A Combined Entropy and Output-based Adaptive Approach Using a Stabilized Continuous Finite Element Formulation

Kevin T. Doetsch* and Krzysztof J. Fidkowski†
University of Michigan, Ann Arbor, MI 48109

Adaptive mesh refinement driven by combining an error indicator derived from using an adjoint on a user-specified engineering scalar output and one derived directly from entropy variables has previously shown benefits in cost and accuracy for many applications. By leveraging indicators obtained from both approaches, the relative downsides of each approach can be limited. Numerous examples of both steady and unsteady simulations using a variety of governing equations highlight the possible benefits to both accuracy and cost that a combined error indicator approach can present. Previous works demonstrating this approach all relied on a discontinuous Galerkin (DG) discretization. In this work, the characteristics of the combined approach are demonstrated using a continuous finite element formulation, the streamline upwind/Petrov Galerkin (SUPG) method. Despite using a completely different discretization, similar advantages of the combined approach are demonstrated for simulations that use the steady-state inviscid Navier-Stokes equations. In addition, direct comparisons of error estimates and adaptive meshes are made for the various adaptive strategies using both SUPG and DG formulations.

I. Introduction

Solution-based adaptive methods that use an error indicator to drive computational mesh adaptation have long been a popular approach in Computational Fluid Dynamics (CFD) to obtain accurate solutions for problems that exhibit a wide range of spatial length scales [1–7]. Numerous examples in the literature detail extensive analysis of various indicators and their relative performance, robustness, and accuracy.

One popular approach involves a user-specified engineering scalar output [4, 8, 9] to drive adaptation through the use of an adjoint. This approach is advantageous since the indicator specifically targets areas of the computational domain that are critical to the prediction of the desired output [3, 4, 10, 11]. The downside of this approach is that it can be computationally expensive, since it requires the implementation of a transpose linear solver and since it involves quantities that must be computed in a refined-space computational domain. Another downside is that the output-based approach is not always effective due to a suboptimal approximation of the adjoint, especially in the presence of adjoint singularities such as on a leading-edge stagnation streamline and near an airfoil trailing edge[12]. These singularities lead to noise in the error estimate, which leads to areas of unnecessary, based on our tests, over-refinement of the computational mesh. Another approach leverages entropy variables instead of a separate adjoint computation [13, 14]. Using entropy variables is less computationally expensive since these variables can be computed from a direct variable transformation of the conservative state. Entropy-variable indicators, unfortunately, target all regions of the domain where spurious entropy is produced, e.g. shocks or 3D vortices. Therefore, this approach is not as globally discriminating as the output-based approach.

Since each of the two aforementioned approaches has its strengths and weaknesses, a new approach was implemented that combines both indicators to drive the adaptation. Extensive research was previously done to show the benefits of various approaches to combining the indicators for both steady-state and unsteady simulations using a variety of governing equations [15–17]. All of the previous work on this subject used a discontinuous Galerkin (DG) formulation for the spatial discretization. In this work, the combined approach is further analyzed, this time using a stabilized continuous finite-element formulation.

The streamline upwind Petrov–Galerkin (SUPG) method [18, 19] is one of many stabilized continuous finite element methods that are increasingly becoming more popular in the high-order CFD community. Discontinuous Galerkin (DG) schemes have been aggressively developed and researched over the last several years [20–23] and continue to be one of the most popular choices for high-order spatial discretizations. In this scheme, each element has its own set of nodes and

*Graduate Research Assistant, Department of Aerospace Engineering, kevintd@umich.edu, AIAA Member
†Professor, Department of Aerospace Engineering, kfid@umich.edu, AIAA Associate Fellow.

solutions associated with those nodes. This leads to an element-wise discontinuous approximation of the solution over the entire computational domain. In stabilized finite element schemes, nodes on edges, vertices, and faces are shared between adjacent elements, which leads to a continuous solution over the domain. This means that for moderate formal orders of accuracy, stabilized finite element methods require far fewer degrees of freedom than DG methods on the same mesh [24, 25]*. For implicit time-marching schemes, DG also requires far more non-zero matrix entries. Adaptation is required for both discretizations to observe the full potential of error reduction at a given degree of freedom count. Consequently, there is a growing benefit to be able to demonstrate mesh adaptation techniques using a continuous finite element formulation such as SUPG.

The outline of this paper is as follows. Section II details the governing equations and the specifics of the SUPG discretization including the stabilization parameter. Additional details on the linear solver and the DG discretization are also discussed. Section III reviews the combined approach, as well as details on the output-based adjoint approach and the entropy variables approach. Section IV details the adaptive mechanics used in the work. Finally, Section V presents the results of the various adaptive strategies for both SUPG and DG.

II. Numerical Implementation

The primary focus of this section is to review the SUPG discretization of the Euler equations. Extra attention is devoted to the stabilization term that is necessary for convection-dominated flows. Additional information regarding the linear solvers and other aspects of the numerical implementation is also discussed.

A. Governing Equations

This work considers solutions of the compressible Euler equations of gas dynamics that govern the flow of a perfect inviscid gas. These equations are obtained from the negligible-viscosity assumption at the high Reynolds number limit of the compressible Navier-Stokes equations. Any viscous-related terms in the Navier-Stokes equations are ignored in this approach. Written in terms of the conservation of mass, momentum, and energy, the Euler equations are

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) &= 0 \\ \frac{\partial \rho \vec{v}}{\partial t} + \nabla \cdot (\rho \vec{v}\vec{v} + P) &= 0 \\ \frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho \vec{v} H) &= 0, \end{aligned} \tag{1}$$

where ρ is density, $\vec{v}$ is velocity, H is total enthalpy per unit mass, and P is pressure. The Euler equations can be written in a compact form as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} = \mathbf{0}, \tag{2}$$

where i indexes the spatial dimension. The state and flux vector in two dimensions can be written as

$$\text{state} : \mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \text{flux} : \vec{\mathbf{F}} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{bmatrix} \hat{x} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{bmatrix} \hat{y}. \tag{3}$$

The divergence of the inviscid flux vector, $\frac{\partial \mathbf{F}_i}{\partial x_i}$, can be rewritten in terms of the unknown state values as

$$\frac{\partial \mathbf{F}_i}{\partial x_i} = \frac{\partial \mathbf{F}_i}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial x_i} = \mathbf{A}_i \frac{\partial \mathbf{u}}{\partial x_i}, \tag{4}$$

where $\mathbf{A}_i$ is the inviscid flux Jacobian. Rewriting the inviscid flux vector in terms of the inviscid flux Jacobian is helpful since the stabilization term used in SUPG is directly related to it.

*this does not mean that continuous methods are universally more accurate for a given number of degrees of freedom, as the error magnitudes could be different at the same order

B. SUPG Discretization

The weak form of the governing system of equations is obtained by using an approximation of the solution with a weighted sum of basis functions of the form

$$\mathbf{u}(\vec{x}) = \sum_{j=1}^{N} \phi_{e,j}(\vec{x})\mathbf{U}_j, \tag{5}$$

where N is the total number of degrees of freedom in the mesh. We also denote by N_e, the number of elements in the domain Ω, and by n_p the number of basis functions that have support over a given element. Using this approximation for the solution, the weak form can be written as

$$\int_\Omega \boldsymbol{\phi}^T \left[\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} \right] d\Omega = \mathbf{0}. \tag{6}$$

where $\boldsymbol{\phi}$ is a general test vector over the states. After integrating Eqn. 6 by parts, the weak form can be written as

$$\int_\Omega \left[\boldsymbol{\phi}^T \frac{\partial \mathbf{u}}{\partial t} - \frac{\partial \boldsymbol{\phi}^T}{\partial x_i}\mathbf{F}_i \right] d\Omega + \int_\Gamma \boldsymbol{\phi}^T \left(\mathbf{F}_i^b n_i \right) d\Gamma = \mathbf{0}, \tag{7}$$

where $\mathbf{F}_i^b n_i$ is the normal component of the flux on the boundary Γ. For the far-field boundaries, the boundary flux is constructed using the Roe scheme [26], where the interior and freestream states values are used in the Riemann solver. For the inviscid wall boundaries, the boundary flux at the wall is given by

$$F^b = [0, p^b n_x, p^b n_y, 0], \quad P^b = (\gamma - 1)\left[\rho E^+ - \frac{1}{2}\rho^+ \left|\vec{v}^b\right|^2 \right], \tag{8}$$

where $\vec{n} = n_x\hat{x} + n_y\hat{y}$ is the normal, P^b is the boundary pressure, and $\vec{v}^b$ is the boundary velocity. The boundary velocity is tangential, meaning that the interior velocity with the wall-normal component is taken out. The boundary pressure is calculated based on the interior density, energy, and boundary velocity.

The Galerkin continuous finite element discretization as shown in Eqn. 7 will be unstable for advection dominated flows. This is because spurious oscillations will lead to instability in the time integration. Several stabilization methods in the past have been studied to correct this instability. One such approach, the Streamline Upwind Petrov-Galerkin (SUPG) method [27–30], was chosen for this discretization. In this approach, a stabilization matrix, τ^{SUPG}, is added to compensate for the lack of dissipation in the streamwise direction [18].

To better understand the stability notation, it is useful to rewrite Eqn. 2 using a differential operator $\mathcal{L}$ [31],

$$\mathcal{L}(\mathbf{u}) = \mathbf{0}, \qquad \mathcal{L} = \frac{\partial}{\partial t} + \mathbf{A}_i \frac{\partial}{\partial x_i}. \tag{9}$$

The added stability that is necessary to obtain converged solutions can be achieved by adding an upwind bias to the test functions $\boldsymbol{\phi}$. In this case, the convective portion of the aforementioned operator, $\mathcal{L}_c$, acting on the basis functions is added to the existing test functions to produce [32, 33]

$$\hat{\boldsymbol{\phi}} = \boldsymbol{\phi} + \boldsymbol{\tau}\mathcal{L}_c(\boldsymbol{\phi}) = \boldsymbol{\phi} + \boldsymbol{\tau}\mathbf{A}_i \frac{\partial \boldsymbol{\phi}}{\partial x_i}, \tag{10}$$

where $\boldsymbol{\tau}$ is the stabilization matrix. The purpose of this term is to limit the amount of numerical dissipation added to the scheme to as low of an amount as possible while preserving stability.

Using this augmented test function, the weak form in Eqn. 7 can be rewritten as [34, 35]

$$\int_\Omega \boldsymbol{\phi}^T \frac{\partial \mathbf{u}}{\partial t} - \frac{\partial \boldsymbol{\phi}}{\partial x_i}\mathbf{F}_i d\Omega + \int_\Gamma \boldsymbol{\phi}^T \left(\mathbf{F}_i^b n_i \right) d\Gamma + \underbrace{\sum_{e=1}^{N_e} \int_\Omega \mathbf{P}_e \left(\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} \right) d\Omega}_{\text{Stabilization term}} = \mathbf{0}. \tag{11}$$

In the above equation, it is important to note that the stabilization term is calculated over all of the elements in the domain Ω. The stabilization term only affects the element interior and does not have an impact on the boundary terms. The elemental perturbation term, $\mathbf{P}_e$, can be expanded as [30, 35, 36] (summation implied on i)

$$\mathbf{P}_e = \mathbf{A}_i \frac{\partial \boldsymbol{\phi}}{\partial x_i}\boldsymbol{\tau}_e. \tag{12}$$

The elemental stabilization matrix $\boldsymbol{\tau}_e$ is obtained using the eigensystem decomposition of the projection of the flux Jacobian matrices onto the spatial gradients of the basis functions [19, 34]. For element e,

$$\begin{aligned} \boldsymbol{\tau}_e^{-1} &= \sum_{j=1}^{n_p} \left| \frac{\partial \phi_{e,j}}{\partial x_i} \mathbf{A}_i \right| \\ \left| \frac{\partial \phi_{e,j}}{\partial x_i} \mathbf{A}_i \right| &= [\mathbf{T}] \, |\boldsymbol{\Lambda}| \, [\mathbf{T}]^{-1} , \end{aligned} \tag{13}$$

where $[\mathbf{T}]$ denotes the matrix of right eigenvectors and $|\boldsymbol{\Lambda}|$ denotes the diagonal matrix of absolute values of the eigenvalues. i indexes the spatial dimension, whereas j indexes the basis functions for the specific element e. The inverse of the stability matrix is evaluated at each Gauss quadrature point for volume integration. This means that there is a unique elemental stability matrix for each quadrature point. Note that even for high-order discretization, $p \geq 2$, only the linear shape functions are used for the computation of the stabilization matrix. It has been previously shown that including the higher-order basis functions will reduce the amount of dissipation added to the scheme [37]. Since the stabilization term was initially developed for linear basis functions [27] and there is no formal derivation of the stability matrix for high-order discretization, the choice was made to include linear basis functions in Eqns. 12 and 13.

C. General Numerics and Solver

For this work, a continuous finite element-based code written in ANSI C was created to solve all cases that incorporate the SUPG discretization. Besides handling the high-order SUPG discretization, the code also has two-dimensional high-order meshing, error estimation, and adjoint capabilities. The code leverages an open-source suite of various data structures and routines, PETSc [38–40], for solving large linear systems of equations. In this work, we use PETSc routines that solve systems using preconditioned Krylov subspace methods. Incomplete LU factorization (ILU) with two levels of fill was chosen as the preconditioner [41–43]. The generalized minimal residual method (GMRES) [44] with Gram-Schmidt orthogonalization was used for the Krylov subspace method.

Our previous works all used a discontinuous Galerkin (DG) finite-element code [45] to handle the discretization of the governing equations. That same code was used in this work as well. It relies on the Roe approximate Riemann solver [26] for the inviscid fluxes. The code also uses a Newton-GMRES implicit solver with element-line-Jacobi preconditioning for solving large linear systems.

III. Combined Approach for Output Error Estimation

This section reviews a previously documented error estimation technique that involves combining two mesh refinement indicators. One indicator is obtained from an output-based adjoint and the other directly from entropy variables [15, 16]. A brief review of the derivation of how each indicator, along with a description of how the indicators are combined, is the primary focus of this section.

A. Adaptation Using Output Adjoints

Output error estimation methods refine areas of the mesh that have a direct impact on the prediction of a targeted engineering output. The error estimates can be localized to refine areas of the mesh that have the greatest impact on the prediction of the chosen output. They 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 perturbations of the residual to the output of interest, $J(\mathbf{u})$, where $\mathbf{u}$ denotes the state vector.

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

$$\delta J = < \delta\mathbf{r}, \boldsymbol{\psi} >, \tag{14}$$

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

$$\mathcal{R}'[\mathbf{u}](\mathbf{w}, \boldsymbol{\psi}) + J'[\mathbf{u}](\mathbf{w}) = 0, \qquad \forall \mathbf{w} \in \mathcal{V}, \tag{15}$$

where the primes denote Fréchét linearization about $\mathbf{u}$, $\mathcal{V}$ is the trial and test function space and $\mathcal{R}(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ is the semilinear operator that is the weak form of the differential equation.

The solution of the adjoint equation can be used to estimate the output error using an adjoint-weighted residual method. For most cases, the solution in a finite-dimensional approximation space $\mathcal{V}_H$ does not exactly satisfy the PDE and results in a residual disturbance in the weak form. Fortunately, the adjoint translates the residual perturbation to an output perturbation via Eqn. 14 [14]

$$\delta J =< \delta \mathbf{r}, \boldsymbol{\psi} >= -\mathcal{R}(\mathbf{u}_H, \boldsymbol{\psi}). \tag{16}$$

The numerical error in the chosen output can be quantified using a weighted residual approximation for the solution of the above expression. However, the continuous adjoint must be solved on a finer finite-dimensional space, $\mathcal{V}_h \supset \mathcal{V}_H$, either directly or iteratively [46–48]. For this work, the fine space is obtained by incrementing the approximation order. The fine-space adjoint $\boldsymbol{\psi}_h$ is then used to obtain an indicator that relates the output error to the local contribution of each element in the computational domain. Eqn. 16 can be approximated as

$$\delta J \approx -\sum_{\kappa_H \in T_H} \mathcal{R}_h\left(\mathbf{u}_H, \boldsymbol{\psi}_h|_{\kappa_H}\right) \approx -\sum_{\kappa_H \in T_H} \mathcal{R}_h\left(\mathbf{u}_H, \delta\boldsymbol{\psi}_h|_{\kappa_H}\right), \tag{17}$$

where $|_{\kappa_H}$ denotes the restriction of an interpolated function to element κ_H of the triangulation T_H. The difference between the fine and coarse space adjoints, $\delta\boldsymbol{\psi}_h \equiv \boldsymbol{\psi}_h - \boldsymbol{\psi}_H$, is taken to minimize the error due to possible order-dependence of the residual [49]. Using the absolute value of the elemental contributions in Eqn. 17, the elemental adaptive indicator can be expressed as [3, 10, 50, 51]

$$\eta_{\kappa_H} = \left|\mathcal{R}\left(\mathbf{u}_H, \delta\boldsymbol{\psi}_h|_{\kappa_h}\right)\right|. \tag{18}$$

Since the output-based adjoint must be computed on the fine-space with a transpose solver, it can be computationally expensive. This is particularly true for unsteady problems where the fine-space adjoint must be computed at each time step in a reverse time integration.

B. Adaptation Using Entropy Variables

Entropy variables can also be used instead of an output-based adjoint to compute error estimates. Regions in the computational domain that exhibit high net production of *spurious* entropy are targeted for refinement. The following derivation of the entropy variable approach to mesh refinement follows our previous work [14]. Note that this derivation includes viscous terms for the sake of completeness.

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

$$\mathbf{A}_i\partial_i\mathbf{u} - \partial_i\left(\mathbf{K}_{ij}\partial_j\mathbf{u}\right) = \mathbf{0}, \quad \partial_i F_i = 0, \tag{19}$$

where i is the spatial index, $\mathbf{A}_i\partial_i\mathbf{u}$ is the inviscid flux, $-\mathbf{K}_{ij}\partial_j\mathbf{u}$ is the viscous flux, and $F_i(\mathbf{u})$ is the entropy flux associated with the entropy function $U(\mathbf{u})$.

The entropy flux and entropy function satisfy the compatibility relation $U_{\mathbf{u}}\mathbf{A}_i = (F_i)_{\mathbf{u}}$. The entropy variables, defined as $\mathbf{v} = U_{\mathbf{u}}^{\mathrm{T}}$, symmetrize the conservation laws in the sense that [32, 52]: the transformation Jacobian matrix, $\mathbf{u}_{\mathbf{v}}$, is symmetric, positive definite; and $\mathbf{A}_i\mathbf{u}_{\mathbf{v}}$ is symmetric. The entropy variables must also symmetrize $\mathbf{K}_{ij}$, in the sense that $\widetilde{\mathbf{K}}_{ij} = \widetilde{\mathbf{K}}_{ji}^{\mathrm{T}}$, where $\widetilde{\mathbf{K}}_{ij} = \mathbf{K}_{ij}\mathbf{u}_{\mathbf{v}}$ [32]. Substituting $\partial_i\mathbf{u} = \mathbf{u}_{\mathbf{v}}\partial_i\mathbf{v}$ into Eqn. 19 and taking the transpose yields the following equation for the entropy variables,

$$\partial_i\mathbf{v}^{\mathrm{T}}\mathbf{A}_i\mathbf{u}_{\mathbf{v}} - \partial_i\left(\partial_j\mathbf{v}^{\mathrm{T}}\widetilde{\mathbf{K}}_{ji}\right) = \mathbf{0}. \tag{20}$$

The entropy variables satisfy an adjoint equation for an output that has no domain integral contribution [14]. This can be proven by using the entropy variables instead of the output-based adjoint,

$$J = \int_{\partial\Omega} F_i n_i ds + \int_{\Omega} \partial_i\mathbf{v}^T\widetilde{\mathbf{K}}_{ij}\partial_j\mathbf{v}d\Omega - \int_{\partial\Omega}\mathbf{v}^T\widetilde{\mathbf{K}}_{ij}\partial_j\mathbf{v}n_i ds. \tag{21}$$

Each of the terms in Eqn. 21 has a physical meaning outlined in previous works [13, 14]. The entropy function, S, that produces 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 [32]. We can write the entropy variables in terms of the entropy functions via

$$\mathbf{v} = U_{\mathbf{u}}^T = \left[\frac{\gamma - S}{\gamma - 1} - \frac{1}{2}\frac{\rho V^2}{P},\ \frac{\rho u_i}{P},\ -\frac{\rho}{P}\right]^T, \qquad U = -\rho S/(\gamma - 1)\,, \qquad S = \ln P - \gamma \ln \rho, \tag{22}$$

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 found via a nonlinear transformation of the conservative variables. The entropy flux function can be defined as $F_i = u_i U$.

Using entropy variables to drive the adaptation is an attractive alternative to using output-based adjoints since the entropy variables can be computed directly from the state. However, since this approach targets all regions of the domain where spurious energy is generated, regardless of the effect on a particular engineering output, over-refinement may result. This is especially true for simulations that contain flow discontinuities.

C. Adaptation Using a Combined Approach

With the ability to produce two different error indicators, one obtained from the fine-space output-based adjoint and the other from entropy variables, a new approach can be derived where the indicators are combined. The simplest approach involves computing the indicators separately and then combining them through direct, elemental multiplication. This can be expressed mathematically using the indicators obtained through Eqn. 18 as

$$\eta_{\kappa_H,\text{combined}} = \eta_{\kappa_H,\text{output}} \cdot \eta_{\kappa_H,\text{entropy}}, \tag{23}$$

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. The output-based indicator is obtained using the fine-space adjoint, $\boldsymbol{\psi}_h$, while the entropy-based indicator is obtained using fine-space entropy variables, $\mathbf{v}_h$.

Combining the indicator through multiplication is a simple option, but certainly not the only one. Various other approaches designed to improve adaptive efficiency and computational cost have been outlined and investigated in previous works [15, 16]. At present, only the basic combined approach was evaluated in this work. In the future, other combined approaches will be vetted as well.

IV. Adaptive Mechanics

In this research, we considered two different strategies to drive the mesh adaptation for each adaptive iteration. The first adaptation approach is a variation of mesh optimization via error sampling and synthesis (MOESS)[49, 53]. The distinguishing feature of this approach to MOESS is an elemental error sampling approach for determining the convergence rate tensor of the elemental error. The second mesh adaptation strategy involves sizing the elements using a priori rate estimates and using Hessians of scalars computed from the state to determine each element's anisotropy. A detailed summary of the approach can be found in reference [54].

A. Mesh Optimization via Error Sampling and Synthesis (MOESS)

As previously stated, the unique aspect of the implementation of MOESS used in this work is an error sampling approach for determining the convergence rate tensor of the error on a single element. The ideal local refinement for a particular element is determined through the local sampling approach by computing error indicator $\Delta\mathcal{E}_{ej}$ for a finite number of refinement options j of the element. The indicators estimate the error between the coarse-space solution and the solution for a given refinement option j, which can be expressed mathematically as

$$\mathcal{E}_{ej} \equiv \mathcal{E}_{e0} - \Delta\mathcal{E}_{ej}. \tag{24}$$

The calculation of $\Delta\mathcal{E}_{ej}$ 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. From there, adjoint-weighted residuals can be calculated on the original element. For the output-based adjoint, mathematically this can be expressed as

$$\Delta\mathcal{E}_{ej} \equiv \left|\mathcal{R}_h^{p+1}\left(\mathbf{u}_h^p, \delta\tilde{\boldsymbol{\psi}}_{hj}^p|_{\Omega_e}\right)\right|, \tag{25}$$

where $\tilde{\boldsymbol{\psi}}^p_{hj}$ is the fine-space adjoint $\boldsymbol{\psi}^{p+1}_h$ projected from $p+1$ to order p on refinement option j, and then back to order $p+1$ on the original element. $\delta\tilde{\boldsymbol{\psi}}^p_{hj}$ is then defined as $\delta\tilde{\boldsymbol{\psi}}^p_{hj} \equiv \tilde{\boldsymbol{\psi}}^p_{hj} - \boldsymbol{\psi}^p_h$. Note, all projections are done in a least-squares sense in element-reference space. The form of Eqn. 25 that corresponds to the entropy variables is identical except that $\mathbf{v}$ replaces $\boldsymbol{\psi}$.

The complete details of applying this error sampling approach to discontinuous finite element methods like DG can be found in previous work [53]. However, there are a few changes that need to be made to this approach when using a continuous finite element method such as SUPG. In DG, the approximation in each element is discontinuous from one another so that it is straightforward to apply a global subtraction over all of the elements of the coarse-space adjoint from the fine-space adjoint. However, for SUPG, information among elements is shared at the element boundaries. Therefore, before calculating $\Delta\mathcal{E}_{ej}$ in Eqn. 25, the coarse-space adjoint must again be subtracted from the fine-space adjoint using the same approach outlined in the previous paragraph. The key difference from before is that the projection from $p+1$ to p on refinement option j is done for each element individually. This eliminates any issues caused by the sharing of information across elements.

In SUPG, the error indicators used to govern the local sampling approach, $\Delta\mathcal{E}_{ej}$, need to be restricted based on how many elements each `dof` is in contact with. This is because the fine-space residuals and adjoints are not tied to each element individually. Instead, they are shared across all elements. To leverage the fine-space residuals and adjoints available from error estimation, they error contributions are distributed as shown in Figure 1.

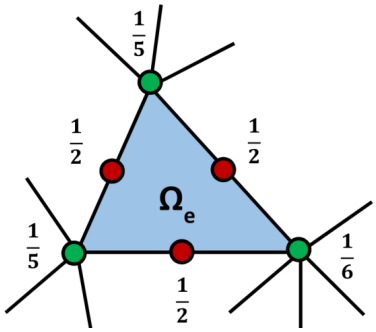


Fig. 1 Weights for distributing residuals and adjoint-weighted residuals from globally-coupled SUPG degrees of freedom to elements, shown for triangles. The node weights are the inverse of the node cardinality while the face weights are simply $\frac{1}{2}$.

By refining each element based on the local sampling approach, anisotropy is created. This produces much better mesh resolution, especially when discontinuities are present in the flow field.

B. *A Priori* Mesh Adaptation

In the *a priori* mesh adaptation approach, the elemental error indicator $\Delta\mathcal{E}_e$ is calculated using the adjoint-weighted residual approach outlined in Eqn. 25. Unlike in MOESS, where this is done for various refinement options using a local sampling approach in order to discern anisotropy, the anisotropy information in the *a priori* approach is found by estimating the direct interpolation error of the solution. We define the interpolation error of a scalar solution u over an edge E in the mesh as [5]

$$\delta_{u,E} = \left|\vec{s}^T\mathbf{H}\vec{s}\right| h^2, \qquad H_{i,j} = \frac{\partial u^2}{\partial x_i \partial x_j},\ i,j \in [1,...,d]\,, \tag{26}$$

where $\vec{s}$ is the edge's unit tangent vector, h is the length, and $\mathbf{H}$ is the Hessian matrix of a chosen scalar. The scalar solution chosen for this work, which is suitable for many types of flows, is the Mach number. Error equidistribution states that the ideal mesh is obtained by equally distributing the squared edge length under the metric $|\mathbf{H}|$, which is a symmetric, positive definite matrix. Using the eigenvalues and eigenvectors from the Hessian, we can find the stretching ratio for a particular element.

With the stretching information known, the desired element sizes can be found using the error indicators obtained from Eqn. 25. To do so, we must estimate the number of adapted elements, n_e, in element e of the original mesh. This

is done by relating the increase in the number of elements to an error reduction factor through an *a priori* estimate [54]

$$\underbrace{n_e \frac{\mathcal{E}^f}{N^f}}_{\text{allowable error}} = \underbrace{\mathcal{E}_e^c \left(\frac{h_{\text{ref}}^f}{h_{\text{ref}}^c} \right)^{\bar{p}_e+1}}_{\text{a priori estimate}} = \mathcal{E}_e^c \left[\prod_{i=1}^{d} \left(\frac{h_i^f}{h_i^c} \right) \right]^{(\bar{p}_e+1)/d}, \tag{27}$$

where i denotes the principal direction, h_i^c and h_i^f are the current and requested element sizes, respectively, $\mathcal{E}_e^c$ is the elemental error indicator, $\bar{p}_e = \min(p_e, \gamma_e)$, and γ_e is the lowest order of any singularity in the element. The error reduction factor, $\mathcal{E}^f/N^f$, is the ratio of the output error tolerance to the number of elements in the adapted mesh. In this work, the output error tolerance is not known. Instead, it is controlled by a target number of elements, N_f, and computed via

$$\mathcal{E}^f = N^f \left(\frac{1}{N^f} \sum_{e=1}^{N_e} \left(\mathcal{E}_e^c \right)^{r'} \right)^{1/r'}, \tag{28}$$

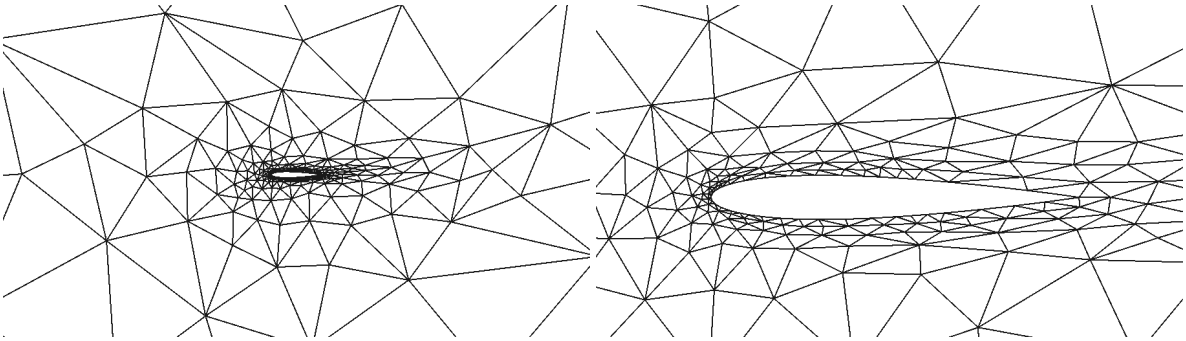
where $r' = d/(p+1+d)$. With $\mathcal{E}^f$ and N^f known, the scaling factor in element size, h_i^f/h_i^c, is determined through the smallest principal stretching, obtained from the Hessian, and the corresponding aspect ratios along the other principal directions. More information can be found in reference [54].

V. Results

The results in this section showcase how the combined approach performs for SUPG relative to DG. Adaptation using either output-based adjoints or entropy variables is shown as well. In total there are two different cases presented in this work that were also analyzed in previous works [15, 16]. Both cases involve solving the steady-state inviscid Navier-Stokes equations.

A. NACA 0012 in Two-Dimensional, Inviscid, Subsonic Flow: $M_\infty = 0.3,\ \alpha = 5°$

The first case is a two-dimensional, inviscid flow over the NACA 0012 airfoil with a closed trailing edge. The initial mesh, shown in Figure 2, is made up of 533 triangular elements with a far-field located 100 chord lengths away from the airfoil. Curved elements, $q = 3$, were used to represent the geometry and solution approximation orders of $p = 1$ and $p = 2$ were used. The following degrees of freedom targets were run with $p = 1$: 750, 1500, 3000, and 6000. In addition, the following degrees of freedom targets were run with $p = 2$: 1500, 3000, and 12000. At each `dof` target, multiple solution iterations were performed so that an average of five solutions could be obtained.



(a) Initial Mesh (b) Initial Mesh (Zoom)

Fig. 2 Initial NACA 0012 airfoil mesh.

The freestream Mach number is 0.3 and the angle of attack is 5°. The flow is subsonic everywhere in the domain. Mach number contours are shown in Figure 3. The drag coefficient and lift coefficient were chosen as the engineering

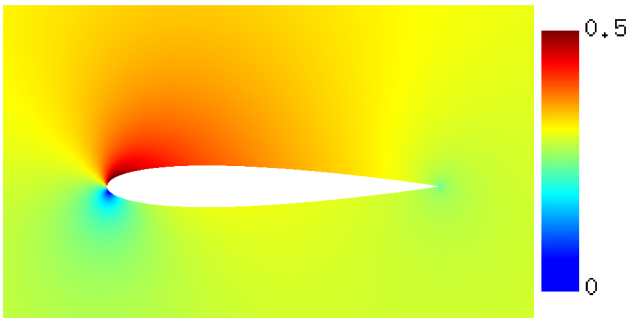
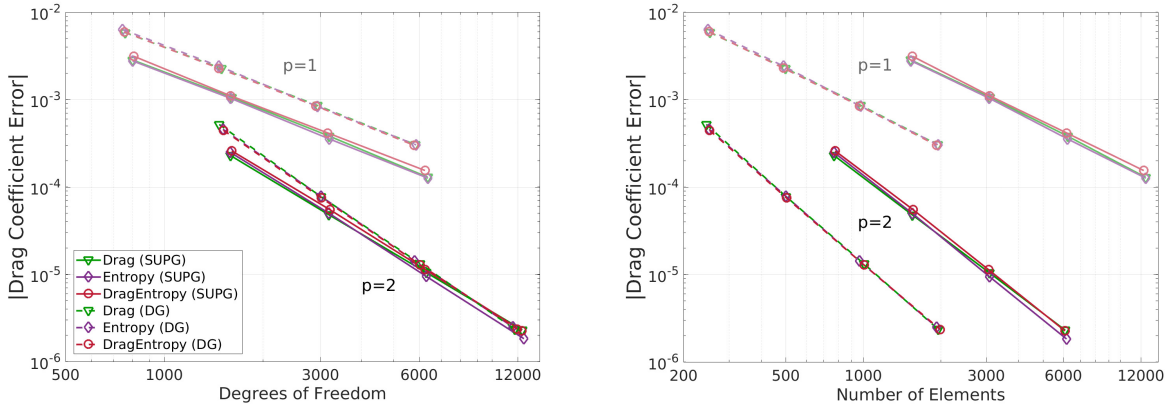


Fig. 3 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Mach number contours (range: 0-0.5).

outputs. Adaptation was governed by adjoints using either the output-based adjoint or entropy variables. In addition, the combined output-based and entropy-based adjoint approach was analyzed.

Figure 4 shows the convergence of the drag coefficient for the various adaptation strategies using both SUPG and DG. One convergence plot compares the drag coefficient error to degrees of freedom, while the other shows the error versus the number of elements. The drag truth solution was obtained by taking the finest mesh obtained using the drag-based adjoint with the DG discretization at $p = 2$, uniformly refining it twice, and solving it with a solution approximation order of $p = 3$. A second truth solution was obtained by taking the finest mesh obtained using the drag-based adjoint with the SUPG discretization and running with a solution approximation order of $p = 3$. The difference between these two solutions was 3×10^{-8}.



(a) Drag Error Versus Degrees of Freedom

(b) Drag Error Versus Number of Elements

Fig. 4 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Comparison of drag coefficient convergence histories for various adaptation methods.

For the same number of degrees of freedom, cases using a solution approximation order of $p = 2$ produced smaller errors compared to cases that used $p = 1$. This conclusion holds regardless of whether SUPG or DG was used. At $p = 1$, there is a much greater difference in error levels between SUPG and DG than at $p = 2$. At each degrees of freedom target, SUPG produces drag coefficient error levels that are about one-third an order of magnitude less than DG. For $p = 2$, the relative difference in error between SUPG and DG is much less, especially for the higher degrees of freedom. Since each element run with DG has its own unique set of degrees of freedom, there are far fewer overall elements in DG compared to simulations using SUPG. At $p = 1$, there are only three unique degrees of freedom for each discontinuous element in DG. This leads to inferior performance relative to SUPG where the degrees of freedom are continuous and there are more elements in the mesh. For $p = 2$, the number of degrees of freedom in DG increases to six, which is

enough to better compensate for the fact that the solution is discontinuous across elements. It is no surprise that for the same number of elements, DG performs much better than SUPG. However, that is not a fair comparison as the number of degrees of freedom per element is much less using SUPG and it is the number of degrees of freedom, not the number of elements, that adds to the computational expense.

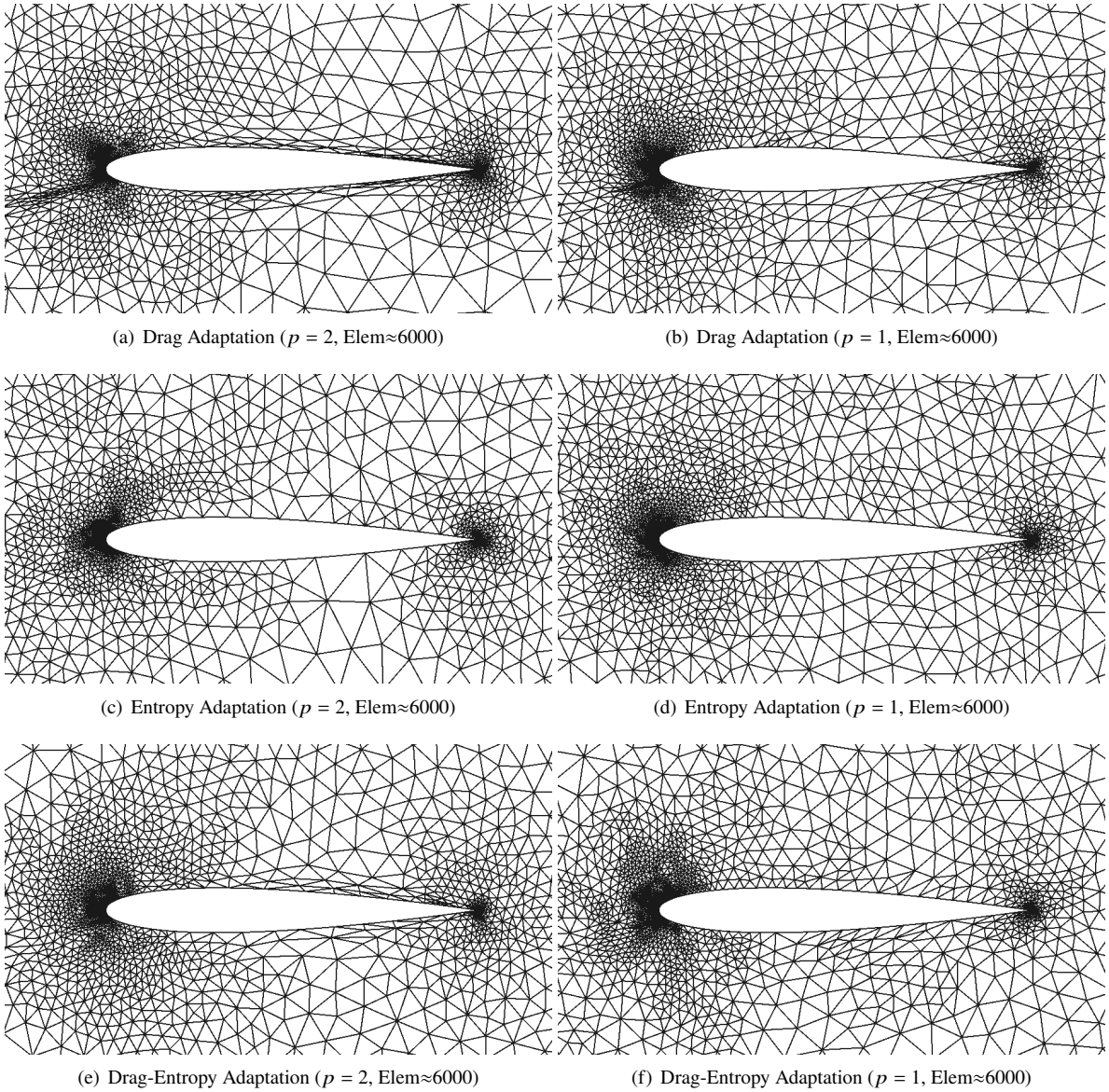
This case is a benign simulation with no major flow discontinuities. Consequently, there is not too much difference between the three different error estimation approaches, especially for DG. For SUPG, there is a small benefit in accuracy using just the entropy-adjoint compared to using just the drag-based adjoint for $p = 2$. This is a particular case that shows no loss in accuracy using entropy variables to drive adaptation compared to the typical output-based adjoint approach. However, as previous works have highlighted, this is only because there are no regions in the domain that generate spurious entropy that are far from the airfoil. Since there is not a major difference in performance between using the drag-based adjoint versus entropy variables, the combined approach does not provide much of an impact. However, it does not negatively impact the performance.

Sample meshes generated using SUPG with the various error indicator approaches for both $p = 2$ and $p = 1$ are shown in Figure 5. All of these meshes have approximately 6000 elements, but the meshes generated using $p = 2$ have far more degrees of freedom. Both meshes obtained using the drag-based adjoint allocate a large number of elements to the stagnation streamline region well in front of the airfoil, where the adjoint has a weak singularity [12]. However, at $p = 2$ significantly more elements are allocated to this region. Neither meshes generated using the entropy variables or the combined approach show this propensity to allocate elements to this region. This explains why these approaches produce slightly lower error levels at $p = 2$. For meshes generated using $p = 2$, there is also more refinement near the surface of the airfoil in the middle region, except for meshes that use the entropy variables approach. When using entropy variables, all refinement is focused on the nose and tip of the airfoil where it is more critical to accurately predict drag.

A comparison of meshes generated using SUPG and DG with the various error indicator approaches for $p = 2$ is shown in Figure 6. For SUPG, there are approximately 1500 elements in each mesh, while there are approximately 2000 elements in the DG meshes. Despite the scarcity in the number of elements in each mesh compared to the previous figure, these are the finest DG meshes shown in Figure 4 since the number of degrees of freedom per element is much higher compared to SUPG. Overall, the meshes are pretty comparable between DG and SUPG for a given error strategy. Since the meshes are much coarser from an element perspective, all of the extra refinement in the stagnation streamline region is not present for the drag-based adjoint cases. This explains where there was minimal difference between the entropy variables and output-based approach for DG. These meshes are not fine enough to warrant the unnecessary stagnation streamline refinement. A reasonable conclusion is that for this case the benefit of the entropy variables approach is much stronger in SUPG compared to DG since SUPG requires far more elements at the same degrees of freedom level. When there are more elements in a given mesh, they are more likely to be placed in unnecessary regions for the drag-based adjoint approach.

Figure 7 shows the convergence of the lift coefficient for the various adaptation strategies using both SUPG and DG. Instead of using the drag-based adjoint for the output-based adjoint and combined approaches, the lift-based adjoint is used instead. Finding the lift truth solution was done in the same manner as was done for the drag truth solution. The difference in truth solutions between SUPG and DG is 3×10^{-7}. When it comes to predicting lift coefficient, the relative difference in error between $p = 2$ and $p = 1$ is still significant for both SUPG and DG. However, there is much more variability in predicting lift coefficient among the different error indicator approaches compared to predicting the drag coefficient, where all of the different approaches were pretty similar. For $p = 1$, all of the approaches produce lower lift coefficient error levels when using SUPG compared to DG. The entropy variables approach in particular outperforms all of the other approaches. However, for $p = 2$ the results obtained using DG and much more comparable, if not slightly better compared, to SUPG. For the higher order of $p = 2$, the entropy variables approach suffers (especially for DG), while the combined approach generally produces lower error levels for both SUPG and DG. Since there is more variability between the output-based and entropy variables approach when calculating lift coefficient, the combined approach makes much more of a positive impact.

The benefits of the combined approach are much more obvious upon closer examination of the meshes. A comparison of meshes for both $p = 2$ and $p = 1$ generated using SUPG is shown in Figure 8. The meshes generated using the lift-based adjoints refine the stagnation streamline area to an even greater degree compared to the meshes generated using the drag-based adjoints. The entropy variables approach does not refine this region, nor does it refine near the surface of the airfoil to the same degree as the output-based adjoint. This was not an issue for calculating the drag coefficient, but this may be more of an issue for calculating lift coefficient. The entropy variables approach in particular lacks refinement near the surface close to the tip and nose. The combined approach may strike a better balance between



(a) Drag Adaptation ($p = 2$, Elem≈6000)

(b) Drag Adaptation ($p = 1$, Elem≈6000)

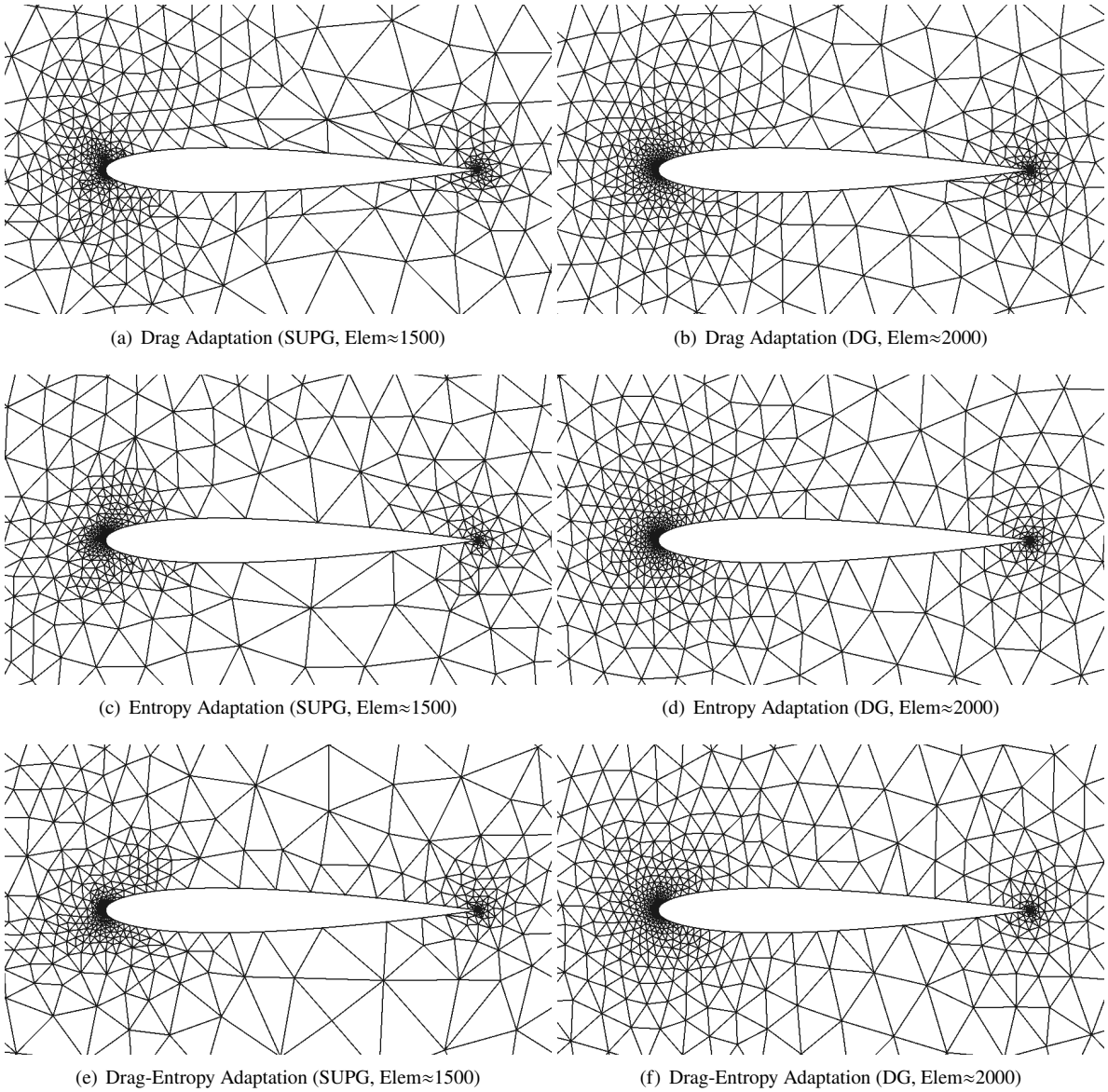
(c) Entropy Adaptation ($p = 2$, Elem≈6000)

(d) Entropy Adaptation ($p = 1$, Elem≈6000)

(e) Drag-Entropy Adaptation ($p = 2$, Elem≈6000)

(f) Drag-Entropy Adaptation ($p = 1$, Elem≈6000)

Fig. 5 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Meshes generated using SUPG for various error indicators.



(a) Drag Adaptation (SUPG, Elem≈1500)

(b) Drag Adaptation (DG, Elem≈2000)

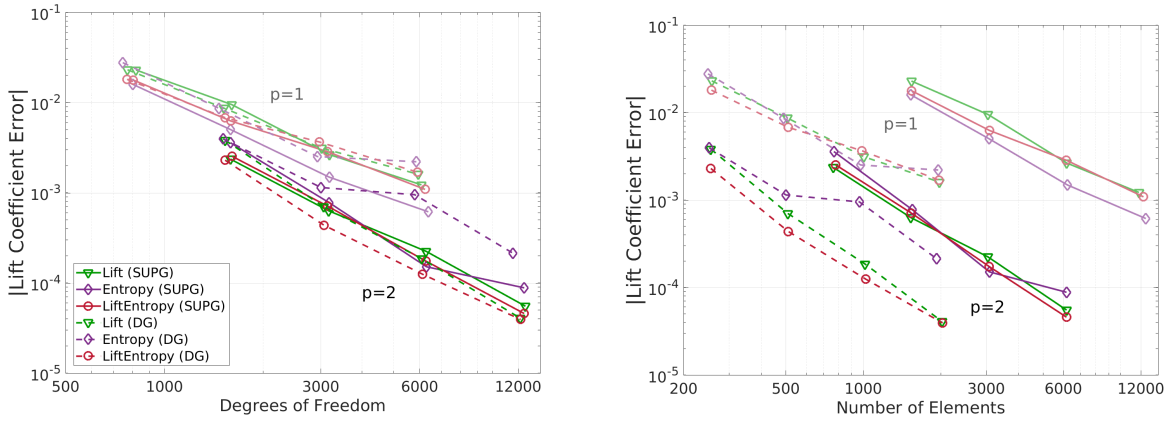
(c) Entropy Adaptation (SUPG, Elem≈1500)

(d) Entropy Adaptation (DG, Elem≈2000)

(e) Drag-Entropy Adaptation (SUPG, Elem≈1500)

(f) Drag-Entropy Adaptation (DG, Elem≈2000)

Fig. 6 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Meshes generated using SUPG and DG for various error indicators at $p = 2$.



(a) Lift Error Versus Degrees of Freedom

(b) Lift Error Versus Number of Elements

Fig. 7 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Comparison of lift coefficient convergence histories for various adaptation methods.

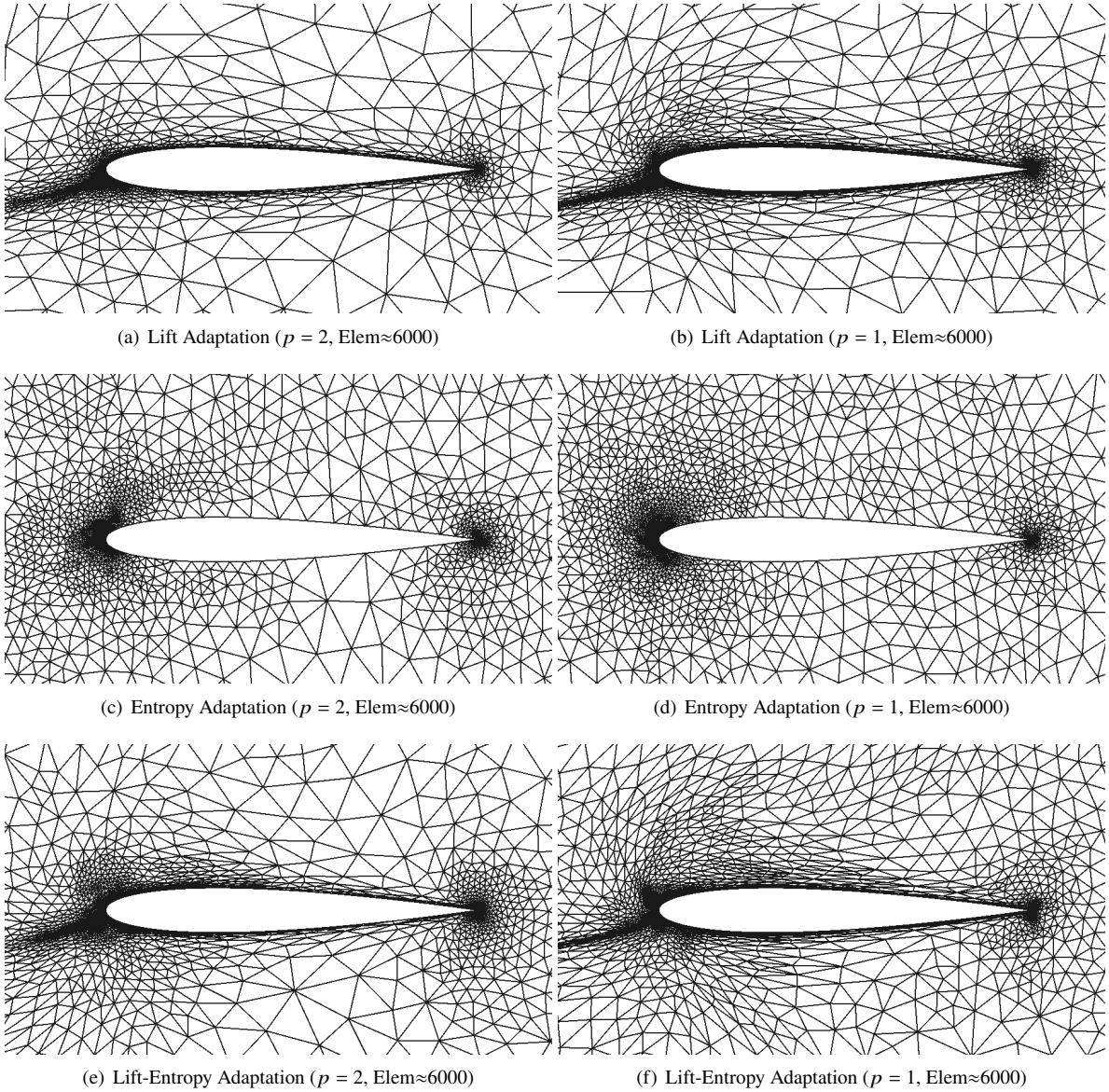
refining near the surface of the airfoil, near the nose and trailing edge especially, while not refining the stagnation streamline as much as the output-based approach. The combined approach at $p = 2$ does not refine the stagnation streamline region far from the airfoil as much as $p = 1$, which explains why it performs better for $p = 2$.

Meshes at $p = 2$ generated using DG and SUPG are directly compared in Figure 9. Just as in Figure 6, for SUPG there are approximately 1500 elements in each mesh, while there are approximately 2000 elements in the DG meshes. As before, the meshes between DG and SUPG are pretty comparable for the same error indicator approach. Unlike in Figure 6, there are much more noticeable differences in the meshes among the different error indicator approaches despite the meshes being coarser than in Figure 8. The stagnation streamline region and regions near the surface of the airfoil are significantly more refined in the output-based approach compared to the entropy variables approach. While the extra refinement in the stagnation streamline region may not be necessary, the entropy variables approach is not refining the regions near the surface of the airfoil enough. This is why the combined approach works so well for calculating lift coefficient, as both the output-based approach and entropy variables approach have limitations in refining the appropriate regions for calculating lift.

All of the results documented up to this point use MOESS for the mesh adaptation strategy. In Figure 10, the SUPG results at $p = 2$ from the previous figures are compared to similar simulations that use the *a priori* mesh adaptation approach instead of MOESS. Overall, the *a priori* approach yields lower drag coefficient error levels compared to MOESS, regardless of the error indicator approach. However, the lift coefficient error performance between MOESS and *a priori* is dependent on chosen error indicator approach. The combined approach and the entropy variables approach yield similar performance between MOESS and *a priori*, whereas the lift-based adjoint approach performs much better using MOESS.

Figure 11 presents sample meshes from the various approaches shown in Figure 10. For both the drag-based and lift-based adjoint approaches, MOESS focuses more refinement in the stagnation streamline region and near the surface of the airfoil. This is because MOESS picks up anisotropic refinement much better than *a priori*. Having refinement near the surface of the airfoil in the middle section of the airfoil is not as essential for accurately predicting the drag coefficient, which is why the *a priori* approach outperforms MOESS. Conversely, the extra refinement in the mid-board section of the airfoil leads to a better estimation of the lift since having a good estimation of the pressure on both the bottom and top surface of the airfoil is critical for calculating the lift coefficient. It is obvious based on the sample mesh generated with the lift-based adjoint approach that the *a priori* approach is unable to produce ideal anisotropic elements on the boundary of the airfoil. This leads to suboptimal performance compared to MOESS. A reasonable conclusion is that if the desired output requires highly anisotropic elements, using MOESS is a more desirable approach compared to the *a priori* approach.

The same conclusions regarding the relative performance between MOESS and *a priori* for the output-based adjoint approach also apply to the combined approaches. However, the contributions from the entropy variable indicator in



(a) Lift Adaptation ($p = 2$, Elem≈6000)

(b) Lift Adaptation ($p = 1$, Elem≈6000)

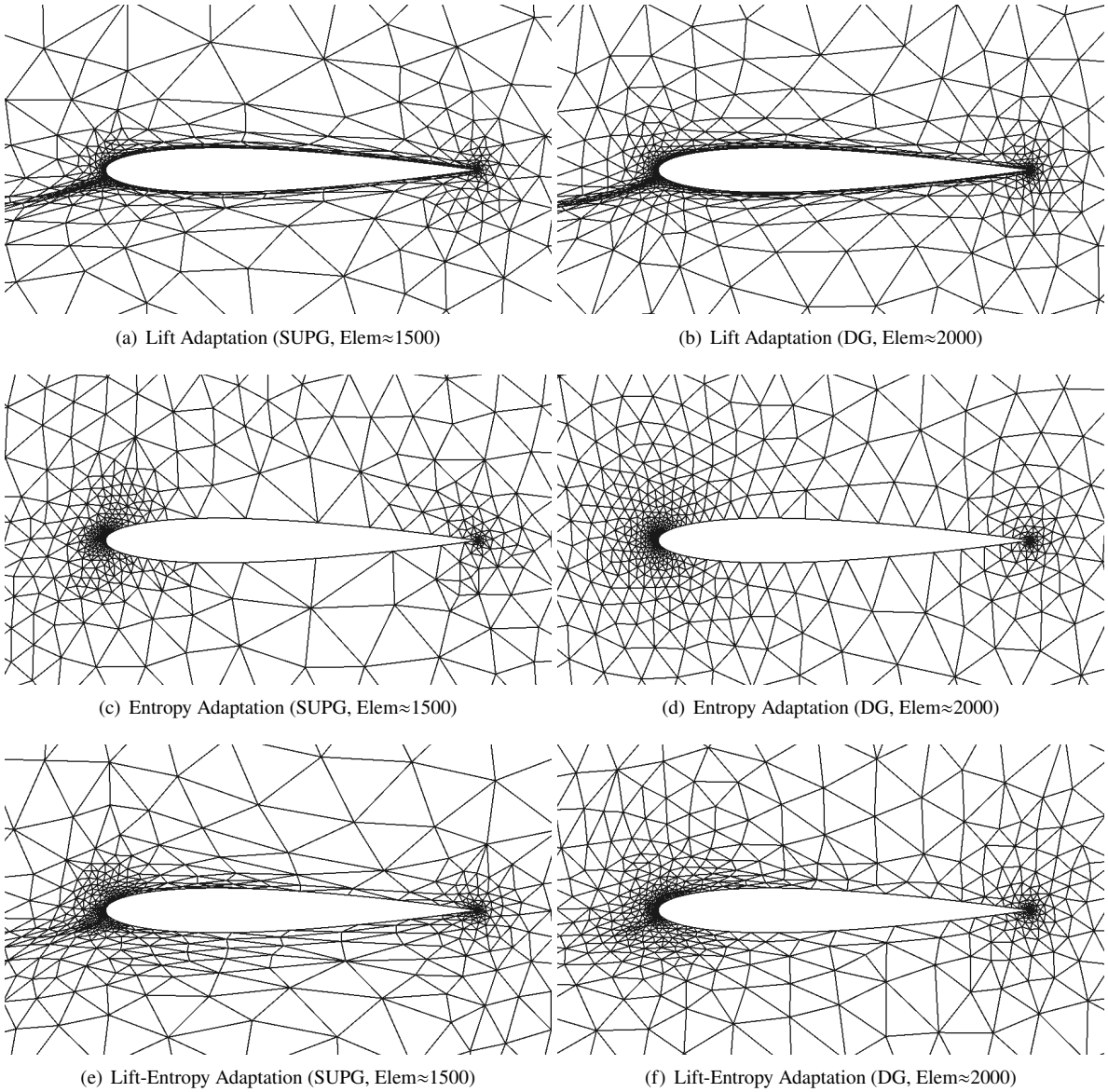
(c) Entropy Adaptation ($p = 2$, Elem≈6000)

(d) Entropy Adaptation ($p = 1$, Elem≈6000)

(e) Lift-Entropy Adaptation ($p = 2$, Elem≈6000)

(f) Lift-Entropy Adaptation ($p = 1$, Elem≈6000)

Fig. 8 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Meshes generated using SUPG for various error indicators.



(a) Lift Adaptation (SUPG, Elem≈1500)

(b) Lift Adaptation (DG, Elem≈2000)

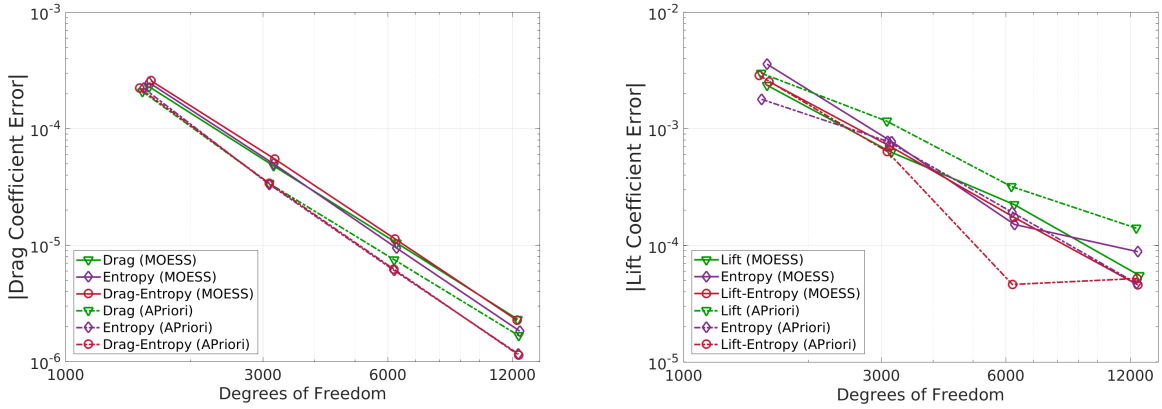
(c) Entropy Adaptation (SUPG, Elem≈1500)

(d) Entropy Adaptation (DG, Elem≈2000)

(e) Lift-Entropy Adaptation (SUPG, Elem≈1500)

(f) Lift-Entropy Adaptation (DG, Elem≈2000)

Fig. 9 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Meshes generated using SUPG and DG for various error indicators at $p = 2$.



(a) Drag Error Versus Degrees of Freedom

(b) Lift Error Versus Degrees of Freedom

Fig. 10 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Comparison of drag and lift coefficient convergence histories for various mesh adaptation strategies using SUPG at $p = 2$.

the combined approach limit the dramatic differences present using the output-based approach. While the stagnation streamline region is still targeted for refinement in both combined approaches using MOESS, the refinement is not as fine as it was for the output-based approaches. This is why for estimating lift coefficient in particular, the combined approaches outperform the output-based approaches in both MOESS and *a priori*. The entropy variables approach using *a priori* produces lower drag and lift errors compared to MOESS because the mesh is much coarser near the surface of the airfoil. While the output-based approaches in MOESS produced too much refinement in this region, the entropy variables approach does not refine this region enough. Since the entropy variables and output-based adjoint approaches target different regions of the domain for both MOESS and *a priori*, there can be a clear benefit using the combined approach so that all regions of the domain required to accurately calculate a particular output are adequately refined.

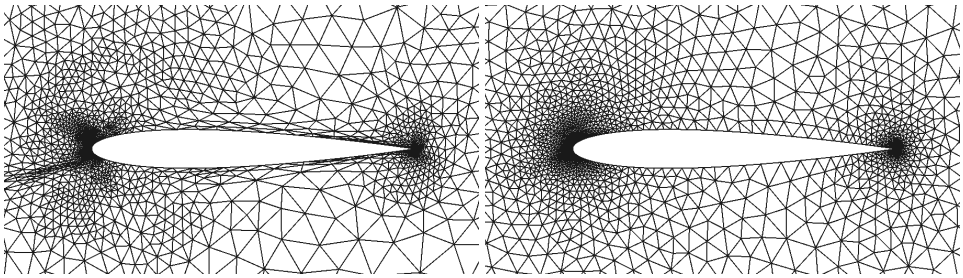
B. Diamond Airfoil in Two-Dimensional, Inviscid, Supersonic Flow: $M_\infty = 1.5,\ \alpha = 2°$

The next case was chosen specifically because there are multiple flow discontinuities present in the computational domain. In this case, a thin diamond airfoil, with a thickness-to-chord ratio of 0.05, is subjected to a supersonic flow of $M_\infty = 1.5$ at an angle of attack of $\alpha = 2°$. The supersonic flow produces shocks emanating from the upper and lower surfaces of the diamond, as shown in Figure 12. The simulation can be solved with an inviscid discretization because the shocks are weak. The weak shocks are due to the small thickness of the diamond airfoil.

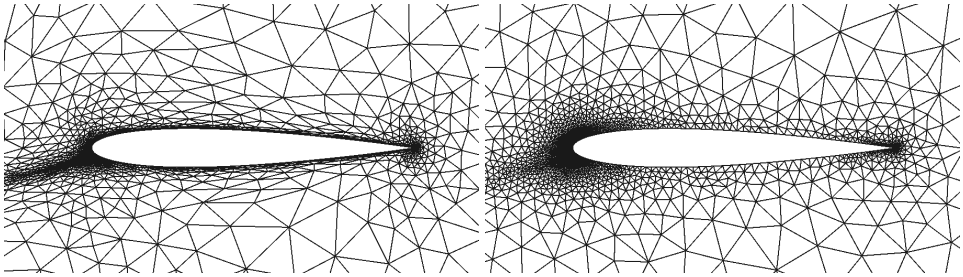
The grid is a square with a side length of ten chords. The diamond is placed slightly off from the center of the domain so that the shocks emanating from the airfoil do not directly impact the corners of the domain. The initial linear, $q = 1$, mesh is made up of unstructured, triangular elements, as shown in Figure 13. Cases run with a solution interpolation order of $p = 1$ used the following three degrees of freedom targets: 375, 750, and 1500. Cases that had a solution interpolation order of $p = 2$ used the following degrees of freedom targets: 750, 1500, and 3000. As before, multiple solution iterations were performed at each target so that an average could be done over five separate iterations. The drag coefficient was again chosen as the analyzed engineering output for this case. Consequently, the drag-based adjoint approach was analyzed along with the entropy variables approach and the combined approach that uses the drag-based adjoint.

Figure 14 presents the drag coefficient error convergence for the adaptation strategies using both DG and SUPG. The mesh adaptation strategy used for these cases was MOESS. The truth solution was obtained by generating an adaptive mesh with approximately 3,000 degrees of freedom using the drag-based adjoint with a DG formulation, uniformly refining that mesh, and generating a solution at $p = 3$. A similar truth solution was obtained using SUPG. The difference in the truth solutions between SUPG and DG was 2×10^{-7}.

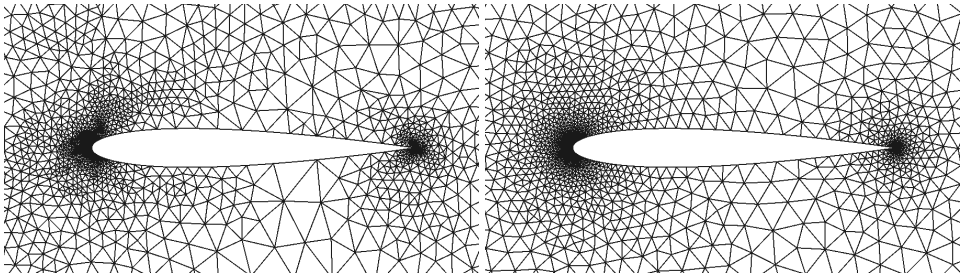
A noteworthy observation from Figure 14 is that generally there is not much of a benefit running this simulation at $p = 2$. Unlike in subsonic airfoil case where simulations at $p = 2$ always produced lower output errors at the same degree of freedom level, regardless of the discretization and adaptation approach used, for this case it is more advantageous to



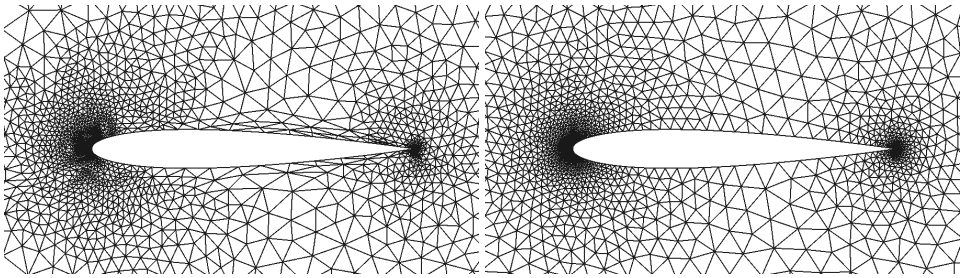
(a) Drag Adaptation (MOESS, Elem≈6000) (b) Drag Adaptation (*A Priori*, Elem≈6000)



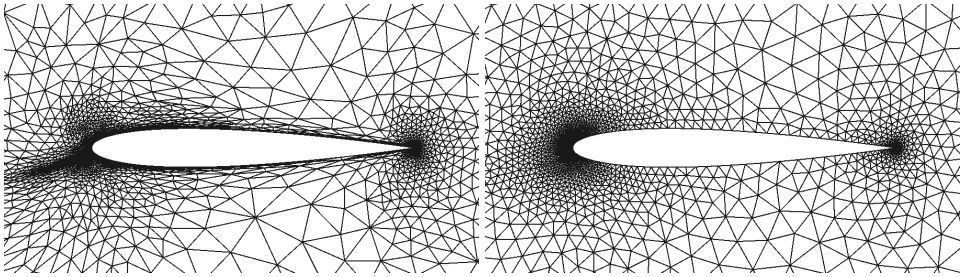
(c) Lift Adaptation (MOESS, Elem≈6000) (d) Lift Adaptation (*A Priori*, Elem≈6000)



(e) Entropy Adaptation (MOESS, Elem≈6000) (f) Entropy Adaptation (*A Priori*, Elem≈6000)



(g) Drag-Entropy Adaptation (MOESS, Elem≈6000) (h) Drag-Entropy Adaptation (*A Priori*, Elem≈6000)



(i) Lift-Entropy Adaptation (MOESS, Elem≈6000) (j) Lift-Entropy Adaptation (*A Priori*, Elem≈6000)

Fig. 11 NACA 0012 $M_\infty = 0.3$, $\alpha = 5°$: Comparing mesh adaptation strategies using SUPG at $p = 2$.

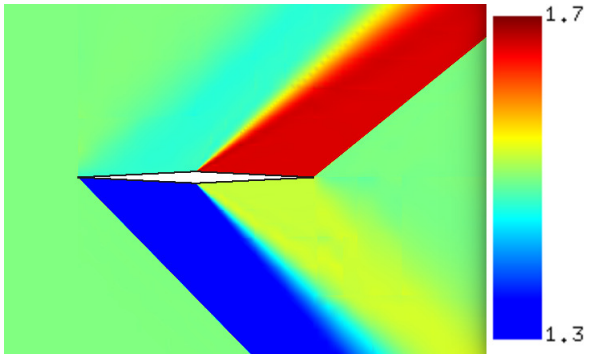


Fig. 12 Diamond $M_\infty = 1.5$, $\alpha = 2°$: Mach number contours (range: 1.3-1.7).

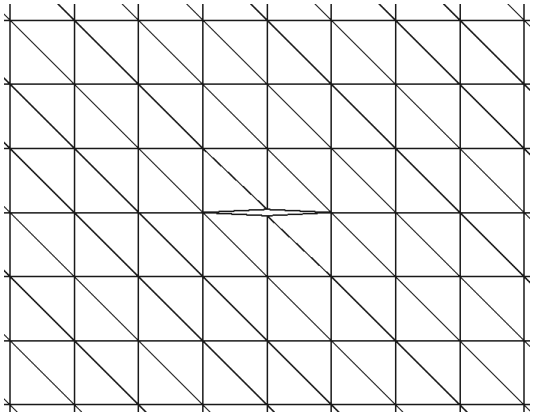
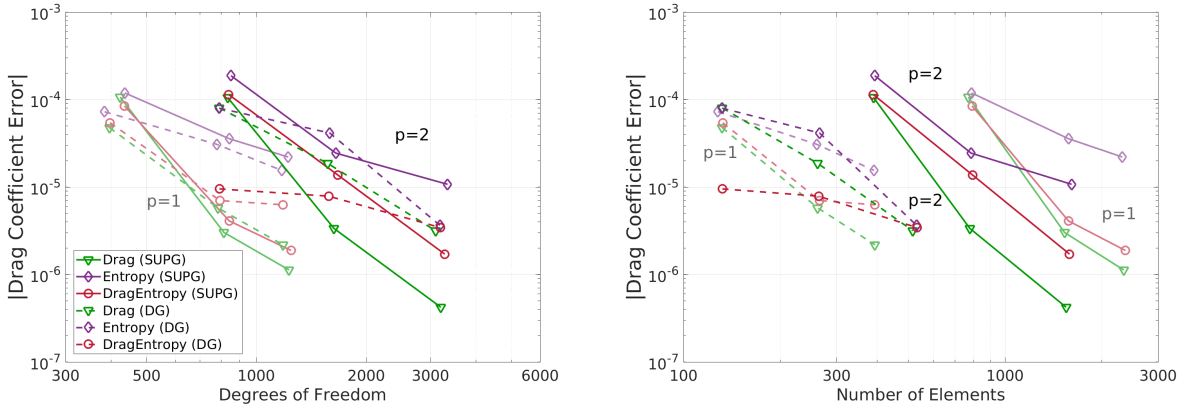


Fig. 13 Initial diamond airfoil mesh.



(a) Drag Error Versus Degrees of Freedom

(b) Drag Error Versus Number of Elements

Fig. 14 Diamond $M_\infty = 1.5$, $\alpha = 2°$: Comparison of drag coefficient convergence histories for various adaptation methods.

allocate degrees of freedom to the number of elements and not to higher elemental orders. A reasonable conclusion is that capturing the flow discontinuities with more elements produces better drag error estimates than using fewer higher order-elements. For the same number of elements, using $p = 2$ does provide a benefit for SUPG. However, that benefit is not as significant as it was for the previous cases.

Another observation, consistent with our previous works [15, 16] is that for this case the entropy variables approach is inferior to the output-based adjoint approach, regardless of whether SUPG or DG is used. This is because the entropy variables approach targets only the shocks where a large amount of entropy is generated. Since the entropy variable approach performs poorly, it negatively impacts the combined approach. This is a situation where a more sophisticated way of combining the indicators, weighted more towards those generated from the output-based adjoint, would likely improve the error estimation. In terms of comparing SUPG versus DG, using the combined approach and the drag-based adjoint approach, respectively, in SUPG produces lower drag error estimates compared to DG. Unlike in the previous airfoil case at subsonic flow, there is much more variation in drag error levels between SUPG and DG. If we consider our previous observation that for this case the number of elements is more essential than the number of degrees of freedom in estimating drag coefficient because outside of the shocks the domain is pretty uniform, it is reasonable to summarize that SUPG performs better than DG for this case because for the same number of degrees of freedom SUPG uses far more elements.

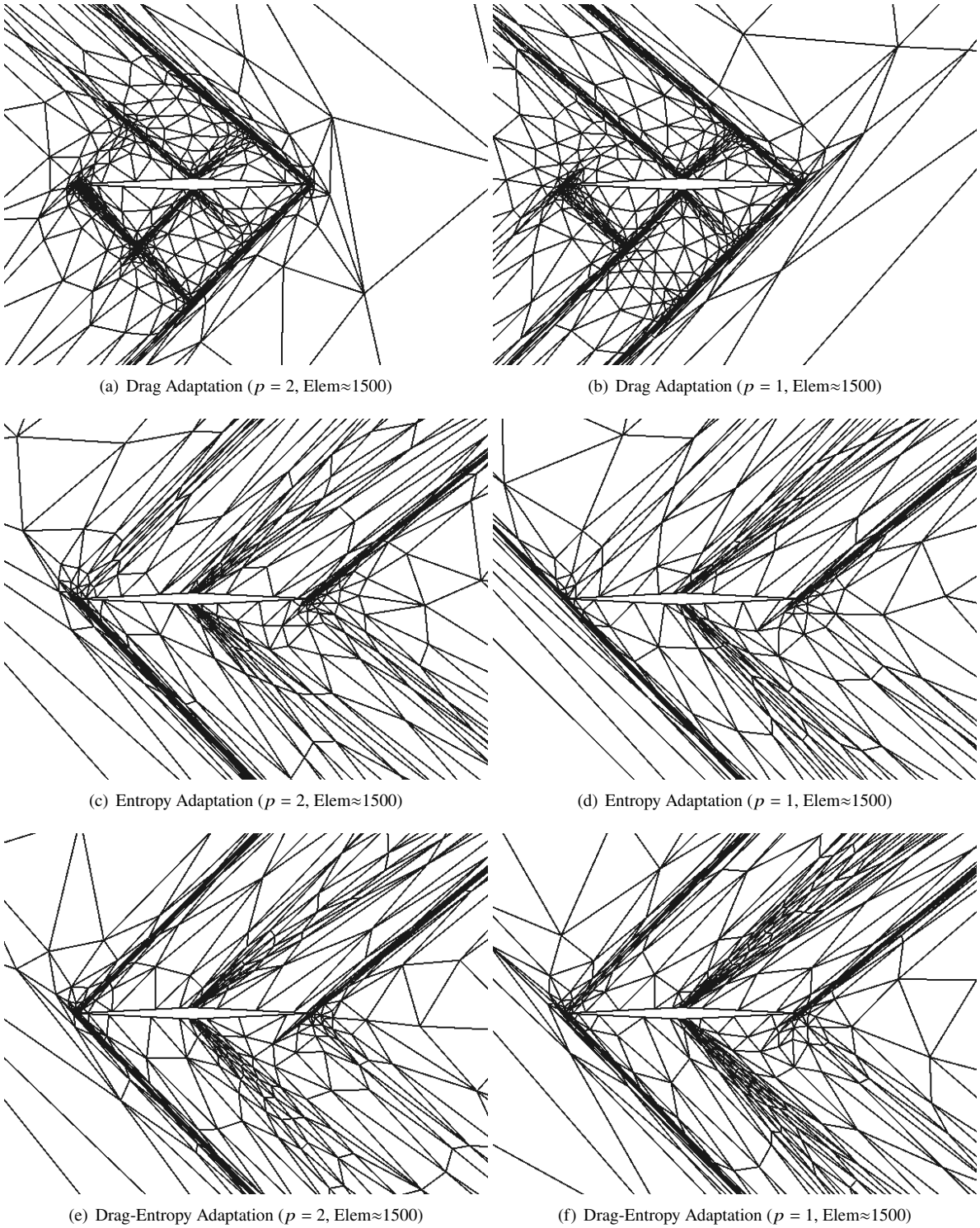
Samples meshes of around 1500 elements generated using SUPG at $p = 2$ and $p = 1$ are shown in Figure 15. For the entropy variables approach, the refinement is targeted in the regions consumed by the discontinuities emanating from the nose and tip of the airfoil. This refinement extends unnecessarily to the far-field boundary because this approach has no way of discriminating regions of the domain that do not have much of an effect on the output of interest. This accounts for why the entropy variables approach produces inferior results relative to the output-based approach. The output-based approach restricts the refinement to regions near the airfoil in the shape of a diamond. Whether all of the refinement that travels upstream from the airfoil is necessary to produce a good drag estimate is a worthwhile question, since the meshes generated from the combined approach do not show this feature. However, since the combined approach views refinement considerations from the output-based and entropy variables approach equally, there is still a lot of unnecessary refinement far from the airfoil present in the combined approach. This leads to inferior performance compared to the output-based approach. This further suggests that an alternative way of combining the indicators to favor the output-based approach is likely necessary to improve performance.

The effect the mesh adaptation strategy has on the various adaptive indicator approaches is shown in Figure 16. In this figure, drag coefficient error comparisons between cases that use MOESS versus cases that use the *a priori* mesh adaptation approach are shown. All of the cases have a solution approximation order of $p = 2$. The most important takeaway from this comparison is how much better the output-based approach performs with MOESS compared to *a priori*. In the subsonic airfoil case, highly anisotropic refinement was not necessary to predict an accurate drag coefficient. In this case, because of the flow discontinuities in the domain that must be captured, anisotropic refinement is much more beneficial and leads to a better drag error estimate for a given cost. The combined approach and entropy variables approach do not show as significant of a difference between MOESS and *a priori* since both are degraded by all of the unnecessary refinement of the discontinuous regions of the flow domain far from the airfoil.

An examination of sample meshes made up of approximately 1500 elements in Figure 17 better illustrates the benefit of MOESS over *a priori* for this case. While the differences between the meshes for the entropy variables and the combined approaches are subtle, there is a significant difference between the meshes for the output-based adjoint approach. Using the *a priori* approach no longer produces the diamond pattern of mesh refinement shown with MOESS. Instead, the *a priori* approach focuses most of its refinement on the discontinuity emanating from the nose. Without the ability to capture the discontinuity with highly anisotropic elements, the *a priori* approach has to refine this region with far more elements. This consequently leads to less refinement in other regions of the domain, which negatively impacts the ability to produce an accurate drag error estimate.

VI. Conclusions and Future Work

In this work, we further demonstrated the potential of the combined approach for mesh adaptation and error estimation by implementing a stabilized continuous finite element formulation. The potential benefits of these methods over discontinuous methods like DG, make the advantages of the combined approach even more pronounced. The same issues that were previously discovered for both the output-based adjoint approach and the entropy variables approach while using DG, are still present in SUPG. Using the combined approach with SUPG offers the same potential benefits as it did with DG. However, there is still more work that needs to be done to demonstrate that the combined approach is



(a) Drag Adaptation ($p = 2$, Elem≈1500)

(b) Drag Adaptation ($p = 1$, Elem≈1500)

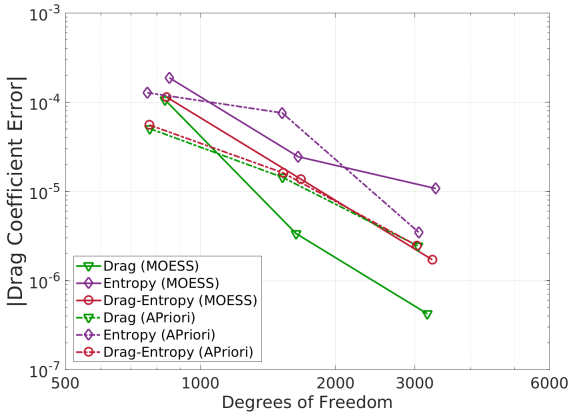
(c) Entropy Adaptation ($p = 2$, Elem≈1500)

(d) Entropy Adaptation ($p = 1$, Elem≈1500)

(e) Drag-Entropy Adaptation ($p = 2$, Elem≈1500)

(f) Drag-Entropy Adaptation ($p = 1$, Elem≈1500)

Fig. 15 Diamond $M_\infty = 1.5$, $\alpha = 2°$: Meshes generated using SUPG for various error indicators.



(a) Drag Error Versus Degrees of Freedom

Fig. 16 Diamond $M_\infty = 1.5$, $\alpha = 2°$: Comparison of drag coefficient convergence histories for various mesh adaptation strategies using SUPG at $p = 2$.

a viable option for mesh adaptation when using SUPG. In cases such as the diamond airfoil, where the entropy variables approach is not a viable option for accurate error estimation, the combined approach suffers.

The only version of the combined approach demonstrated in this work is the most basic approach where the unmodified indicators from both the output-based and entropy-based adjoint approaches are multiplied directly. The problem with this approach is that it is more computationally expensive than either of the individual approaches alone. Previous works highlighted approaches where the output-based indicator came from a coarse-space output-based adjoint, which is far less expensive to compute. In addition, masking techniques were shown to have a substantial benefit as well. These combined approaches need to be analyzed as well using the SUPG formulation and compared to similar simulations using DG.

As was demonstrated in our previous work [17], the benefits of the combined approach extend to unsteady simulations. For unsteady simulations, the fine-space output-based adjoint has to be computed for each time step. This is computationally expensive and makes a combined approach that uses a coarse-space output-based adjoint more desirable. Future work will focus on demonstrating the combined approach for an unsteady simulation using the SUPG discretization. This effort will provide further verification that the combined approach can be used for multiple types of spatial discretizations and often yield superior performance compared to the output-based adjoint approach.

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.

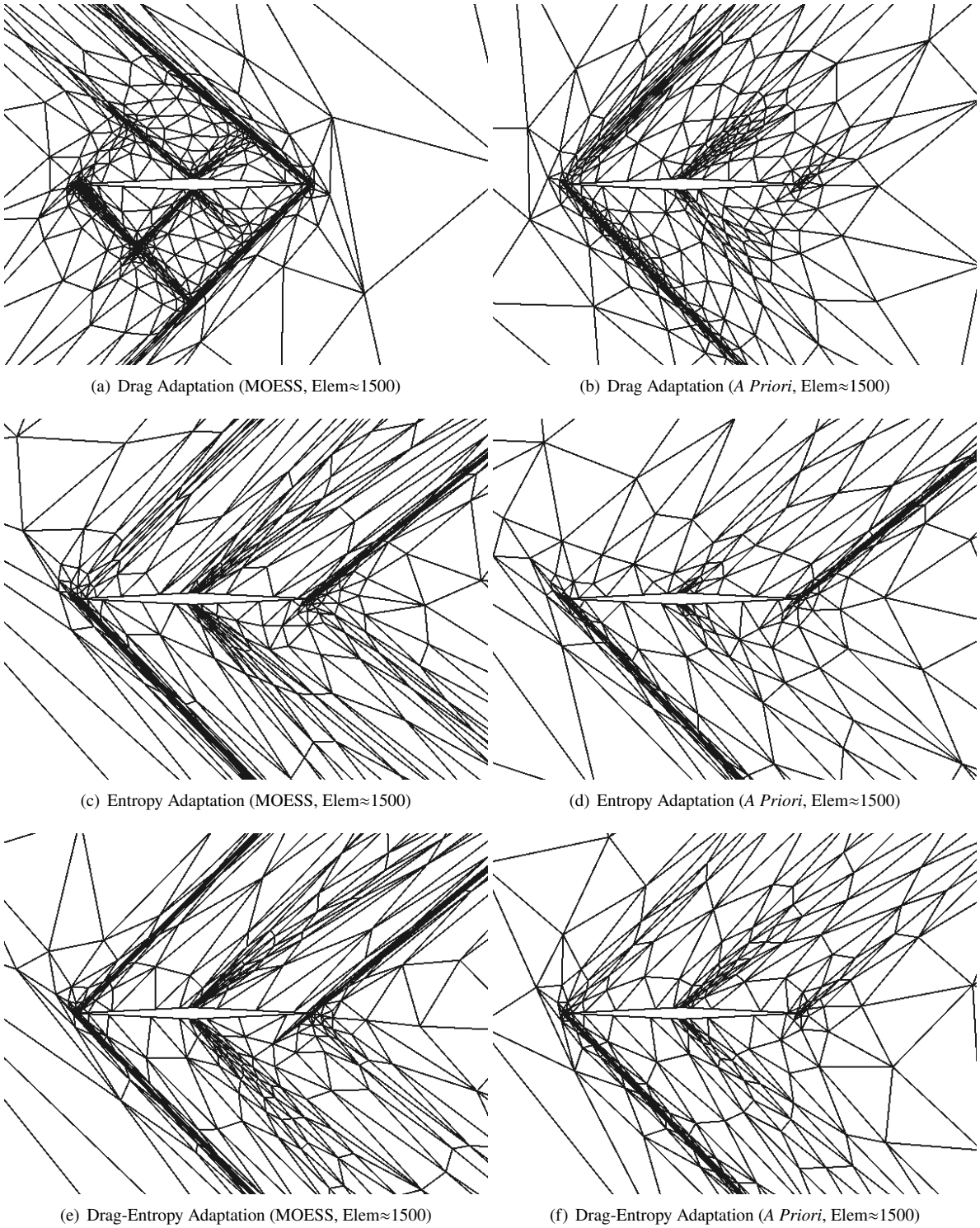
[2] Dompierre, J., Vallet, M.-G., Bourgault, Y., Fortin, M., and Habashi, W. G., "Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part III: Unstructured Meshes," *International Journal for Numerical Methods in Fluids*, Vol. 39, 2002, pp. 675–702.

[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. doi:10.1016/j.jcp.2007.02.007.

[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.



(a) Drag Adaptation (MOESS, Elem≈1500)

(b) Drag Adaptation (*A Priori*, Elem≈1500)

(c) Entropy Adaptation (MOESS, Elem≈1500)

(d) Entropy Adaptation (*A Priori*, Elem≈1500)

(e) Drag-Entropy Adaptation (MOESS, Elem≈1500)

(f) Drag-Entropy Adaptation (*A Priori*, Elem≈1500)

Fig. 17 Diamond $M_\infty = 1.5$, $\alpha = 2°$: Comparing mesh adaptation strategies using SUPG at $p = 2$.

[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. doi:10.2514/6.2009-1303.

[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. doi: 10.2514/1.J050073.

[10] 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.

[11] 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.

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

[13] Fidkowski, K. J., and Roe, P. L., "Entropy-based Mesh Refinement, I: The Entropy Adjoint Approach," AIAA Paper 2009-3790, 2009. doi:10.2514/6.2009-3790.

[14] 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. doi:10.1137/090759057.

[15] Doetsch, K. T., and Fidkowski, K. J., "A Combined Entropy and Output-based Adjoint Approach for Mesh Refinement and Error Estimation," AIAA Paper 2018-0918, 2018.

[16] Doetsch, K., and Fidkowski, K. J., "Combined Entropy and Output-based Adjoint Approach for Mesh Refinement and Error Estimation," *AIAA Journal*, Vol. 57, No. 8, 2019.

[17] Doetsch, K. T., and Fidkowski, K. J., "Unsteady Combined Entropy and Output-based Adjoint Approach for Mesh Refinement and Error Estimation," AIAA Paper 2019–2951, 2019.

[18] Brooks, A., and Hughes, T., "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, 1982, pp. 199–259.

[19] Shakib, F., Hughes, T. J., and Johan, Z., "A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 89, No. 1, 1991, pp. 141–219. Second World Congress on Computational Mechanics.

[20] Bassi, F., and Rebay, S., "A High–order discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 131, 1997, pp. 267–279.

[21] Houston, P., Hartmann, R., and Süli, E., "Adaptive discontinuous Galerkin finite element methods for compressible fluid flows," *Numerical Methods for Fluid Dynamics VII, ICFD*, edited by M. Baines, 2001, pp. 347–353.

[22] Oliver, T. A., Fidkowski, K. J., and Darmofal, D. L., "Multigrid Solution for High–order Discontinuous Galerkin Discretization of the Compressible Navier-Stokes Equations," *Third International Conference on Computational Fluid Dynamics, Toronto, Canada*, 2004. doi:10.1007/3-540-31801-1_64.

[23] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "*p*-Multigrid solution of high–order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 207, 2005, pp. 92–113. doi:10.1016/j.jcp.2005.01.005.

[24] Anderson, W. K., Wang, L., Kapadia, S., Tanis, C., and Hilbert, B., "Petrov–Galerkin and discontinuous-Galerkin methods for time-domain and frequency-domain electromagnetic simulations," *Journal of Computational Physics*, Vol. 230, No. 23, 2011, pp. 8360–8385.

[25] Anderson, K., Newman, J., and Karman, S., "Stabilized Finite Elements in FUN3D," 2017. doi:10.2514/6.2017-0077.

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

[27] Brooks, A. N., and Hughes, T. J., "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, No. 1, 1982, pp. 199–259.

[28] Tezduyar, T., and Hughes, T., "Finite Element Formulations for Convection Dominated Flows with Particular Emphasis on the Compressible Euler Equations," 1983. doi:10.2514/6.1983-125.

[29] Hughes, T. J., and Mallet, M., "A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 58, No. 3, 1986, pp. 305–328. doi:https://doi.org/10.1016/0045-7825(86)90152-0.

[30] Hughes, T., Scovazzi, G., and Tezduyar, T., "Stabilized Methods for Compressible Flows," *Journal of Scientific Computing*, Vol. 43, 2010, pp. 343–368. doi:10.1007/s10915-008-9233-5.

[31] Whiting, C., and Jansen, K., "A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis," *International Journal for Numerical Methods in Fluids - INT J NUMER METHOD FLUID*, Vol. 35, 2001, pp. 93–116. doi:10.1002/1097-0363(20010115)35:13.0.CO;2-G.

[32] 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.

[33] Kirk, B. S., "Adaptive Finite Element Simulation of Flow and Transport Applications on Parallel Computers," Ph.D. thesis, The University of Texas, Austin, 2005.

[34] Reza Ahrabi, B., "An hp-Adaptive Petrov-Galerkin Method for Steady-State and Unsteady Flow Problems," Ph.D. thesis, 08 2015.

[35] Reza Ahrabi, B., Anderson, K., and III, J., "An adjoint-based hp-adaptive stabilized finite-element method with shock capturing for turbulent flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 318, 2017, pp. 1030–1065. doi:10.1016/j.cma.2017.02.001.

[36] Wang, Q., "Convergence of the least squares shadowing method for computing derivative of ergodic averages," *SIAM Journal on Numerical Analysis*, Vol. 52, No. 1, 2014, pp. 156–170.

[37] Ahrabi, B. R., and Mavriplis, D. J., "A scalable solution strategy for high-order stabilized finite-element solvers using an implicit line preconditioner," *Computer Methods in Applied Mechanics and Engineering*, Vol. 341, 2018, pp. 956–984.

[38] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Web page," `https://www.mcs.anl.gov/petsc`, 2021. URL `https://www.mcs.anl.gov/petsc`.

[39] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.15, Argonne National Laboratory, 2021. URL `https://www.mcs.anl.gov/petsc`.

[40] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhäuser Press, 1997, pp. 163–202.

[41] Dupont, T., Kendall, R. P., and Rachford, H. H., Jr., "An Approximate Factorization Procedure for Solving Self-Adjoint Elliptic Difference Equations," *SIAM Journal on Numerical Analysis*, Vol. 5, No. 3, 1968, pp. 559–573. doi:10.1137/0705045.

[42] Oliphant, T. A., "An Implicit, Numerical Method for Solving Two-dimensional time-dependent diffusion problems," *Quarterly of Applied Mathematics*, Vol. 19, No. 3, 1961, pp. 221–229.

[43] Chan, T., Van, H., and Van der Vorst, H., "Approximate And Incomplete Factorizations," 1970.

[44] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[45] Fidkowski, K. J., "Output error estimation strategies for discontinuous Galerkin discretizations of unsteady convection-dominated flows," *International Journal for Numerical Methods in Engineering*, Vol. 88, No. 12, 2011, pp. 1297–1322. doi:10.1002/nme.3224.

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

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

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

[54] Fidkowski, K. J., and Chen, G., "Metric-based, goal-oriented mesh adaptation using machine learning," *Journal of Computational Physics*, Vol. 426, 2021, p. 109957.