Adjoint-Based Error Estimation and Mesh Adaptation for Problems with Output Constraints

Benjamin A. Rothacker,[*] Marco Ceze,[†] and Krzysztof J. Fidkowski[‡]

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

In this paper we extend adjoint-based output error estimation and mesh adaptation to computational fluid dynamics simulations that involve trim constraints – i.e. constraints on certain outputs. These constraints may involve outputs other than those directly targeted for error estimation and adaptation. A representative example used in this work is the reduction of numerical error in drag under fixed-lift conditions. The lift output comes in only through the constraint, but numerical errors in the lift will indirectly affect the drag calculation. We present a method for including this effect in the output error calculation, and for adapting the discretization to better predict both the target output and the trim constraint. The method relies on a discrete adjoint solution, which is also used to efficiently trim the simulations. We demonstrate the method on a discontinuous finite element discretization of the two-dimensional compressible Navier-Stokes equations.

I.   Introduction

Over the last decade, output-based error estimation and mesh adaptation techniques have been applied to a wide range of aerospace engineering problems.[1–7] Numerous challenges have already been addressed, including turbulence modeling, unsteady flows, mesh optimization, and *hp*-refinement. However, one practical aspect that has not received much attention is the problem of handling simulations with trim conditions; that is, ones with output constraints. We tackle this problem in the present work in the context of a high-order finite element discretization of the compressible Navier-Stokes equations.

Output-based error estimation and mesh adaptation have proven themselves as methods capable of both reducing the cost and quantifying the expected accuracy of a flow simulation. Such methods can be designed in a robust manner that is able to handle complex geometries[8] with little or no user intervention. Previous work in multi-output error estimation and mesh adaptation[9] has employed a user-prescribed linear combination of the outputs of interest to form a single output that drives adaptation. Here, we are interested in a single output (*e.g.* drag) and the multi-target nature of the problem arises naturally due to the boundary condition parameters (*e.g.* angle of attack) being dependent on an output – the constraint – of the solution (*e.g.* lift).

Adaptive methods with error estimation have already shown promise for optimization problems of practical interest.[10] Our work complements previous optimization work with a method for rigorously estimating the contribution of constraints to the objective function error. The relative importance of each output in the adaptive procedure comes directly from the contribution of the individual outputs to the objective function error.

In Section II, we define the problem and the notation used throughout the paper. We describe the constraint solver in Section III and the discontinuous Galerkin discretization in Section IV. Section V describes output error estimation for unconstrained problems and its extension to constrained problems. Section VI describes the mesh adaptation procedure and we show our results in Section VII. We conclude in Section VIII.

---

[*]Graduate Research Assistant, AIAA Member
[†]Postdoctoral Research Fellow, AIAA Member
[‡]Assistant Professor, AIAA Senior Member

American Institute of Aeronautics and Astronautics

II.   Problem Statement

We consider a system of N nonlinear equations resulting from a computational fluid dynamics discretization,

$$\mathbf{R}(\mathbf{U}, \boldsymbol{\alpha}) = \mathbf{0}, \tag{1}$$

where

- $\mathbf{R} \in \mathbb{R}^N$ is a vector of N residuals that must be driven to zero.
- $\mathbf{U} \in \mathbb{R}^N$ is a state vector that encodes the flow state.
- $\boldsymbol{\alpha} \in \mathbb{R}^{N_\alpha}$ is a parameter vector that constitutes the trimming "knobs."

We now define one scalar output and one vector of outputs,

- $J^{\text{adapt}}(\mathbf{U}, \boldsymbol{\alpha})$ is a scalar output on which we want to adapt.
- $\mathbf{J}^{\text{trim}}(\mathbf{U}, \boldsymbol{\alpha}) \in \mathbb{R}^{N_\alpha}$ is a vector of outputs that are used to define trimming constraints.

Our problem statement now reads: predict $J^{\text{adapt}}(\mathbf{U}, \boldsymbol{\alpha})$ to a user-specified numerical accuracy of $\varepsilon^{\text{adapt}}$, subject to the flow equations in Eqn. 1 and the following N_α constraints:

$$\mathbf{J}^{\text{trim}}(\mathbf{U}, \boldsymbol{\alpha}) - \overline{\mathbf{J}}^{\text{trim}} = \mathbf{0}, \tag{2}$$

where $\overline{\mathbf{J}}^{\text{trim}} \in \mathbb{R}^{N_\alpha}$ is a set of user-specified trim outputs. For example, in a lift-constrained simulation, $N_\alpha = 1$, $\overline{\mathbf{J}}^{\text{trim}}$ is just the single target lift, and the single parameter may be the angle of attack. Finally, we note that in later sections, we will add a subscript h or H to the above quantities to differentiate different discretization levels (e.g. fine versus coarse).

III.   Trim-Constraint Solver

Frequently in aerospace applications, CFD simulations are conducted under trimmed conditions, meaning under fixed, user-defined values of certain outputs, such as lift or pitching moment. This means that certain parameters, *e.g.* angle of attack, depend on outputs computed from the flow solution. One approach to handling this dependence is to use a feedback loop to correct the input parameters when using a standard solver that takes these parameters as inputs. We take this approach in the present work, and Figure 1 shows the details of the loop, where the various quantities are defined in Section II.
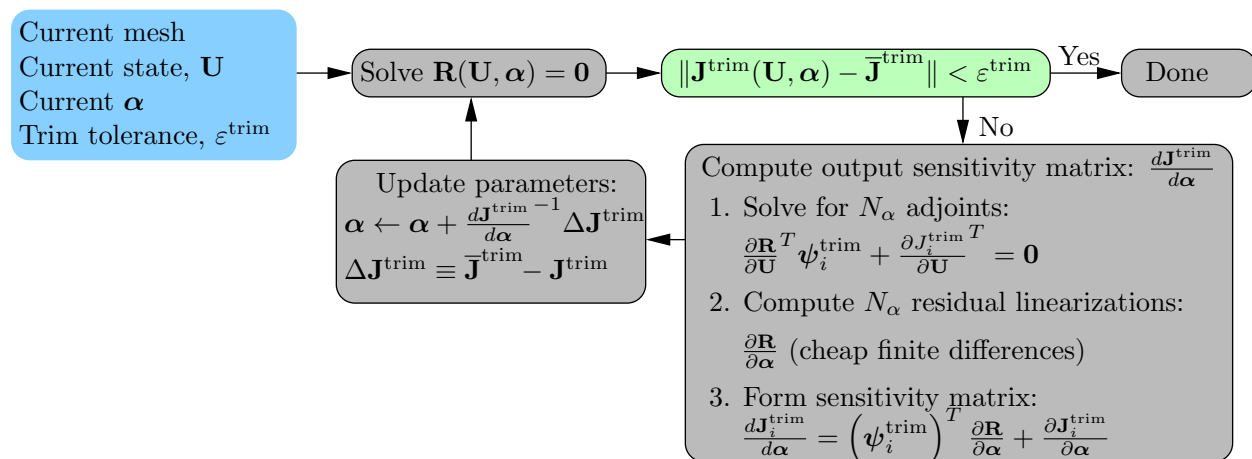

Figure 1. Feedback loop used in adjoint-based boundary-condition parameter correction, i.e. trimming.

The cycle starts by solving the flow equations using an initial guess for the inputs $\boldsymbol{\alpha}$. Then, $\mathbf{J}^{\text{trim}}$ is computed from the solution and checked against the target, $\overline{\mathbf{J}}^{\text{trim}}$, under a trimming tolerance, $\varepsilon^{\text{trim}}$. In

American Institute of Aeronautics and Astronautics

cases of multiple trimming outputs, the norm used for this check is usually the discrete L_2 norm. Until the tolerance is met, $\boldsymbol{\alpha}$ is corrected using Newton's method for which the sensitivities of $\mathbf{J}^{\text{trim}}$ with respect to $\boldsymbol{\alpha}$ are needed. The components of this sensitivity matrix are computed via inner products between an adjoint for $\mathbf{J}^{\text{trim}}$ and a residual linearization $\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}$, which is computed using inexpensive finite differences.

In cases where the target value for the output is not achievable or the initial guess is bad, the cycle in Figure 1 may not converge. In those cases, a contingency plan is needed, such as a maximum number of iterations or a restart of the cycle with a better initial guess. We choose to limit the number of iterations in this work.

We employ a discrete adjoint approach in which $\boldsymbol{\psi}_i^{\text{trim}} \in \mathbb{R}^N$ is the sensitivity of output J_i^{trim}, $1 \leq i \leq N_\alpha$, to residual perturbations added to the left-hand side of Eqn. 1. This vector is computed with one linear solve involving the transpose of the residual Jacobian matrix and driven by the linearization of J_i^{trim}.

An alternative approach to the feedback loop in Figure 1 is to solve for the flow and trimming equations simultaneously. If we define the trimming residual, $\mathbf{R}^{\text{trim}}(\mathbf{U}, \boldsymbol{\alpha})$ as the left-hand side of Eqn. 2, we obtain the coupled system

$$\begin{cases} \mathbf{R}(\mathbf{U}, \boldsymbol{\alpha}) & = & \mathbf{0} & \leftarrow & N \text{ flow equations} \\ \mathbf{R}^{\text{trim}}(\mathbf{U}, \boldsymbol{\alpha}) & = & \mathbf{0} & \leftarrow & N_\alpha \text{ trim conditions} \end{cases} \tag{3}$$

Note that typically $N_\alpha \ll N$ and that in general, both of these will be nonlinear equations. Applying Newton's method,

$$\begin{bmatrix} \frac{\partial \mathbf{R}}{\partial \mathbf{U}} & \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} \\ \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{U}} & \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \boldsymbol{\alpha}} \end{bmatrix}_{\mathbf{U}, \boldsymbol{\alpha}} \begin{bmatrix} \Delta \mathbf{U} \\ \Delta \boldsymbol{\alpha} \end{bmatrix} + \begin{bmatrix} \mathbf{R} \\ \mathbf{R}^{\text{trim}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{4}$$

Imagine forming the Schur-complement system, i.e. solving for $\Delta \mathbf{U}$ from the first block row and substituting into the second block row. The resulting equation for $\Delta \boldsymbol{\alpha}$ is

$$\left[\frac{\partial \mathbf{R}^{\text{trim}}}{\partial \boldsymbol{\alpha}} - \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{U}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right)^{-1} \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} \right] \Delta \boldsymbol{\alpha} + \left[\mathbf{R}^{\text{trim}} - \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{U}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right)^{-1} \mathbf{R} \right] = \mathbf{0}. \tag{5}$$

Now, we recognize that

$$\frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{U}} = \frac{\partial \mathbf{J}^{\text{trim}}}{\partial \mathbf{U}}, \quad \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \boldsymbol{\alpha}} = \frac{\partial \mathbf{J}^{\text{trim}}}{\partial \boldsymbol{\alpha}}, \quad -\frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{U}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right)^{-1} = \boldsymbol{\Psi}^{\text{trim}} \equiv \left[\boldsymbol{\psi}_1^{\text{trim}}, \boldsymbol{\psi}_2^{\text{trim}}, \ldots, \boldsymbol{\psi}_{N_\alpha} \right]. \tag{6}$$

Substituting these expressions into Eqn. 5 we obtain

$$\frac{d\mathbf{J}^{\text{trim}}}{d\boldsymbol{\alpha}} \Delta \boldsymbol{\alpha} + \left[\mathbf{J}^{\text{trim}} - \overline{\mathbf{J}}^{\text{trim}} - \left(\boldsymbol{\Psi}^{\text{trim}} \right)^T \mathbf{R} \right] = \mathbf{0}, \tag{7}$$

where the matrix multiplying $\Delta \boldsymbol{\alpha}$ became $\frac{d\mathbf{J}^{\text{trim}}}{d\boldsymbol{\alpha}}$ according to the definition of the total derivative, e.g. as in Figure 1. With $\Delta \boldsymbol{\alpha}$ in hand, we could then go back and solve for $\Delta \mathbf{U}$. If we have already solved the flow problem, $\mathbf{R} = \mathbf{0}$, and we find that Eqn. 7 is equivalent to the update in the feedback loop in Figure 1.

In addition, in the case of linear flow equations and outputs, only one iteration would be required in the feedback loop in Figure 1 and the two approaches would be equivalent. For nonlinear equations/outputs, which is what we predominantly encounter, the coupled approach may reduce the total number of nonlinear iterations as the trimming can be performed concurrently in the context of one set of Newton outer iterations. However, this approach also requires a concurrent adjoint solve[11] – we will investigate the efficacy of such an idea in future work.

IV. Discretization

Of interest in this work are computational fluid dynamics applications, and to model the flow, we use the Spalart-Allmaras (SA) Reynolds-averaged compressible Navier-Stokes equations,

$$\partial_t \mathbf{u} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}) + \vec{\mathbf{G}}(\mathbf{u}, \nabla \mathbf{u}) + \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0}, \tag{8}$$

where

American Institute of Aeronautics and Astronautics

- $\mathbf{u} = [\rho, \rho\vec{v}, \rho E, \rho\tilde{\nu}]^T \in \mathbb{R}^s$ is the conservative state vector. Note ρ is the density, $\vec{v}$ is the spatial velocity, E is the specific total energy, $\tilde{\nu}$ is the working variable for the SA model, and s is the state rank.

- $\vec{\mathbf{F}}(\mathbf{u}) \in [\mathbb{R}^s]^d$ is the inviscid flux. Note d is the spatial dimension.

- $\vec{\mathbf{G}}(\mathbf{u}, \nabla\mathbf{u}) \in [\mathbb{R}^s]^d$ is the viscous flux.

- $\mathbf{S}(\mathbf{u}, \nabla\mathbf{u}) \in \mathbb{R}^s$ is a source term that is active when turbulent flow is modeled.

We discretize Eqn 8 using a discontinuous Galerkin (DG) finite element method, for various reasons including ease of high order, hanging-node adaptation, and hp-refinement. However, this is not restrictive as the output-based adaptive method is applicable to other discretizations. A DG discretization approximates the solution in a space $\mathcal{V}^h$ of piecewise polynomials of degree p with local support on each element resulting from a subdivision of the spatial domain.

The Riemann flux involved in the convective term is approximated with Roe's[12] solver in which the SA working variable, $\tilde{\nu}$, is transported as a conserved scalar. The diffusion term is discretized using the second form of Bassi & Rebay[13] (BR2) and the SA treatment follows Allmaras and Oliver's implementation.[14] These modifications ensure stability of the model at negative $\tilde{\nu}$ and they are specifically suited for discontinuous Galerkin discretizations.

The discrete system is obtained by expanding the state in terms of polynomial trial functions on each element and testing, in a finite-element weak sense, Eqn. 8 with functions that are the same as the trial basis functions. We denote by $\mathcal{V}_h$ the approximation/test space, and by $\mathbf{U}_h \in \mathbb{R}^N$ the corresponding discrete vector of coefficients on the basis functions. The resulting set of equations is

$$\mathbf{M}_h \frac{d\mathbf{U}_h}{dt} + \mathbf{R}_h(\mathbf{U}_h, \boldsymbol{\alpha}) = \mathbf{0}, \tag{9}$$

where the state and residual have already been defined, and $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$ is the block diagonal mass matrix that corresponds to the volume integral of basis function products on each element in the mesh. Note that the input parameters $\boldsymbol{\alpha} \in \mathbb{R}^{N_\alpha}$ can enter the equations in various ways, including as parameters describing the boundary conditions that affect the fluxes on boundary elements. Although we are interested here in steady-state solutions, we have retained the unsteady term in Eqn. 9 as this term is used in our Constrained Pseudo-Transient Continuation (CPTC) solver.[15] This method incorporates physical realizability constraints in the solution path, and thus it improves the robustness of the DG solver.

Input parameters could come in various forms, such as boundary conditions, geometry parameters, or physical constants. In the present work we will focus on angles of attack of possibly multiple components. To allow for relative angle changes between components, such as two airfoils, we employ a mesh deformation strategy based on the arbitrary Lagrangian-Eulerian formulation.[7] Specifically, we solve the problem on an undeformed reference configuration in which components are at fixed relative positions. A parameter-dependent mapping transforms this reference configuration into the "physical" configuration in which components may be rotated relative to each other via a rigid-body deformation that blends to zero in a fixed disk around the component.

V.   Output Error Estimation

Numerical errors due to insufficient mesh resolution can affect outputs, often in seemingly-subtle but significant ways. The latter point is especially relevant for the convection-dominated flows common to aerospace engineering applications. Our goals are to quantify the effect of these numerical errors and to reduce them through mesh adaptation, all in an output-constrained setting. First we briefly review the adjoint-weighted residual approach to error estimation, the details of which can be found in.[3]

V.A.   The Adjoint-Weighted Residual

As a surrogate for the true numerical error in an output, we use the difference between outputs computed on two finite dimensional spaces: a coarse approximation space ($\mathcal{V}_H \to \mathbf{U}_H \in \mathbb{R}^{N_H}$) on which we calculate the state and output, and a fine space ($\mathcal{V}_h \to \mathbf{U}_h \in \mathbb{R}^{N_h}$) relative to which we estimate the error:

$$\text{output error:} \quad \delta J \equiv J_H(\mathbf{U}_H) - J_h(\mathbf{U}_h) \tag{10}$$

In this work we construct the fine space by increasing each element's approximation order, $p \rightarrow p+1$.

We can estimate the output error without solving for the fine-space state by using an adjoint-weighted residual. From the point of view of the fine space, the coarse state injected in to the fine space, $\mathbf{U}_h^H$ will not satisfy the fine space equations. So we expect $\mathbf{R}_h(\mathbf{U}_h^H) \neq \mathbf{0}$. However, the fine-space adjoint, $\boldsymbol{\psi}_h$, as a sensitivity to residuals, tells us how to weight the fine-space residuals to obtain an output perturbation,

$$\delta J \approx J_h(\mathbf{U}_h^H) - J_h(\mathbf{U}_h) = \boldsymbol{\psi}_h^T \delta \mathbf{R}_h = -\boldsymbol{\psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H). \tag{11}$$

This derivation assumes small perturbations in $\mathbf{U}$ and $\mathbf{R}$ when the output or equations are nonlinear. Calling the left-hand side δJ assumes $J_H(\mathbf{U}_H) = J_h(\mathbf{U}_h^H)$, which is true if the output definition (e.g. geometry) does not change between the coarse and fine spaces.

In summary, we have

$$\delta J \approx -\boldsymbol{\psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H). \tag{12}$$

Note that this error estimate does not require the fine-space primal solution, $\mathbf{U}_h$. It does require a fine-space adjoint vector, which, for large cases, we approximate by injecting into the fine space a coarse-space adjoint vector, and applying $\nu_{\mathrm{smooth}} = 5$ (unless otherwise noted) iterations of an element block-Jacobi smoother.

V.B.   Output Adjoint Solution Under Trim Constraints

For our scalar output of interest J^{adapt}, the discrete adjoint equation corresponding to the coupled system in Eqn. 3 is obtained by using the transpose of the coupled system Jacobian,

$$\begin{bmatrix} \frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T & \frac{\partial \mathbf{R}^{\mathrm{trim}}}{\partial \mathbf{U}}^T \\ \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}^T & \frac{\partial \mathbf{R}^{\mathrm{trim}}}{\partial \boldsymbol{\alpha}}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi} \\ \boldsymbol{\phi} \end{bmatrix} + \begin{bmatrix} \frac{\partial J^{\mathrm{adapt}}}{\partial \mathbf{U}}^T \\ \frac{\partial J^{\mathrm{adapt}}}{\partial \boldsymbol{\alpha}}^T \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{13}$$

where $\boldsymbol{\psi} \in \mathbb{R}^N$ is the discrete adjoint associated with the flow residuals, and $\boldsymbol{\phi} \in \mathbb{R}^{N_\alpha}$ is the discrete adjoint associated with the trimming residuals. We calculate the Schur-complement system by solving for $\boldsymbol{\psi}$ from the first block row and substituting into the second block row. The resulting equation for $\boldsymbol{\phi}$ is

$$\left[\frac{d\mathbf{J}^{\mathrm{trim}}}{d\boldsymbol{\alpha}}\right]^T \boldsymbol{\phi} + \frac{dJ^{\mathrm{adapt}}}{d\boldsymbol{\alpha}}^T \quad = \quad \mathbf{0}, \tag{14}$$

where the key equalities in the derivation are the same as in Eqn. 6. Note, $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ are components of the adjoint solution to the coupled flow/trim problem. Taken individually, neither of these are adjoints to either J or J^{trim}, but we can relate $\boldsymbol{\psi}$ to $\boldsymbol{\psi}^{\mathrm{adapt}}$ and $\boldsymbol{\Psi}^{\mathrm{trim}}$. To see this, we substitute for $\boldsymbol{\phi}$ back into the first block row in Eqn. 13. The result is

$$\boldsymbol{\psi} \quad = \quad -\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^{-T}\left[\frac{\partial J^{\mathrm{adapt}}}{\partial \mathbf{U}}^T + \frac{\partial \mathbf{R}^{\mathrm{trim}}}{\partial \mathbf{U}}^T \boldsymbol{\phi}\right] \tag{15}$$

$$= \quad \underbrace{-\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^{-T}\frac{\partial J^{\mathrm{adapt}}}{\partial \mathbf{U}}^T}_{\boldsymbol{\psi}^{\mathrm{adapt}}} \underbrace{-\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^{-T}\frac{\partial \mathbf{J}^{\mathrm{trim}}}{\partial \mathbf{U}}^T}_{\boldsymbol{\Psi}^{\mathrm{trim}}} \boldsymbol{\phi} \tag{16}$$

$$= \quad \boldsymbol{\psi}^{\mathrm{adapt}} + \boldsymbol{\Psi}^{\mathrm{trim}}\boldsymbol{\phi} \tag{17}$$

Finally, we remark that the solution for $\boldsymbol{\phi}$ in Eqn. 14 is

$$\boldsymbol{\phi} = -\left[\frac{d\mathbf{J}^{\mathrm{trim}}}{d\boldsymbol{\alpha}}\right]^{-T}\frac{dJ^{\mathrm{adapt}}}{d\boldsymbol{\alpha}}^T = -\left[\frac{dJ^{\mathrm{adapt}}}{d\boldsymbol{\alpha}}\left(\frac{d\mathbf{J}^{\mathrm{trim}}}{d\boldsymbol{\alpha}}\right)^{-1}\right]^T. \tag{18}$$

V.C.   Output Error Estimation Under Trim Constraints

To estimate the output error in a solution with trim constraints, we apply the adjoint-weighted residual to the system in Eqn. 3. The adjoint equation for this system is given in Eqn. 13, and so a straightforward extension of the adjoint-weighted residual in Eqn. 12 gives,

$$\delta J \approx -\boldsymbol{\psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H, \boldsymbol{\alpha}_H) - \boldsymbol{\phi}_h^T \mathbf{R}_h^{\mathrm{trim}}(\mathbf{U}_h^H, \boldsymbol{\alpha}_H). \tag{19}$$

Note, since the number of inputs does not change from the coarse to the fine space, there is no injection operator for $\boldsymbol{\alpha}_H$. Using the definition of $\mathbf{R}_h^{\text{trim}}$, the residual in the second term of the error estimate is

$$
\begin{aligned}
\mathbf{R}_h^{\text{trim}}(\mathbf{U}_h^H, \boldsymbol{\alpha}_H) &= \mathbf{J}_h^{\text{trim}}(\mathbf{U}_h^H, \boldsymbol{\alpha}_H) - \overline{\mathbf{J}}_h^{\text{trim}} \\
&= \underbrace{\mathbf{J}_H^{\text{trim}}(\mathbf{U}_H, \boldsymbol{\alpha}_H) - \overline{\mathbf{J}}_h^{\text{trim}}}_{=0} + \mathbf{J}_h^{\text{trim}}(\mathbf{U}_h^H, \boldsymbol{\alpha}_H) - \mathbf{J}_H^{\text{trim}}(\mathbf{U}_H, \boldsymbol{\alpha}_H).
\end{aligned} \tag{20}
$$

This term could be nonzero, i.e. $\mathbf{J}_h^{\text{trim}}(\mathbf{U}_h^H, \boldsymbol{\alpha}_H) \neq \mathbf{J}_H^{\text{trim}}(\mathbf{U}_H, \boldsymbol{\alpha}_H)$, if, for example, the geometry changes from the coarse to the fine space or if there are errors between different quadrature rules. However, we assume that these effects are negligible, as in the non-trimmed case. Substituting for $\boldsymbol{\psi}$ using Eqn. 17, the remaining first term in the error estimate yields

$$
\delta J \approx -\left[\boldsymbol{\psi}_h^{\text{adapt}} + \boldsymbol{\psi}_h^{\text{trim}} \boldsymbol{\phi} \right]^T \mathbf{R}_h(\mathbf{U}_h^H, \boldsymbol{\alpha}_H). \tag{21}
$$

Defining $\delta J^{\text{adapt}} \equiv -(\boldsymbol{\psi}_h^{\text{adapt}})^T \mathbf{R}_h(\mathbf{U}_h^H, \boldsymbol{\alpha}_H)$ (a scalar) and $\delta \mathbf{J}^{\text{trim}} \equiv -(\boldsymbol{\Psi}_h^{\text{trim}})^T \mathbf{R}_h(\mathbf{U}_h^H, \boldsymbol{\alpha}_H)$ (a vector of N_α values), we obtain our final form for the error estimate,

$$
\delta J \approx \delta J^{\text{adapt}} + \boldsymbol{\phi}^T \delta \mathbf{J}^{\text{trim}}. \tag{22}
$$

Recall that $\boldsymbol{\phi} \in \mathbb{R}^{N_\alpha}$, given via Eqn. 18, resembles a ratio of parameter sensitivities of the adaptation output to the trim outputs. Its role in the error estimate is to convert numerical errors in the trim output calculations into errors in the adaptation output. For example, in a drag (D) adaptation at fixed lift (L) and trimmed angle of attack (α), $\boldsymbol{\phi}$ would just be a scalar and it would be equal to $\frac{dD}{d\alpha} / \frac{dL}{d\alpha}$.

VI.   Mesh Adaptation

The localization of the output error estimate in Eqn. 22 to individual elements is simple in a DG discretization: the residual vector inside each element dotted with the discrete adjoint vector in that element gives the contribution of that element to the error estimate. The absolute value of this contribution can be taken as the error indicator for the element, and in our work we take one further conservative measure by grouping the residuals according to their state equation and then summing absolute values of each weighted residual.

When trim outputs are involved, we have another decision to make: whether we would allow cancellation between the adaptive and trim output error estimates. In this work we do not let such cancellation happen when calculating the error indicator, so that the final error indicator on element k is given by

$$
\eta_k = \eta_k^{\text{adapt}} + |\boldsymbol{\phi}|^T \boldsymbol{\eta}_k^{\text{trim}}, \tag{23}
$$

where η_k^{adapt} is the adaptation output error indicator, and $\boldsymbol{\eta}_k^{\text{trim}} \in \mathbb{R}^{N_\alpha}$ is a vector of trim output error indicators. Note, $|\boldsymbol{\phi}|$ is the entry-wise absolute value of $\boldsymbol{\phi}$. This error indicator then drives a fixed-fraction, isotropic, hanging-node, mesh refinement strategy. Anisotropic and hp-refinement[6] options are also possible.

VII.   Results

VII.A.   Supersonic Biplane

As a first, somewhat contrived, demonstration of the importance of incorporating output constraint information into adaptation, we consider a supersonic biplane. Specifically, we look at two half-diamond airfoils in $M = 1.5$ supersonic flow, as illustrated in Figure 2. The airfoils are separated sufficiently in the vertical direction such that they do not affect each other in this supersonic flow. For the trim constraint, we require that the airfoils produce a combined lift coefficient of $J^{\text{trim}} \equiv c_\ell = L_{\text{total}}/(0.5\rho V_\infty^2 c) = 0.25$. For the adapted output we care about the drag coefficient on the lower airfoil, $J^{\text{adapt}} \equiv c_{d,\text{lower}}$. As the trimming parameter, we take the angle of attack of the biplane, α, which is set through the ALE mesh motion procedure described in Section IV.

We now compare two methods for output-based adaptation. The first method is the one proposed in this work, in which the adaptive indicator includes the contribution of the trim output, as given in Eqn. 23. The
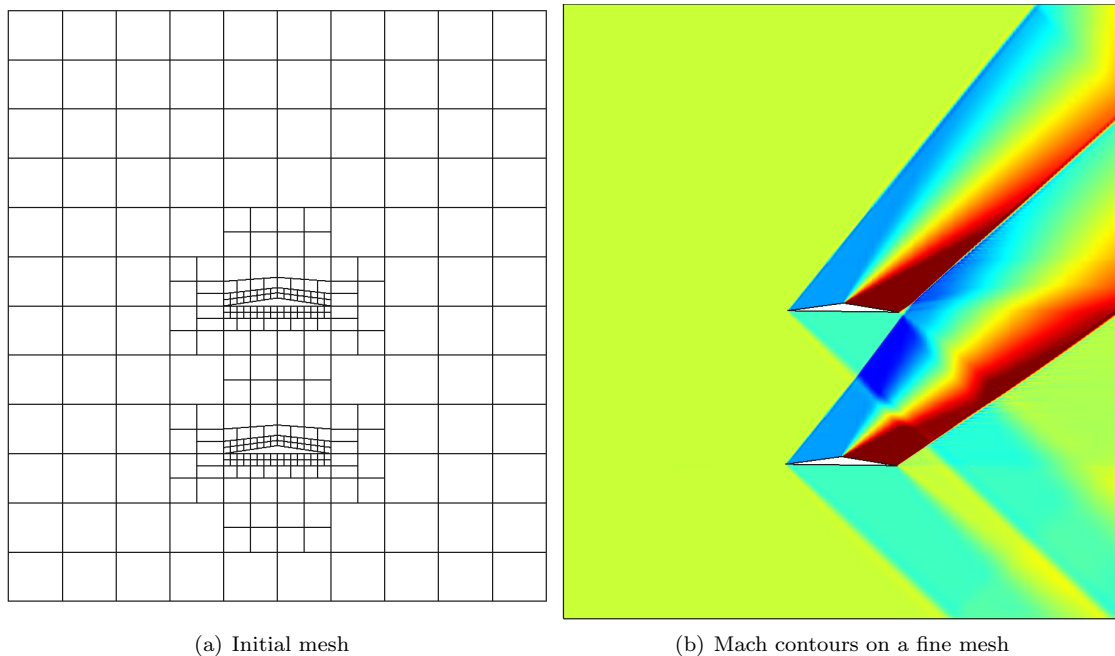
(a) Initial mesh

(b) Mach contours on a fine mesh

Figure 2. Supersonic biplane: initial mesh and Mach number contours for a lift-constrained drag-adaptive simulation.

second method, which we call "unconstrained adaptation", uses only the adaptive indicator derived from the adapted output, *i.e.* only η_k^{adapt}. The runs are still trimmed at every adaptive iteration to meet the trim output constraint, but the adaptation does not incorporate the error in the trim output.

We do not expect the unconstrained strategy to perform very well for this problem because the mesh around the top airfoil will not be adapted (those elements do not affect the lower airfoil drag), which means that the total lift trim constraint output will always be polluted by some numerical error. Indeed, Figure 3 shows that this is the case: both the lower airfoil drag coefficient and the angle of attack converge to different values with the two methods. Figure 4 illustrates why the unconstrained adaptation produces

(a) Drag coefficient convergence

(b) Angle of attack convergence
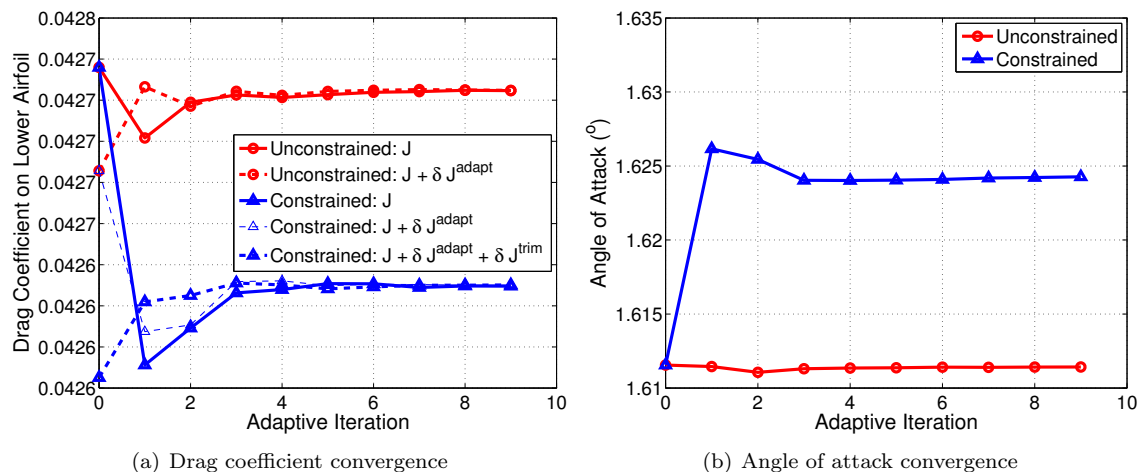
Figure 3. Supersonic biplane: convergence of the adapted output, the lower-airfoil drag coefficient, and the trimming parameter, the angle of attack, for two adaptive methods: the one proposed in this work versus an unconstrained version that only targets the error in the adapted output.

incorrect results. As expected, without the contribution of the total lift trim output to the error indicator, elements near the upper airfoil are never targeted for refinement. The result is an incorrect value for the angle of attack trim parameter and the resulting lower airfoil drag coefficient.

American Institute of Aeronautics and Astronautics

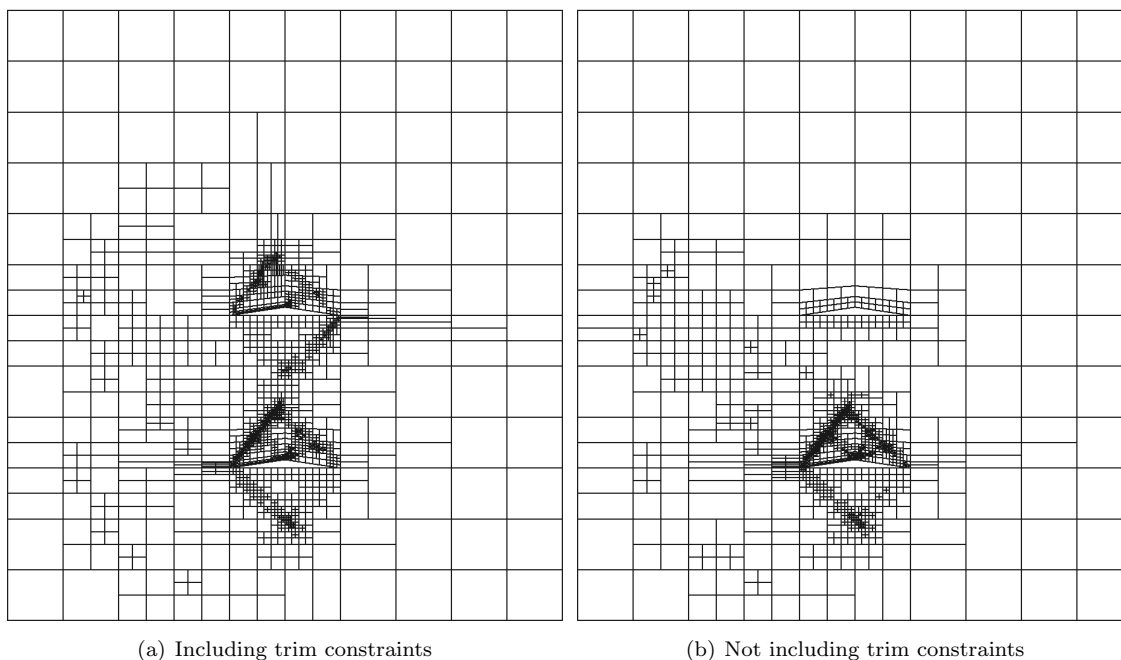(a) Including trim constraints   (b) Not including trim constraints

Figure 4. Supersonic biplane: meshes adapted using the two tested indicators. Including the trim output contribution in the adaptive indicator causes the upper airfoil to be adapted because those elements affect the total lift which then indirectly affects the lower airfoil drag through the trim parameter, α. Note that in both cases, the shocks are only resolved to the minimum extent necessary to predict forces on the airfoils.

Because of the disjoint relationship between the outputs used for trimming and adaptation in this case, the results represent an extreme scenario in which the adaptation converges to different values. In the cases considered next, the outputs will not be perfectly disjoint, so that the results eventually converge to the same value as adaptation progresses. However, as we will see, including the trim output contribution to the error and the indicator accelerates convergence to the true output at the trimmed conditions.

VII.B.   Laminar NACA 0012

The second problem we consider is drag prediction of the NACA 0012 airfoil at fixed lift, for which we use two adaptation strategies. One includes only the drag error to drive the adaptation and the other uses the combination of the drag error with the lift-constraint error according to Eqns. 22 and 23. In both cases, we use $p = 1$ approximation order and we trim the angle of attack to yield the desired lift.

In order to properly compare the two strategies we establish a reference *exact* solution. This reference is computed via 10 successive lift and drag adaptation steps with $p = 2$ approximation order at a fixed angle of attack. Table 1 summarizes the reference flow conditions and output values.

Flow Parameter	Value
Reynolds Number	5000
Mach Number	0.5
Angle of Attack	1°
Lift coefficient	0.01825032256
Drag coefficient	0.05533250

Table 1. Reference values for NACA 0012 case.

The initial grid for both the reference calculation and trimming simulations is composed of 374 quadrilateral elements arranged in a C-grid pattern, with the boundary 50 chord-lengths away from the airfoil, as shown in Figure 5. This initial grid is intentionally coarse to allow many levels of refinement with reasonable

American Institute of Aeronautics and Astronautics

computational resources.

Similarly to the previous case, we use mesh deformation for trimming. Since the objective here is to compare the adaptive strategies against a reference solution on an undeformed mesh, we set the free stream boundary condition at an angle of attack of $1°$. Hence, as the adapted solutions approach the reference solution, the mesh deformation vanishes.
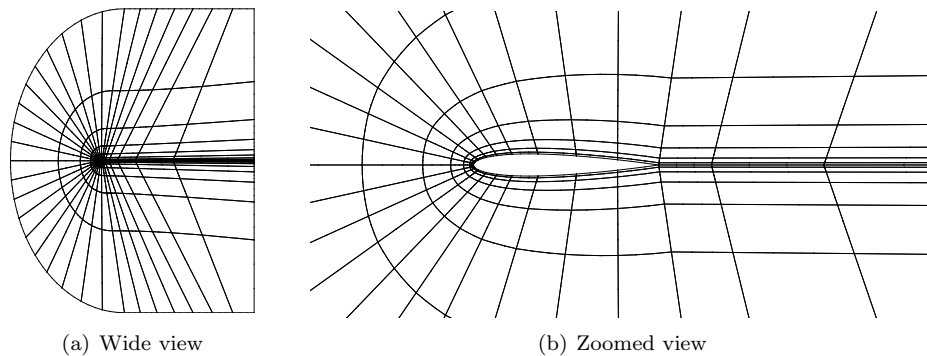

(a) Wide view

(b) Zoomed view

Figure 5. Laminar NACA 0012 case: initial grid consisting of 374 cubic-geometry quadrilaterals with a farfield distance of approximately 50 chords.

At each adaptive step of both drag prediction strategies, we trim the angle of attack such that the lift coefficient is within 2×10^{-7} of the reference value in Table 1. Figure 6 shows the evolution of angle of attack for both adaptation strategies. Note that the angle of attack converges more quickly with the additional error due to the lift constraint included in the error estimate. The error in the trimming parameter exhibits an oscillatory behavior for both strategies. Towards the end of the adaptation process, the unconstrained adaptation method oscillates around an error roughly 5 times larger than the value for the constrained strategy. This is related to the accuracy of the lift adjoint on the adapted meshes. Figure 7 compares the conservation of mass component of the drag and lift adjoints for both strategies. Note that the adjoint contours for drag are virtually the same for both strategies as they include drag error in the adaptive indicator. The difference between the strategies is clear in the lift adjoint contours for which the unconstrained strategy shows poorer approximation in the trailing edge region than its counterpart that includes lift error in the adaptive indicator.

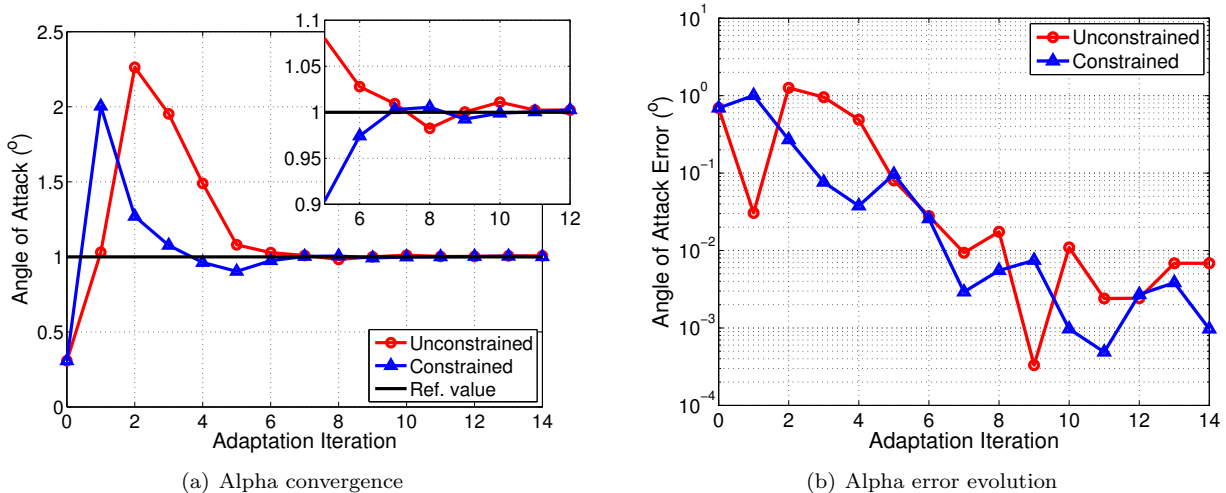(a) Alpha convergence

(b) Alpha error evolution

Figure 6. Laminar NACA 0012 case: angle of attack convergence for the two drag prediction strategies.

Figure 8 shows the drag convergence histories for both strategies. Note that including the lift-constraint error in the drag error estimate provides a significantly better output correction. Two main aspects are important here. One is the fact that including constraint error in the adaptive indicator improves the

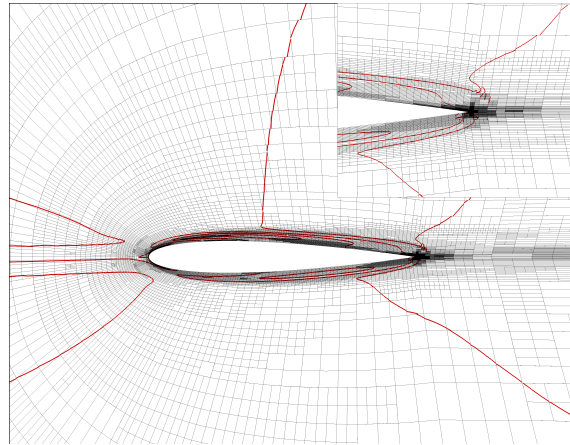American Institute of Aeronautics and Astronautics
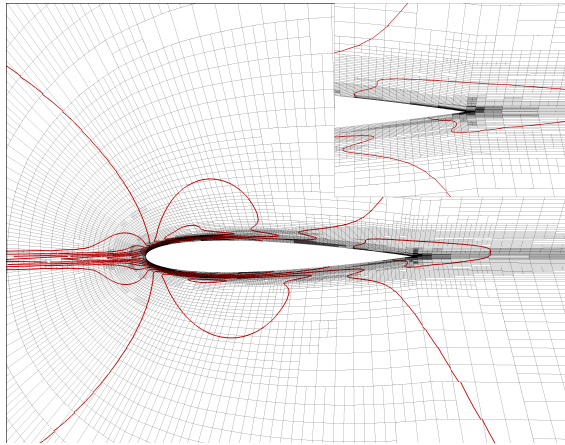
(a) Conservation of mass component of drag adjoint with unconstrained adaptation.
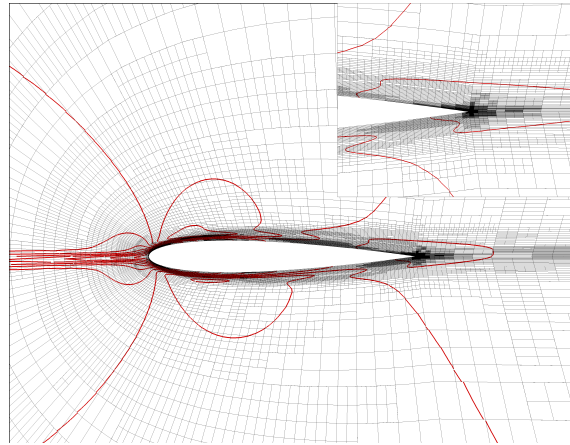
(b) Conservation of mass component of drag adjoint with constrained adaptation.
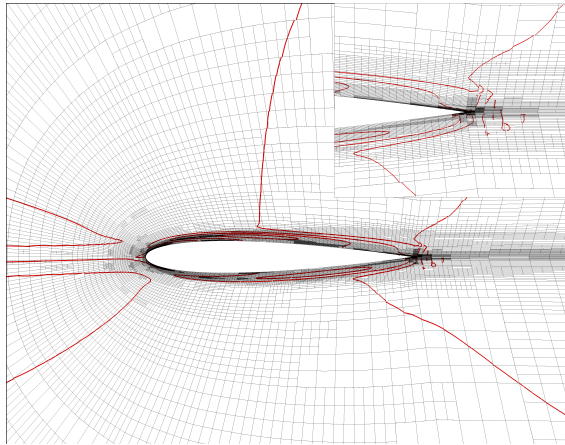
(c) Conservation of mass component of lift adjoint with unconstrained adaptation.

(d) Conservation of mass component of lift adjoint with constrained adaptation.

Figure 7. Laminar NACA 0012 case: drag and lift adjoints on the final adapted meshes for both strategies.

American Institute of Aeronautics and Astronautics

convergence of the trimming parameter which, in turn, improves the quality of the error estimate for J^{adapt}. This is supported by the lighter blue dashed drag convergence line in Figure 8(a), which corresponds to correcting the drag output solely by the first term in Eqn. 22. The darker blue dashed line in Figure 8(a) is the drag output corrected by both terms in Eqn. 22. The second effect is that the error estimate for J^{trim} improves as we adapt the mesh because the fine-space adjoint for the trimming output becomes better approximated.

(a) Drag convergence
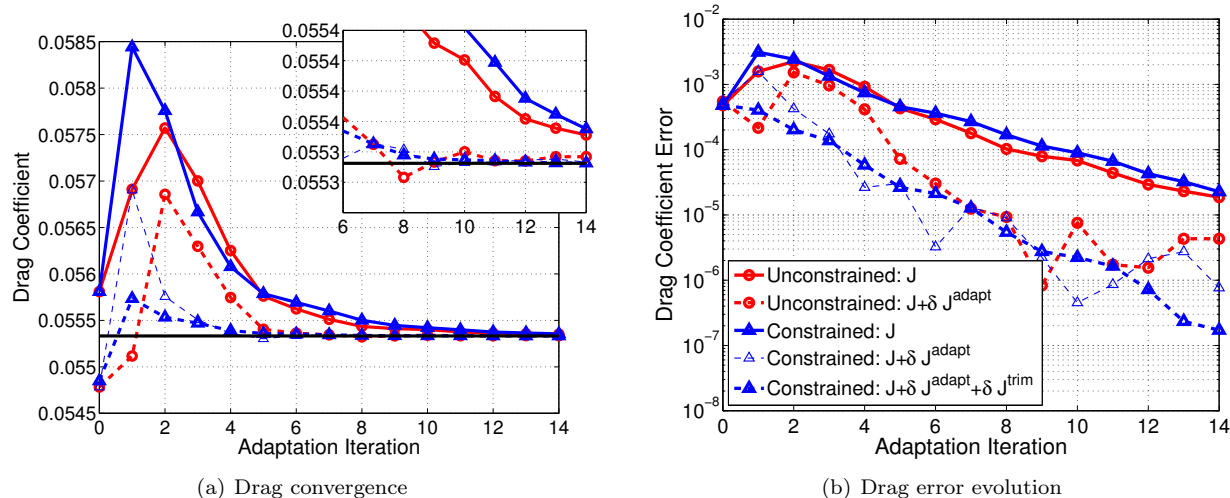

(b) Drag error evolution

Figure 8. Laminar NACA 0012 case: drag convergence for both adaptive strategies.

Another notable contrast between both strategies is shown in the zoomed box in Figure 8(a): δJ^{adapt} for the unconstrained strategy does not provide the full output correction. This is reflected in Figure 8(b) where the error for the corrected output obtained by the unconstrained strategy (red dashed line) stagnates at $\sim 10^{-6}$ after the 8th adaptation step. Conversely, the fully-corrected drag for the constrained strategy (thick dashed blue line) continues to converge past that level to $\sim 10^{-7}$.

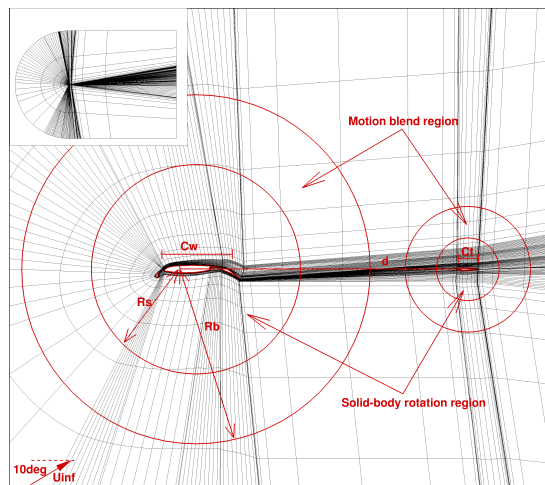## VII.C. High-lift tandem MDA 30P-30N and NACA 0012

This case consists of a high-lift configuration of the MDA 30P-30N multi-element airfoil in tandem with a NACA 0012 airfoil, which simulates a horizontal tail. Figure 9(a) shows the arrangement of the two airfoils and the initial mesh composed of 8195 cubic elements generated via agglomeration of linear quadrilaterals. Table 2 lists the parameter values for this case. Turbulence is simulated by the Spalart-Allmaras model with the modifications specified in Ref. 16 and the CPTC method solves the nonlinear system resultant from a $p = 1$ discretization of the flow equations used in this case.

We are interested in accurately computing the drag coefficient of both airfoils, $J^{\text{adapt}} \equiv c_{d,\text{total}}$, under the conditions of a prescribed total lift and static stability of the configuration about the main wing's quarter-chord point, i.e.:

$$\mathbf{J}^{\text{trim}} = \left[\begin{array}{c} c_\ell \equiv \dfrac{L_{\text{total}}}{0.5\rho V_\infty^2 C_w} = 3.0 \pm 3 \times 10^{-6} \\[2mm] c_m \equiv \dfrac{M_{\text{total},0.25C_w}}{0.5\rho V_\infty^2 C_w^2} = 0.0 \pm 6 \times 10^{-6} \end{array} \right]. \tag{24}$$

The trimming parameters are respectively the angles of attack of the wing and tail. These parameters control the solid-body rotation angles of the inner circles in Figure 9(a). The region between the inner and outer circles smoothly blends the inner, rotated region with the undeformed portion of the mesh outside the outer circles via a quintic polynomial in the radial coordinate. We apply this mesh deformation via the ALE mapping described in Section IV. In order to reduce the amount of mesh deformation required to satisfy the trimming constraints, we specify the free stream at 10° with respect to the undeformed domain.

Figure 9(b) shows the flow and trimming residuals for the first solve. Each flow residual drop in that figure corresponds to one flow solve with an updated trimming parameter. To improve the robustness of the

American Institute of Aeronautics and Astronautics

(a) Arrangement of MDA 30P-30N and NACA 0012 airfoils in tandem. Dimensions are given in Table 2.
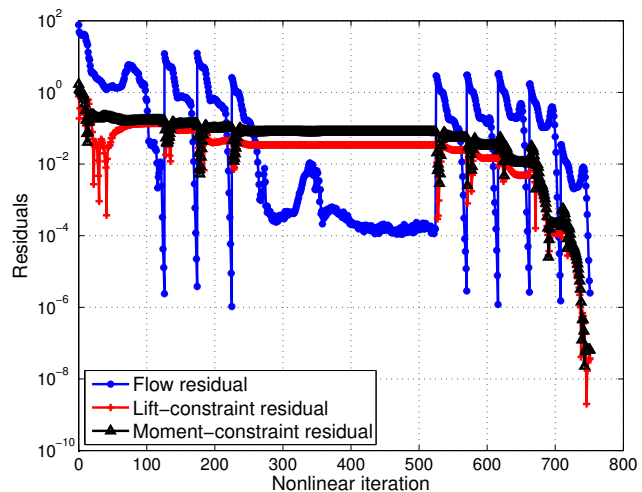
(b) Residuals for the initial flow and constraint solves.

Figure 9. Tandem MDA 30P-30N–NACA 0012 case: initial mesh and residual histories.

Parameter	Description	Value
C_w	Wing chord	0.5588
C_t	Tail chord	$0.3 \cdot C_w$
Wing R_s	Solid body rotation radius for wing	$1.5 \cdot C_w$
Wing R_b	Blend region radius for wing	$2.5 \cdot C_w$
Tail R_s	Solid body rotation radius for wing	$1.5 \cdot C_t$
Tail R_b	Blend region radius for wing	$3 \cdot C_t$
d	Distance between quarter-chord points	$4 \cdot C_w$
M_∞	Free stream Mach number	0.2
Re_{C_w}	Reynolds number based on C_w	9×10^6

Table 2. Dimensions and flow conditions for the tandem MDA 30P-30N and NACA 0012 case.

trimming process, we limit the updates in the wing and tail angles to $|\Delta \alpha_i| \leq 1.5°$. Note that the fourth flow solve stalls with the residual at $\sim 10^{-4}$. However, the adjoint solves for $\mathbf{J}^{\text{trim}}$ at that stage still provide accurate trimming sensitivities, thus not disrupting the trimming process.

As in the previous cases, we consider both the constrained and unconstrained adaptation methods and at each adaptive step we iterate until the trimming constraints are satisfied. Figure 10(a) shows the drag convergence of both adaptation strategies. As in an engineering application, we do not know the exact drag value for this problem, hence we examine how quickly the corrected output from each strategy converges to an output value. The main benefit of the constrained adaptation strategy in this case is that it provides a corrected drag value that is less oscillatory than the unconstrained method. This comes from the fact that including the δJ^{trim} in the adaptive indicator provides a smooth convergence for the trimming parameters (Figure 10(b)).

Figure 11 shows the final adapted meshes for both strategies. They do not exhibit notable differences as both strategies converge to virtually the same drag and trimming parameters values. The small mesh differences are due to the different histories for the trimming parameters. Note the rings of adapted mesh in the blend region for the mesh deformation. This is because the ALE formulation introduces small errors that both adaptation strategies target.
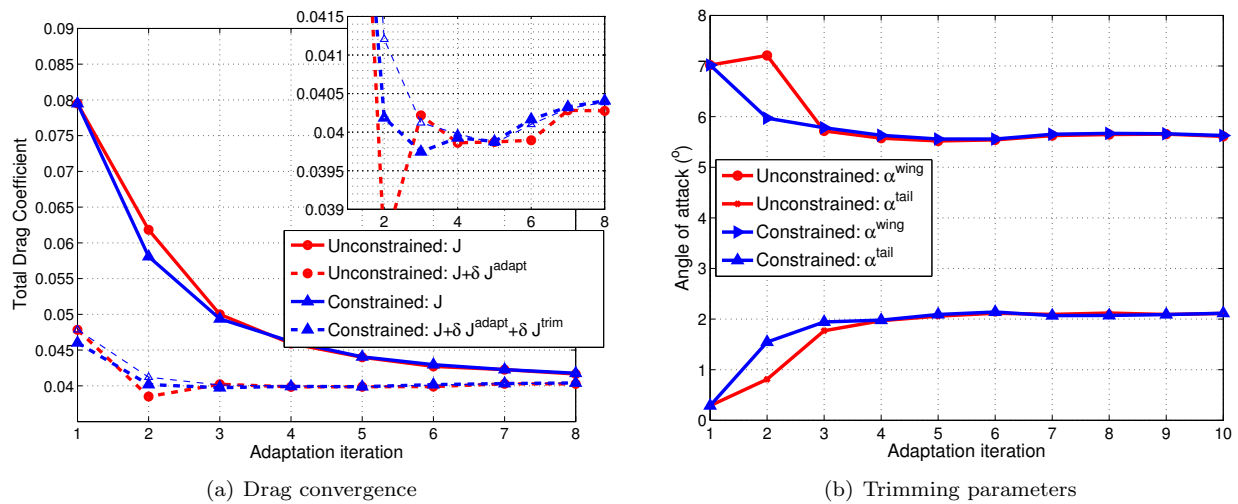
(a) Drag convergence

(b) Trimming parameters

Figure 10. Tandem MDA 30P-30N–NACA 0012 case: drag convergence for both adaptive strategies. Solid lines: raw drag output; dashed lines: drag output corrected by error estimate. The lighter blue dashed line includes only the direct drag error estimate.
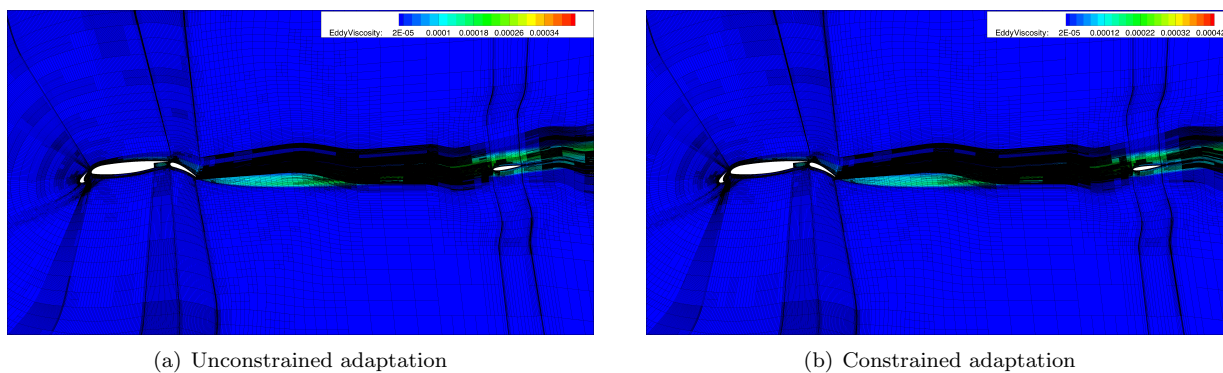

(a) Unconstrained adaptation

(b) Constrained adaptation

Figure 11. Tandem MDA 30P-30N–NACA 0012 case: final adapted meshes for both adaptation strategies with turbulent viscosity contours.

American Institute of Aeronautics and Astronautics

VIII.   Conclusions

We extended the adjoint-weighted residual error estimation method to problems with output constraints. We demonstrated this extension in several problems of varying complexity: a drag prediction problem under a fixed-lift constraint for both "academic" supersonic biplane and airfoil configurations, as well as a drag prediction problem under fixed lift and moment constraints for a practical high-lift configuration. Even when there are significant overlaps between the outputs, including the trim output contribution to the error and indicator accelerates convergence. In more extreme cases with disjoint outputs, we showed that not including the trim output in the adaptive indicator makes the adaptive process converge to a different solution which still contains trimming errors. We observed that the main benefit of including the constraint error in the adaptive indicator is in the convergence of the trimming parameters. In future work, we plan to pursue ways of reducing the computational cost of the constraint solver.

Acknowledgments

References

[1]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.

[2]Nemec, M. and Aftosmis, M. J., "Error Estimation and Adpative Refinement for Embedded-Boundary Cartesian Meshes," AIAA Paper 2007-4187, 2007.

[3]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.

[4]Hartmann, R., Held, J., and Leicht, T., "Adjoint-based error estimation and adaptive mesh refinement for the RANS and $k - \omega$ turbulence model equations," *Journal of Computational Physics*, Vol. 230, No. 11, 2011, pp. 4268 – 4284.

[5]Yano, M., Modisette, J., and Darmofal, D., "The Importance of mesh adaptation for higher-order discretizations of aerodynamics flows," AIAA Paper 2011-3852, 2011.

[6]Ceze, M. A. and Fidkowski, K. J., "An anisotropic hp-adaptation framework for functional prediction," *American Institute of Aeronautics and Astronautics Journal*, Vol. 51, 2013, pp. 492–509.

[7]Kast, S. M. and Fidkowski, K. J., "Output-based Mesh Adaptation for High Order Navier-Stokes Simulations on Deformable Domains," *Journal of Computational Physics*, Vol. 252, No. 0, 2013, pp. 468–494.

[8]Nemec, M., Aftosmis, M. J., and Wintzer, M., "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," AIAA Paper 2008-0725, 2008.

[9]Leicht, T. and Hartmann, R., "Multitarget Error Estimation and Adaptivity in Aerodynamic Flow Simulations," *International Journal for Numerical Methods in Fluids*, Vol. 56, 2008, pp. 2111–2138.

[10]Nemec, M. and Aftosmis, M. J., "Output Error Estimates and Mesh Refinement in Aerodynamic Shape Optimization," AIAA Paper 2013-0865, 2013.

[11]Lu, J. and Darmofal, D. L., "A Quasi-Minimal Residual Method for Simultaneous Primal-Dual Solutions and Superconvergent Functional Estimates," *SIAM Journal on Scientific Computing*, Vol. 24, No. 5, 2003, pp. 1693–1709.

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

[13]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.

[14]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.

[15]Ceze, M., *A Robust hp-Adaptation Method for Discontinuous Galerkin Discretizations Applied to Aerodynamic Flows*, Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, 2013.

[16]Allmaras, S. R., Johnson, F. T., and Spalart, P. R., "Modifications and Clarifications for the Implementation of the Spalart-AllmarasTurbulence Model," *Seventh Intenational Conference on Computational Fluid Dynamics (ICCFD7)*, 2012.

American Institute of Aeronautics and Astronautics