unstructured, quadrilateral mesh made up of linear, $q = 1$, elements. As evident from the figure, the initial mesh is quite coarse. Just as with the inviscid NACA 0012 case, the solution approximation order for all of the cases is $p = 2$. All of the cases ran for exactly fourteen adaptive iterations, with the last iteration resulting in a mesh made up of approximately 70,000 degrees of freedom. This was done by varying f^{adapt} for each case. In addition, for these runs, a new parameter, $f^{\text{error}} = 0.95$ was set. This parameter, called the adaptation fixed error fraction, indicates the fraction of total error (sum of indicators) that will be targeted for refinement during adaptation. Specifically, the fraction of elements targeted for adaptation, $\tilde{f}$, is not constant in this approach, but instead chosen as the value that makes the sum of the error indicators in the targeted elements (starting with those possessing the highest errors) a fixed fraction of the total error indicator sum. However, the actual fraction of elements adapted will still be limited by f^{adapt}, so that

$$\text{fraction of elements adapted } = f = \min(\tilde{f}, f^{\text{adapt}}) \tag{16}$$

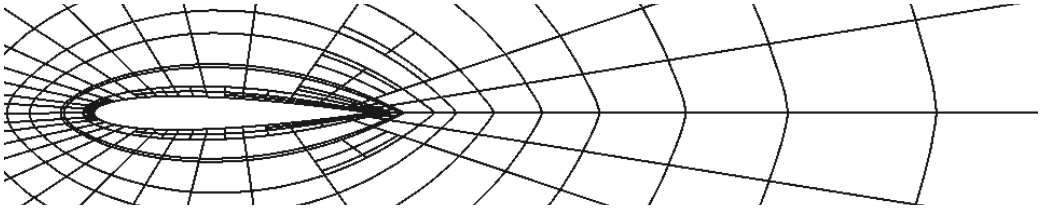
For this case, the drag coefficient and the lift coefficient were again used as outputs to govern the mesh adaptation. Adaptation runs using adjoints based on these two outputs, as well as runs using



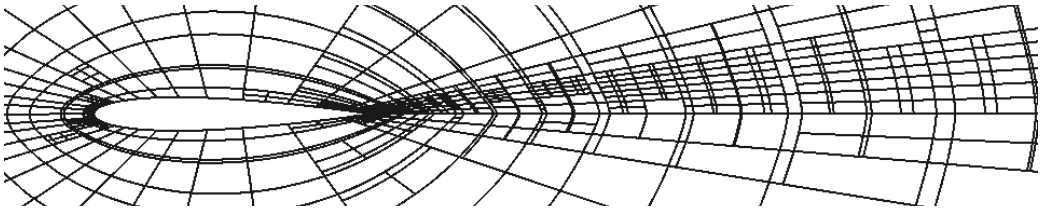
(a) Initial Mesh



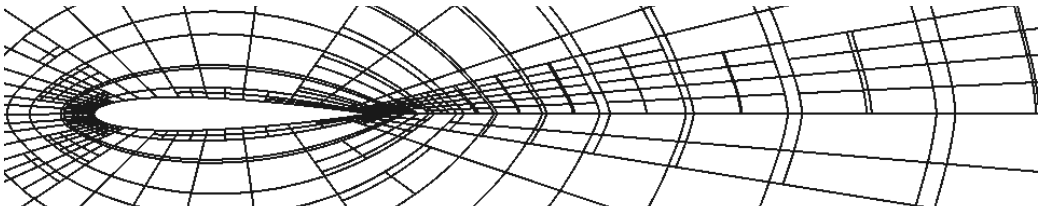
(b) Drag Adaptation



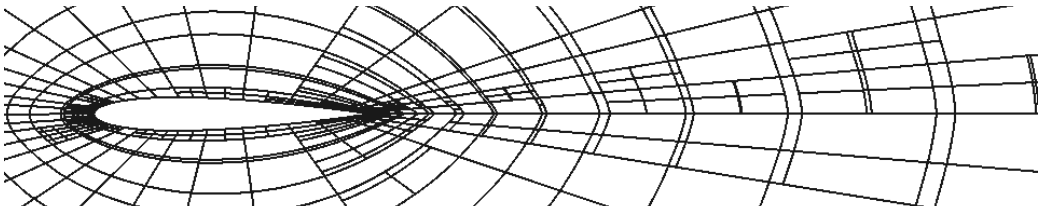
(c) Lift Adaptation



(d) Entropy Adaptation



(e) Drag-Entropy Adaptation



(f) Lift-Entropy Adaptation

Fig. 7 NACA 0012 Wing $M_\infty = 0.4$, $\alpha = 3^\circ$: Cut-plane at wing tip from initial mesh and meshes after five adaptation iterations for various error indicator strategies.

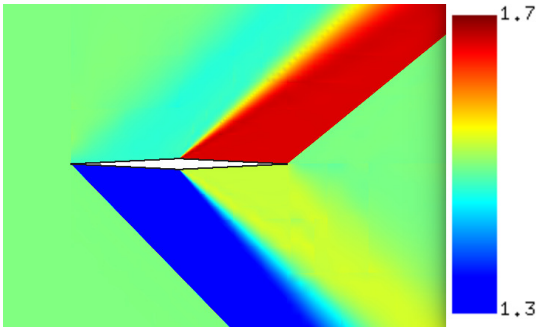


Fig. 8 Diamond $M_\infty = 1.5$, $\alpha = 2^\circ$: Mach number contours (range: 1.3-1.7).

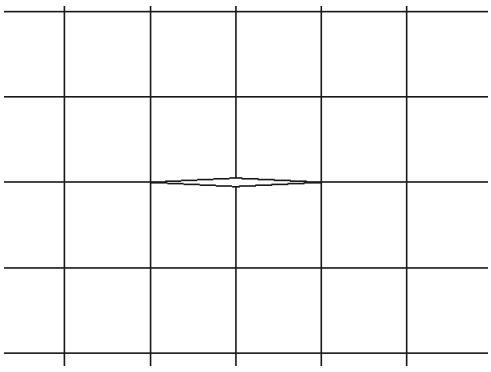
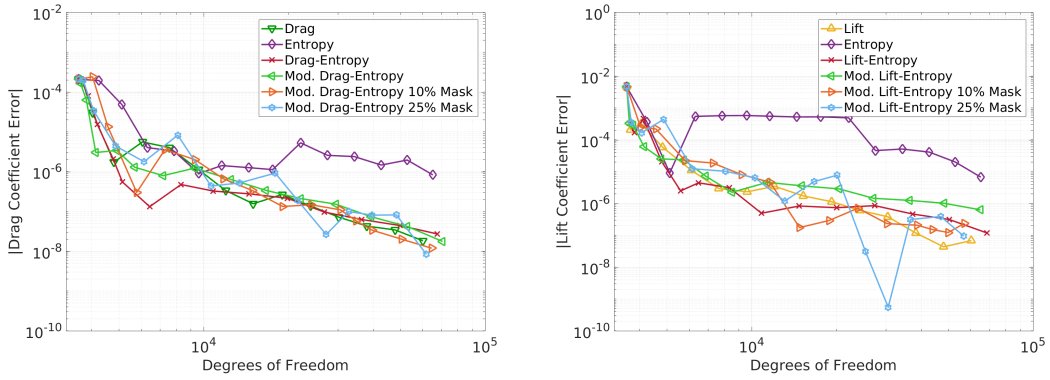


Fig. 9 Initial diamond airfoil mesh.

both the entropy-based adjoint and the combined entropy and output-based adjoint were generated. In addition, three modified forms of the combined approach were considered for this case. The first modified approach, described in Subsection III A, entails not solving for the fine-space output-based adjoint. Instead, the adjoint and the state are projected down one order, to $p = 1$ for this case, so that the output-based indicator can be obtained at the $p = 2$ space. The second and third modified approaches build off of the first modified approach by adding a mask on the entropy-based indicator using different masking percentages, in this case 10% and 25%.

Figure 10 shows the drag and lift coefficient convergence results for all of the various error indicator strategies. As previously discussed, by varying f^{adapt} for each case, we were able to run each case with fourteen adaptive iterations that produced similar mesh sizes. The truth outputs for both lift and drag coefficient were obtained from a case at $p = 3$ on a mesh obtained by uniformly refining the finest output-adapted mesh. Another set of truth solutions was obtained by uniformly refining

the initial truth solution and running a new case at $p = 3$. The difference in lift and drag coefficient between these two truth solutions was on the order of 2×10^{-9}. The modified combined approach, in which no fine-space output-based adjoints were solved for and the fine-space primal problem was approximated with an iterative solver, shows slightly downgraded performance, especially in terms of predicting lift coefficient. However, the performance of this modified approach improves significantly with the addition of the mask on the entropy-based indicator. Unfortunately, no one particular masking percentage yields superior results for both lift and drag error estimation. The 10% mask is generally superior in terms of measuring drag coefficient. However, for lift coefficient error estimation, the 25% mask appears to do better. It crosses the truth solution, which explains the sudden drop in error magnitude.



(a) Drag Output

(b) Lift Output

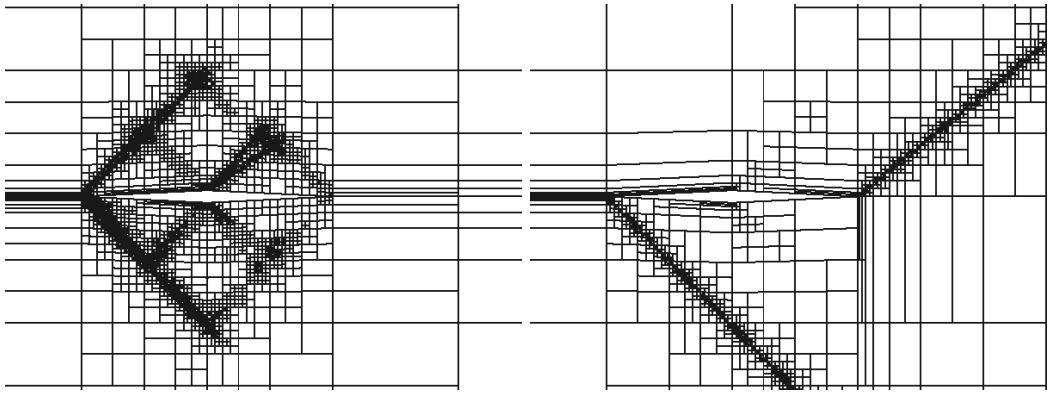
Fig. 10 Diamond $M_\infty = 1.5$, $\alpha = 2°$: Comparison of output convergence histories for various adaptation methods.

Figure 11 presents the meshes after the final adaptation iteration for the various error indicator strategies that, outside of the entropy-based adjoint alone case, all include indicators based on the drag-based adjoint. The entropy-based adjoint indicator mostly focuses on the discontinuities emanating from the leading edge and trailing edge of the diamond. This refinement extends unnecessarily all the way to the farfield boundaries. This accounts for why the error convergence of the entropy-based approach produces the least desirable error convergence. However, the entropy-variable approach still produces a better error estimate compared to the example in Subsection VI A where the error never improved as the mesh was refined. In that case, the shocks were

nowhere near the leading edge, leading to no refinement there. For this supersonic case, there are shocks/expansions close to the airfoil that do get refined. Thus the error estimate predicted by the entropy-based approach is much better for this case. The drag-based adjoint indicator does not suffer from this issue, instead producing regions of refinement that when examined as a whole, resemble the shape of a diamond. The regions of refinement that originate near the airfoil and then travel downstream are necessary for good error estimation. However, by examining the four meshes obtained using a variation of the combined approach, it is reasonable to conclude that the regions of refinement that travel upstream may not be necessary for accurate error estimation. This is because of the singularities in the output-based adjoint in the vicinity of the leading-edge stagnation streamline. The four combined approaches produce fairly similar drag coefficient error estimates, despite their meshes looking quite a bit different. The two masking cases, in particular, show a significant difference in refinement on the top of the airfoil. It is reasonable to assume that the refinement from the shocks above the airfoil is necessary since only at the last adaptive iteration does the 25% mask case outperform the 10% case. Of course, it is clear that the discontinuity emanating from the nose that is below the airfoil is far more important since this region of the mesh is refined regardless of the method. Overall, a lower masking percentage appears to produce much more reliable meshes.

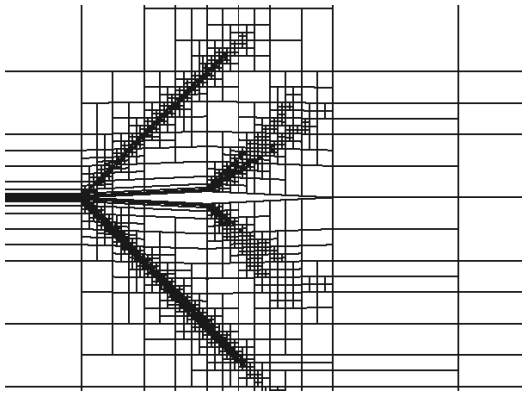
Cell-wise adaptive indicator plots on meshes after seven adaptive iterations are shown in Figure 12. The cases shown in this figure are identical to those shown in Figure 11, the main difference being these meshes are after seven adaptive iterations, not fourteen. For all cases, the area of the mesh that gets refined first is the shock emanating from the nose that is below the airfoil. After seven iterations this region is not nearly as refined in the entropy variables case compared to the drag-based adjoint case. This is because the mesh is refined in regions where the discontinuity is far from the airfoil. This region appears to be the most refined for the masking cases, which might explain their slight improvement in performance. Overall though most of the cases produce similar drag coefficient error levels despite the clear differences in these cell-wise adaptive indicator plots.

Figure 13 shows meshes after the final adaptation iteration for the various error indicator strategies that rely on the lift-based adjoint. For lift coefficient estimation, the modified combined approach does not perform nearly as well without the mask. The issue with this case is that many

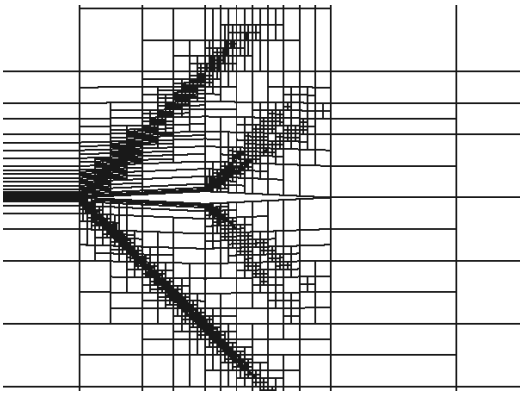


(a) Drag Adaptation

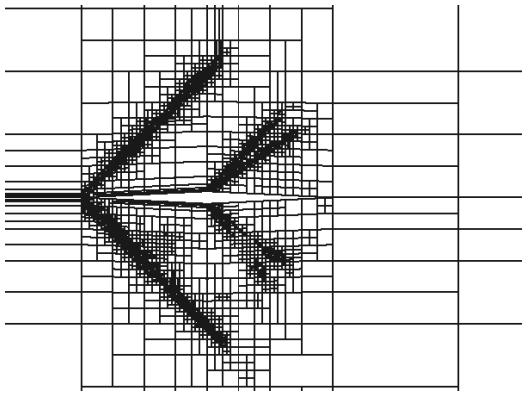
(b) Entropy Adaptation



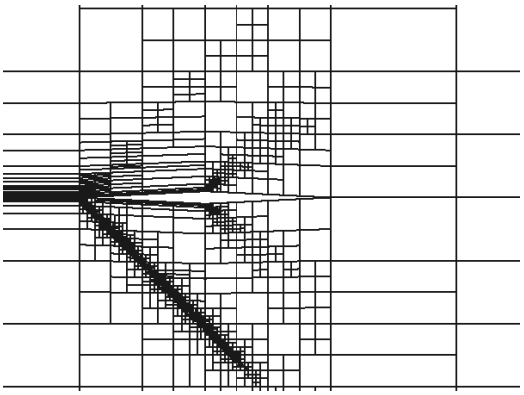
(c) Drag-Entropy Adaptation



(d) Mod. Drag-Entropy Adaptation



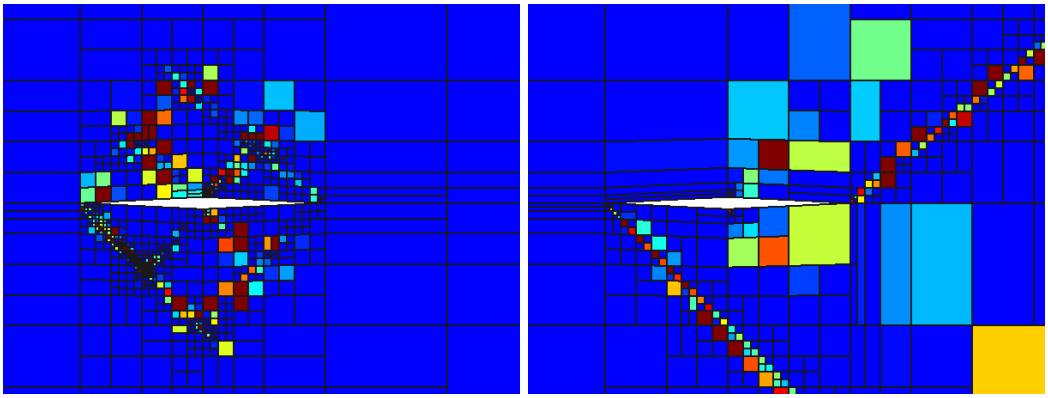
(e) Mod. Drag-Entropy Adaptation 10% Mask



(f) Mod. Drag-Entropy Adaptation 25% Mask

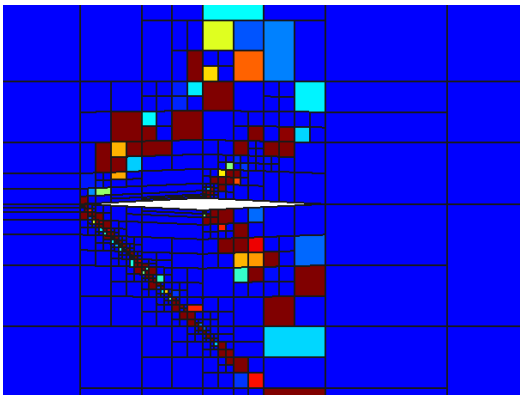
Fig. 11 Diamond $M_\infty = 1.5$, $\alpha = 2^\circ$: Final mesh after adaptation iterations for various error indicator strategies, most of which require the drag coefficient adjoint.

of the elements, particularly those in the region above the airfoil, have high aspect ratios. Since hanging node adaptation requires certain ratios between cell dimensions, the very small grid cells near the top point and leading tip of the diamond airfoil lead to the propagation of large aspect ratio

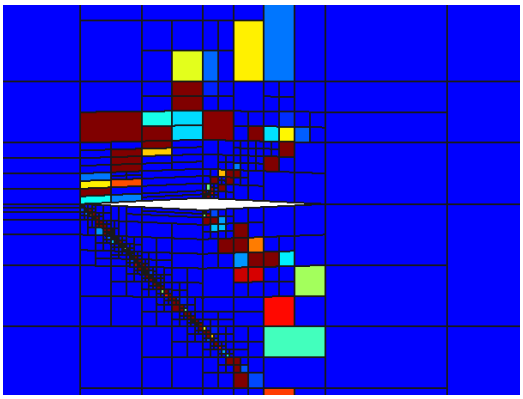


(a) Drag Adaptation

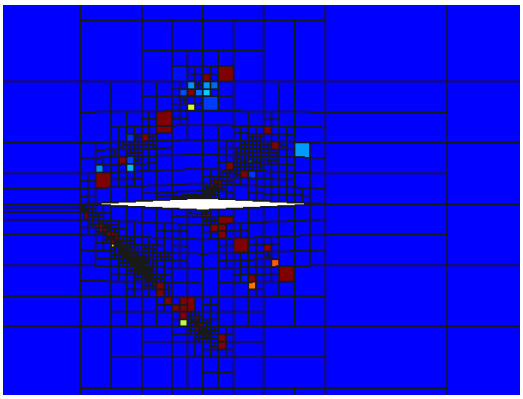
(b) Entropy Adaptation



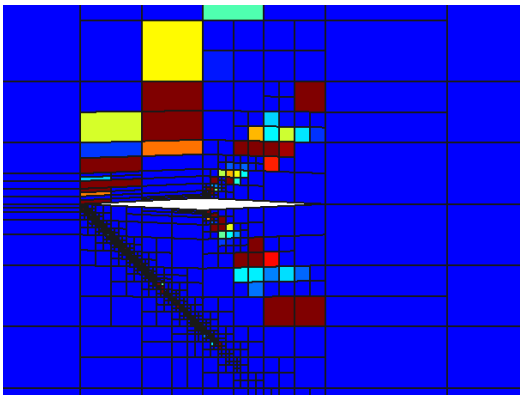
(c) Drag-Entropy Adaptation



(d) Mod. Drag-Entropy Adaptation



(e) Mod. Drag-Entropy Adaptation 10% Mask



(f) Mod. Drag-Entropy Adaptation 25% Mask

Fig. 12 Diamond $M_\infty = 1.5$, $\alpha = 2°$: Adaptive indicator at seven adaptation iterations for various error indicator strategies, most of which require the drag coefficient adjoint.

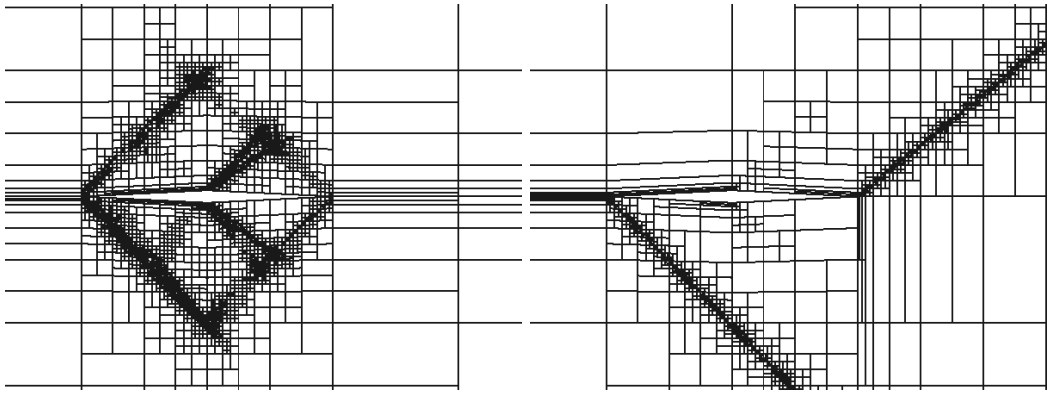
cells above the airfoil. These cells do a poor job of capturing the very thin discontinuity. The case with the 25% has many poor aspect ratio cells too, which is likely why the lift coefficient estimation is so erratic with each adaptive iteration. The high aspect ratio cells are not as prevalent in the output-based case and the combined case with the 10%, hence the improvement in the estimation of the lift coefficient.

It is important to note that many masking percentages, as well as various values for f^{adapt} and f^{error}, were studied using this case. Additionally, masking was attempted on the output-based indicator as opposed to the entropy-based indicator. The authors have chosen to include combinations of inputs that yielded the best results for this particular case. However, it is clear that no one particular combination of inputs always yields the best error convergence. Part of the issue in determining the optimal approach is the inherent limitations of the hanging node adaptation discussed in the previous paragraph. To eliminate these limitations, the subsequent results will focus on adaptation using MOESS.

D. NACA 0012 in Two-Dimensional Viscous Flow: $M_\infty = 0.5$, $\alpha = 5^\circ$, $Re = 5000$

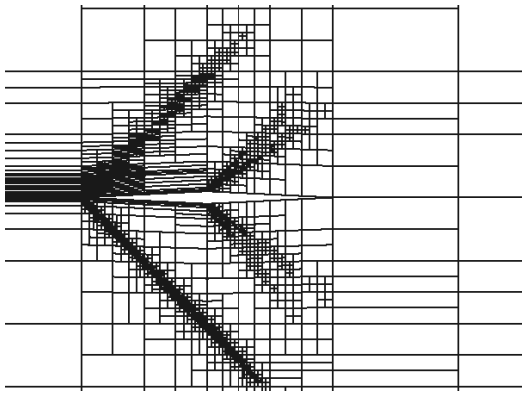
To illustrate how using MOESS adaptation compares to hanging node adaptation, we consider two-dimensional, viscous flow over the NACA 0012 airfoil with a closed trailing edge. This case differs from the previous examples in that it is a viscous simulation. This allows us to demonstrate the effectiveness of the combined approach for a problem with much more challenging governing flow equations. The initial meshes are shown in Figure 14. The initial mesh used with hanging node adaptation is a quadrilateral mesh with quartic, $q = 4$, curved elements representing the geometry. The mesh is made up of 818 elements and the farfield is approximately 40 chord lengths away. The extra refinement near the surface is necessary to enable a solution on the coarsest mesh. The adaptation fraction was set to $f^{\text{adapt}} = 0.1$ and the adaptation fixed error fraction was set to $f^{\text{error}} = 1$ for each of the nine adaptation iterations. Finally, the solution was set to $p = 2$ for all cases.

The initial mesh used with MOESS consists of 356 unstructured triangles. The farfield is over 2000 chord-lengths away, and curved elements of geometry order $q = 4$ are used to represent the

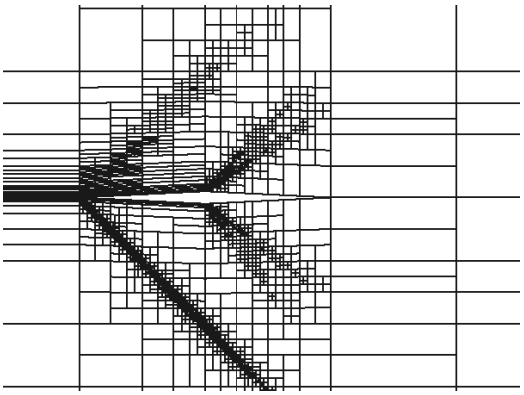


(a) Lift Adaptation

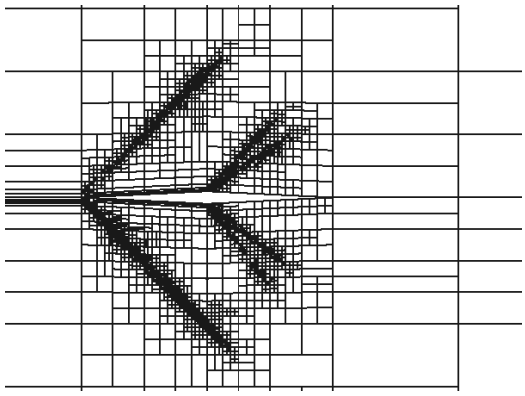
(b) Entropy Adaptation



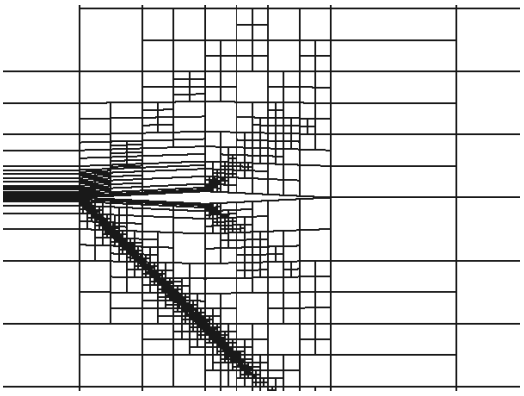
(c) Lift-Entropy Adaptation



(d) Mod. Lift-Entropy Adaptation



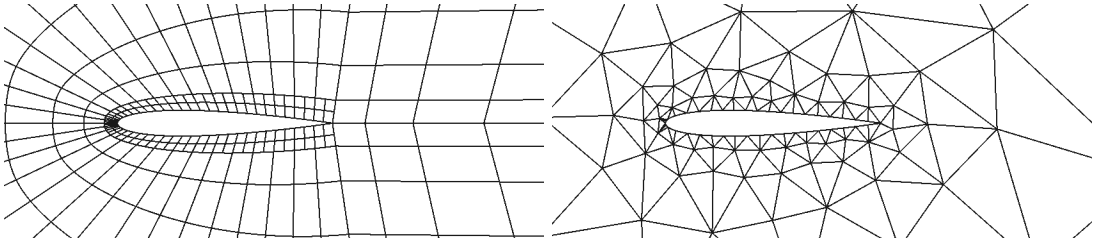
(e) Mod. Lift-Entropy Adaptation 10% Mask



(f) Mod. Lift-Entropy Adaptation 25% Mask

Fig. 13 Diamond $M_\infty = 1.5$, $\alpha = 2°$: Final mesh after adaptation iterations for various error indicator strategies, most of which require the lift coefficient adjoint.

airfoil. The cases are run with a solution interpolation order of $p = 2$ with the following six degrees of freedom targets: 2000, 4000, 8000, 16000, 32000, and 64000. At each `dof` target, fifteen solution iterations are performed before moving to the next `dof` target.

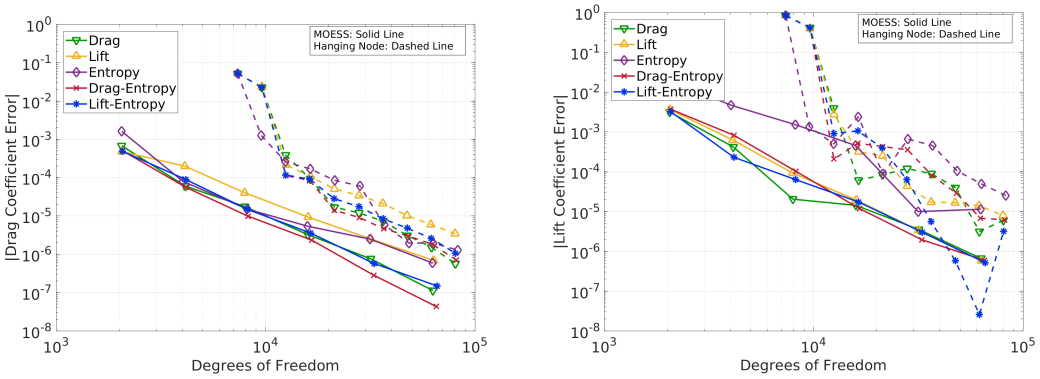


(a) Hanging Node (b) MOESS

Fig. 14 Initial viscous NACA 0012 airfoil meshes

For both sets of adaptation strategies, the drag coefficient and lift coefficients were used as outputs to drive the mesh adaptation. Adaptation runs for these outputs, the entropy variables alone, and the combined entropy and output-based adjoint were considered. None of the previously detailed modifications were made to the combined approach in this case. Figure 15 shows the convergence results for all of the error indicators using both sets of mesh adaptation strategies. The solid lines represent solutions using MOESS, while the dashed lines represent solutions using hanging node adaptation. The output and `dof` values reported from the cases incorporating MOESS are averages over the last 5 solution iterations at each target `dof`. For the viscous case, the truth outputs for both the lift and drag coefficients were obtained by refining the final output-adapted mesh and running cases at orders $p = 4$ and $p = 5$. The difference in drag coefficient between these two sets of cases was of the order 4×10^{-8}, while the difference in lift coefficient was of the order 8×10^{-8}. For the MOESS case, truth solutions were run at $p = 3$ and $p = 4$. The difference in outputs between these two sets of cases was of the order 1×10^{-8}.

Unsurprisingly the MOESS adapted cases for the most part perform much better than the hanging node adapted cases since MOESS has the ability to coarsen regions of the mesh that do not affect the output error estimation. When implementing hanging node refinement, the error indicator strategies' drag coefficient error convergence does not vary significantly, outside of the lift-based adjoint approach whose relative error estimation is far inferior. However, for MOESS a clear distinction is present. The entropy variables based adaptation solution is quite poor, comparable to the lift-based adjoint approach, while the combined approach that incorporates the drag-based adjoint yields better performance than the drag coefficient based adaptation solution. In addition,



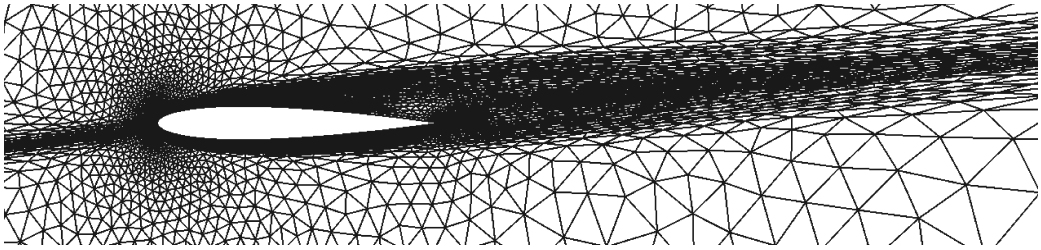
(a) Drag Output

(b) Lift Output

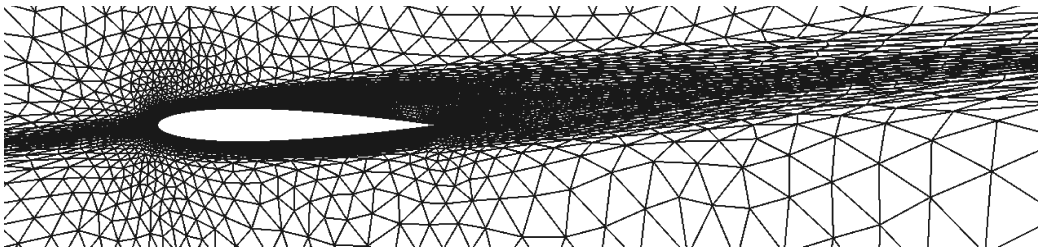
Fig. 15 NACA 0012 $M_\infty = 0.5$, $\alpha = 5°$, $Re = 5000$: Comparison of output convergence histories for various adaptation methods using both hanging node and MOESS adaptation.

combined approach that uses the lift-based adjoint compares very well to the drag-based approach. For the lift coefficient error convergence, the combined approach using hanging node adaptation is superior to output-based hanging node adaptation. However, for MOESS there is not much difference between the combined approach and output-based approach.

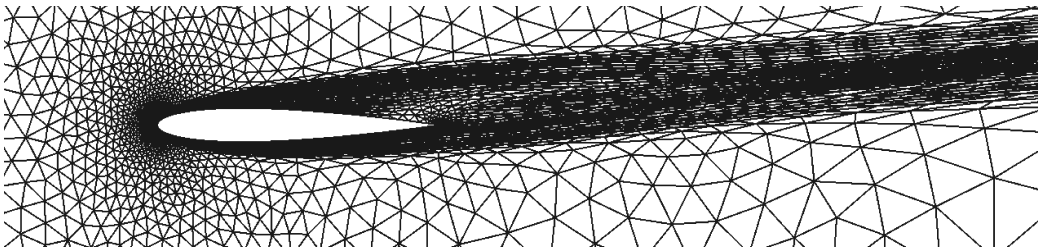
Figure 16 presents the meshes at 64000 `dof` for the cases that used MOESS adaptation. The individual output-based approaches target regions of the domain near the airfoil, as well as the portions of the wake near the airfoil and areas near the stagnation streamline. In fact, there is substantial refinement ahead of the stagnation streamline that is multiple chord lengths long. The entropy-based adjoint approach does not target regions of the mesh well ahead of the stagnation point. Instead, it targets the entire wake, all the way to the farfield boundary since there is substantial spurious entropy generation in an under-resolved wake. The entropy indicator also leaves the aft portion of the upper surface of the airfoil relatively unrefined. The combined approach achieves a balance between refining ahead of the airfoil and in the wake. The meshes obtained from these approaches are refined in the wake, but not to the excessive extent present in the mesh produced from the entropy-based approach. In addition, these meshes have refinement ahead of the stagnation streamline, but the refinement does not extend well ahead of the airfoil as it does for the output-based approaches. This balance in refinement for this particular case clearly shows the benefit of using a combined approach to govern the mesh adaptation.



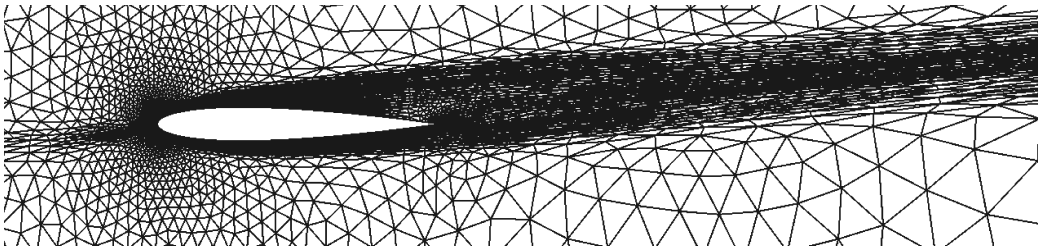
(a) Drag Adaptation



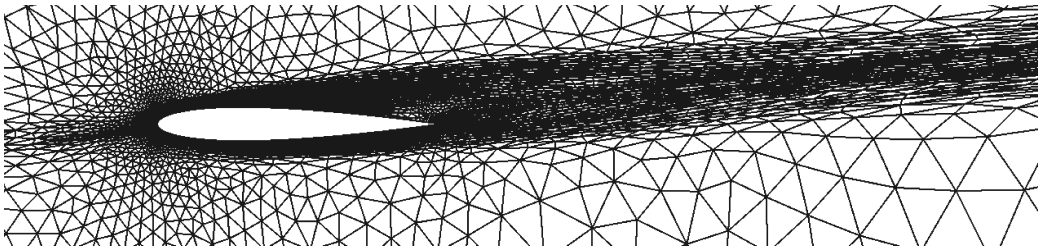
(b) Lift Adaptation



(c) Entropy Adaptation



(d) Drag-Entropy Adaptation



(e) Lift-Entropy Adaptation

Fig. 16 NACA 0012 $M_\infty = 0.5$, $\alpha = 5^\circ$, $Re = 5000$**: Meshes after final adaptation iteration (approximately 64000** `dof`**) for various error indicator strategies.**

E. NACA 0004 in two-dimensional RANS flow: $M_\infty = 0.5$, $\alpha = 3^\circ$, $Re = 1 \times 10^4$

To test the combined approach's effectiveness for even more challenging numerics, the final case is the NACA 0004 airfoil in turbulent flow with a modified Spalart-Allmaras turbulence model [37]. A relatively low freestream Mach number of $M = 0.5$ was chosen so that the Mach number in the domain never approaches sonic conditions. The rationale for this is the concern that the present shock capturing techniques, through artificial viscosity stabilization, can adversely affect the entropy variables. Additionally, the wall distance function interpolation order, which is the order of polynomials used to approximate the wall distance within each element [38], was set to 2. For curved geometries, this order needs to be sufficiently high to not introduce low-order errors in the wall distance calculation, which is required for the source terms of the turbulence model. Experiments with linear wall distance function approximation (in an element's reference space) showed degradation of the output convergence, whereas higher wall distance function orders did not appreciably affect the results for the solution orders used in this example.

Figure 17 shows the initial coarse mesh for this case, which consists of 660 triangular elements. Curved elements of order $q = 4$ represent the airfoil geometry, and the farfield boundary is 2000 chord-lengths away. The cases were run with a solution approximation order of $p = 2$ with the following five degrees of freedom targets: 5000, 10000, 20000, 40000, and 80000. Just as with the previous MOESS simulations, for each `dof` target, fifteen solution iterations were performed before moving to the next `dof` target.

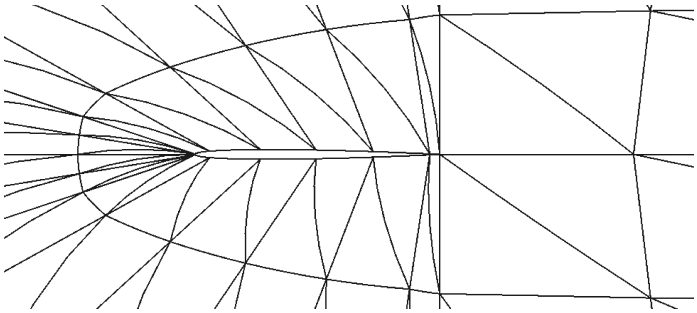


Fig. 17 Initial NACA 0004 airfoil mesh.

As before, the drag coefficient and lift coefficients were used as outputs to govern the mesh adaptation. Adaptation runs for these outputs, the entropy variables alone, and the combined entropy

and output-based adjoint solution were generated. Two modified versions of the combined approach were also investigated for this case. The first is a modification to the combined approach in which only the output-based indicator is used to target which elements get refined while using MOESS. Therefore the output-based indicator alone governs the local sampling. The second modification involves using a modified residual made up of only the convective terms to obtain the entropy-based adjoint when using the combined approach.

Previous work[39] indicates that for RANS cases, the optimal drag coefficient error estimate using entropy variables requires two inviscid residual evaluations, one with the approximate solution and one with the exact solution. This is motivated by relating the output J from Eqn. 10 directly to one output of interest, the drag coefficient, through Oswatitsch's formula[40] via

$$\delta c_{d,\text{osw}} \approx K\delta J^{\text{inv}} = K\left(J^{\text{inv}}\left(\mathbf{u}_H\right) - J^{\text{inv}}\left(\mathbf{u}\right)\right) = K\int_{\Omega}\mathbf{v}^T\left(\mathbf{r}^{\text{inv}}\left(\mathbf{u}_H\right) - \mathbf{r}^{\text{inv}}\left(\mathbf{u}\right)\right)d\Omega, \tag{17}$$

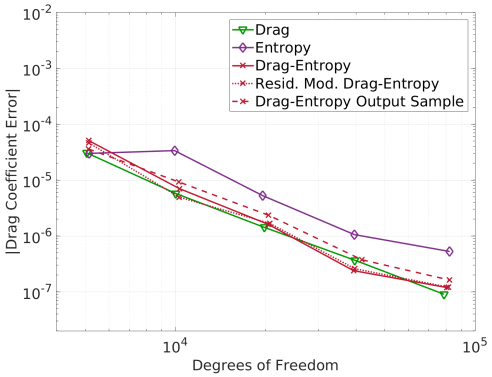
where the superscript "inv" denotes inviscid residuals and fluxes. For RANS cases, the term $\mathbf{r}^{\text{inv}}\left(\mathbf{u}\right)$ needs to be approximated. The discrete analogue of Eqn. 17 is

$$\delta c_{d,\text{osw}} \approx K\mathbf{V}_h^T\left[\mathbf{R}_h^{\text{inv}}\left(\mathbf{U}_h^H\right) - \mathbf{R}_h^{\text{inv}}\left(\mathbf{U}_h\right)\right], \tag{18}$$

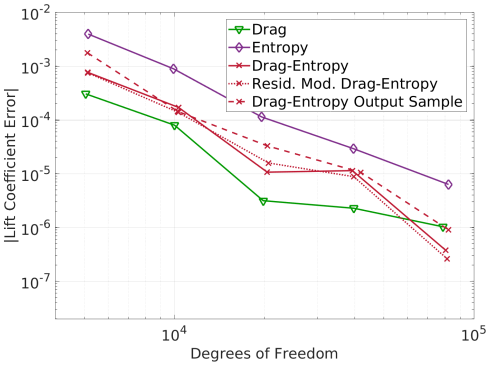
where $\mathbf{U}_h^H$ is the discrete coarse-space state vector injected into the fine space, and $\mathbf{U}_h$ is the discrete fine-space state. The residual operators in Eqn. 18 include only the convective terms of the RANS equations.

Figure 18 shows the convergence results for all of the error indicator strategies. In this figure, we also present how the drag-based indicators perform in predicting lift coefficient, and vice versa, which yields a total of four output error convergence plots. Again, the output and `dof` values reported from these results are averages over the last 5 solution iterations at each target `dof`. The truth outputs for both the drag and lift coefficients were obtained by uniformly refining the final output-adapted mesh and running at $p = 3$. An identical set of truth solutions was obtained for both truth solutions using $p = 4$. The difference in the outputs between these two solutions is of the order 1×10^{-9}. The dashed red line corresponds to the case where the output-based adjoint alone controls the local sampling. The dotted red line corresponds to the combined approach that

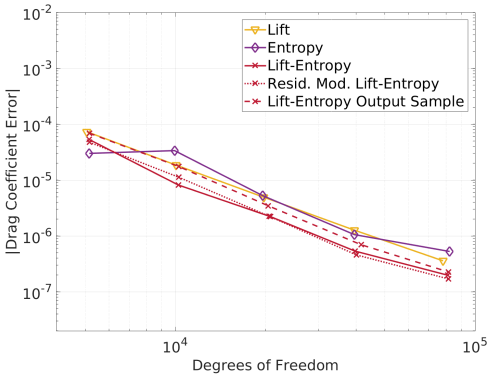
incorporates the modified residual approach to obtain the entropy-based indicator.



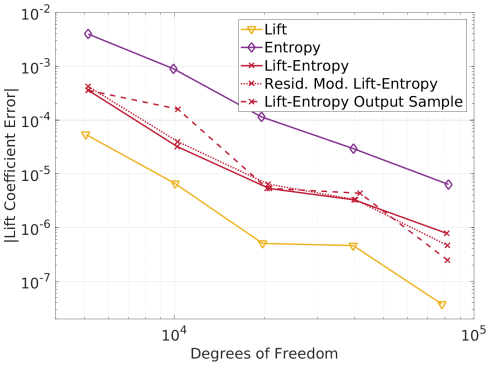
(a) Drag Output Using Drag-based Adjoints



(b) Lift Output Using Drag-based Adjoints



(c) Drag Output Using Lift-based Adjoints



(d) Lift Output Using Lift-based Adjoints

Fig. 18 NACA 0004 $M_\infty = 0.5$, $\alpha = 3°$, $Re = 1 \times 10^4$: Comparison of output convergence histories.

When it comes to predicting the drag coefficient, none of the combined approaches yield significantly different results compared to the drag-based approach. In fact, several other RANS simulations were analyzed with multiple geometries and flow conditions. It was found that the drag-based adjoint indicator performs similarly, or sometimes a little better than the combined approach indicators. However, this is not overly discouraging, since these are steady-state simulations. For unsteady simulations, the cost of fine output adjoint calculations will be dramatically larger, and we expect more benefit from using the much cheaper entropy variables.

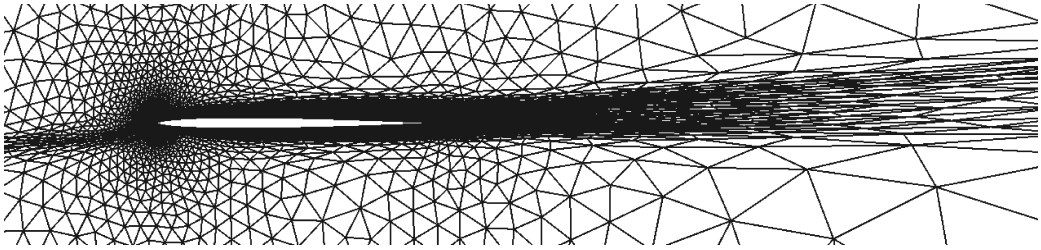
With regards to predicting lift coefficient, there is much more variety among the different adaptive indicator approaches. The lift-based indicator predicts a much more accurate lift coefficient

compared to the combined approach. This is not surprising given that we have observed the entropy-based indicator is more reliable with regards to its prediction of drag coefficient since regions of the flow field that produce spurious entropy generation are those that subsequently produce more drag. On the other hand, the combined approach that uses the drag-based indicator obtains more accurate lift coefficients than the drag-based indicator for the final degrees of freedom target. Unfortunately, since the combined approaches do not produce as accurate lift coefficient errors at the lower degrees of freedom targets, it is difficult to definitively state that the combined approaches are better.

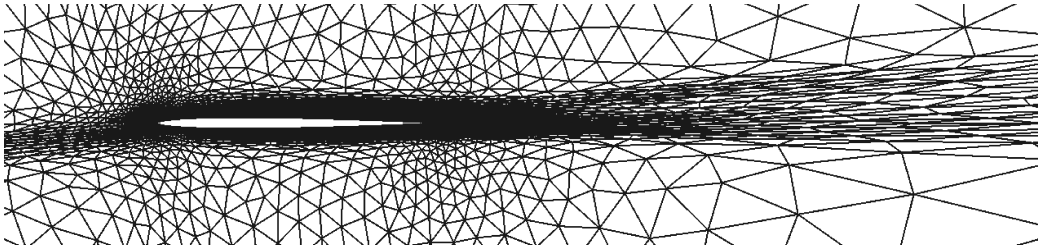
Figure 19 presents the meshes at 80000 `dof` for the output-based adjoint approaches, the entropy-based approach, and the combined approaches with no modifications. Although the drag coefficient errors at this `dof` are not that different among all of the adaptation indicator strategies, the meshes do show visible differences consistent with those in Figure 16. Clearly, the output-based adjoint approaches focus refinement ahead of the airfoil and less in the wake, while the entropy-based approach focuses significant refinement in the wake. Interestingly, the drag coefficient error levels specifically do not vary significantly among the different strategies, which suggests that either the set of meshes that yield nearly optimal drag predictions is large, or that an optimal mesh has not yet been identified by the adaptive approaches considered. The resolution of this question is an area of ongoing research.

VII. Conclusions

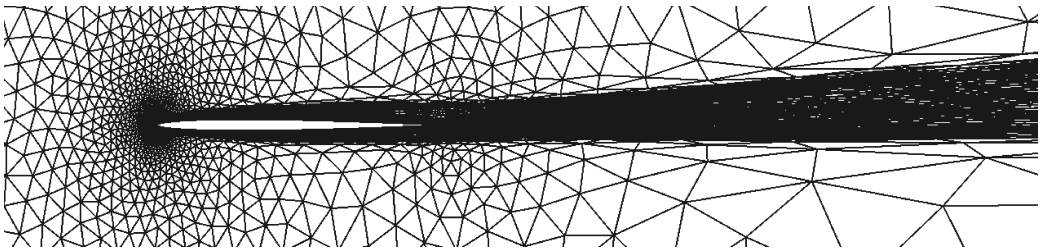
By combining the output-based adjoint indicator with the entropy-variable indicator, an appreciable benefit over the entropy-variable approach was found in several examples of aerodynamic interest. The entropy-variable approach targets areas of the mesh where spurious entropy is generated, even though those areas may not significantly affect the engineering output of interest. This sometimes prevents refinement in more critical regions of the mesh. By combining the entropy-variable indicator with the output-based adjoint indicator, this problem is marginalized since output-based adjoint methods do not target these unnecessary areas of the mesh for refinement. In addition, the combined indicator approach sometimes yields improved performance compared to the standard output-based indicator approach due to the previously-discussed inefficiencies of the output-based



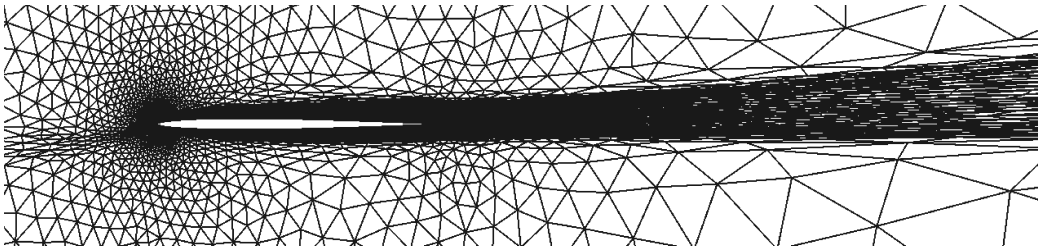
(a) Drag Adaptation



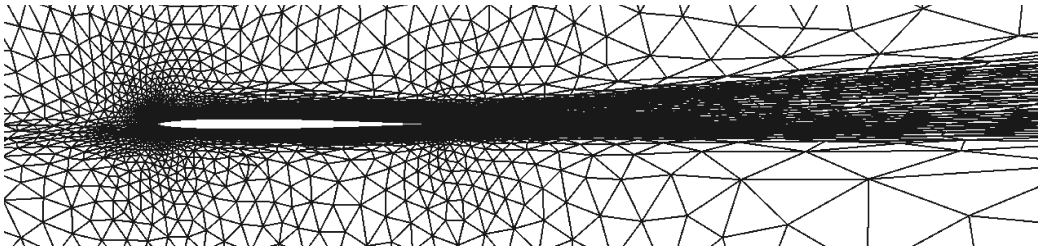
(b) Lift Adaptation



(c) Entropy Adaptation



(d) Drag-Entropy Adaptation



(e) Lift-Entropy Adaptation

Fig. 19 NACA 0004 $M_\infty = 0.5$, $\alpha = 3^\circ$, $Re = 1 \times 10^4$: Meshes after final adaptation iteration (approximately 80000 DOF) for various error indicator strategies.

approach. Since the output-based indicator is derived from approximate adjoint solutions, which may be under-resolved and noisy, these methods may end up targeting areas of the domain where additional refinement is not necessary. This is particularly the case with the flow ahead of the body of interest, near the stagnation streamline. Combining the indicators limits excessive refinement both upstream of and downstream from the body.

When using hanging node mesh adaptation, it was evident that applying a mask on the entropy-based indicator can yield superior error convergence when just direct, elemental combination of indicators is not sufficient. However, the optimal mask is not always clear for a specific case, as changing what percentage of elements are masked can yield varying results.

The benefits of using the combined indicator approach are not exclusive to hanging node adaptation, as benefits to the combined approach were shown for cases using MOESS. While the combined approach shows an appreciable benefit for inviscid and viscous cases, the benefit is not as significant for RANS cases. The combined approaches produce comparable drag coefficient errors compared to the drag-coefficient based approach. However, the combined approaches fail to produce as reliable lift coefficient errors. We expect greater differences for unsteady simulations, especially when factoring in the cost difference of calculating the entropy and output adjoints.

As previously mentioned, obtaining the fine-space output-based adjoint can be quite expensive, especially for unsteady cases. Future work will focus on applying the combined indicator approach to unsteady problems. To mitigate the cost associated with obtaining the output-based adjoint indicator, the indicator will be obtained on a coarser space without an exact solution. When combined with the entropy-based indicator, this should lead to results that are superior to those obtained using just entropy-based adaptation and not significantly more computationally expensive. The combined approach will always be more computationally expensive than using the entropy-based approach since it requires an output-based indicator, but because of the limitations with the entropy-based indicator, should often be more accurate. In addition, the combined approach using a coarse space approximate output-based indicator will be less expensive than an output-based approach using a fine-space adjoint for unsteady problems. These observations suggest that the benefits of the combined-approach will be much more pronounced for unsteady problems.

VIII. 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] Fidkowski, K. J. and Darmofal, D. L., "Output-Based Error Estimation and Mesh Adaptation: Overview and Recent Results," AIAA Paper 2009-1303, 2009.

[9] Fidkowski, K. J. and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *American Institute of Aeronautics and Astronautics Journal*, Vol. 49, No. 4, 2011, pp. 673–694.

[10] Fidkowski, K. J. and Roe, P. L., "Entropy-based Mesh Refinement, I: The Entropy Adjoint Approach," AIAA Paper 2009-3790, 2009.

[11] Fidkowski, K. J. and Roe, P. L., "An Entropy Adjoint Approach to Mesh Refinement," *SIAM Journal on Scientific Computing*, Vol. 32, No. 3, 2010, pp. 1261–1287.

[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] Baker, T. J., "Mesh Adaptation Strategies for Problems in Fluid Dynamics," *Finite Elements in Analysis and Design*, Vol. 25, 1997, pp. 243–273.

[15] 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. Kröner, M. Ohlberger, and C. Rhode, Springer-Verlag, 1999.

[16] Warren, G. P., Anderson, W. K., Thomas, J. L., and Krist, S. L., "Grid Convergence for Adaptive Methods," AIAA Paper 1991-1592, 1991.

[17] 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.

[18] 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.

[19] Giles, M. B. and Pierce, N. A., "Adjoint equations in CFD: duality, boundary conditions and solution behavior," AIAA Paper 97-1850, 1997.

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

[21] 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.

[22] 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.

[23] Yano, M., *An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2012.

[24] 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. 40–69.

[25] 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.

[26] 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.

[27] Fidkowski, K. J., "A Local Sampling Approach to Anisotropic Metric-Based Mesh Optimization," AIAA Paper SciTech, 2016.

[28] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[29] Bassi, F. and Rebay, S., "GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations," *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by K. Cockburn and Shu, Springer, Berlin, 2000, pp. 197–208.

[30] Park, M. A., "Three–Dimensional Turbulent RANS Adjoint–Based Error Correction," AIAA Paper 2003-3849, 2003.

[31] Aftosmis, M. J., Berger, M. J., and Melton, J. M., "Adaptive Cartesian mesh generation," *Handbook of Grid Generation*, edited by J. F. Thompson, B. K. Soni, and N. P. Weatherill, CRC Press, 1998.

[32] 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.

[33] Hartmann, R., "Adjoint consistency analysis of Discontinuous Galerkin discretizations," *SIAM Journal on Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696.

[34] 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.

[35] Persson, P.-O. and Peraire., J., "Sub-cell shock capturing for discontinuous Galerkin methods," AIAA Paper 2006-112, 2006.

[36] Spalart, P. R., "Airplane Trailing Vortices," *Annual Review of Fluid Mechanics*, Vol. 30, 1998, pp. 107–138.

[37] Allmaras, S., Johnson, F., and Spalart, P., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," Seventh International Conference on Computational Fluid Dynamics (ICCFD7) 1902, 2012.

[38] Ceze, M. A. and Fidkowski, K. J., "High-order output-based adaptive simulations of turbulent flow in two dimensions," *AIAA Journal*, Vol. 54, No. 9, 2016.

[39] Fidkowski, K. J., Ceze, M. A., and Roe, P. L., "Entropy-Based Drag Error Estimation and Mesh Adaptation in Two Dimensions," *AIAA Journal of Aircraft*, Vol. 49, No. 5, September-October 2012, pp. 1485–1496.

[40] Oswatitsch, K., *Gas Dynamics*, Academic, New York, 1956.