51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
07 - 10 January 2013, Grapevine (Dallas/Ft. Worth Region), Texas

AIAA 2013-0051

Drag Prediction Using Adaptive Discontinuous Finite Elements

Marco Ceze* and Krzysztof J. Fidkowski†

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

This paper presents the results obtained with the XFlow solver for the Fifth Drag Prediction Workshop. The discontinuous Galerkin finite element method is used for the spatial discretization of the Reynolds-Averaged Navier-Stokes (RANS) equations with a modified version of the Spalart-Allmaras (SA) turbulence model. Drag convergence is sought via mesh adaptation driven by an adjoint-weighted residual method. We present results for the drag polar of the NACA 0012 airfoil under subsonic flow conditions and for the Common Research Model (CRM) wing-body geometry under transonic flow conditions and fixed lift. The angle-of-attack that yields the desired lift is obtained via a Newton solve and a lift adjoint. We discuss how this lift constraint adds an additional term to the drag error estimate.

I. Introduction

The presence of Computational Fluid Dynamics (CFD) tools in the engineering environment has steadily increased in the past few decades. With the evolution of algorithms and the substantial enhancement of computational power, CFD tools now provide the ability to explore new configurations and test flow conditions that may be otherwise difficult to produce experimentally. As the range of applications becomes wider and the number of simulations increases, requirements of high-accuracy and robustness present challenges for the CFD development community.[1]

One application of CFD that demands high-accuracy is drag prediction of a large transport aircraft. The reason being that seemingly small variations in drag significantly impact the aircraft's payload.[2,3] Aerodynamic flow over an aircraft, as for many other cases, exhibits features with unknown spatial distribution, and the range of the features' length scales can easily span six orders of magnitude. Furthermore, flows can exhibit singularities that pose additional challenges for the prediction of drag and other outputs. The trivial solution to these problems is to globally refine the mesh. However, this strategy is generally inefficient due to very large grid sizes required to appropriately resolve the relevant flow features and to accurately predict the scalars of engineering interest.

The American Institute of Aeronautics and Astronautics (AIAA) organizes drag and lift prediction workshops (DPW and HLPW) with the purpose of assessing the capability of state-of-the-art computational methods and turbulence modeling for predicting forces and moments on relevant geometries in the aeronautical industry. In these workshops, starting meshes are generated based on industry's best practices and mesh independence is generally sought via uniform refinement studies. Nevertheless, the spread of results can be significant.[4–7]

Mesh adaptation methods present an attractive alternative for accurate calculations on affordable grid sizes. These methods rely on the definition of an adaptive indicator which localizes the regions of the computational domain that need mesh modification through refinement, coarsening, or node movement. An effective indicator is obtained through adjoint-based error estimation methods, which have already been demonstrated for many complex problems, including those in aerospace applications.[8] The goal of these methods is to provide confidence measures in the form of error bars for scalar outputs of engineering interest. In addition, one can use the error contributions of different elements or volumes of the computational mesh as an adaptive indicator that specifically targets errors in the outputs of interest.[9–14]

[*] PhD Candidate, AIAA Member

[†] Assistant Professor, AIAA Member

Another important aspect of efficient output prediction is the ability to produce high-order approximations of the solution where smooth features – both in the primal and in the adjoint solutions – are present. One high-order method that gained popularity in aerodynamic applications is the discontinuous Galerkin (DG) method. Arguably, its popularity is due to its suitability to both unstructured meshes and to convection-dominated problems. In addition, DG's finite element formulation naturally embeds both output error estimation and the ability to handle meshes with hanging nodes. It is precisely these characteristics that this work builds upon.

The structure of this paper is as follows. Section II describes the flow solver and the discretization method. Section III outlines the output error estimation process that drives the mesh adaptation described in Section IV. Results are presented in Section V and conclusions and future work are discussed in Section VI.

II. Solver Description

The XFlow code is a high-order discontinuous Galerkin (DG) finite element solver for general equation sets with MPI-based, distributed-memory parallel capabilities. XFlow serves as a platform for development in research areas such as error estimation, mesh adaptation, and solver algorithms.[15–17] In the case of this work, the equations are the Reynolds-Averaged Navier-Stokes (RANS) equations with a modified version of the Spalart-Allmaras (SA) turbulence model.

The RANS-SA equations are written in their compact, conservative form as

$$\partial_t \mathrm{u}_s + \partial_i \mathcal{C}_{is}(\mathbf{u}) - \partial_i \mathcal{D}_{is}(\mathbf{u}) = \mathcal{S}_s(\mathbf{u}), \tag{1}$$

where $\mathcal{C}_{is}$ and $\mathcal{D}_{is}$ are the convective and diffusive fluxes respectively, $\mathcal{S}_s$ is the SA source term, $i \in [1, .., \mathrm{dim}]$ indexes the spatial dimensions, and s indexes the equations of conservation of mass, momentum, energy, and turbulent viscosity. Accordingly, the state vector is denoted by $\mathbf{u} = [\rho, \rho v_i, \rho E, \rho\tilde{\nu}]^T$, where ρ is the density, v_i are the spatial components of the velocity, E is the specific total energy, and $\tilde{\nu}$ is the working variable for the SA model.

The discontinuous Galerkin (DG) spatial discretization of the flow equations approximates the solution in a space $\mathcal{V}^{H,p}$ of piecewise polynomials of degree p with local support on each element $\kappa^H \in T^H$, where T^H is the set of elements resulting from a subdivision of the spatial domain. The resulting weak form reads:

$$\partial_t(\mathbf{u}^{H,p}, \mathbf{w}^{H,p}) + \mathbb{R}(\mathbf{u}^{H,p}, \mathbf{w}^{H,p}) = 0 \qquad \mathbf{w}^{H,p} \in \mathcal{V}^{H,p}, \tag{2}$$

where $(\cdot, \cdot)$ denotes an inner product and $\mathbb{R}(\mathbf{u}^{H,p}, \mathbf{w}^{H,p})$ is a weighted residual statement that includes source, convective, and diffusive terms.

The Riemann flux involved in the convective term is approximated with Roe's[18] 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[19] (BR2) and the SA source term is discretized according to Allmaras and Oliver's[20] modifications to the original SA model.[21] 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 components the state $\mathbf{u}^{H,p}$ and the weight functions $\mathbf{w}^{H,p}$ in terms of the basis functions $\phi^{H,p}(\mathbf{x}) \in \mathcal{V}^{H,p}$. The result has the form:

$$\mathbf{M}\frac{d\mathbf{U}}{dt} = -\mathbf{R}(\mathbf{U}), \tag{3}$$

where $\mathbf{U}$ is the discrete state, $\mathbf{R}$ is the discrete residual operator and $\mathbf{M}$ is the block diagonal mass matrix that corresponds to the volume integral of basis function products on each element in the mesh.

We use the Constrained Pseudo-Transient Continuation (CPTC)[22] method for marching Eqn. 3 in time. This method incorporates physical realizability constraints in the solution path, and thus it improves the robustness of the DG solver.

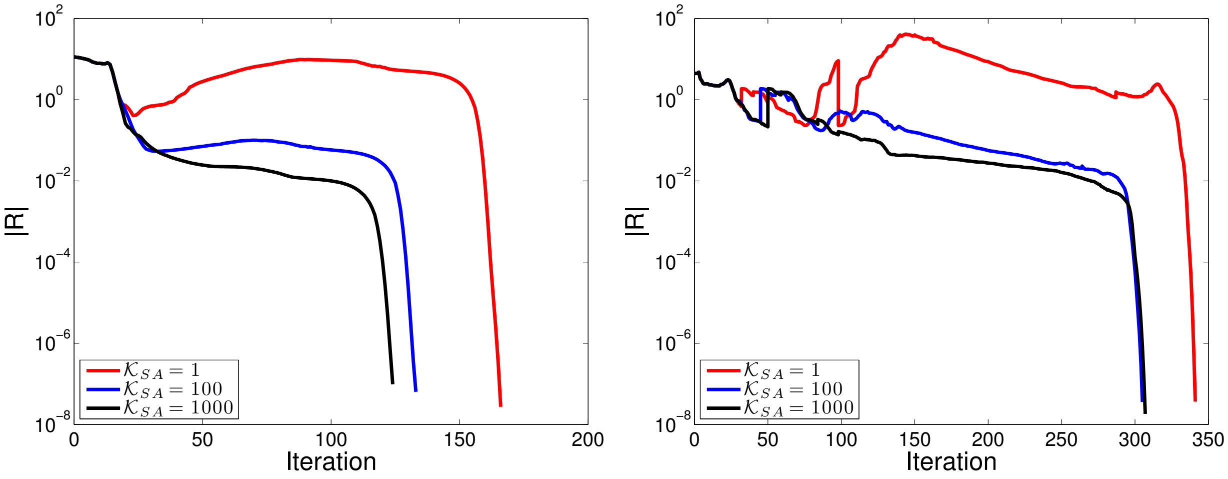
A. Scaling of the SA Discrete Equation

Most practical cases in the aeronautical industry are in the Reynolds number regime of $10^6 \to 10^7$. In this regime, the SA working variable, $\tilde{\nu}$, typically spans several orders of magnitude. Therefore, it is also desirable to choose an appropriate scale for $\tilde{\nu}$. The scale used in this work is

$$(\rho\tilde{\nu})' = \frac{\rho\tilde{\nu}}{\kappa_{\mathrm{SA}}\mu_\infty}, \tag{4}$$

where $(\rho\tilde{\nu})'$ is the scaled conserved variable that is stored and evolved by the solver, κ_{SA} is a user-prescribed scaling factor, and μ_∞ is the freestream dynamic viscosity. Essentially, we are non-dimensionalizing $\rho\tilde{\nu}$ by a factor larger than the physical viscosity.

To exemplify the effect of κ_{SA}, we show in Figure 1 the residual history for two flows at $Re = 6.5 \times 10^6$, one subsonic and one transonic. For each case, three scaling factors are used, $\kappa_{\mathrm{SA}} = 1,\ 100,\ 1000$. Note that κ_{SA} significantly affects the convergence history. Specifically, the larger values of κ_{SA} ameliorate the secondary transient observed in RANS computations using DG.[23]



(a) RAE2822 - $M_\infty = 0.3$, $Re = 6.5 \times 10^6$, $\alpha = 2.31°$.

(b) RAE2822 - $M_\infty = 0.734$, $Re = 6.5 \times 10^6$, $\alpha = 2.79°$.

Figure 1. Residual convergence using $p = 1$ for different $\tilde{\nu}$ scaling factors (κ_{SA}).

The drag and lift coefficients (C_D and C_L respectively) for both flow conditions are shown in Table 1 and Table 2. As expected, the scaling factor has virtually no effect on the results. However, it makes the conserved variables closer in magnitude which, in turn, helps implicit time integration methods.

Table 1. RAE2822 - $M_\infty = 0.3,\ Re = 6.5 \times 10^6,\ \alpha = 2.31° -$ Comparison of force coefficients and maximum values of x-momentum and SA working variable for different scaling factors.

Quantity	$\kappa_{\mathrm{SA}} = 1$	$\kappa_{\mathrm{SA}} = 100$	$\kappa_{\mathrm{SA}} = 1000$
C_D	0.0122	0.0122	0.0122
C_L	0.4507	0.4507	0.4506
$(\rho v_x)_{\max}$	1.25182	1.25182	1.25192
$(\rho\tilde{\nu})'_{\max}$	1.03775×10^3	1.03775×10^1	1.03783

Table 2. RAE2822 - $M_\infty = 0.734,\ Re = 6.5 \times 10^6,\ \alpha = 2.79° -$ Comparison of force coefficients and maximum values of x-momentum and SA working variable for different scaling factors.

Quantity	$\kappa_{\mathrm{SA}} = 1$	$\kappa_{\mathrm{SA}} = 100$	$\kappa_{\mathrm{SA}} = 1000$
C_D	0.0198	0.0198	0.0198
C_L	0.7334	0.7334	0.7334
$(\rho v_x)_{\max}$	1.11808	1.11808	1.11811
$(\rho\tilde{\nu})'_{\max}$	1.64361×10^3	1.64362×10^1	1.64378

B. C_L Driver

Frequently in the aeronautical industry, CFD simulations are conducted under trimmed conditions, meaning, under fixed, user-defined values of certain outputs – typically lift or pitching moment. This means that certain boundary condition parameters, *e.g.* angle of attack, depend on outputs computed from the flow solution. Thus, a feedback loop must be used to correct those input parameters.

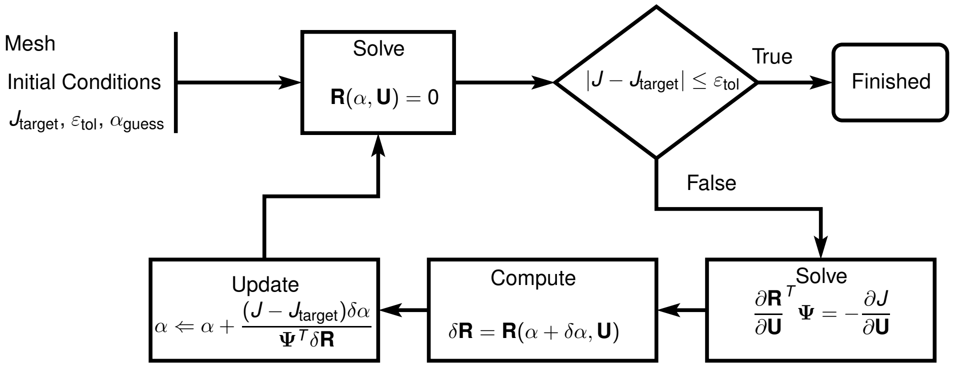

Mesh
Initial Conditions
$J_{\text{target}}, \varepsilon_{\text{tol}}, \alpha_{\text{guess}}$
Solve
$\mathbf{R}(\alpha, \mathbf{U}) = 0$
$|J - J_{\text{target}}| \le \varepsilon_{\text{tol}}$
True
Finished
False
Solve
$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T \boldsymbol{\Psi} = -\frac{\partial J}{\partial \mathbf{U}}$
Compute
$\delta\mathbf{R} = \mathbf{R}(\alpha + \delta\alpha, \mathbf{U})$
Update
$\alpha \Leftarrow \alpha + \frac{(J - J_{\text{target}})\delta\alpha}{\boldsymbol{\Psi}^T \delta\mathbf{R}}$

Figure 2. Adjoint-based boundary-condition parameter correction, here shown for the angle of attack, α.

The feedback loop used in this work is illustrated in Figure 2, where J_{target} is the target value of the output for which the parameter, α, is trimmed. The cycle starts by solving the flow equations using an initial guess for α. Then, J is computed and checked against J_{target} under a trimming tolerance, ε_{tol}. Until this tolerance is met, α is corrected using Newton's method for which the sensitivity of J with respect to α is needed. This sensitivity is computed via an inner product between an adjoint for J and a residual perturbation δR resultant from a perturbation in α. This residual perturbation is computed by evaluating the residual with the boundary condition perturbed by a small user-defined $\delta\alpha$. In this work, $\delta\alpha$ is one thousandth of a radian.

In cases where the target value for the output is not achievable or the initial guess is bad, the cycle in Figure 2 may not converge. In those cases, a contingency plan is needed, *e.g.*, a maximum number of iterations is assigned or the cycle is restarted with a better initial guess. In the output-based adaptation framework presented in this work, the boundary conditions are only trimmed if the error estimate for J is smaller than its trimming tolerance, ε_{tol}.

III. Output Error Estimation

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

$$\mathbb{R}(\mathbf{u}, \mathbf{w}) = 0, \qquad \forall \mathbf{w} \in \mathcal{V}, \tag{5}$$

the adjoint $\boldsymbol{\psi} \in \mathcal{V}$ is the sensitivity of J to an infinitesimal source term added to the left-hand side of the original PDE. $\boldsymbol{\psi}$ satisfies a linear equation,

$$\mathbb{R}'[\mathbf{u}](\mathbf{w}, \boldsymbol{\psi}) + J'[\mathbf{u}](\mathbf{w}) = 0, \qquad \forall \mathbf{w} \in \mathcal{V}, \tag{6}$$

where the primes denote Fréchét linearization with respect to the arguments in square brackets. Details on the derivation of the adjoint equation can be found in many sources, including the review in Ref.[8] Specifically, in the present work we employ the discrete adjoint method, in which the system is derived systematically from the discretized primal system.[24, 25]

An adjoint solution can be used to estimate the numerical error in the corresponding output of interest. The resulting adjoint-weighted residual method is based on the observation that a solution $\mathbf{u}^{H,p}$ in a finite-dimensional approximation space $\mathcal{V}^{H,p}$, polynomials of order p on a subdivision T^H of the domain into elements κ_H, will generally not satisfy the original PDE. The adjoint $\boldsymbol{\psi} \in \mathcal{V}$ translates the residual perturbation to an output perturbation via,

$$\delta J = J(\mathbf{u}^{H,p}) - J(\mathbf{u}) \approx -\mathbb{R}(\mathbf{u}^{H,p}, \boldsymbol{\psi}). \tag{7}$$

This expression is based on a linear analysis, and hence for nonlinear problems and finite-size perturbations, the result is approximate.

Although the continuous solution $\mathbf{u}$ is not required directly, the continuous adjoint $\boldsymbol{\psi}$ must be approximated to make the error estimate in Eqn. 7 computable. In practice, $\boldsymbol{\psi}^{h,p^+}$ is solved approximately or exactly on a finer finite-dimensional space $\mathcal{V}^{h,p^+} \supset \mathcal{V}^{H,p}$.[26–28] This finer space can be obtained either through mesh subdivision or approximation order increase[20, 29, 30] – denoted here by changes in the superscript H and p, respectively.

The adjoint-weighted residual evaluation in Eqn. 7 can be localized to yield an adaptive indicator consisting of the relative contribution of each element to the total output error. In this work, the finer space is obtained by approximation order increment, $\mathcal{V}^{H,p+1} \supset \mathcal{V}^{H,p}$, and $\boldsymbol{\psi}^{H,p+1}$ is approximated by injecting $\boldsymbol{\psi}^{H,p}$ into $\mathcal{V}^{H,p+1}$ and applying 5 – unless otherwise noted – element block-Jacobi smoothing iterations. The output perturbation in Eqn. 7 is approximated as

$$\delta J \approx -\sum_{\kappa^H \in T^H} \mathbb{R}_{\kappa^H}(\mathbf{I}_{H,p}^{H,p+1}(\mathbf{u}^{H,p}), \boldsymbol{\psi}^{H,p+1} - \mathbf{I}_{H,p}^{H,p+1}(\boldsymbol{\psi}^{H,p})), \tag{8}$$

where $\mathbf{I}_{H,p}^{H,p+1}(\cdot)$ is an injection operator from p to $p+1$ in the coarse mesh T^H, and $\mathbb{R}_{\kappa^H}$ corresponds to the elemental residual as defined in Eqn. 2. Note, the difference between the coarse-space and fine-space adjoints is not strictly necessary due to Galerkin orthogonality.[8] However, when the primal residual is not fully-converged to machine precision levels the use of the adjoint perturbation gives better error estimates. Equation 8 expresses the output error in terms of contributions from each coarse element. A common approach for obtaining an adaptive indicator is to take the absolute value of the elemental contribution in Eqn. 8,[14, 27, 31–34]

$$\eta_{\kappa^H} = \left| \mathbb{R}_{\kappa^H}(\mathbf{I}_{H,p}^{H,p+1}(\mathbf{u}^{H,p}), \boldsymbol{\psi}^{H,p+1} - \mathbf{I}_{H,p}^{H,p+1}(\boldsymbol{\psi}^{H,p})) \right|. \tag{9}$$

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

A. Drag Error Estimation with Fixed Lift

In a fixed-lift run, we directly solve the discrete residual equations and indirectly, via a feedback loop (Section B), solve for α to satisfy the lift constraint. This can be written as the following discrete system:

$$\begin{cases} \mathbf{R}(\alpha, \mathbf{U}) & = 0 \\ L(\alpha, \mathbf{U}) & = 0, \end{cases} \tag{10}$$

where $L(\alpha, \mathbf{U}) = C_L(\alpha, \mathbf{U}) - C_{L_{\text{target}}}$. Here, we introduce α as an argument in the discrete residual operator to explicitly denote the parameterization of the free-stream boundary condition.

As we are interested in computing drag with a solution that satisfies the constraints in Eqn. 10, we form a Lagrangian by introducing adjoint variables:

$$\mathcal{L}(\alpha, \mathbf{U}, \mathbf{\Psi}_R, \Psi_L) = D(\alpha, \mathbf{U}) + \mathbf{\Psi}_R^T \mathbf{R}(\alpha, \mathbf{U}) + \Psi_C L(\alpha, \mathbf{U}), \tag{11}$$

where $D(\alpha, \mathbf{U})$ is the drag function, Ψ_R is the drag adjoint as defined previously and Ψ_C represents the discrete sensitivity of drag with respect to perturbations in the lift constraint. In the continuum limit, this would represent the slope of a tangent to the drag polar curve.

We seek variations of the drag function, *i.e.* a drag error estimate, that satisfy the constraints in Eqn. 10. This corresponds to setting $\delta\mathcal{L} = 0$ for general perturbation in the input parameters. Taking the variation of the Lagrangian yields:

$$\delta\mathcal{L} = \delta D + \underbrace{\boldsymbol{\Psi}_R^T \delta\mathbf{R}}_{\text{(a)}} + \underbrace{\Psi_C \delta L}_{\text{(b)}} = 0, \tag{12}$$

where "a" is the drag error estimate for fixed α described previously,"b" is the influence of the lift error in the drag error due to the lift constraint and δL is the lift error estimate.

We compute Ψ_C according to:

$$\Psi_C = \frac{\partial D/\partial\alpha}{\partial L/\partial\alpha}, \tag{13}$$

where the sensitivities with respect to α are approximated via an inner product between the drag and lift adjoints and a residual perturbation due to a perturbation in α.

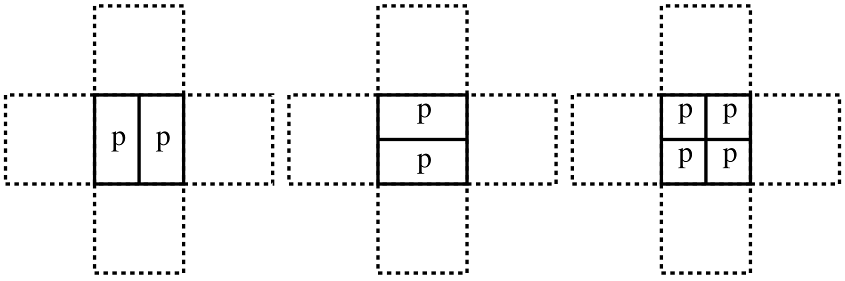
The final expression for the drag error estimate under a fixed-lift constraint is:

$$\delta D = -\underbrace{\boldsymbol{\Psi}_R^T \delta\mathbf{R}}_{(a)} - \Psi_C \underbrace{\delta L}_{(b)}, \tag{14}$$

where the terms "a" and "b" are respectively approximated using drag and lift adjoints in Eqn. 8. Note that appropriate sign reversals are required so that Eqn. 14 and Eqn. 8 are compatible.

IV. Mesh Adaptation Mechanics

The elemental adaptive indicator, η_{κ^H}, drives a fixed-fraction hanging-node adaptation strategy. In this strategy, which was chosen for simplicity and predictability of the adaptive algorithm, a certain fraction, f^{adapt}, of the elements with the largest values of η_{κ^H} is marked for refinement. Marked elements are refined according to discrete options which correspond to subdividing the element in different directions or increasing the approximation order. For quadrilaterals, the discrete options are: x-refinement, y-refinement, and xy-refinement, as depicted in Figure 3. Although the option of modifying the local polynomial approximation order is possible in this framework,[17] we consider only h-adaptation. The directions x and y refer to reference-space coordinates of elements that can be arbitrarily oriented and curved in physical space. Also, the subelements created through refinement inherit the approximation order from the original element. In three dimensions a hexahedron can be refined in seven ways: three single-plane cuts, three double-plane cuts, and isotropic refinement.



(a) x-refinement (b) y-refinement (c) xy-refinement

Figure 3. Quadrilateral h-refinement options. The dashed lines indicate the neighbors of the refined element.

In this work, h-refinement is performed in an element's reference space by employing the coarse element's reference-to-global coordinate mapping in calculating the refined element's geometry node coordinates. The refined elements inherit the same geometry approximation order and quadrature rules as the parent coarse element. As a result, there is no loss of element quality when a nonlinear mapping is used to fit the element to a curved geometry. Therefore, curved elements near a boundary can be efficiently refined to capture boundary layers in viscous flow. For simplicity of implementation, the initial mesh is assumed to capture the geometry sufficiently well, through a high enough order of geometry interpolation on curved boundaries, such that no additional geometry information is used throughout the refinements. That is, refinement of elements on the geometry boundary does not change the geometry. We note that for highly-anisotropic meshes, curved elements may be required away from the boundary, and for simplicity we use meshes with curved elements throughout the domain.

Note that elements created in a hanging-node refinement can be marked for h-refinement again in subsequent adaptation iterations. In this case, neighbors will be cut to keep one level of refinement difference between adjacent cells. This is illustrated in Figure 4.

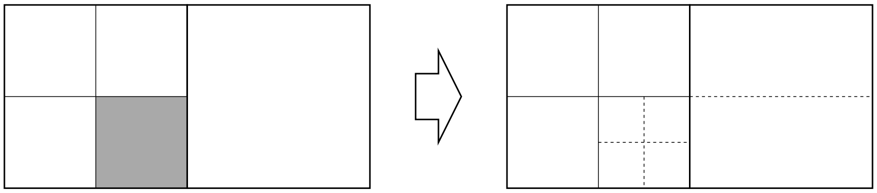


Figure 4. Hanging-node adaptation for a quadrilateral mesh, with a maximum of one level of refinement separating two elements. The shaded element on the left is marked for isotropic refinement, and the dashed lines on the right indicate the additional new edges formed.

Downloaded by UNIVERSITY OF MICHIGAN on April 1, 2014 | http://arc.aiaa.org | DOI: 10.2514/6.2013-51

A. Merit Function

The choice of a particular refinement option is made locally in each element flagged for refinement. This choice is made by defining a merit function $m(i)$ that ranks each available refinement option i. This function is defined as

$$m(i) = \frac{b(i)}{c(i)} \ , \tag{15}$$

where b and c respectively correspond to measures of the benefit and the computational cost of the refinement option indexed by i. These measures depend on the method used for solving the flow equations and they should be tailored for each specific solver.

During calculation of the merit function, local mesh and data structures are created, one for each element, that include the flagged element and its first-level neighbors along with the corresponding primal and adjoint states. In these local structures, the central element is refined in turn according to each of the discrete options. On the refined local mesh, the merit function is computed and the refinement option with the largest value of $m(i)$ is chosen.

Since we seeking the most efficient way of locally refining an element amongst the available options, it is important that the cost and benefit measures in Eqn. 15 are accurate but tractable representations of the computational expense and gain in accuracy respectively. In Refs. [17, 22], we discuss these aspects at length.

In an output-based mesh adaptation cycle, the steady-state residual is driven to zero at each adaptive step. Therefore, mesh modification on the element level can be interpreted as a local residual perturbation. Since an adjoint solution represents the sensitivity of an output with respect to a residual perturbation, we define our benefit function as:

$$b(i) = \sum_{\kappa^h \in \kappa^H} |\mathrm{R}_{\kappa^h}(\mathrm{U}_k \mathrm{T}_{kl}(i))_j| |\Psi_k \mathrm{T}_{kj}(i)|, \tag{16}$$

where $\mathrm{R}_{\kappa^h}(\cdot)_j$ is a discrete residual component in the embedded element, $\mathrm{T}(i)$ is a matrix that transfers the discrete primal and adjoint states to the local meshes for each refinement i, and Ψ is the discrete coarse-space adjoint solution. Note that the adjoint variables act as positive weights for each of the perturbations.

In this work, most of the computational time is spent in the GMRES algorithm used to solve the linear systems that arise at each step of the pseudo-transient continuation method. These systems are sparse, hence we approximate the number of floating point operations in applying GMRES by the number of non-zero entries in the residual Jacobian matrix. Based on this observation, we define the cost measure as:

$$c(i) = \sum_{\kappa^h \in \kappa^H} \left\{ (p+1)^{2\cdot\text{dim}} + \sum_{N_{\text{iface}}(i)} (p+1)^{2\cdot\text{dim}} \right\}, \tag{17}$$

where p denotes the polynomial approximation order and $N_{\text{iface}}(i)$ is the number of internal faces due to refinement option i. The first term in Eqn. 17 accounts for the self-blocks of the residual Jacobian matrix corresponding to each of the subelements. The second term corresponds to the dependence of the subelements' residual on the neighboring states.

V. Results

A. NACA 0012, $M_\infty = 0.15,\ Re = 6 \times 10^6$, Drag Polar

This case is one of the NASA's Turbulence Modeling Resource cases.[35] The purpose of this case is to validate the modifications made to the SA model. As suggested by NASA's Turbulence Modeling Resource, the domain's outer boundary is located 500 chord-lengths away from the airfoil. We consider eight angles of attack in the drag polar: $\alpha = 0°$, $2°$, $4°$, $6°$, $8°$, $10°$, $12°$, and $15°$. For each angle of attack, an initial quartic mesh is generated by agglomerating 16 quadrilaterals from a linear mesh. The linear meshes are generated so that the cells downstream from the airfoil are approximately aligned with the wake. Figure 5 shows an example of an initial quartic mesh.

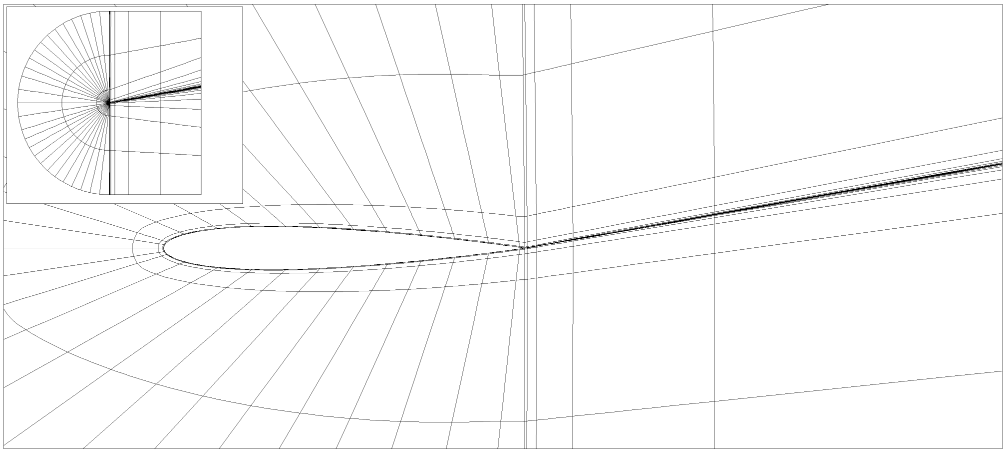
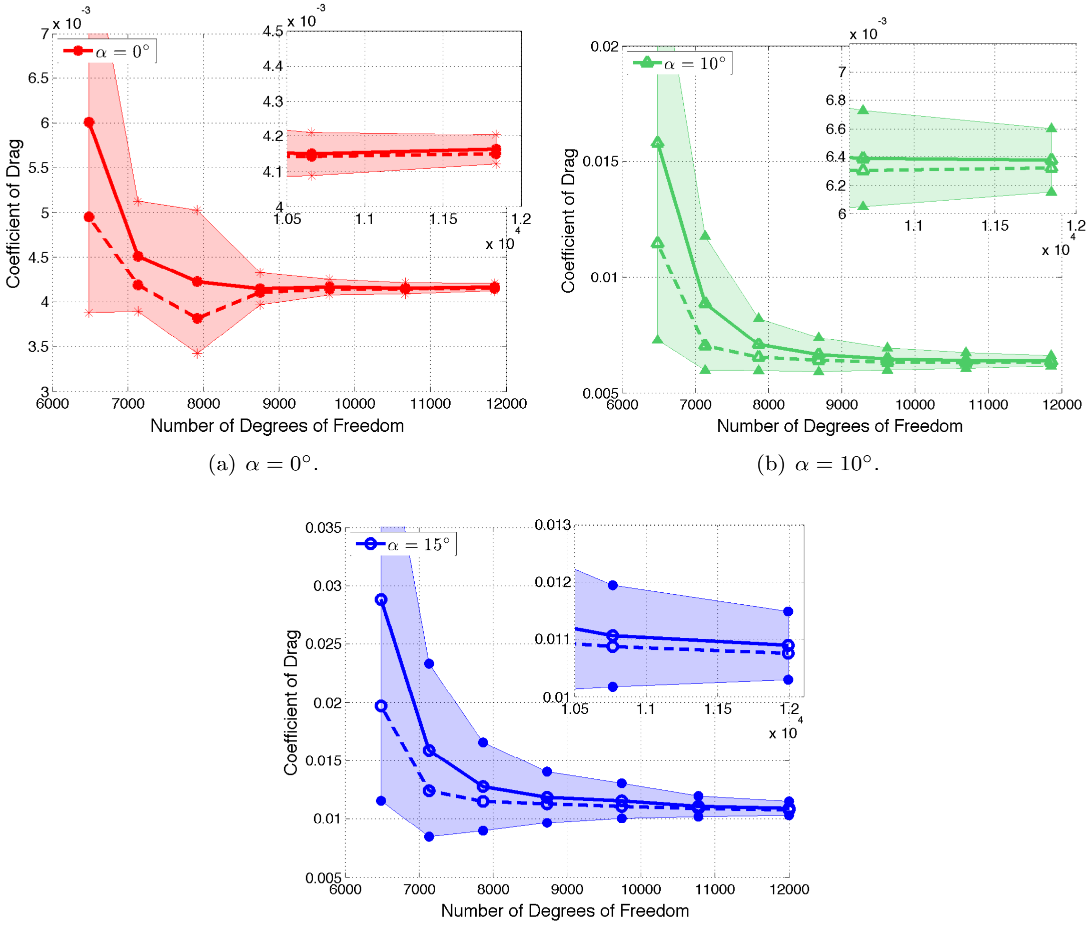


Figure 5. NACA 0012, $M_\infty = 0.15,\ Re = 6 \times 10^6$, drag polar: Initial mesh for $\alpha = 10°$ (720 quartic elements).

The polynomial order is $p = 2$ and the discretized SA equation is scaled by $\kappa_{\text{SA}} = 1000$. The adaptation is driven by drag error with $f^{\text{adapt}} = 10\%$ and the residual norm is reduced by 8 orders of magnitude at each adaptive step. To simplify our analyses, we limit the number of the adaptive steps to 6 for all the angles and measure the error level of the final result. Figure 6 shows the drag convergence for three representative angles of attack. The largest final error estimate over all the angles of attack is approximately 3 drag counts ($\sim 3\%$) in the $\alpha = 15°$ case.

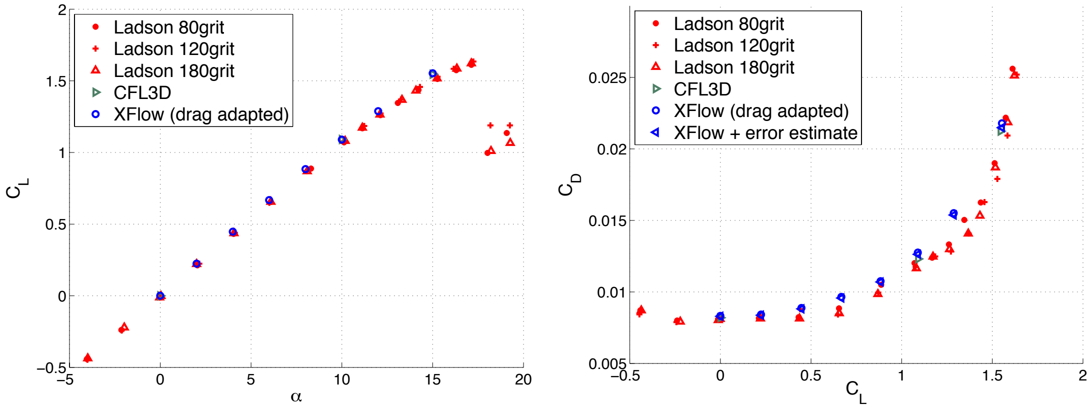


(a) $\alpha = 0°$.

(b) $\alpha = 10°$.

(c) $\alpha = 15°$.

Figure 6. NACA 0012, $M_\infty = 0.15,\ Re = 6 \times 10^6$, drag polar: drag convergence for three angles of attack; solid lines: drag values; dashed lines: drag corrected by its error estimate; shading: magnitude of the sum of error indicators.



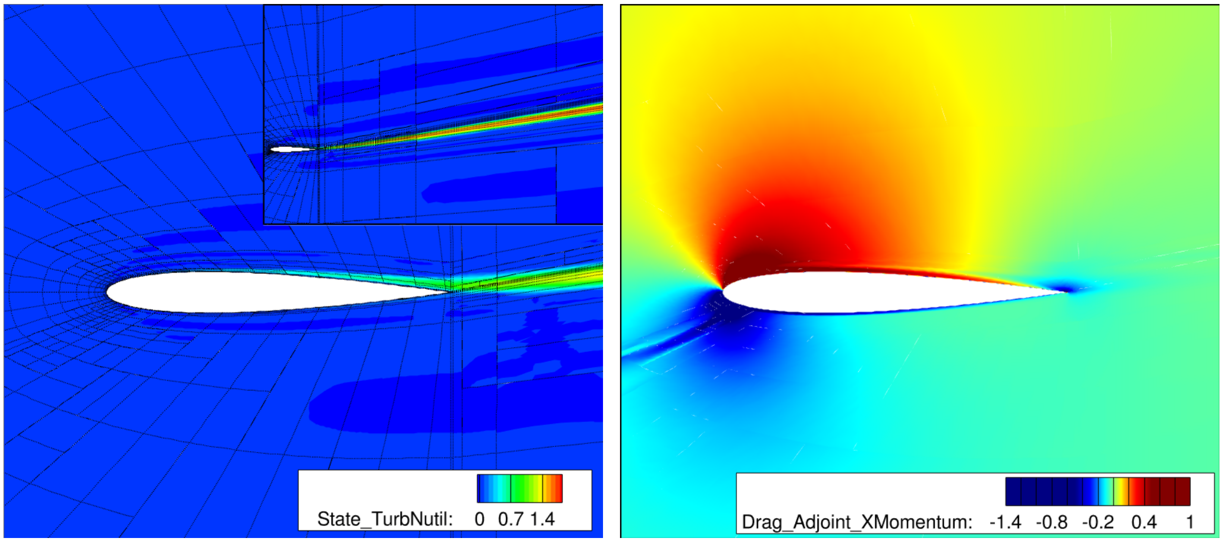
(a) Lift coefficient versus angle of attack.

(b) Drag polar

Figure 7. NACA 0012, $M_\infty = 0.15,\ Re = 6 \times 10^6$, drag polar: comparison with experimental data.

Figure 7 compares our adaptive results with Ladson's experimental data[36] and with results computed with CFL3D[37] on a fine, 897 × 257 element, structured grid.[35] The experimental data consists of three sets of wind tunnel runs with varying roughness of carborundum strips to force transition to turbulence at the 5% position along the chord. This reduces transition effects and allows for a more adequate comparison with fully turbulent simulations.

In spite of the adaptation being driven by drag error, the lift values in Figure 7(a) are in close agreement with the experimental data. Our computed drag values are within 3% difference with respect to CFL3D's results which is within the spread of 4% in the CFD results with the SA model presented in Ref. [35]. With respect to the experimental values in Figure 7(b), the simulations show slightly larger drag values. We attribute these differences to the turbulence model and possibly to experimental measurement precision as the adjoint-based error estimation and adaptation only targets, and provides an error estimate for, the discretization error.



(a) 6th drag-adapted mesh and SA-working variable contours. (b) 6th drag-adapted x-momentum adjoint solution for drag.

Figure 8. NACA 0012, $M_\infty = 0.15$, $Re = 6 \times 10^6$, drag polar: final mesh, $\rho\tilde{\nu}$ contours, and drag adjoint for $\alpha = 10°$.

The adjoint solution offers insight on regions of the computational domain where discretization errors affect the output of interest. Figure 8(b) shows the x-momentum drag-adjoint solution for the $\alpha = 10°$ case. The most notable feature of this adjoint solution is the stagnation streamline which, in the inviscid limit, is a weak inverse-square-root singularity.[38] This sharp variation of the adjoint is reflected in the adapted mesh in Figure 8(a).

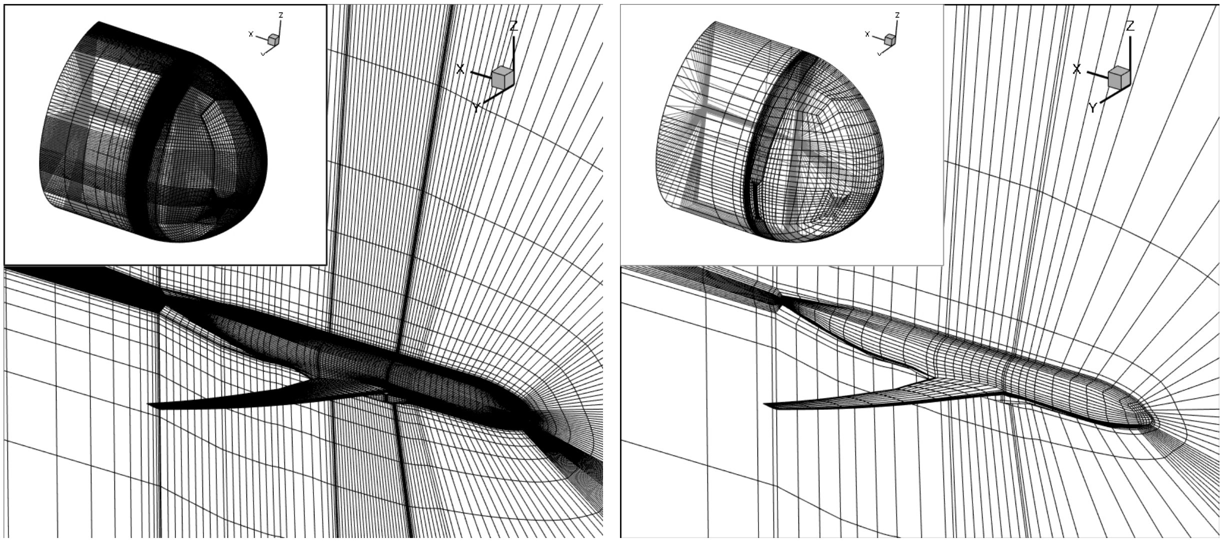
Other features that are important for accurate prediction of drag are the boundary layer, the upper flow acceleration region, the trailing edge, and the wake. These regions are also frequently targeted for refinement as they present large magnitudes of the adjoint variables.

B. CRM - wing-body geometry, $M_\infty = 0.85$, $C_L = 0.5$, $Re_{\mathbf{MAC}} = 5 \times 10^6$

This case consists of transonic, turbulent flow over NASA's Common Research Model.[39] This wing-body geometry mimics a modern passenger aircraft and its purpose is to establish a reference for testing computational tools for simulation and design. This case is considered very challenging for discontinuous finite element methods and certain aspects of the results presented here could lead to additional investigations in the future.

The cubic mesh used in this case was generated via agglomeration of linear cells. The initial linear mesh was generated with the tradeoff of being coarse to use in our adaptation routine but fine-enough to represent the geometry adequately. Figure 9 shows the linear and the agglomerated meshes. The off-wall spacing in the agglomerated mesh is such that $y^+ \approx 100$, based on a flat-plate correlation for the coefficient of friction and with the Reynolds number based on the mean aerodynamic chord ($Re_{\text{MAC}} = 5 \times 10^6$).

The discretized SA equation is scaled by $\kappa_{\text{SA}} = 100$ and a slightly modified version[22] of Persson and Peraire's[40] element-wise constant viscosity shock-capturing scheme is included in the residual operator. The



(a) Linear mesh used for agglomeration (1218375 elements). (b) Cubic mesh generated via agglomeration (45125 elements).

Figure 9. CRM - wing-body geometry, $M_\infty = 0.85,\ C_L = 0.5,\ Re_{\mathbf{MAC}} = 5 \times 10^6$: linear and agglomerated cubic meshes.

convergence criterion is a residual-norm reduction of 8 orders of magnitude from its initial value.

We consider anisotropic h-adaptation at fixed $p = 1$ with $f^{\text{adapt}} = 10\%$. Converging the initial solution for this problem is difficult. The physicality-constrained solver with line-search and the mRDM CFL strategy[22] is used for the first primal solve. In addition, one step of mesh adaptation based on the physicality constraints[41] is taken to help the solver to converge. In subsequent solves, converging the residual is significantly easier.

The output used for adaptation is the total drag at a fixed lift. That is, at each primal solve, the angle of attack is trimmed so that the coefficient of lift is $C_{L_{\text{target}}} = 0.5 \pm 0.001$. The method for trimming α is described in Section B.

Due to lack of spatial resolution in the initial mesh, the flow separates (Figure 11(a)) before the lift requirement is achieved. The solution in the initial mesh is nearly unsteady which makes the adjoint problem very ill-conditioned and, consequently, causes the error-estimates to be very large as shown in Figures 12(a) and 12(b). In this situation, the lift requirement is relaxed and the adaptive process proceeds. This decision is not yet automated and is one of the aspects of this problem that could benefit from further research.

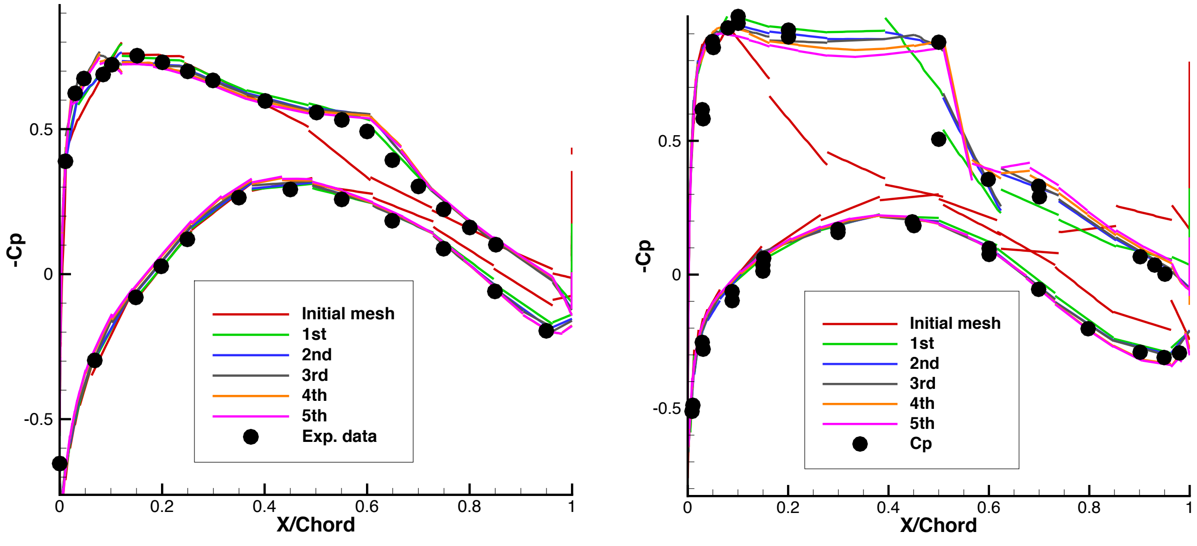
After the first drag-based adaptation step, the flow field is significantly different (Figure 11(b)). The supersonic region is larger and no visible flow separation is present. The lift requirement is now satisfied and the error estimates for lift and drag are significantly smaller (Figure 12).

The Mach number contours shown in Figure 11 do not present large differences after the second adaptation step. Also, the areas targeted for adaptation are similar to the regions observed in the DPW III - W1 case presented in Ref. [17]. These regions are: the stagnation streamline, the sonic transition, the shock-boundary-layer interaction, and the wake.

Figure 10 compares the pressure coefficient at two span locations with the corresponding experimental data[a].[42] Note that the initial result is very far from the experiments. However, after one adaptation step the pressure distribution is much closer to the experimental data and as the adaptation progresses, the shock profile becomes sharper and the changes in pressure distribution become smaller.

Figure 12 shows the convergence history for drag, lift, and pitching moment. Note that our results for pitching moment are within the range of data submitted to the workshop, while the drag values are above the range of results from the workshop. However, it is worth emphasizing that the finest solution presented here has a factor of 5 to 10 fewer degrees of freedom than the mid-range meshes used in the uniform refinement studies in DPW-V.

[a] Experimental data was digitized from the 5th Drag Prediction Workshop summary presentation.



(a) Pressure coefficient at 13.06% of the reference span.

(b) Pressure coefficient at 50.24% of the reference span.

Figure 10. CRM - wing-body geometry, $M_\infty = 0.85$, $C_L = 0.5$, $Re_{\mathbf{MAC}} = 5 \times 10^6$: comparison of pressure coefficient with experimental data.

1. High-Order Mesh and Geometric Irregularities

The mesh agglomeration algorithm used in this work receives as input a linear, multi-block, mesh where each block has a number of nodes that satisfies the following law:

$$N_{\text{node}} = (q \cdot N_i + 1) \cdot (q \cdot N_j + 1) \cdot (q \cdot N_k + 1), \tag{18}$$

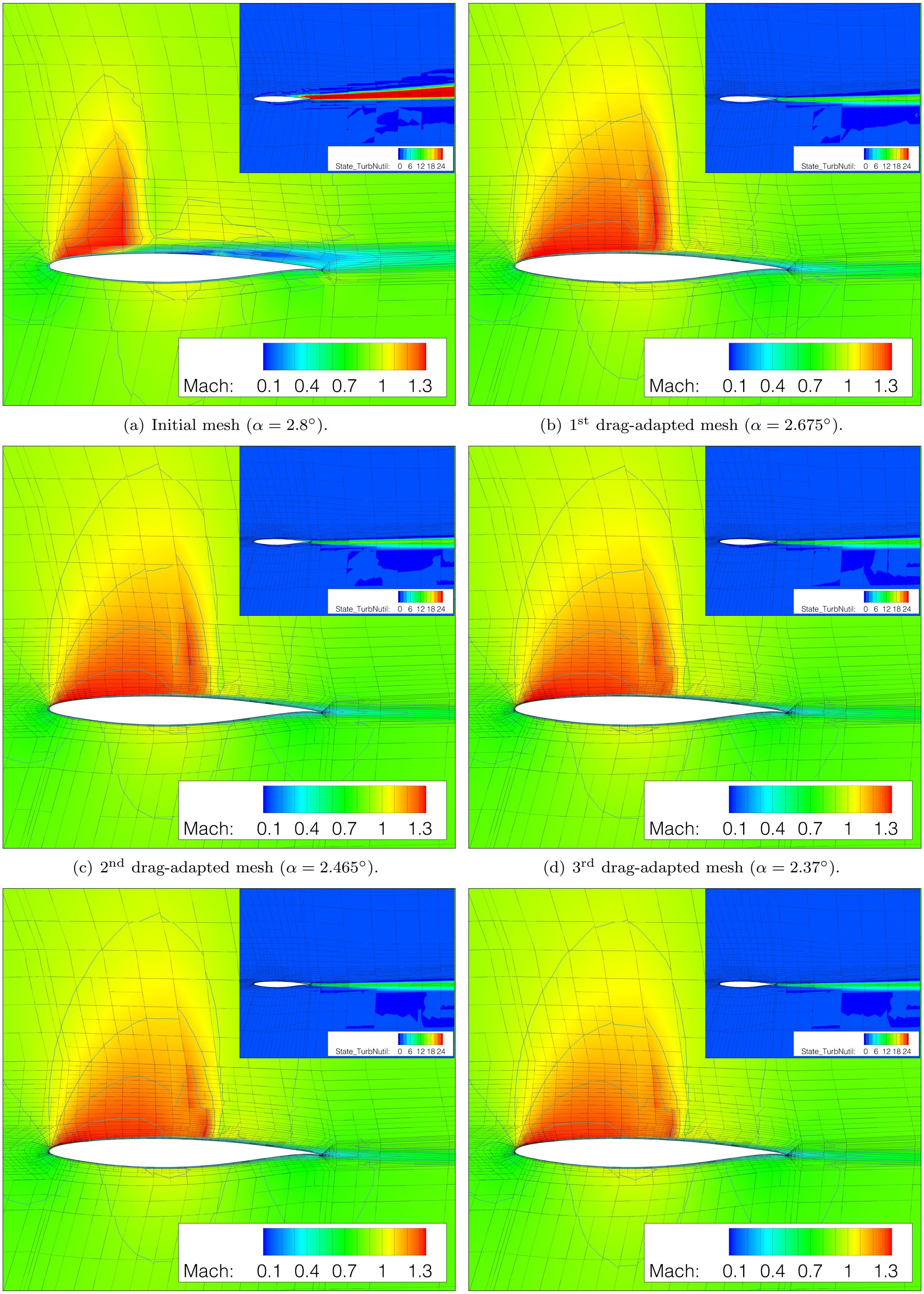
where q is the desired polynomial order for the geometry representation and $[N_i, N_j, N_k]$ are the number of q-th order elements in the $[i, j, k]$ directions. Note that with the above rule for the number of nodes, geometric irregularities of order $g \leq q$, on the agglomerated mesh, can only exist at element borders. The relevance of this observation is that the block boundaries in the linear mesh should coincide with the geometric irregularities of order $g \leq q$ in the source geometry. A mitigation to this problem is to generate locally finer meshes or to put block boundaries close to geometric irregularities. However, this is not always possible or easy to achieve.

The irregularities identified on the CRM geometry are: wing trailing edge, wing-fairing junction, cockpit-nose-cone junction, and fairing-body junction. In the case of our block topology for the initial mesh, a block boundary is not aligned with the aft portion of the fairing-body junction (Figure 13). This causes oscillations in the geometry of the agglomerated mesh that affect the computed drag and the robustness of the error estimates. In general, the slightly wavy geometry where $g \leq q$ is a possible source of larger drag in our results (Figure 12(a)), as the geometry is held fixed throughout the adaptation.

VI. Conclusions and Ongoing Work

We demonstrated the use of an adaptive discontinuous finite element method to predict drag. The two-dimensional results show that adjoint-based adaptation quickly – after six adaptation steps – reduces the discretization error to acceptable levels and allows for a separation between errors in drag due to discretization and inaccuracies in the turbulence modeling and/or in the experimentation.

The drag prediction results for the CRM geometry show that adjoint-based mesh adaptation can significantly save degrees of freedom in comparison to uniform refinement methods. The variational formulation of the discontinuous Galerkin method allows for straightforward output error estimation. In the 2D results, the drag error estimates provide a better correction for the output than in the CRM results. This is related



(a) Initial mesh ($\alpha = 2.8°$).

(b) 1st drag-adapted mesh ($\alpha = 2.675°$).

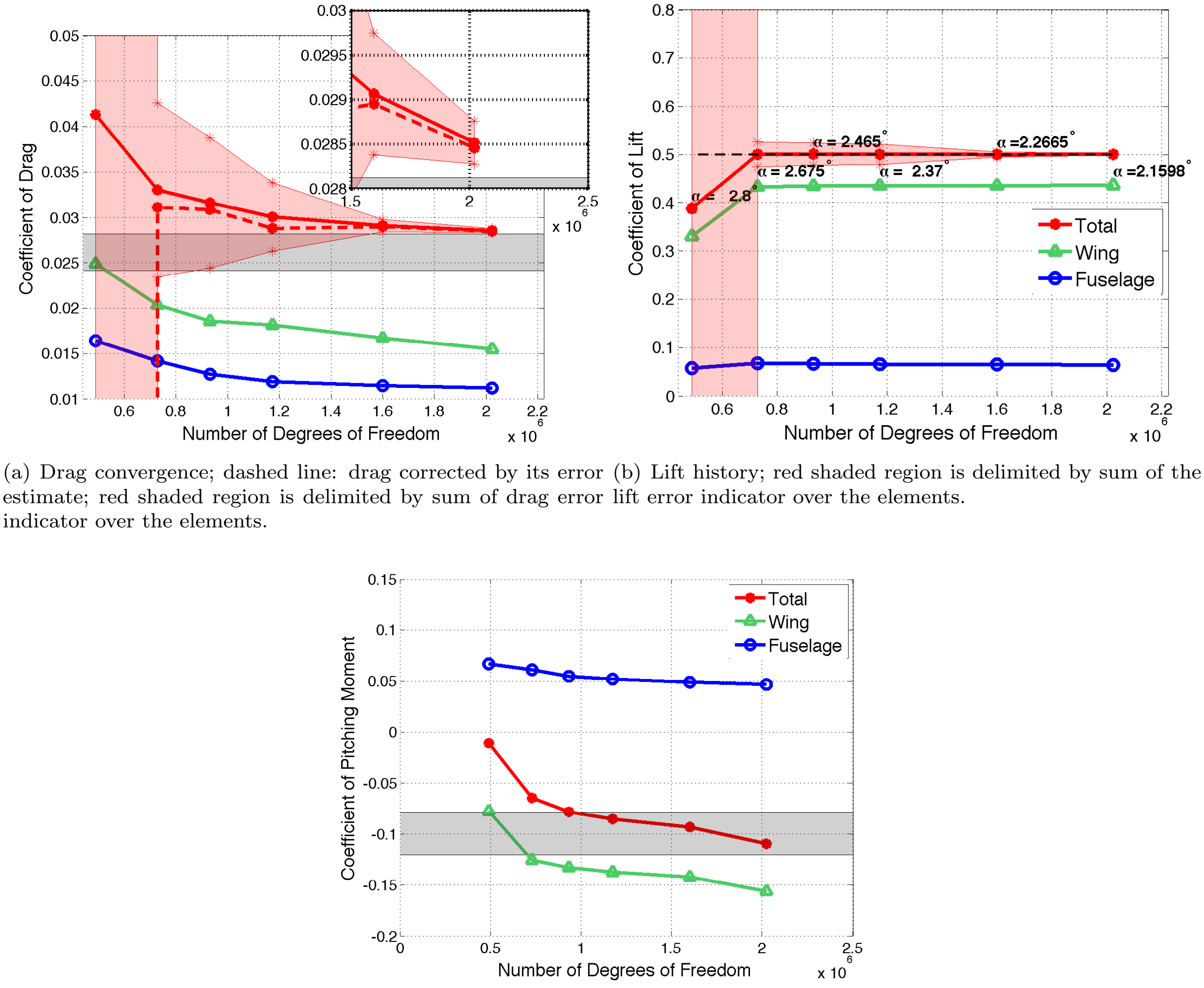
(c) 2nd drag-adapted mesh ($\alpha = 2.465°$).

(d) 3rd drag-adapted mesh ($\alpha = 2.37°$).

(e) 4th drag-adapted mesh ($\alpha = 2.2665°$).

(f) 5th drag-adapted mesh ($\alpha = 2.1598°$).

Figure 11. CRM - wing-body geometry, $M_\infty = 0.85$, $C_L = 0.5$, $Re_{\mathbf{MAC}} = 5 \times 10^6$: slice at 37% of the span (428 inches).

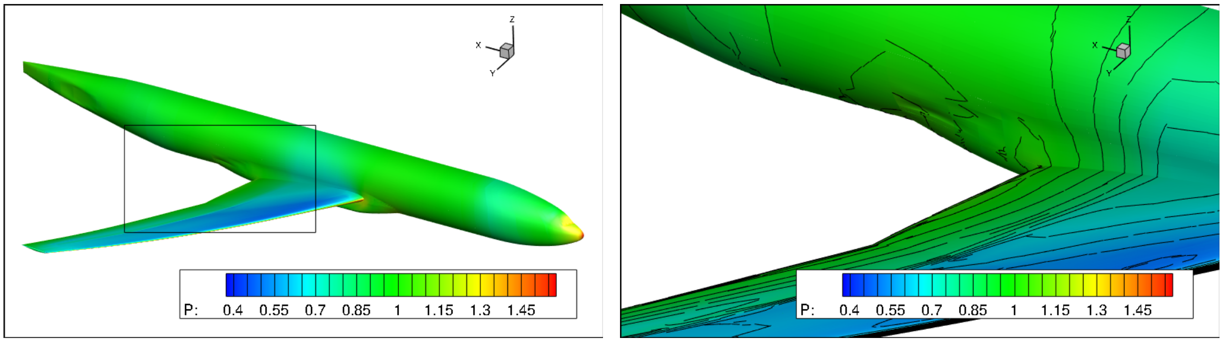


(a) Drag convergence; dashed line: drag corrected by its error estimate; red shaded region is delimited by sum of drag error indicator over the elements.

(b) Lift history; red shaded region is delimited by sum of the lift error indicator over the elements.

(c) Pitching moment history.

Figure 12. CRM - wing-body geometry, $M_\infty = 0.85,\ C_L = 0.5,\ Re_{\mathbf{MAC}} = 5 \times 10^6$: drag, lift, and pitching moment for the sequence of adapted meshes; gray shaded region: range of data submitted to DPW-V.



(a) Pressure contours on final adapted mesh.

(b) Waviness on fairing-body junction.

Figure 13. CRM - wing-body geometry, $M_\infty = 0.85,\ C_L = 0.5,\ Re_{\mathbf{MAC}} = 5 \times 10^6$: waviness on fairing-body junction.

to the robustness of the fine-space approximation of the adjoint solution in the CRM case. Investigating this issue is ongoing work.

Another challenge is managing geometric irregularities as oscillations may occur in the high-order geometry representation. A way to address this problem is to adapt the surface elements of the initial mesh based on the integrated distance between the high-order elements' solid boundaries and the source geometry and to re-project the new nodes created in this adaptation onto the source geometry. This allows for explicit control over the geometry representation error with the caveat that we need to ensure volume positivity of the curved elements at the boundary. The development of this algorithm is ongoing work.

The fixed-lift constraint adds an extra term in the drag error estimate. This suggests that the adaptation should be driven by both drag and lift error. The results presented here were obtained before this realization and we intend to evaluate the impact of combining lift and drag in the adaptive results.

Finally, three-dimensional RANS simulations using discontinuous finite elements is still very challenging both in terms of robustness and in terms of computational expense. The results, however, are very promising and further research in these topics will certainly be beneficial to aerodynamic performance prediction.

Acknowledgments

The authors acknowledge the support given by the University of Michigan to the development of this work and appreciate the computational resources provided by NASA which were essential to obtaining the results presented here. Funding for this research is provided by the Air Force Office of Scientific Research under contract number FA9550-10-C-0040.

References

[1] Buttazzo, G., Frediani, A., Allmaras, S. R., Bussoletti, J. E., Hilmes, C. L., Johnson, F. T., Melvin, R. G., Tinoco, E. N., Venkatakrishnan, V., Wigton, L. B., and Young, D. P., "Algorithm Issues and Challenges Associated with the Development of Robust CFD Codes," *Variational Analysis and Aerospace Engineering*, Vol. 33 of *Springer Optimization and Its Applications*, Springer New York, 2009, pp. 1–19.

[2] Vassberg, J. C., DeHaan, M. A., and Sclafani, T. J., "Grid Generation Requirements for Accurate Drag Predictions Based on OVERFLOW Calculations," *16th AIAA Computational Fluid Dynamics*, No. 2003-4124, 2003.

[3] Fidkowski, K. J., *A High–order Discontinuous Galerkin Multigrid Solver for Aerodynamic Applications*, MS thesis, M.I.T., Department of Aeronautics and Astronautics, June 2004.

[4] Levy, D. W., Zickuhr, T., Vassberg, J., Agrawal, S., Wahls, R. A., Pirzadeh, S., and Hemsch, M. J., "Data Summary from the First AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 40, No. 5, 2003, pp. 875–882.

[5] Laflin, K. R., Klausmeyer, S. M., Zickuhr, T., Vassberg, J. C., Wahls, R. A., Morrison, J. H., Brodersen, O. P., Rakowitz, M. E., Tinoco, E. N., and Godard, J.-L., "Data Summary from the Second AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 42, No. 5, 2005.

[6] Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Laflin, K. R., and Mavriplis, D. J., "Abridged Summary of the Third AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 45, No. 3, 2008.

[7] Vassberg, J. C., Tinoco, E. N., Mani, M., Rider, B., Zickuhr, T., Levy, D. W., Brodersen, O. P., Eisfeld, B., Crippa, S., Wahls, R. A., Morrison, J. H., Mavriplis, D. J., and Murayama, M., "Summary of teh Fourth AIAA CFD Drag Prediction Workshop," *28th AIAA Applied Aerodynamics Conference*, 2010.

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

[9] 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.

[10] Park, M. A., "Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation," AIAA Paper 2002-3286, 2002.

[11] Hartmann, R. and Houston, P., "Goal-Oriented A Posteriori Error Estimation for Multiple Target Functionals," *Hyperbolic Problems: Theory, Numerics, Applications*, edited by T. Hou and E. Tadmor, Springer-Verlag, 2003, pp. 579–588.

[12] Fidkowski, K. J. and Darmofal, D. L., "An Adaptive Simplex Cut-Cell Method for Discontinuous Galerkin Discretizations of the Navier-Stokes Equations," AIAA Paper 2007-3941, 2007.

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

[14] Ceze, M. and Fidkowski, K. J., "Output-Driven Anisotropic Mesh Adaptation for Viscous Flows using Discrete Choice Optimization," *48th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA 2010-170, 2010.

[15] 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.

[16] Fidkowski, K. J. and Luo, Y., "Output-Based Space-Time Mesh Adaptation for the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 230, 2011, pp. 5753–5773.

[17] Ceze, M. and Fidkowski, K. J., "Anisotropic *hp*-adaptation framework for functional prediction," *AIAA Journal*, 2012 Accepted.

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

[19] 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.

[20] Oliver, T. A., *A High–order, Adaptive, Discontinuous Galerkin Finite Elemenet Method for the Reynolds-Averaged Navier-Stokes Equations*, PhD dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.

[21] Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *30th Aerospace Sciences Meeting and Exhibit*, No. AIAA-92-0439, AIAA, 1992.

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

[23] Burgess, N. K. and Mavriplis, D. J., "Robust Computation of Turbulent Flows using a Discontinuous Galerkin Method," *50th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA 2012-0457, 2012.

[24] Giles, M. B., Duta, M. C., Müller, J.-D., and Pierce, N. A., "Algorithm Developments for Discrete Adjoint Methods," *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 198–205.

[25] Mader, C. A., Martins, J. R., Alonso, J. J., and van der Weide, E., "ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers," *AIAA Journal*, Vol. 46, No. 4, 2008, pp. 863–873.

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

[27] 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.

[28] 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.

[29] Lu, J., *An a Posteriori Error Control Framework for Adaptive Precision Optimization Using Discontinuous Galerkin Finite Element Method*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

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

[31] Venditti, D. A. and Darmofal, D. L., "Grid adaptation for functional outputs: application to two-dimensional inviscid flows," *Journal of Computational Physics*, Vol. 176, No. 1, 2002, pp. 40–39.

[32] 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.

[33] 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.

[34] 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.

[35] National Aeronautics and Space Administration, "Turbulence Modeling Resource," NASA Langley, 2012, http://turbmodels.larc.nasa.gov/.

[36] Ladson, C. L., "Effects of Independent Variation of Mach and Reynolds Numbers on the Low-Speed Aerodynamic Characteristics of the NACA 0012 Airfoil Section," Technical Memorandum 4074, NASA, 1988.

[37] National Aeronautics and Space Administration, "CFL3D Home Page," NASA Langley, 2012, http://cfl3d.larc.nasa.gov/.

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

[39] Vassberg, J. C., DeHaan, M. A., Rivers, M., and Wahls, R. A., "Development of a Common Research Model for Applied CFD Validation Studies," *26th AIAA Applied Aerodynamics Conference*, 2008.

[40] Persson, P.-O. and Peraire., J., "Sub-cell shock capturing for discontinuous Galerkin methods," *44th AIAA Aerospace Sciences Meeting and Exhibit*, No. 2006-112, 2006.

[41] Ceze, M. and Fidkowski, K. J., "A Robust Adaptive Solution Strategy for High-Order Implicit CFD Solvers," *20th AIAA Computaional Fluid Dynamics Conference*, No. AIAA 2011-3696, AIAA, 2011.

[42] Tinoco, E. N., Levy, D., and Brodersen, O. P., "DPW 5 Summary of Participant Data," June 2012, http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/.