

# Drag Output Error Estimation Using the Entropy Adjoint Approach

Krzysztof J. Fidkowski\*, Marco A. Ceze† and Philip L. Roe‡

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

This paper presents a method for estimating the drag error in a computational fluid dynamics simulation using the adjoint property of entropy variables. The error of interest is that caused by the numerical discretization in the calculation of drag, including effects of finite mesh size and approximation order. Since the entropy variables are obtained by a direct transformation of the state, no separate adjoint solution is required. The proposed error estimate applies to a drag calculation that involves a far-field integration of the entropy. Error estimates are derived separately for inviscid and viscous flows, and results are presented for examples in two dimensions. These results show that the error estimate is consistently within a factor of 1.0 and 1.5 of the true numerical error in the drag. The results also indicate that the error estimate applies equally well to near-field drag calculations on meshes adapted using the entropy adjoint approach. As such, this estimate provides a useful stopping criterion for adaptation using the entropy adjoint.

## I. Introduction

Numerical error due to finite-dimensional discretization is present in virtually all practical computational fluid dynamics (CFD) solutions. In large, complex, simulations this error can be difficult to control, especially if computational resources are already taxed. Solution-based adaptive methods address this problem by allocating degrees of freedom to areas where they are deemed necessary based on some automated interrogation of the solution, in order to produce numerically accurate solutions at reduced computational cost. Even when these methods are used, however, solution accuracy is not guaranteed. One complication is that accuracy can be measured in different ways, for example through norms of the residual or solution error, or through errors in scalar outputs. Yet very few of the adaptive methods used today provide bounds or estimates of any of these measures of accuracy. Indeed, many heuristic methods have been shown, at least in certain cases, to produce remarkably incorrect solutions under one or more relevant accuracy measures.

A notable exception is output-based adaptation, in which error estimates are computed for targeted scalar quantities during the adaptive process. These methods have been applied successfully to aerodynamic flows in two and three dimensions,<sup>1-5</sup> resulting in robust solution procedures. The output error estimates rely on adjoint solutions, which although not prohibitively expensive computationally, do require code-intrusive changes and are not yet widely available.

---

\*Assistant Professor, AIAA Member

†Graduate Research Assistant

‡Professor, AIAA Fellow

Recent work by the authors investigated an alternative adaptation method based on the “entropy adjoint”.<sup>6</sup> This method appealed to the observation that the entropy state variables act as an adjoint solution for an output that expresses an entropy balance statement in the computational domain. The residual weighted by the entropy variables then provides an error estimate for this output. That is, we can compute the numerical error in an output without calculating a separate adjoint solution, as the entropy variables are calculated directly from the conservative or primitive state vector.

Adaptation driven by the entropy adjoint approach was demonstrated to produce “all-around” good solutions that competed with engineering output adjoint solutions in terms of output accuracy for several representative aerodynamic cases. However, the entropy adjoint method did not provide error estimates for these engineering outputs. Without the exact solution, we would not know whether the adaptation converged to the right value and when the adaptation could be terminated. The only measure of error in the entropy adjoint approach is the error in the entropy balance output, which is generally not of direct engineering interest.

In this work we propose a relationship between the entropy-adjoint error estimate and an estimate of one engineering output of interest: drag, which is the free-stream aligned force on an object in an external flow. Drag is critical for analysis and design of aerospace vehicles, but it is notoriously difficult to predict. It is the subject of a large body of previous work,<sup>7–15</sup> including an ongoing workshop by the American Institute of Aeronautics and Astronautics.<sup>16–19</sup> The present work relies on the observation that drag in CFD, as in experiments, can be measured in more than one way: in particular via a near-field and via a far-field integration. The latter can be expressed in terms of entropy through a formula due to Oswatitsch,<sup>20</sup> and it is this formula that provides the basis for our relationship between drag and the entropy balance output. Moreover we show that for the cases considered, entropy-adjoint adaptation produces meshes in which the near-field and far-field drag errors are comparable, and we attribute this property to the conservative discretization and to the targeting of areas of spurious entropy generation by the entropy adjoint indicator.

The organization of this paper is as follows. Section II presents the near-field and far-field drag expressions, including Oswatitsch’s formula. The discretization of the Euler and compressible Navier-Stokes equations used in this work is given in Section III. Section IV discusses the adjoint-based numerical error estimation procedure using engineering output adjoints and using entropy variables. Section V presents the mesh adaptation procedure used in this study. Results for inviscid and viscous flows in two dimensions are given in Section VI.

## II. Drag Calculation in Computational Fluid Dynamics

The drag force on a body in an external flow governed by the steady Navier-Stokes equations can be calculated using a direct integration of the stress on the body surface,  $S_{\text{body}}$ ,

$$D_{\text{near}} = \int_{S_{\text{body}}} [p\mathbf{n} - \mathbf{n} \cdot \boldsymbol{\tau}] \cdot \hat{\mathbf{x}} dS, \quad (1)$$

where  $p$  is the pressure,  $\boldsymbol{\tau}$  is the viscous stress tensor,  $\mathbf{n}$  is the surface normal vector pointing outward from the fluid, and  $\hat{\mathbf{x}}$  is the unit vector pointing along the  $x$  axis, which is aligned with the free-stream. Figure 1 illustrates these definitions schematically. As the subscript on  $D$  indicates, we will refer to Eqn. 1 as the near-field drag calculation.

By conservation of momentum in steady state, the drag on the body can also be computed without approximation via an integral on the surface of a far-field control volume enclosing the

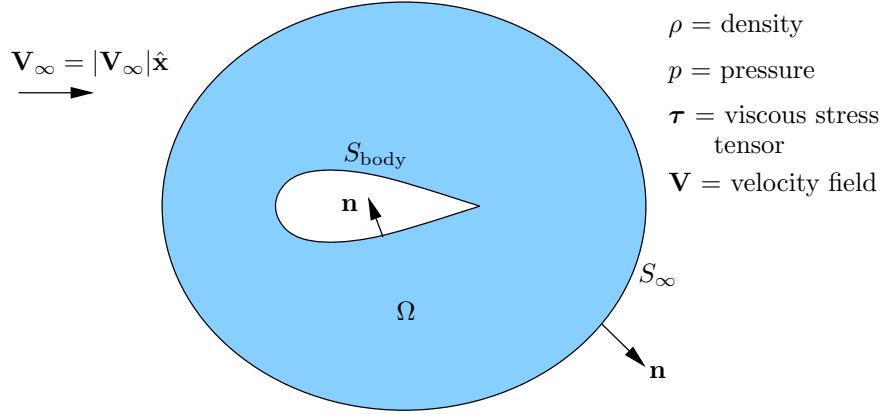


Figure 1. Schematic diagram for near- and far-field drag calculations.

body,  $S_\infty$ ,

$$D_{\text{far}} = \int_{S_\infty} [-\rho u \mathbf{V} \cdot \mathbf{n} - p n_x + \mathbf{n} \cdot \boldsymbol{\tau} \cdot \hat{\mathbf{x}}] dS, \quad (2)$$

where  $\mathbf{V}$  is the velocity field,  $\mathbf{n}$  is again the normal vector pointing outward from the fluid,  $n_x$  is the  $x$ -component of  $\mathbf{n}$ , and  $u$  is the velocity component in the  $x$  direction.  $D_{\text{near}}$  and  $D_{\text{far}}$  will be identical for the exact solution and equivalent to machine precision for a discrete solution when using a conservative scheme in which the residual is converged to machine precision.

In the case of no trailing vortices, if  $S_\infty$  is sufficiently far from the body such that  $p = p_\infty$  and the flow is parallel to the free-stream, the far-field expression in Eqn. 2 can be re-written as,<sup>11</sup>

$$D_{\text{osw}} = \int_{S_\infty} u_\infty \left[ 1 - \sqrt{1 + \frac{2}{(\gamma - 1)M_\infty^2} (1 - e^{\Delta s/c_p})} \right] \rho \mathbf{V} \cdot \mathbf{n} dS, \quad (3)$$

where  $u_\infty$  is the  $x$ -component of the free-stream velocity,  $M_\infty$  is the free-stream Mach number,  $\gamma$  is the specific heat ratio,  $c_p = \gamma R/(\gamma - 1)$  is the specific heat at constant pressure, and  $R$  is the gas constant. The entropy change is measured relative to the freestream,

$$\Delta s \equiv s - s_\infty, \quad s \equiv c_v \ln p - c_p \ln \rho,$$

where  $c_v = R/(\gamma - 1)$  is the specific heat at constant volume. Eqn. 3 will be referred to as the exact Oswatitsch expression.<sup>20,21</sup> It accounts for drag produced either by shocks or by boundary layers. When the far-field boundary is not sufficiently far from the body, correction terms can be added to account for “mid-field” changes in pressure and enthalpy. However, these terms will not be considered in the present work.

Another difference between the drag values obtained using Eqns. 1 and 3 is due to numerical error. While in an exact solution the drag computed using Eqn. 3 is equal, up to the above assumptions, to that computed using Eqn. 1, this thermodynamic equivalence does not necessarily hold for a discrete numerical solution. Specifically, most solution schemes do not conserve entropy, so that the entropy measured downstream of the body is polluted by spurious entropy generation. Eqn. 3 will then suffer the effects of this spurious entropy pollution.

Under the additional assumption that  $\Delta s/R \ll 1$ , a first-order Taylor-series expansion of Eqn. 3 yields the ‘‘approximate Oswatitsch’’ drag expression,<sup>20</sup>

$$D_{\text{osw}} \approx \frac{u_\infty}{\gamma R M_\infty^2} \int_{S_\infty} \Delta s \rho \mathbf{V} \cdot \mathbf{n} dS. \quad (4)$$

In this integral  $\Delta s$  can be replaced by  $s$  since the far-field surface is assumed to be closed and  $s_\infty$  is constant.

Finally, in two dimensions, we define the non-dimensional drag coefficient as

$$c_d = \frac{D}{\frac{1}{2} \rho_\infty |\mathbf{V}_\infty|^2 c},$$

where  $D$  is the drag computed using any of the above formulas,  $\rho_\infty$  is the freestream density,  $\mathbf{V}_\infty$  is the freestream velocity, and  $c$  is a characteristic length of the body. In the present results for an airfoil,  $c$  is the airfoil chord length.

### III. Discretization

We consider the steady compressible Navier-Stokes equations,

$$\nabla \cdot \mathbf{F}^i(\mathbf{u}) - \nabla \cdot \mathbf{F}^v(\mathbf{u}, \nabla \mathbf{u}) = 0, \quad (5)$$

where  $\mathbf{u} = [\rho, \rho \mathbf{V}, \rho E]$  is the conservative state vector and  $\mathbf{F}^i/\mathbf{F}^v$  are the inviscid/viscous fluxes, respectively. In this work we discretize Eqn. 5 using a discontinuous Galerkin (DG) finite element method that employs the Roe approximate Riemann solver<sup>22</sup> for the inviscid flux and the second form of Bassi and Rebay for the viscous flux.<sup>23</sup> The solution is obtained via a Newton-GMRES implicit solver with element-line Jacobi preconditioning and local pseudo-time stepping. While a DG finite element method was used in this work, the conclusions are not strictly tied to the discretization.

The result of the discretization is a system of nonlinear algebraic equations,

$$\mathbf{R}_H(\mathbf{u}_H) = 0, \quad (6)$$

where  $\mathbf{u}_H$  is the discrete solution vector and  $\mathbf{R}_H$  is the discrete residual vector. This will be referred to as the primal system.

The error estimation strategy outlined in the next section relies on output adjoint solutions. In this work, a discrete adjoint  $\psi_H$  corresponding to an output of interest  $J(\mathbf{u}_H)$  is obtained by solving the linearized transpose system,

$$\left( \frac{\partial \mathbf{R}_H}{\partial \mathbf{u}_H} \right)^T \psi_H - \left( \frac{\partial J}{\partial \mathbf{u}_H} \right)^T = 0, \quad (7)$$

where the linearizations are performed about  $\mathbf{u}_H$ . We note that the sign on  $\left( \frac{\partial J}{\partial \mathbf{u}_H} \right)^T$  in Eqn. 7 is arbitrary and is chosen to make the entropy variables correspond to the adjoint for the output defined in Eqn. 12, as derived in.<sup>6</sup> The same element-line Jacobi preconditioned GMRES solver used in the primal solve is used for the linear adjoint solve.

## IV. Drag Error Estimation

In this work we are interested in estimating the numerical error in the calculation of drag from a computational fluid dynamics simulation. We consider two approaches:

- a direct adjoint-based error estimation applied directly to the drag expressions in Section II.
- an indirect error estimation based on the entropy adjoint method.

These approaches are outlined in the following subsections.

### A. Output Adjoint Approach

The discretization discussed in the previous section yields a solution approximation,  $\mathbf{u}_H$ , in some finite dimensional space  $\mathcal{V}_H$ . We are interested in the effect of the solution approximation on the error in a scalar output,  $J(\mathbf{u}_H)$ . In the present work, this output will be the coefficient of drag computed using either Eqn. 1 or Eqn. 3. To make the error estimation tractable, we resolve to compare  $J(\mathbf{u}_H)$  to the output calculated from a “fine” solution,  $\mathbf{u}_h$ , on a richer space  $\mathcal{V}_h$ . In a discontinuous Galerkin discretization, the solution space can be enriched by increasing the approximation order or by refining the elements, and in this work we consider increasing the approximation order from  $p$  in  $\mathcal{V}_H$  to  $p + 1$  in  $\mathcal{V}_h$ .

We denote by  $\mathbf{u}_h^H$  the coarse solution injected into the fine space. There is no additional approximation in this injection as  $\mathcal{V}_H \subset \mathcal{V}_h$ . The fine-space residual computed with  $\mathbf{u}_h^H$  will generally not be zero. The adjoint on the fine space weights this residual to yield an estimate of the output error,<sup>5</sup>

$$\delta J \approx \delta \boldsymbol{\psi}_h^T \mathbf{R}_h(\mathbf{u}_h^H), \quad (8)$$

where  $\delta \boldsymbol{\psi}_h = \boldsymbol{\psi}_h - \boldsymbol{\psi}_h^H$ ,  $\boldsymbol{\psi}_h$  is the fine-space adjoint,  $\boldsymbol{\psi}_h^H$  is the injection of the coarse adjoint into the fine space, and  $\mathbf{R}_h$  is the fine-space residual of the coarse projection. If the discretization is not equipped with Galerkin orthogonality, an additional computable correction term arises from the inner product between  $\boldsymbol{\psi}_h^H$  and the residual.

A critical part of the error estimate in Eqn. 8 is the fine-space adjoint solution,  $\boldsymbol{\psi}_h$ , and this solution can be obtained in several ways. It can be reconstructed via least-squares or solved approximately or exactly in the fine space, linearizing about  $\mathbf{u}_h^H$  or about an exact or approximate  $\mathbf{u}_h$ . We compare an exact adjoint solve about the exact  $\mathbf{u}_h$ , a somewhat expensive proposition, to an approximate adjoint solve about an approximate  $\mathbf{u}_h$ , where the approximate solve consists of  $\nu_{\text{fine}} = 5$  iterations of element-block Jacobi relaxation. The results show that an approximate adjoint solution yields error estimates that are virtually indistinguishable from those obtained with the exact adjoint solve for the cases tested, and this observation is supported by the fact that the error estimate is insensitive to coarse-space errors in the adjoint.<sup>5</sup>

For the drag coefficient based on the near-field calculation in Eqn. 1, we denote the corresponding fine-space output adjoint as  $\boldsymbol{\psi}_{h,\text{near}}$ . For the drag coefficient based on the exact Oswatitsch expression in Eqn. 3, the adjoint is  $\boldsymbol{\psi}_{h,\text{osw}}$ . The error estimates for the drag coefficients when computed with a discrete solution  $\mathbf{u}_H$  are then

$$\text{Near-field drag error} = \delta c_{d,\text{near}} \approx \delta \boldsymbol{\psi}_{h,\text{near}}^T \mathbf{R}_h(\mathbf{u}_h^H), \quad (9)$$

$$\text{Oswatitsch drag error} = \delta c_{d,\text{osw}} \approx \delta \boldsymbol{\psi}_{h,\text{osw}}^T \mathbf{R}_h(\mathbf{u}_h^H). \quad (10)$$

We note that due to discrete conservation of momentum, the adjoint for the far-field drag output in Eqn. 2 is equal to the near-field drag adjoint. However, because entropy is generally not conserved at the discrete level,  $\psi_{h,\text{near}}$  and  $\psi_{h,\text{osw}}$  will be different.

## B. Entropy Adjoint Approach

The Navier-Stokes equations admit an entropy function for which the corresponding entropy variables symmetrize both the inviscid and viscous terms.<sup>24</sup> This entropy function, unique up to additive and multiplicative constants, is

$$U = -\rho s/R, \quad s = c_v \ln p - c_p \ln \rho,$$

where  $p$  is the pressure,  $\rho$  is the density,  $s$  is the entropy, and  $\gamma$  is the ratio of specific heats. Differentiating with respect to the conservative state  $\mathbf{u}$  yields the entropy variables,

$$\mathbf{v} = U_{\mathbf{u}}^T = \left[ \frac{\gamma}{\gamma-1} - \frac{s}{R} - \frac{1}{2} \frac{\rho V^2}{p}, \frac{\rho \mathbf{V}}{p}, -\frac{\rho}{p} \right]^T, \quad (11)$$

where  $V^2 = \mathbf{V} \cdot \mathbf{V}$  is the square of the velocity magnitude. Note that the entropy variables are obtained via a nonlinear transformation of the conservative variables. The corresponding entropy flux is  $\mathbf{F} = \mathbf{V}U = -s\rho\mathbf{V}/R$ .

As shown in,<sup>6</sup> the entropy variables satisfy an adjoint equation for one specific output,

$$J = \underbrace{\int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} dS}_{J^i} - \underbrace{\int_{\Omega} \mathbf{v}^T \nabla \cdot (\mathbf{K} \nabla \mathbf{u}) d\Omega}_{J^v}, \quad (12)$$

where  $\Omega$  is the computational domain with boundary  $\partial\Omega$ , and  $\mathbf{K}$  is the linearization of the viscous flux with respect to the state gradient. This output is an entropy balance statement for the computational domain: the inviscid contribution  $J^i$  represents the net inflow of physical entropy through  $\partial\Omega$ , while the viscous contribution  $J^v$  is the negative total generation (i.e. the destruction) of entropy in  $\Omega$ . For the exact solution,  $J$  will be zero, and  $J^i = J^v$ . However, for an approximate solution  $J$  will generally not be zero due to spurious entropy generation.

We now make the observation that the ‘‘inviscid’’ portion of this output,  $J^i$ , is directly related to the approximate form of Oswatitsch’s drag formula, Eqn. 4,

$$D_{\text{osw}} \approx \frac{u_{\infty}}{\gamma R M_{\infty}^2} \int_{S_{\infty}} s \rho \mathbf{V} \cdot \mathbf{n} dS = -\frac{u_{\infty}}{\gamma M_{\infty}^2} \int_{S_{\infty}} \mathbf{F} \cdot \mathbf{n} dS = -\frac{u_{\infty}}{\gamma M_{\infty}^2} \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} dS = -\frac{u_{\infty}}{\gamma M_{\infty}^2} J^i, \quad (13)$$

where in the second-to-last step the integral of  $\mathbf{F} \cdot \mathbf{n}$  over the body,  $S_{\text{body}}$ , is zero due to no flow through the body, so that the integral over  $S_{\infty}$  is equal to the integral over  $\partial\Omega$ . The drag coefficient is then approximated by

$$c_{d,\text{osw}} = \frac{D_{\text{osw}}}{\frac{1}{2} \rho u_{\infty}^2 c} \approx -\frac{2}{\underbrace{\rho u_{\infty} \gamma M_{\infty}^2 c}_K} J^i. \quad (14)$$

This is a direct relationship between the entropy balance output and an output of engineering interest: the drag coefficient measured using Oswatitsch’s approximate formula. Note that the constant  $K$  depends only on freestream conditions.

The error in the drag coefficient is then approximated by

$$\delta c_{d,\text{osw}} \approx K \delta J^i. \quad (15)$$

In the inviscid case,  $J = J^i$ , and we can directly apply the adjoint-weighted residual formula using the entropy variables as the adjoint,

$$\begin{aligned} \delta J^i = \delta J &\approx \delta \mathbf{v}_h^T \mathbf{R}_h(\mathbf{u}_h^H) \\ \delta c_{d,\text{osw}} &\approx K \delta \mathbf{v}_h^T \mathbf{R}_h(\mathbf{u}_h^H) \quad (\text{inviscid case}) \end{aligned} \quad (16)$$

That is, the output error obtained when using the entropy variables as adjoints corresponds approximately to  $1/K$  times the output error in the drag coefficient computed using Oswatitsch's formula. Since the entropy variables are readily computable from the primal state, this error estimate has a key advantage that it does not require a separate adjoint solution.

The viscous case is more complicated because the output  $J$  in Eqn. 12 consists of both  $J^i$  and  $J^v$ . The adjoint weighted residual gives us the error in  $J$ , but for drag error prediction via Eqn. 15, we are interested in the error in  $J^i$ . We can break up the discrete residual as

$$\mathbf{R}_h(\mathbf{u}_h^H) = \mathbf{R}_h^i(\mathbf{u}_h^H) + \mathbf{R}_h^v(\mathbf{u}_h^H). \quad (17)$$

However, we cannot just replace the residual in Eqn. 16 with the inviscid portion, because  $\mathbf{R}_h^i(\mathbf{u}_h)$  is not in general zero, as required in the derivation of the adjoint-weighted residual error estimate.<sup>5</sup> With the understanding that  $\mathbf{R}_h(\mathbf{u}_h) = 0$ , the complete form of the adjoint weighted residual is

$$\begin{aligned} \delta J &\approx \delta \mathbf{v}_h^T \mathbf{R}_h(\mathbf{u}_h^H) - \delta \mathbf{v}_h^T \mathbf{R}_h(\mathbf{u}_h) \\ &= \underbrace{\delta \mathbf{v}_h^T [\mathbf{R}_h^i(\mathbf{u}_h^H) - \mathbf{R}_h^i(\mathbf{u}_h)]}_{\delta J^i} + \underbrace{\delta \mathbf{v}_h^T [\mathbf{R}_h^v(\mathbf{u}_h^H) - \mathbf{R}_h^v(\mathbf{u}_h)]}_{\delta J^v}. \end{aligned} \quad (18)$$

Thus, the error estimate for  $\delta J^i$  requires two inviscid residual evaluations on the fine space, one with the coarse solution and one with the fine solution. We note that in this work we investigate the performance of the error estimate when  $\mathbf{u}_h$  is solved only approximately, since the error estimate becomes trivial when the exact  $\mathbf{u}_h$  is available. The drag coefficient error estimate becomes

$$\delta c_{d,\text{osw}} \approx K \delta \mathbf{v}_h^T [\mathbf{R}_h^i(\mathbf{u}_h^H) - \mathbf{R}_h^i(\mathbf{u}_h)] \quad (\text{viscous case}) \quad (19)$$

## V. Mesh Adaptation

The output error estimate drives an adaptive process in which the problem is solved multiple times on successively refined meshes. The iterative process begins with a forward solution, and if necessary an adjoint solution, on a coarse mesh. The drag error is estimated using the formulas described in the previous section, and if the error is below a specified tolerance, the iterative process terminates. Otherwise, the drag error is localized and the mesh is refined as described below.

Since the degrees of freedom in the discretization are associated with individual elements, the inner products in Eqns. 9,10,16, and 19 can be written as a sum over elements. For example, in the adjoint case,

$$\delta J \approx \delta \boldsymbol{\psi}_h^T \mathbf{R}_h(\mathbf{u}_h^H) = \sum_k \delta \boldsymbol{\psi}_{h,k}^T \mathbf{R}_{h,k}(\mathbf{u}_h^H) \quad (20)$$

where  $k$  is an element index that, as a subscript, indicates degrees of freedom restricted to that element. The error indicator for element  $k$  is then defined as the absolute value of the contribution to this sum from that element

$$\eta_k = \left| \delta \boldsymbol{\psi}_{h,k}^T \mathbf{R}_{h,k}(\mathbf{u}_h^H) \right|. \quad (21)$$

The elemental adaptive indicator,  $\eta_k$ , drives a fixed-fraction, isotropic, hanging-node adaptation strategy. In this strategy, a fraction  $f^{\text{adapt}} = 0.1$  of the elements with the largest adaptive indicators are marked for refinement. Marked elements are adapted uniformly, with a maximum difference of one level of refinement between adjacent elements.

The steps involved in each adaptation iteration can be summarized as follows:

1. Solve the primal problem on the current mesh at order  $p$  to obtain  $\mathbf{u}_H$ . If adapting using a drag adjoint, solve the adjoint problem to obtain  $\boldsymbol{\psi}_H$ .
2. Inject  $\mathbf{u}_H$  into an order  $p + 1$  space and either solve the primal problem exactly or iteratively smooth  $\nu_{\text{fine}}$  times to obtain  $\mathbf{u}_h$ .
3. If adapting using a drag adjoint, solve or iterate the fine-space adjoint problem to obtain  $\boldsymbol{\psi}_{h,\text{near}}$  or  $\boldsymbol{\psi}_{h,\text{far}}$ . If instead adapting using entropy variables, compute  $\mathbf{v}_h(\mathbf{u}_h)$  using Eqn.11.
4. Calculate the adaptive indicator,  $\eta_k$ , for each element using either the output adjoint or the entropy adjoint approach.
5. Refine a fraction  $f^{\text{adapt}}$  of the elements with the largest indicator.
6. Initialize the solution on the adapted mesh with a projection of  $\mathbf{u}_H$  and return to step 1.

## VI. Results

The following results present two demonstrations of adaptation and error estimation using output adjoints and using the entropy adjoint. The cases are both in two dimensions: the first is inviscid flow, and the second is viscous flow.

### A. Inviscid Flow

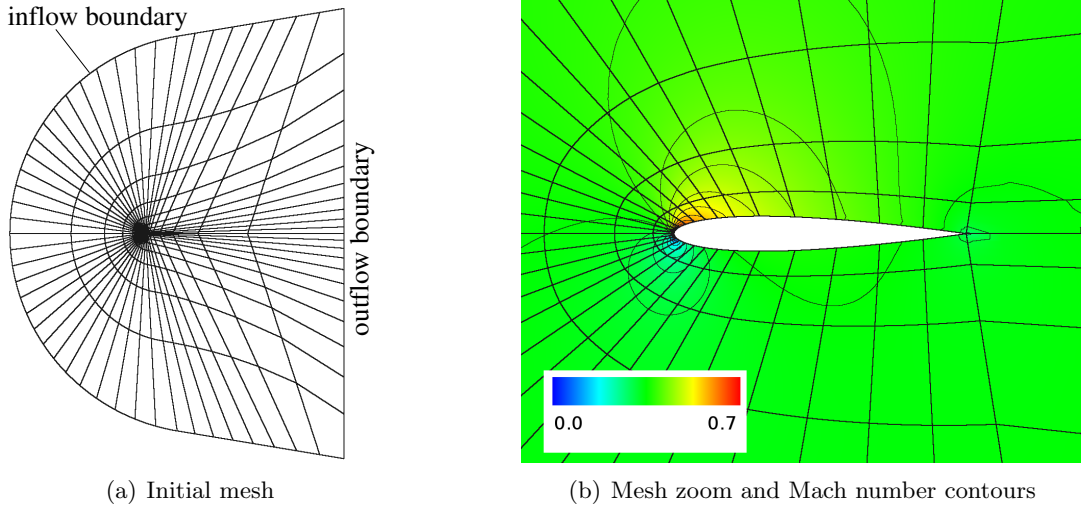
The first example is inviscid, subcritical flow over a NACA 0012 airfoil. Although drag prediction for this flow is of no great engineering consequence, as we expect it to be zero, discretized simulations will yield some spurious drag and the accuracy to which the above methods predict the correct drag error is of interest in verifying the proposed approach.

The airfoil geometry for this example has a closed trailing edge, and the far-field is approximately 100 chord-lengths away from the airfoil. The initial mesh is illustrated in Figure 2, with Mach number contours in the near-field view. This mesh consists of quadrilaterals, with cubic ( $q = 3$ ) elements representing the geometry. Although the initial mesh appears structured, this structure disappears with the first adaptation iteration and the mesh storage is always fully unstructured. For the following results, quadratic solution approximation,  $p = 2$ , is used in the discretization.

The free-stream Mach number for this case is  $M_\infty = 0.4$ , and the angle of attack is  $\alpha = 5^\circ$ . Stagnation quantities and flow direction are specified on the inflow boundary, and static pressure is specified on the outflow boundary. Figure 3 shows the convergence of the drag coefficient calculated using the near-field and Oswatitsch expressions. Three strategies for adaptation are compared:

- adjoint-based adaptation on the near-field drag coefficient,  $c_{d,\text{near}}$ ,





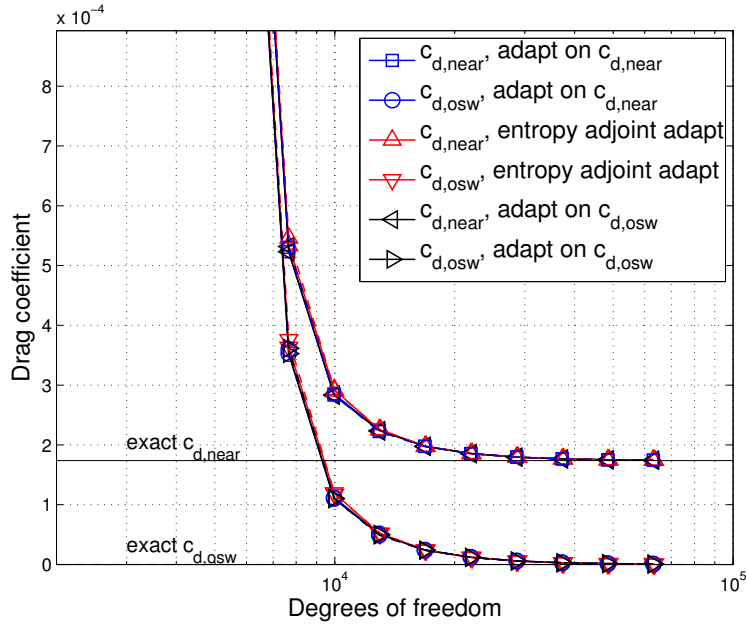
**Figure 2.** NACA 0012,  $M = 0.4$ , inviscid,  $\alpha = 5^\circ$ : far-field and near-field views of the initial mesh for adaptation, together with Mach number contours.

- adjoint-based adaptation on the Oswatitsch drag coefficient,  $c_{d,osw}$ ,
- entropy-adjoint based adaptation.

For each strategy, two adaptive runs are performed: one with an exact fine-space solve (solid lines) and one with an approximate fine-space solve (dashed lines). As shown in Figure 3, there is a difference between the exact values of the two drag coefficients, and this is due to the finite distance of the farfield from the body. The “exact” values are obtained by calculating the outputs on the final output-adapted meshes that have been uniformly refined, using approximation order  $p = 3$ . However, all of the adaptive methods produce nearly identical values for  $c_{d,near}$  and for  $c_{d,osw}$  versus degrees of freedom.

The next question is whether the error estimates correctly predict the true error in the drag coefficients. Figure 4 shows the convergence of the near and Oswatitsch drag coefficient errors with adaptive iteration for each of the three strategies. Several observations deserve attention. First, all of the methods result in monotonic drag coefficient convergence of about 2.5 orders of magnitude  $c_d$  reduction for one order of magnitude increase in degrees of freedom. Second, for all of the strategies, the errors in  $c_{d,near}$  are virtually identical to the errors in  $c_{d,osw}$ , so that an error estimate of the near-field drag error does well for the Oswatitsch drag error, and vice-versa. Third, the entropy adjoint error estimate is nearly as accurate (less than 10% difference) as the error estimates from the output-adjoint strategies.

We emphasize that the entropy adjoint approach requires no separate adjoint solution. However, it does require a fine-space solution for the entropy variables,  $\mathbf{v}_h$ , and hence  $\mathbf{u}_h$ . One could argue that for this effort, the output error is also available as the difference between the output computed with the coarse solution and that computed with the fine solution. To test this hypothesis, Figure 4 also plots precisely this quantity,  $c_d(\mathbf{u}_H) - c_d(\mathbf{u}_h)$ . As shown, if the fine space solution  $\mathbf{u}_h$  is obtained through an exact solve (solid line), this quantity yields an accurate error estimate. However, if  $\mathbf{u}_h$  is obtained only approximately (dashed line), a more reasonable proposition in practice, then  $c_d(\mathbf{u}_H) - c_d(\mathbf{u}_h)$  is no longer an accurate error estimate, with differences of over 1-2 orders of



**Figure 3.** NACA 0012,  $M = 0.4$ , inviscid,  $\alpha = 5^\circ$ : convergence of drag coefficients computed using near ( $c_{d, \text{near}}$ ) and Oswatitsch ( $c_{d, \text{osw}}$ ) drag formulas. Solid/dashed lines (indiscernible difference) indicate exact/approximate fine-space solutions, respectively.

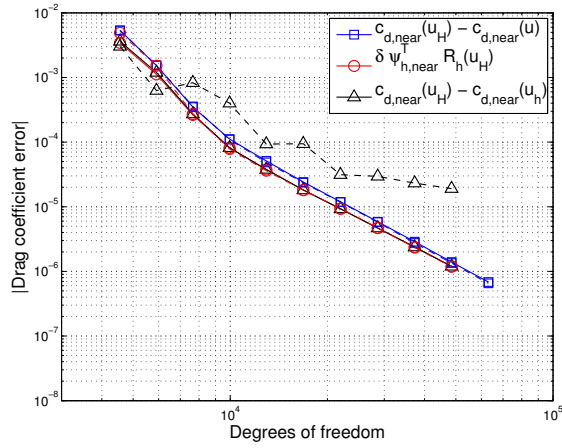
magnitude compared to the true error. In contrast, the adjoint-weighted and entropy-variable weighted residual error estimates perform equally well for the approximate and for the exact fine space solutions. This phenomenon can be explained by the observation that the error estimates rely on  $\delta\psi_h$  or  $\delta\mathbf{v}_h$ , and these quantities often converge with only a few smoothing iterations, in contrast to  $\psi_h$  and  $\mathbf{v}_h$  which generally require a full solve to predict accurately.

Figure 5 shows the final (tenth iteration) adapted meshes from the three strategies. These are shown for the adaptations using the exact fine-space solve, but there was virtually no difference between these meshes and those obtained using the approximate fine-space solve. As shown, for this problem, the three strategies yield nearly identical meshes, indicating that the localized error estimates are targeting the same areas of the computational domain: the leading edge, trailing edge, and upper surface of the airfoil.

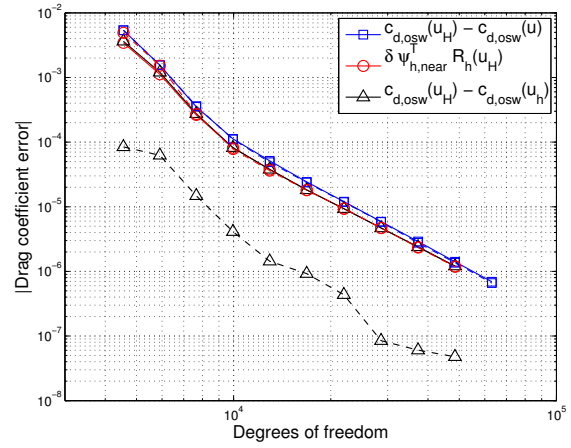
## B. Viscous Flow

The second example consists of a NACA 0012 airfoil in viscous flow at  $M_\infty = 0.5$ ,  $Re = 5000$ , and  $\alpha = 0^\circ$ . The initial mesh and Mach number distribution is illustrated in Figure 6. The farfield boundary is again located at approximately 100 chord-lengths away from the airfoil, and  $p = 1$  solution approximation was used for these results.

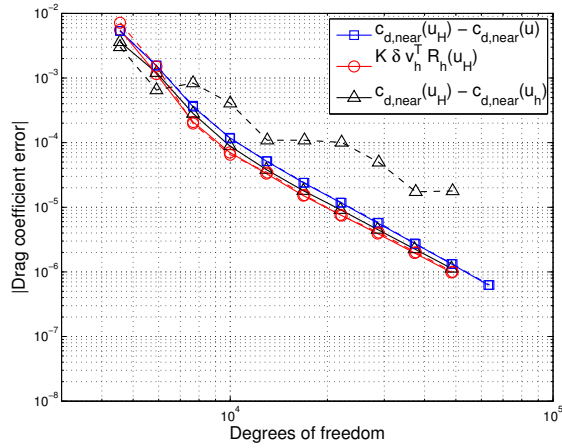
Figure 7 shows the convergence of the drag coefficient calculated using the near-field and Oswatitsch expressions. The three adaptive strategies presented in the previous example are again compared. In addition, runs with approximate and exact fine space solves are compared for each strategy. As shown in Figure 7, there is a difference of about 2 counts between the exact values of the two drag coefficients. This is again due to the finite distance of the farfield from the body, and it could be reduced by introducing additional terms into Eqn. 3.



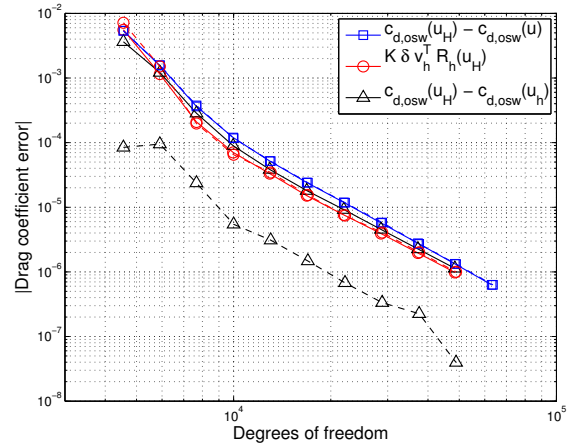
(a) Adapted on  $c_{d,near}$



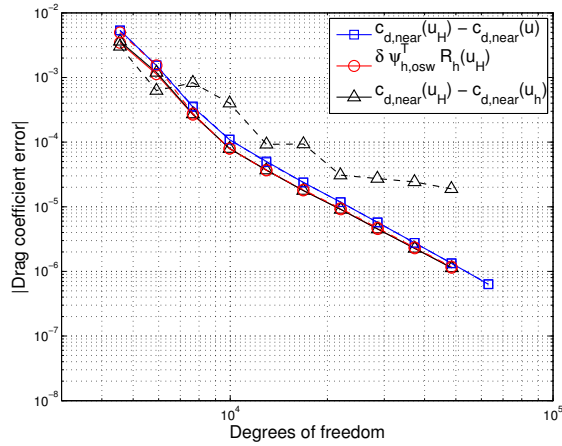
(b) Adapted on  $c_{d,osw}$



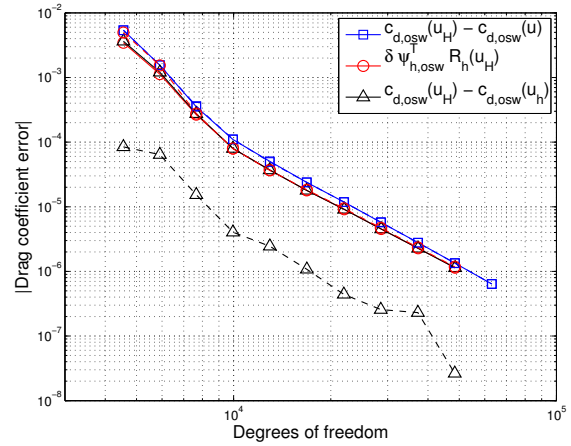
(c) Adapted on entropy adjoint



(d) Adapted on entropy adjoint

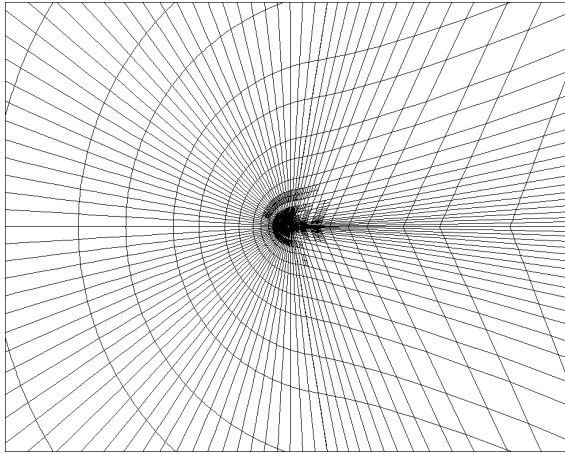


(e) Adapted on  $c_{d,osw}$

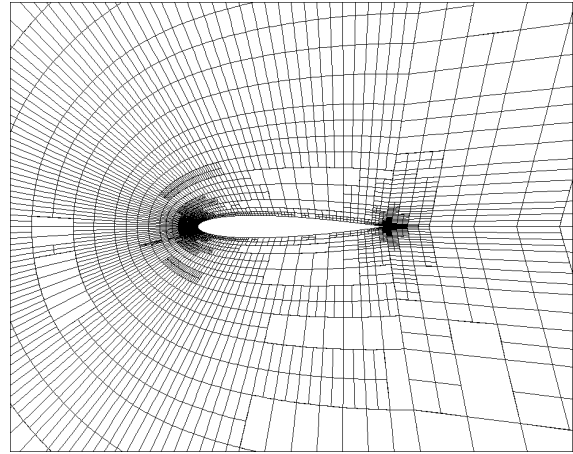


(f) Adapted on  $c_{d,osw}$

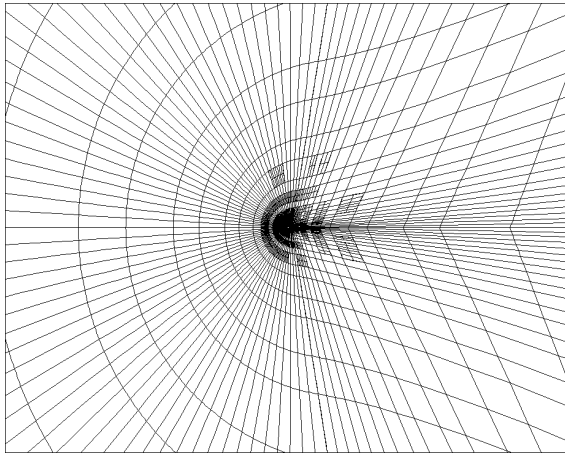
Figure 4. NACA 0012,  $M = 0.4$ , inviscid,  $\alpha = 5^\circ$ : near (left) and Oswatitsch (right) drag coefficient errors for three adaptation strategies. Solid/dashed curves correspond to exact/approximate fine-space primal and adjoint solutions for error estimation.



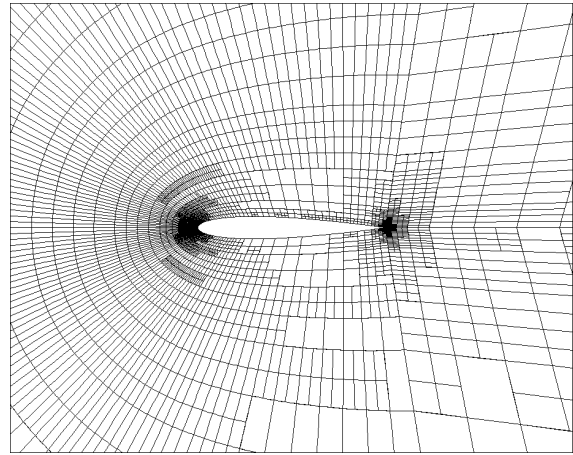
(a) Adapted on  $c_{d,near}$



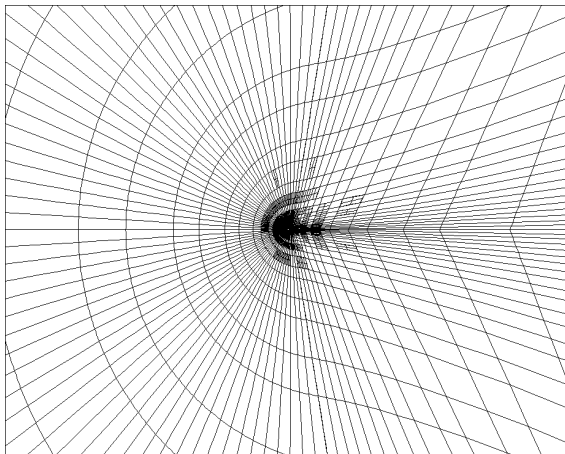
(b) Adapted on  $c_{d,near}$  (zoom)



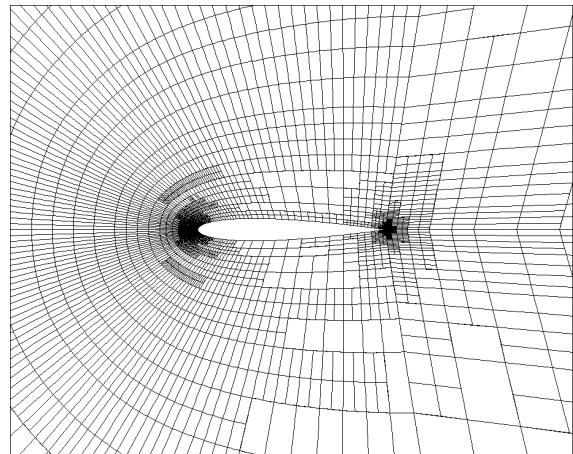
(c) Adapted on entropy adjoint



(d) Adapted on entropy adjoint (zoom)



(e) Adapted on  $c_{d,osw}$



(f) Adapted on  $c_{d,osw}$  (zoom)

**Figure 5. NACA 0012,  $M = 0.4$ , inviscid,  $\alpha = 5^\circ$ : far-field and near-field views of final ( $10^{th}$  iteration) meshes for three adaptation strategies.**

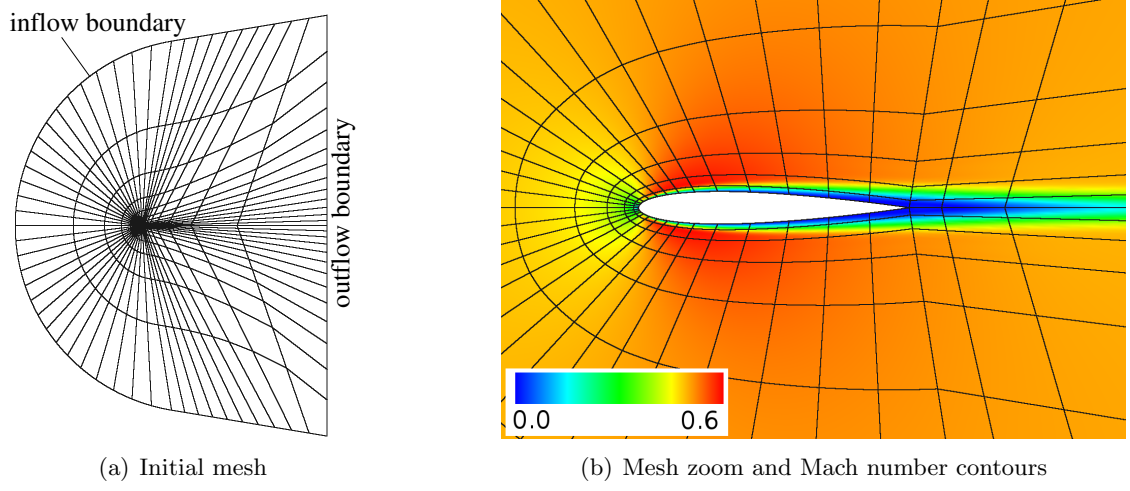


Figure 6. NACA 0012,  $M = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0$ : far-field and near-field views of the initial mesh for adaptation, together with Mach number contours.

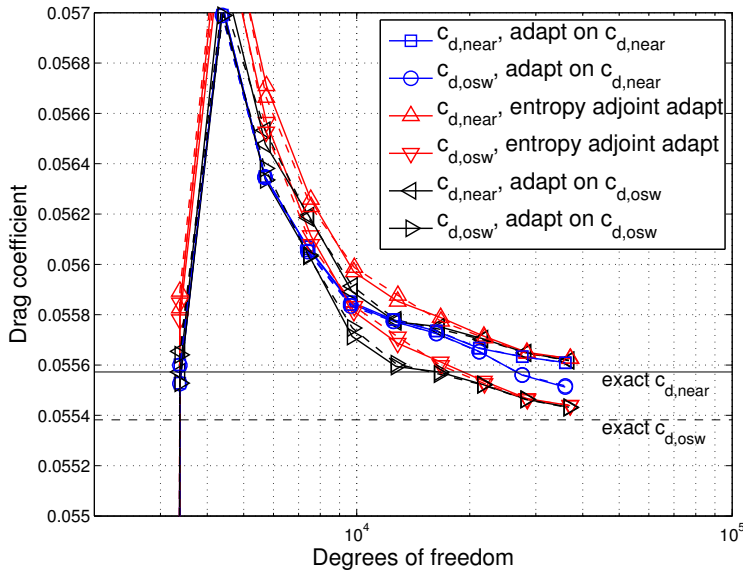


Figure 7. NACA 0012,  $M = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0$ : convergence of drag coefficients computed using near ( $c_{d, near}$ ) and Oswatitsch ( $c_{d, osw}$ ) drag formulas.

In this case, the adaptive strategies no longer produce exactly the same drag convergence histories versus degrees of freedom. The entropy adjoint strategy and adaptation on  $c_{d,osw}$  perform equally well, by the last few iterations, for both outputs. Adaptation on  $c_{d,near}$  performs slightly better for  $c_{d,near}$ , as expected, but significantly worse in the far-field drag,  $c_{d,osw}$ .

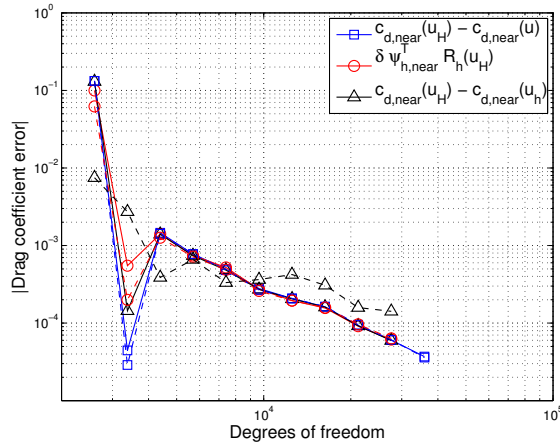
Figure 8 shows the convergence of the near and Oswatitsch drag coefficient errors with adaptive iteration for each of the three strategies. We make several observations. First, as in the previous example, all of the methods yield monotonic drag coefficient convergence after the first few iterations. Second, the Oswatitsch drag error estimate yields an excellent prediction of both the near-field and the far-field drag errors. That is, the error in  $c_{d,near}$  is comparable to the error in  $c_{d,osw}$ , and predicted well using the adjoint for  $c_{d,osw}$  on the resulting adapted sequence of meshes. Third, the drag error estimates provided by the entropy adjoint approach,  $K\delta\mathbf{v}_h^T [\mathbf{R}_h^i(\mathbf{u}_h^H) - \mathbf{R}_h^i(\mathbf{u}_h)]$ , perform very well in predicting the error in  $c_{d,osw}$  which is nearly identical to the error in  $c_{d,near}$ . We note that the simple approach of including the viscous term in the entropy adjoint drag error estimate, i.e. using  $\delta J$  instead of  $\delta J^i$  in Eqn. 15, results in a consistent over-estimation of the output error by a factor of 4-5. Fourth, adapting on the near-field drag coefficient produces the fastest convergence of  $c_{d,near}$ , but the slowest convergence of, and lack of a good error estimate for,  $c_{d,osw}$  (see Figure 8b). This result supports the intuitive conclusion that not all areas of the domain important for the calculation of  $c_{d,osw}$  (e.g. the wake) are important for the calculation of  $c_{d,near}$ . Fifth, the drag coefficient errors computed simply via  $c_d(\mathbf{u}_H) - c_d(\mathbf{u}_h)$  are again seen to be accurate only when the fine space problem is solved exactly, in contrast to the weighted-residual error estimates which remain accurate even if the fine space solution is approximate.

Finally, Figure 9 shows the final (tenth iteration) adapted meshes from the three strategies. Both of the output adjoint strategies allocate some cells to the stagnation streamline region well in front of the airfoil, whereas the entropy adjoint strategy leaves this area relatively coarse. All strategies target the leading edge, boundary layers, and at least some of the wake. The entropy adjoint approach and the adaptation on  $c_{d,osw}$  both target the wake, which is important for accurate entropy propagation to the farfield boundary. We note that the resolution of the wake does not preclude refinement near the airfoil, which is of interest as any spurious entropy production there would pollute the solution downstream. Adaptation on  $c_{d,near}$  does not target a large extent of the wake, and as expected its  $c_{d,osw}$  computations contain the most error.

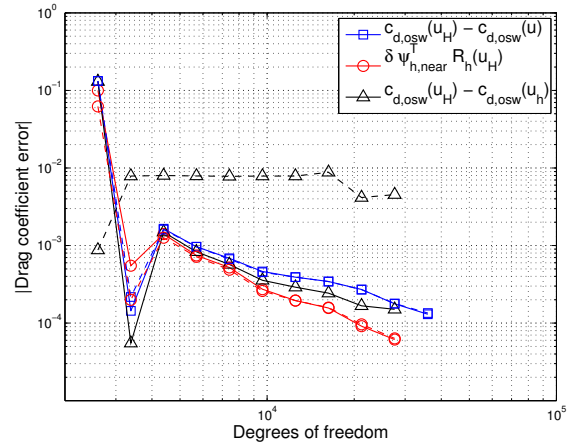
## VII. Conclusions

This paper presents a relationship between the error estimate computed in the entropy-adjoint approach and the drag error in subcritical, two-dimensional simulations of the Euler and Navier-Stokes equations. The relationship holds for the particular case of far-field drag calculated via an integration of entropy, originally due to Oswatitsch. Specifically, this calculation casts the momentum integration on the far-field boundary in terms of the local entropy and free-stream quantities. We relate the leading order term in this drag expression to the entropy balance output available from the entropy adjoint approach. In the inviscid case, the relationship is direct as the entropy adjoint output is an integral of the entropy flux through the domain boundary. In the viscous case, the inviscid residual term has to be treated separately from the viscous residual term to produce an accurate drag error estimate.

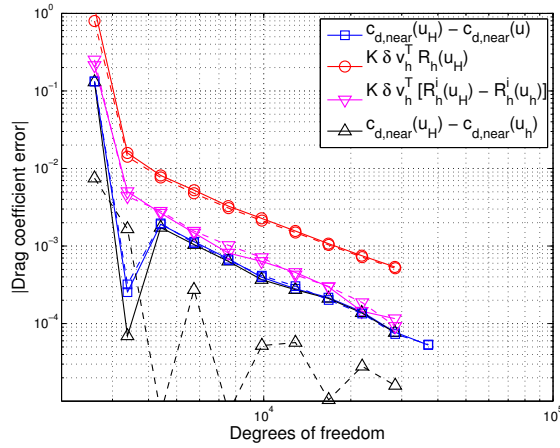
In the two cases tested, the accuracy of the drag error estimate derived from the entropy adjoint approach is on par with that from the output adjoints. Even though the entropy adjoint output is associated with a far-field drag measurement, the near-field drag calculations on the entropy



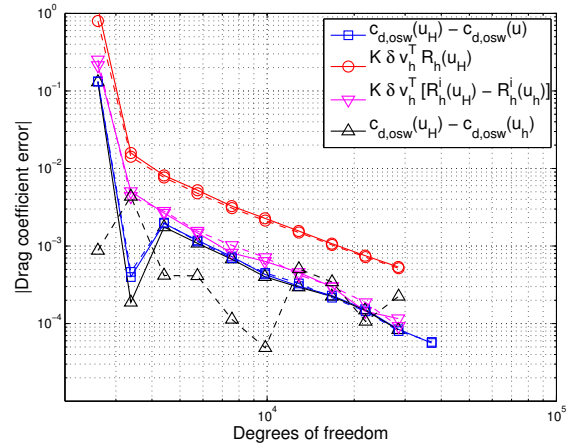
(a) Adapted on  $c_{d,near}$



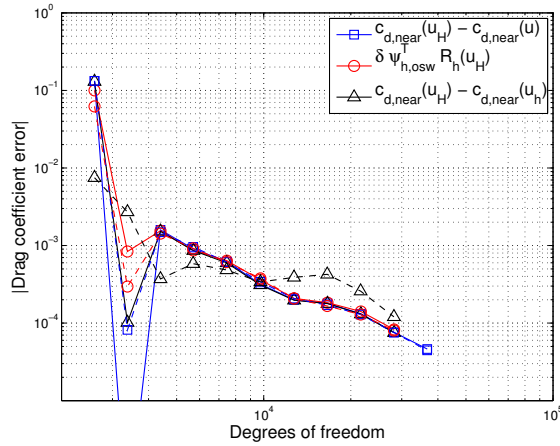
(b) Adapted on  $c_{d,near}$



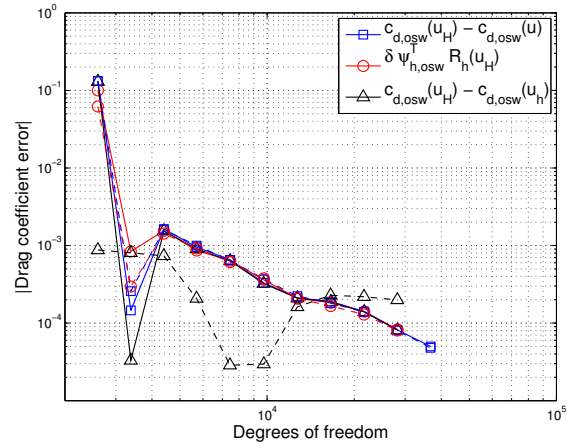
(c) Adapted on entropy adjoint



(d) Adapted on entropy adjoint

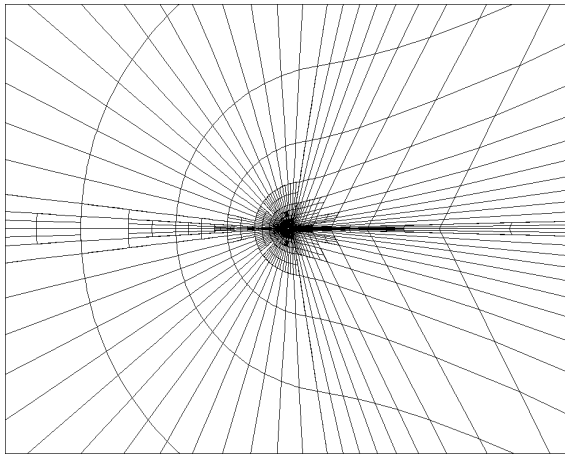


(e) Adapted on  $c_{d,osw}$

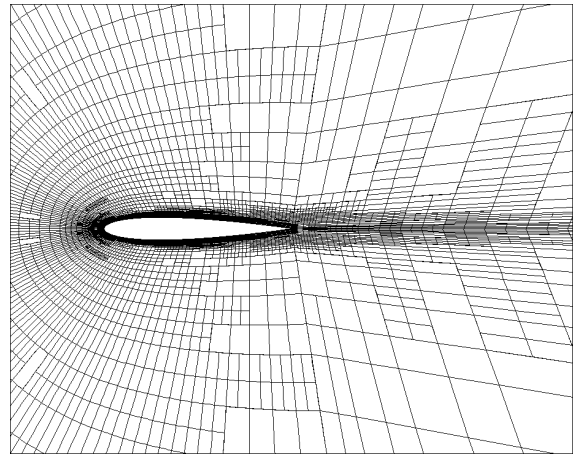


(f) Adapted on  $c_{d,osw}$

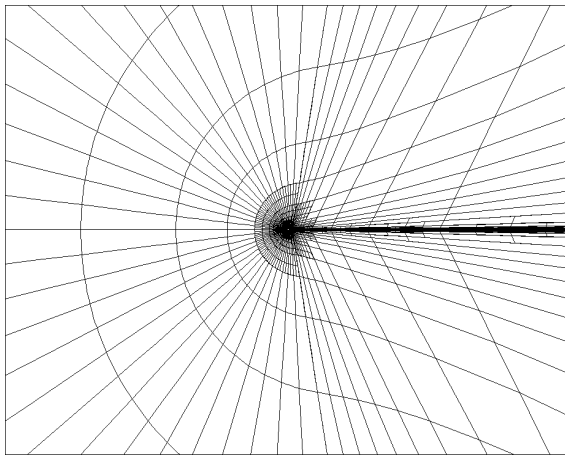
Figure 8. NACA 0012,  $M = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0$ : near (left) and Oswatitsch (right) drag coefficient errors for three adaptation strategies. Solid/dashed curves correspond to exact/approximate fine-space primal and adjoint solutions for error estimation.



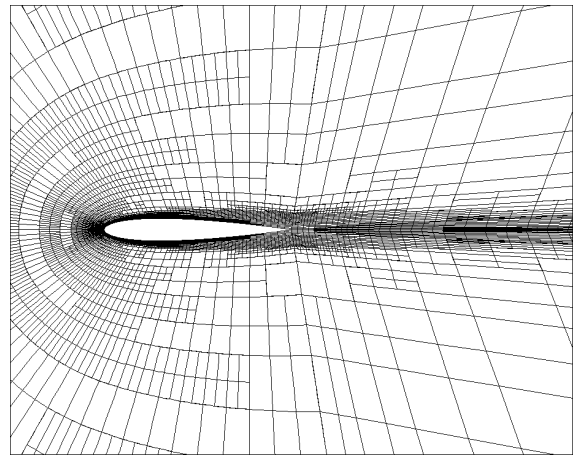
(a) Adapted on  $c_{d,near}$



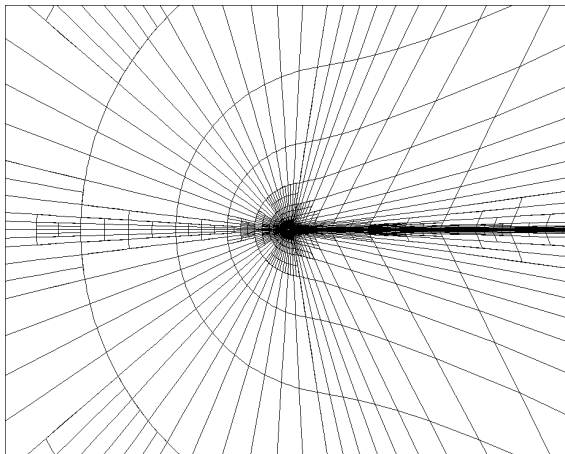
(b) Adapted on  $c_{d,near}$  (zoom)



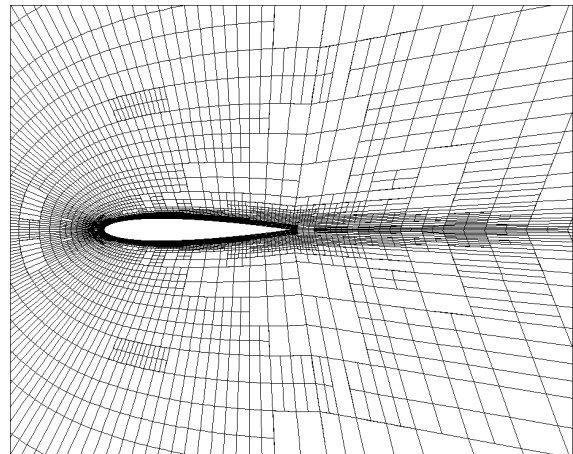
(c) Adapted on entropy adjoint



(d) Adapted on entropy adjoint (zoom)



(e) Adapted on  $c_{d,osw}$



(f) Adapted on  $c_{d,osw}$  (zoom)

**Figure 9. NACA 0012,  $M = 0.5$ ,  $Re = 5000$ ,  $\alpha = 0$ : far-field and near-field views of final ( $10^{th}$  iteration) meshes for three adaptation strategies.**



adjoint adapted meshes show nearly the same error as the far-field calculations. This observation can be reasoned by noting that the entropy adjoint approach targets areas of spurious entropy production, and in a conservative scheme it is the spurious entropy production that is responsible for the difference in the near-field and Oswatitsch drag evaluations. Although the proposed entropy adjoint approach uses a fine space state solution,  $\mathbf{u}_h$ , the drag error estimate is shown to perform well when  $\mathbf{u}_h$  is calculated approximately, e.g. through inexpensive smoothing, in contrast to a simple difference between the coarse-solution drag and one calculated with the approximate  $\mathbf{u}_h$ . Most importantly, the adaptation and drag error estimation using the entropy adjoint approach does not require adjoint solution capability in the code, as the required entropy variables are computed directly from the conservative state vector.

The proposed drag error estimate could be applied to two dimensional, viscous simulations to yield a useful stopping criterion for adaptation with the entropy adjoint approach. An area of future work is extension to three dimensions, where induced drag plays a key role and drag decomposition using a farfield approach becomes desirable. The applicability of this approach to flows with shocks will also be of interest, especially in the transonic regime.

## References

- <sup>1</sup>Pierce, N. A. and Giles, M. B., “Adjoint recovery of superconvergent functionals from PDE approximations,” *SIAM Review*, Vol. 42, No. 2, 2000, pp. 247–264.
- <sup>2</sup>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.
- <sup>3</sup>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.
- <sup>4</sup>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.
- <sup>5</sup>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.
- <sup>6</sup>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.
- <sup>7</sup>van Dam, C. and Nikfetrat, K., “Accurate prediction of drag using Euler methods,” AIAA Paper 91-0038, 1991.
- <sup>8</sup>Giles, M. B. and Cummings, R. M., “Wake integration for three-dimensional flowfield computations: Theoretical development,” *Journal of Aircraft*, Vol. 36, No. 2, 1999, pp. 357–365.
- <sup>9</sup>Giles, M. B. and Cummings, R. M., “Wake integration for three-dimensional flowfield computations: Applications,” *Journal of Aircraft*, Vol. 36, No. 2, 1999, pp. 357–365.
- <sup>10</sup>Chao, D. and van Dam, C., “Airfoil drag prediction and decomposition,” *Journal of Aircraft*, Vol. 36, No. 4, 1999, pp. 675–681.
- <sup>11</sup>D.Destarac, “Far-field/near-field drag balance and applications of drag extraction in CFD,” *von Karman Institute for Fluid Dynamics, Lecture Series 2003-02*, edited by H. Deconinck, K. Sermeus, and C. V. Dam, Rhode-Saint-Genèse, Belgium, Feb. 3-7, 2003.
- <sup>12</sup>Paparone, L. and Tognaccini, R., “Computational fluid dynamics-based drag prediction and decomposition,” *AIAA Journal*, Vol. 41, No. 9, 2003, pp. 1647–1657.
- <sup>13</sup>Chao, D. and van Dam, C., “Wing drag prediction and decomposition,” *Journal of Aircraft*, Vol. 43, No. 1, 2006, pp. 82–90.
- <sup>14</sup>ZiQiang, Z., XiaoLu, W., Jie, L., and Zhou, L., “Comparison of predicting drag methods using computational fluid dynamics in 2d/3d viscous flow,” *Science in China Series E: Technological Sciences*, Vol. 50, No. 5, 2007, pp. 534–549.
- <sup>15</sup>Yamazaki, W., Matsushima, K., and Nakahashi, K., “Drag prediction, decomposition and visualization in unstructured mesh CFD solver of TAS-code,” *International Journal for Numerical Methods in Fluids*, Vol. 57, 2008, pp. 417–436.
- <sup>16</sup>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.

<sup>17</sup>Lafin, K. R., Vassberg, J. C., Wahls, R. A., Morrison, J. H., Brodersen, O., Rakowitz, M., Tinoco, E. N., and Godard, J.-L., “Summary of data from the second AIAA CFD drag prediction workshop,” AIAA Paper 2004-0555, 2004.

<sup>18</sup>Morrison, J. H. and Hensch, M. J., “Statistical analysis of CFD solutions from the third AIAA drag prediction workshop,” AIAA Paper 2007-254, 2007.

<sup>19</sup>Frink, N. T., “Test case presentations from the 4th AIAA drag prediction workshop,” NASA Langley, 2010, <http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop4/presentations/DPW4Presentations.htm>.

<sup>20</sup>Oswatitsch, K., *Gas Dynamics*, Academic, New York, 1956.

<sup>21</sup>Masson, C., Veilleux, C., and Paraschivoiu, I., “Airfoil wave-drag prediction using Euler solutions of transonic flows,” *Journal of Aircraft*, Vol. 35, No. 5, 1998, pp. 748–753.

<sup>22</sup>Roe, P. L., “Approximate Riemann solvers, parameter vectors, and difference schemes,” *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

<sup>23</sup>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.

<sup>24</sup>Hughes, T. J. R., Franca, L. P., and Mallet, M., “A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics.” *Computer Methods in Applied Mechanics and Engineering*, Vol. 54, 1986, pp. 223–234.