

Output-Driven Anisotropic Mesh Adaptation for Viscous Flows using Discrete Choice Optimization

Marco Ceze* and Krzysztof J. Fidkowski†

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

This paper presents a mesh adaptation scheme for direct minimization of output error using a selection process for choosing the optimal refinement option from a discrete set of choices. The scheme is geared for viscous aerodynamic flows, in which solution anisotropy makes certain refinement options more efficient compared to others. No attempt is made, however, to measure the solution anisotropy directly or to incorporate it into the scheme. Rather, mesh anisotropy arises naturally from the minimization of a cost function that incorporates both an output error estimate and a count of the additional degrees of freedom for each refinement option. The method is applied to output-based adaptive simulations of the laminar and Reynolds-averaged compressible Navier-Stokes equations on body-fitted meshes in two and three dimensions. Two-dimensional results for laminar flows show a factor of 2-3 reduction in the degrees of freedom on the final adapted meshes when the discrete choice optimization is used compared to pure isotropic adaptation. Preliminary results on a wing-body configuration show that these savings improve in three dimensions and for higher Reynolds-number flows.

I. Introduction

Improvements in computational power and numerical methods have solidified the foothold of Computational Fluid Dynamics (CFD) in the engineering community. CFD simulations boast rapid startup and turnaround times for new configurations, and they allow for test conditions that may be difficult to produce experimentally. Yet compared to experimental results, many CFD answers are still treated with skepticism, or at least caution. Arguably, such caution is well-founded, considering, for example, the spread of results for representative aerodynamic geometries in the AIAA drag prediction workshops.¹⁻³ Although the accuracy of CFD solutions is improving with increases in computational power, the grid sizes currently used to approach acceptable engineering solutions on representative geometries still prohibit CFD from being used as a high-fidelity design tool.

With the growth in complexity of CFD configurations, managing the liability of accurate solutions is no longer possible solely at the user level. A robust approach to managing this liability is through output-based error estimation, which has already been demonstrated for many complex problems, including those in aerospace applications.⁴⁻⁹ The goal of these methods is to provide

*Graduate Research Assistant, AIAA Member

†Assistant Professor, AIAA Member

confidence measures in the form of error bars for scalar output quantities of engineering interest. Moreover, the theory behind output error estimation allows for the attribution of output error contributions to different elements or volumes of the computational mesh. The resulting adaptive indicator can then be used to drive mesh adaptation that specifically targets the output of interest.^{7,8,10-12} Although output error estimation techniques require more computation, the resulting adapted meshes yield cost savings, in terms of mesh size for a given output accuracy, that generally more than offset the cost of the additional computation.

One challenge for mesh adaptive techniques is creating the least expensive mesh that attains the desired goal. The optimality of the mesh depends on various factors, including the refinement strategy^{8,13} and the available adaptation mechanics. An additional important ingredient for aerodynamic computations is the ability to generate stretched elements in areas where the solution exhibits anisotropy; that is, variations of disparate magnitudes in different directions. These areas include boundary layers, wakes, and shocks. The disparity of length scales in these areas is such that stretching ratios in the hundreds or thousands are common. Using isotropic elements instead quickly becomes prohibitive for such simulations, especially in three dimensions.

The dominant method for detecting anisotropy in aerospace applications has relied on estimates of the directional interpolation error of a representative scalar, such as the Mach number.^{14,15} When used alone, this technique reduces to equidistributing the interpolation error of the chosen scalar over the computational domain, with the absolute level of interpolation error prescribed by the user.^{16,17} Alternately, this technique can be combined with output-based error estimation by using the output adaptive indicator to set the element size and the directional interpolation error to set the element stretching.^{7,18} The same idea can be extended to high-order discretizations,^{19,20} although the measurement of directional interpolation error becomes more tedious. A more fundamental problem with this approach in the context of output-based adaptation is the assumption that mesh anisotropy should be governed by the directional interpolation error of one scalar quantity. This assumption is heuristic because it does not take into account the process by which interpolation errors create residuals that affect the output of interest. As a result, recent research has turned to adaptation algorithms that directly target the output error.

Formaggia *et al.*²¹⁻²³ combine Hessian-based interpolation error estimates with output-based *a posteriori* error analysis to arrive at an output-based error indicator that explicitly includes the anisotropy of each element. Schneider and Jimack²⁴ calculate the sensitivities of the output error estimate with respect to node positions and formulate an optimization problem to reduce the output error estimate by redistributing the nodes. They then combine this node repositioning with isotropic local mesh refinement sequentially in a hybrid optimization/adaptation algorithm. Park²⁵ introduces an algorithm that directly targets the output error through local mesh operators of element swapping, node movement, element collapse, and element splitting. Using the output error indicator to rank elements and nodes, these operations are performed in sequence and automatically result in mesh anisotropy.

Following a similar approach presented by Houston *et al.*,²⁶ in this work we present a direct mesh optimization technique in which a particular mesh refinement is chosen from a discrete number of possible choices in a manner that directly targets the reduction of the output error. This strategy is specifically suited for hanging-node meshes, in which a handful of refinement options are typically available for each element and in which the adaptation mechanics are relatively simple. In the interest of preserving element shape quality, we consider only quadrilateral and hexahedral meshes. Although this choice is more restrictive compared to general triangular and tetrahedral meshes,

many body-fitted quadrilateral and hexahedral meshes already exist as these are the predominant element types for high Reynolds number viscous flows. As such, our goal is to apply the direct optimization technique to practical aerodynamic flows to gauge the importance of anisotropy in output-driven mesh adaptation.

The structure of this paper is as follows. Section II presents the output-based error estimation framework that drives the mesh adaptation. Section III then introduces the discrete-choices direct optimization strategy. Implementation details are discussed in Section IV, followed by representative results in Section V. We conclude in Section VI with discussions of extensions and ongoing work.

II. 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, $\psi(x) \in \mathcal{V}$, 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 (PDE): determine $\mathbf{u} \in \mathcal{V}$ such that

$$\mathcal{R}(\mathbf{u}, \mathbf{w}) = 0, \quad \forall \mathbf{w} \in \mathcal{V}, \quad (1)$$

where $\mathcal{R}(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ is a semilinear form, the adjoint $\psi \in \mathcal{V}$ is the sensitivity of J to an infinitesimal source term $\delta \mathbf{r} \in \mathcal{V}$ added to the left-hand side of the original PDE,

$$\delta J = (\delta \mathbf{r}, \psi), \quad (2)$$

where (\cdot, \cdot) is an inner product over the computational domain, Ω . The output adjoint equation can be derived by linearizing Eqn. 1 to relate infinitesimal state perturbations to infinitesimal residual perturbations, and by requiring the sensitivity property in Eqn. 2 to hold. The result is the adjoint equation for ψ ,

$$\mathcal{R}'[\mathbf{u}](\mathbf{w}, \psi) + J'[\mathbf{u}](\mathbf{w}) = 0, \quad \forall \mathbf{w} \in \mathcal{V}, \quad (3)$$

where the primes denote Fréchet 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.²⁷

An adjoint solution can be used to estimate the numerical error in the corresponding output of interest. The error estimation process, termed the adjoint-weighted residual method, is based on the following observations:

- An approximate solution \mathbf{u}_H in a finite-dimensional approximation space $\mathcal{V}_H \subset \mathcal{V}$ will generally not satisfy the original PDE. Instead it will satisfy a perturbed PDE whose weak form reads: find $\mathbf{u}' \in \mathcal{V}$ such that

$$\mathcal{R}(\mathbf{u}', \mathbf{w}) + (\delta \mathbf{r}, \mathbf{w}) = 0, \quad \forall \mathbf{w} \in \mathcal{V}, \quad \text{where } (\delta \mathbf{r}, \mathbf{w}) = -\mathcal{R}(\mathbf{u}_H, \mathbf{w}).$$

- The adjoint $\psi \in \mathcal{V}$ translates the residual perturbation to an output perturbation via Eqn. 2:

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

This output perturbation quantifies the numerical error in the output when the output is computed with the approximate solution \mathbf{u}_H as compared to when it is computed with the exact solution \mathbf{u} . For non-infinitesimal perturbations, the above expression is approximate and yields an estimate of the numerical error.

Although the continuous solution \mathbf{u} is not required directly, the continuous adjoint ψ must be approximated to make the error estimate in Eqn. 4 computable. In practice, ψ_h is solved approximately or exactly on a finer finite-dimensional space $\mathcal{V}_h \supset \mathcal{V}_H$.^{28–30} This finer space can be obtained either through mesh subdivision or approximation order increase. The procedure for obtaining a consistent set of discrete adjoint equations is described in.^{31–33}

The adjoint-weighted residual evaluation in Eqn. 4 can be localized to yield an adaptive indicator consisting of the relative contribution of each element to the total output error. Working with $\psi_h \in \mathcal{V}_h$, the output perturbation in Eqn. 4 is approximated as

$$\delta J \approx - \sum_{\kappa_H \in T_H} \sum_{\kappa_h \in \kappa_H} \mathcal{R}_h(\mathbf{u}_H, \psi_h|_{\kappa_h}), \quad (5)$$

where T_H is the coarse mesh triangulation, κ_H/κ_h is an element of the coarse/fine triangulation respectively, and $|_{\kappa_h}$ refers to restriction to element κ_h . Note that the coarse and fine spaces can consist of the same triangulations, in which case $\kappa_H = \kappa_h$. Eqn. 5 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. 5,^{4,6,29,34,35}

$$\eta_{\kappa_H} = \left| \sum_{\kappa_h \in \kappa_H} \mathcal{R}_h(\mathbf{u}_H, \psi_h|_{\kappa_h}) \right|. \quad (6)$$

In this work a discontinuous Galerkin finite element discretization is employed and Eqn. 6 is used directly to compute the error indicator. For non-variational discretizations, such as the finite volume method, ψ_h in Eqn. 6 is replaced with $\psi_h - \psi_H$, where ψ_H is an adjoint solution computed with the coarse discretization, in order to target the remaining error in the output.²⁷ The two approaches are identical in a variational formulation with local Galerkin orthogonality. For systems of equations, indicators can be 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.

III. Anisotropic Mesh Adaptation

The elemental adaptive indicator, η_{κ_H} , drives a fixed-fraction hanging-node adaptation strategy. In this strategy, 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 refinement directions. For quadrilaterals, the discrete options are: x -refinement, y -refinement and xy -refinement, as depicted in Figure 1. Note, x and y refer to reference-space coordinates of elements

that can be arbitrarily oriented and curved in physical space. In three dimensions a hexahedron can be refined in seven ways: three single-plane cuts, three double-plane cuts, and isotropic refinement.

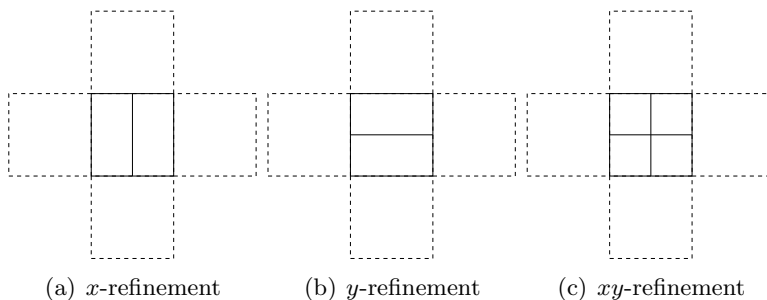


Figure 1. Quadrilateral refinement options. The dashed lines indicate the neighbors of the refined element.

Since the refinement is performed in an element’s reference space, 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 high enough order of geometry interpolation for 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.

Note that elements created in a hanging-node refinement can be marked for refinement in subsequent adaptation iterations. In this case, neighbors will be refined to keep one level of refinement difference between adjacent cells. This is illustrated in Figure 2.

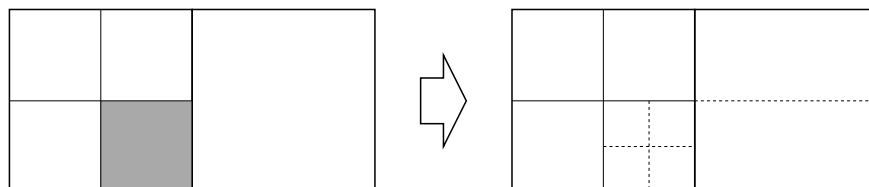


Figure 2. Hanging-node adaptation for a quadrilateral mesh, with a maximum of one level of refinement separating two elements. The shaded element on the left is marked for refinement, and the dashed lines on the right indicate the additional new edges formed.

The choice of a particular refinement option is made locally in each element flagged for refinement. This choice is made by defining a cost function \mathcal{C} that ranks each available refinement option. This function may be generally defined as:

$$\mathcal{C}(i) = \frac{C(i)}{B(i)}, \quad (7)$$

where C and B respectively correspond to measures of the computational cost and the benefit of a refinement option indexed by i . These measures are dependent on the method used for solving the

flow equations and they should be tailored for each specific solver.

In this work, a backwards Euler formulation is used for pseudo-time evolution to steady state in conjunction with a generalized minimal residual (GMRES) linear solver. For the stiff problems considered, the computational effort is found to be dominated by matrix-vector multiplications in the linear solves. The number of operations in these multiplications is proportional to the number of nonzero entries in the residual Jacobian matrix. Assuming an element-wise compact stencil, the number of nonzero elemental blocks in the Jacobian is given by the sum of the number of elements and (two times) the number of internal faces in the mesh. We use this number as the definition of the computational cost, C . Note that this definition does not take into account possible sparsity within the element blocks, as we do not take advantage of any such sparsity in our implementation.

The benefit B of a given refinement option is an assessment of the accuracy of a solution. Similarly to the computational cost evaluation, the benefit is dependent on the method used for solving the flow equations and on the specific application. In general, B should be inversely proportional to some measure of numerical error in the solution. This could be a residual norm, a global interpolation error estimate, or an output error estimate. In this work, the latter is used in the form of an adjoint-weighted residual calculation.

The cost C of each discrete refinement option, in terms of additional elements and internal faces, is straightforward to compute. On the other hand, the error reduction must be estimated, since it is not practical to solve the global system of equations for every refinement option for every element. This error estimation is performed by considering local sub-problems for each element, as proposed in.²⁶

Specifically, local mesh and data structures are created that include the flagged element and its first-level neighbors along with the corresponding coarse and fine primal and adjoint states used in the error estimation. These structures are then used in local sub-problems to approximately solve for the new state in the flagged element upon refinement, assuming that the solution on the first-level neighbors does not change. This assumption is key to making the computation tractable and efficient; it could be relaxed with a wider mesh stencil for the local problems, at a corresponding increase in computational cost.

In the local sub-problems, the central element is refined in turn according to each of the discrete options. On the refined local mesh, one iteration of an element block-implicit Jacobi solver is used to update the coarse primal and adjoint states. These updated states are used as approximations to the solutions that would result if the global system were solved again after refining the element. Note, however, that for the purpose of error estimation, the fine-space adjoint state, ψ_h , is not updated. Instead, it is kept unaltered from the global data structure so that it is possible to establish a common basis for comparison between the various refinement options. In contrast, if different ψ_h were used for each refinement option, the error estimates would be influenced by the different resolutions of the fine space adjoint. Various choices are still allowable for the fine space, \mathcal{V}_h , including order enrichment and mesh refinement.

With the approximate primal and fine-space adjoint solutions available in the local data structure, it is possible to compute the adaptive indicator for each element in the sub-mesh, using the adjoint-weighted residual approach. The adaptive indicator is a surrogate for the indicator that would result from a global solve on the adapted mesh, since the local primal state is only computed approximately.

The final form of the cost function may now be defined in the local sub-problems as:

$$\mathcal{C}_{\kappa_H}(i) = \underbrace{(n_{\kappa_H}(i) + f_{\kappa_H}(i))}_{C(i)} \underbrace{\sum_{\kappa_r \in T_{\kappa_H}(i)} \eta_{\kappa_r}}_{1/B(i)}, \quad (8)$$

where κ_H is an element of the original mesh, i indexes the refinement options for that element, $n_{\kappa_H}(i)$ and $f_{\kappa_H}(i)$ are respectively the number of additional sub-elements and the number of additional internal faces that are created for refinement option i , $T_{\kappa_H}(i)$ is the set of sub-elements, κ_r is one of the sub-elements, and η_{κ_r} is the adaptive indicator on κ_r . This cost function is constructed such that it balances between the extremes of high-cell-count-low-error and low-cell-count-high-error solutions. In particular, the product of $n_{\kappa_H}(i)$ with the sum of the adaptive indicator penalizes the refinement options which produce numerous elements but typically present low output error. Conversely, low-cell-count options which refine the element in non-optimal directions most likely will not tackle the output error correctly, and therefore, these options will be penalized by a high adaptive indicator sum.

It is worth to emphasize, however, that the definition of the cost function in Eqn. 8 is not unique and that any function that is characterized by the general form of Eqn. 7 will feature the characteristics mentioned above. In addition, an exponent could be included on the benefit measure to reflect an *a priori* expected convergence rate of the indicator. However, the validity of this exponent would rely on the solution being in the asymptotic regime, and on no unaccounted singularities. As such, no exponent was included in this work, although the sensitivity of the results to variations in the cost function definition is a topic of ongoing research.

With a cost function defined, a simple optimization problem can be formulated as the minimization of $\mathcal{C}_{\kappa_H}(i)$ among all the refinement options i . The solution to this problem selects the refinement option that is a trade-off between low cell count and low output error contribution. In regions of solution anisotropy, a directional refinement may have nearly the same output error measure as isotropic refinement, but at a cost of only two sub-elements and one internal face compared to four elements and four internal faces, in two dimensions. We stress however that no *a priori* measure of anisotropy is required; an anisotropic refinement option will simply be recognized as the most efficient choice from the cost function perspective. For example, Figure 3 presents an example of the cost function evaluated at a boundary-layer element on a flat plate, using drag as the output of interest. It is seen in this figure that the y -refinement option is optimal as expected from the nature of this type of fluid flow.

The method for selecting an anisotropic refinement option presented in this paper is similar that presented in Houston *et al*²⁶ for quadrilateral meshes. Houston *et al* use a heuristic in which the sum of the sub-element error indicators is computed for each refinement option, and in which the ratio, θ , of the maximum to minimum sum is used to make the decision of adapting isotropically or in one direction. Anisotropy is only deemed important when θ is larger than a user-prescribed threshold, for which a value of 3 is found to work well. The method proposed in the present work differs from that approach in that it does not employ such a user-prescribed parameter.

IV. Implementation

The adaptive indicator in Eqn. 6 was implemented in a discontinuous Galerkin (DG) finite element code and was used to drive a fixed-fraction hanging-node mesh adaptation strategy. The DG

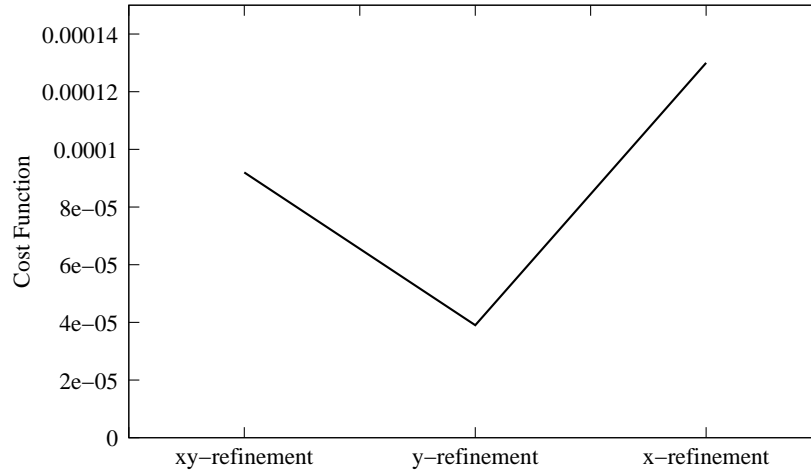


Figure 3. Cost function at a boundary-layer element on a flat-plate computed with the drag output.

discretization of the compressible Navier-Stokes equations employs the Roe approximate Riemann solver³⁶ for the inviscid fluxes and the second form of Bassi and Rebay,³⁷ BR2, for the viscous discretization. For turbulent cases, the Spalart-Allmaras (SA) closure model is used, discretized as described by Oliver.³³ The steady-state solution is obtained via a Newton-GMRES implicit solver with line-Jacobi preconditioning and backward Euler with local time stepping for improved robustness during initial iterations.

A discrete adjoint solution for an engineering output of interest was obtained by solving one additional linear system corresponding to the discrete version of Eqn. 3. Since the full Jacobian matrix is calculated and stored in the Newton-GMRES primal solve, the additional complexity of the transpose solve for the adjoint problem was minimal. The same line-Jacobi preconditioned GMRES solver was used for the adjoint solve. Careful attention was given to the various discretization and output calculation terms to ensure adjoint consistency^{31–33} for the laminar discretization. For the turbulence model, the source terms were discretized in a dual-inconsistent manner for simplicity. We do not expect a dual consistent treatment to significantly change the results, and moreover we are comparing adaptation strategies and not adaptive indicators.

The fine approximation space, \mathcal{V}_h , required for the adjoint solution ψ_h in Eqns. 5 and 6 is obtained by increasing the approximation order from p to $p + 1$ on the same mesh. In this work, the fine-space primal and adjoint solutions are computed by running ν_{fine} element block Jacobi smoothing iterations on \mathcal{V}_h . This approach is more affordable than solving the fine-space problems to machine precision and it was found to yield similar adaptive results. The number of fine-space smoothing iterations used for all the results presented here is $\nu_{\text{fine}} = 5$.

The proposed adaptive strategy was attempted not only using adjoint-based error estimates for engineering outputs, but also using the entropy adjoint error estimate.³⁸ The estimate in this case corresponds to the error in an output that measures the entropy generation in the computational domain. The appealing aspect of this indicator is that it does not require a separate solution of an adjoint equation: the required adjoint is equal to the state represented in entropy variables, which are computed directly from the conservative variables. Specifically, the entropy variables are used

in place of ψ_h , and they are calculated from the fine-space solution \mathbf{u}_h on each element by least-squares projection in \mathcal{V}_h . This “entropy adjoint” indicator was not used for the RANS simulations, as the entropy variables that symmetrize the Navier-Stokes equations do not symmetrize the SA turbulence model.

Both the isotropic and anisotropic adaptation methods are implemented in a distributed-memory multi-processor framework. In this framework, the initial output error estimation is parallelized so that each processor performs its own adjoint-weighted residual calculation, with communication required only when iterating the fine-space problems. For the discrete choice optimization on elements flagged for refinement (Section III), the solution of each sub-problem is carried out by the processor responsible for computing the original element, i.e., before the adaptation. As a result of the parallelization, the time consumed by the error estimation and the discrete choice optimization is very small relative to the solution of the primal problem.

V. Results

A. Laminar Flat Plate, $M = 0.25$, $Re = 100,000$

The first example of this adaptation framework consists of laminar viscous flow over a flat plate. Quadratic approximation, $p = 2$, was used in the discretization and the truth solution was computed by uniformly refining the finest adapted mesh and solving the flow equations with $p = 3$ approximation. The drag was calculated by integrating the shear stress over the flat plate, where the shear stress was computed from the viscous flux with appropriate adjoint-consistency terms included in the output.³¹

The drag coefficient was the engineering output of interest for this case. The coarse initial mesh is illustrated in Figure 5(a). In this mesh, the lower boundary of the domain is divided into two parts. The first part consists of the first five element faces on which a symmetry boundary condition is applied. On the second part, which is composed of the remaining faces on the bottom boundary, a no-slip, adiabatic wall boundary condition is enforced. Four different output-based adaptation strategies were used: isotropic and anisotropic adaptation, each driven by both drag and entropy adjoint error estimates. Figure 4 shows the error convergence for these strategies. Uniform refinement results are given for comparison.

The anisotropic drag-adjoint strategy typically consumes fewer degrees of freedom than its isotropic counterpart. However, this behavior is not observed for the entropy-adjoint strategies. For these strategies, the anisotropic method performs very similarly to anisotropic drag output adaptation for the first two cycles and then it starts targeting the leading edge as well as regions far from the surface, while the isotropic entropy-adjoint strategy seems to be focused on regions close to the surface and therefore performs better. Several of the adapted meshes are shown in Figure 5 for comparison.

B. Laminar NACA 0012, $M = 0.5$, $\alpha = 2^\circ$, $Re = 5,000$

The second example is laminar viscous flow over a NACA 0012 airfoil with 2° angle of attack. The outer boundary is located approximately 50 chord-lengths from the airfoil. The mesh consists of quadrilaterals with quartic ($q = 4$) geometry representation. Two engineering outputs were considered: drag coefficient and lift coefficient. The adaptation runs were performed using adjoints associated with each of these outputs and using the entropy adjoint. All adaptation runs started

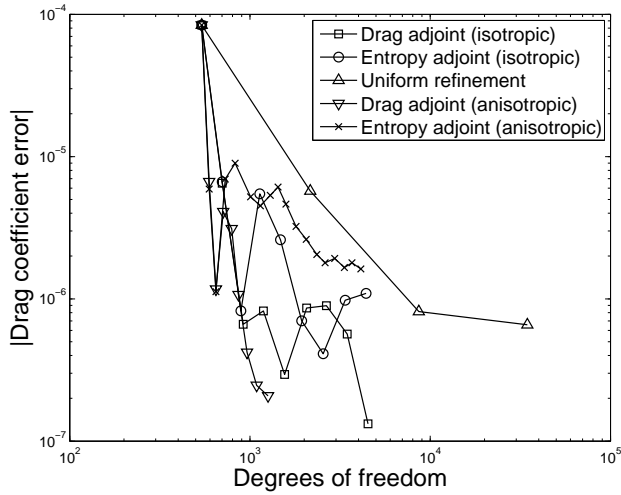


Figure 4. Flat plate, $M = 0.25$, $Re = 100,000$: Drag coefficient error convergence for the tested refinement strategies.

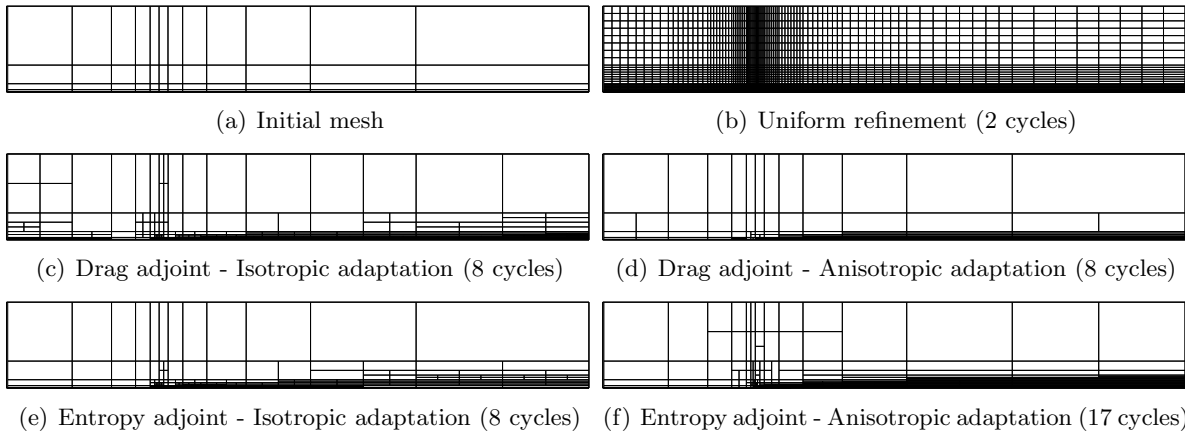


Figure 5. Flat plate, $M_\infty = 0.25$, $Re = 100,000$: Adapted meshes for the tested refinement strategies.

from the baseline mesh depicted in Figure 7(a).

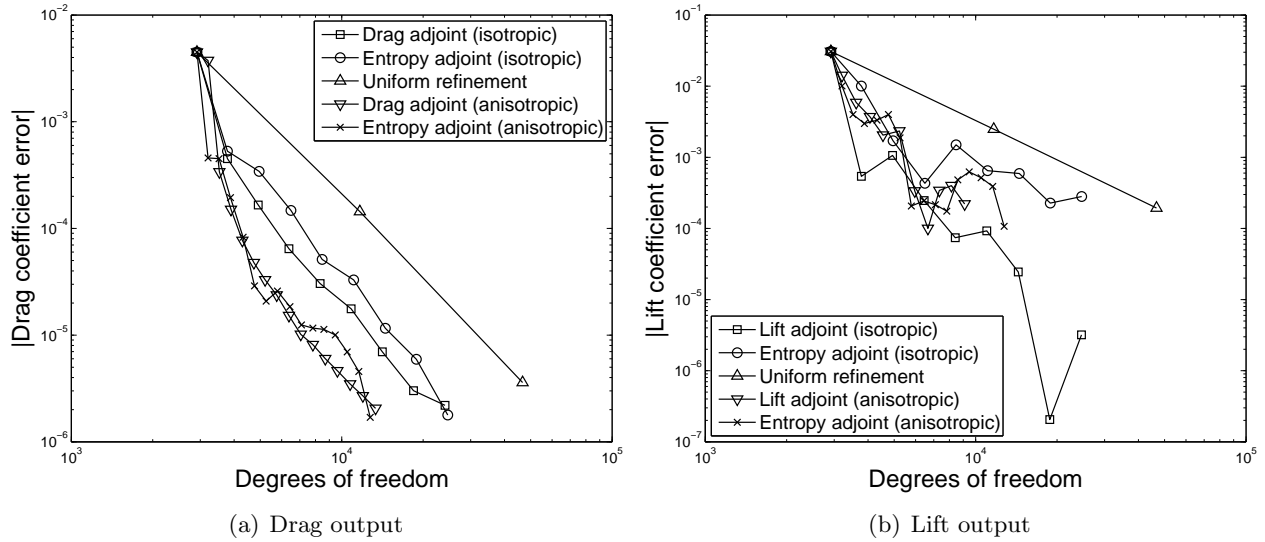


Figure 6. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$: Comparison of output convergence histories for the different adaptation strategies.

Figure 6 presents the results of the adaptation runs for the different strategies along with uniform mesh refinement for comparison purposes. As in the first example, the truth outputs were computed by uniformly refining the finest output adapted mesh and increasing the approximation order. In this example, anisotropic adaptation presented itself most valuable for the drag output

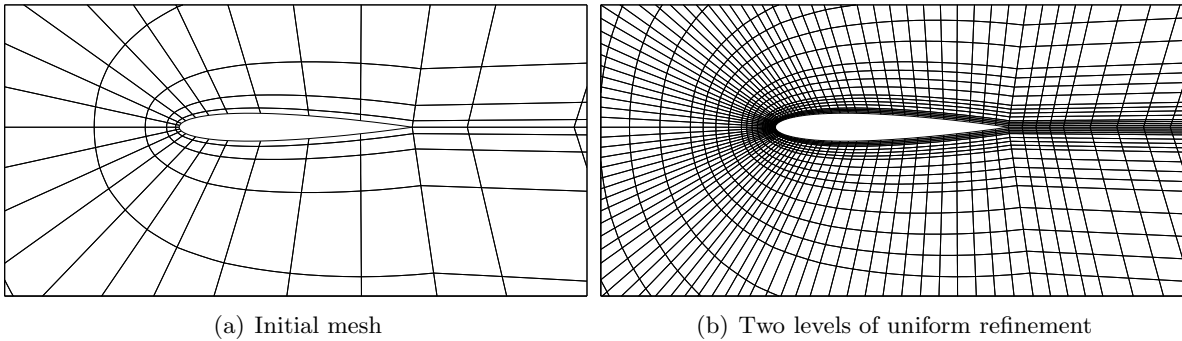


Figure 7. NACA 0012: Initial and uniformly refined meshes..

error convergence. Figure 8 shows that both isotropic and anisotropic methods targeted for adaptation some common regions of the flow domain, namely the stagnation streamline, the boundary layer and the wake. Despite the fact that both methods are based on the same output adjoint, isotropic adaptation refined more on the leading edge and its surroundings while the anisotropic option focused more on the wake region. A possible reason for this difference is the non-linearity of the mesh adaptation problem. That is, at each adaptation cycle different refinement schemes may produce different adapted meshes which subsequently can lead to different adaptive indicator fields. In addition, the initial mesh anisotropy plays an important role. Whereas isotropic refine-

ment preserves the initial mesh anisotropy, which may not be ideal in many locations, anisotropic refinement can choose to reduce this anisotropy through directional refinements. This effect is likely responsible for the apparent over-refinement of the area in front of the airfoil in the isotropic case, where the horizontal cuts in the elements are a byproduct of adapting isotropically and are not strictly necessary, as seen by comparing to the anisotropically-adapted mesh. Finally, note that in this case, the entropy adjoint adaptation performs similarly to the drag adjoint adaptation. The final meshes for these runs are shown in Figure 10.

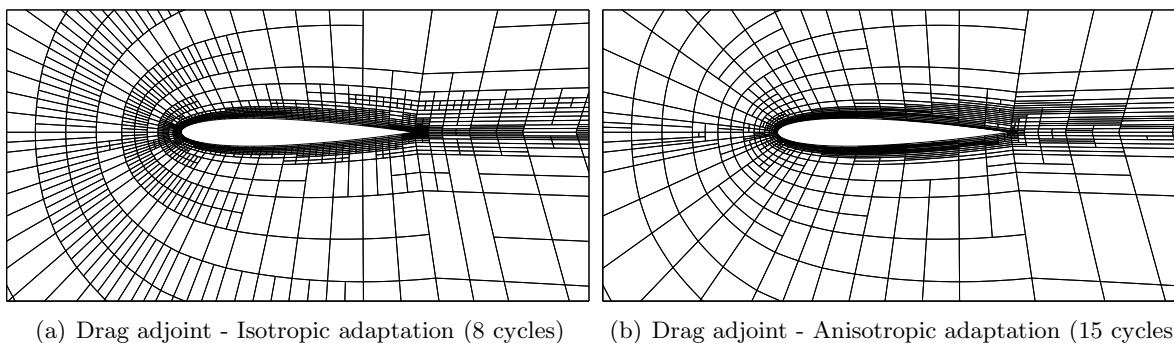


Figure 8. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$: Drag output-based adapted meshes.

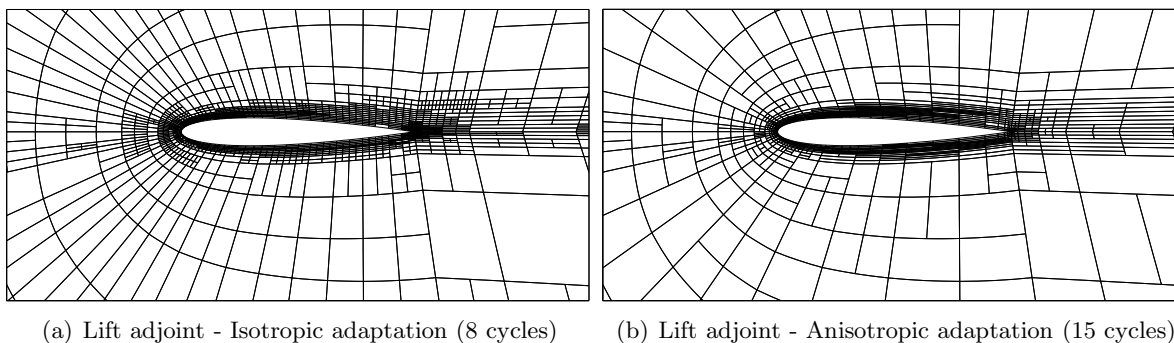
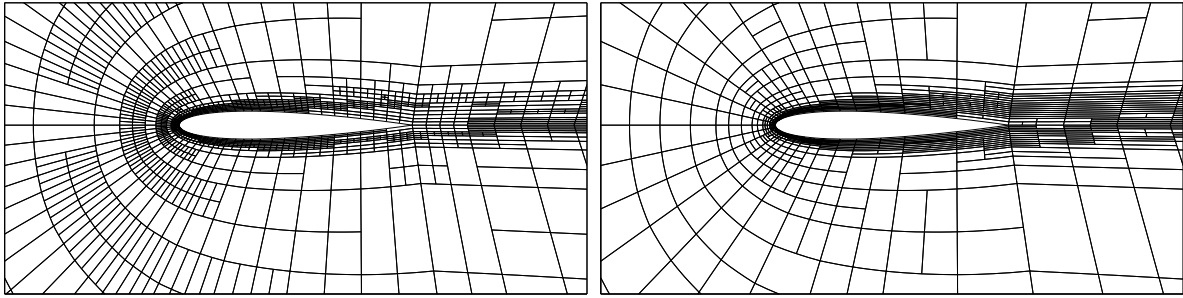


Figure 9. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$: Lift output-based adapted meshes.

Figure 6(b) shows that the lift coefficient error did not benefit significantly from the directional refinement as both isotropic and anisotropic methods presented similar error convergence rates for the lift adjoint. This is supported by the mesh in Figure 9(b) which shows that most elements were refined isotropically. It is worth noting, however, that the anisotropic method needed more adaptation iterations to achieve the same level of lift coefficient error as the isotropic method, as a result of elements being adapted anisotropically first in one direction, and next in the other. This suggests that the cost function $\mathcal{C}_{\kappa_H}(i)$ might be over-penalizing the isotropic refinement option, possibly due to the form of the benefit function $B(i)$ in Eqn. 8 which is a sum of absolute values that exhibits stronger triangle-inequality effects with an increasing number of sub-elements. Therefore, the specific form of the cost function needs further investigation.



(a) Entropy adjoint - Isotropic adaptation (8 cycles) (b) Entropy adjoint - Anisotropic adaptation (15 cycles)

Figure 10. NACA 0012, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$: Entropy-adjoint-based adapted meshes.

C. Laminar NACA 0004, $M = 0.4$, $\alpha = 0^\circ$, $Re = 50,000$

The third example is subsonic high Reynolds number, but still laminar, flow over a thin airfoil. As in the previous example, the outer boundary of the domain is located approximately 50 chord-lengths from the airfoil and the mesh is composed of quartic quadrilaterals.

The thin boundary layer present in this type of fluid flow strongly suggests that the use of the anisotropic adaptation scheme will be more efficient compared to isotropic adaptation. Similarly to the previous examples, uniform refinement was performed for comparison purposes. The output of interest in this case is the drag coefficient.

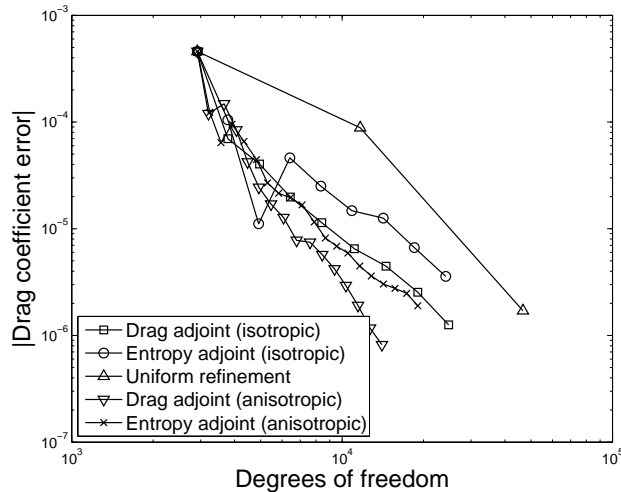


Figure 11. NACA 0004, $M = 0.4$, $\alpha = 0^\circ$, $Re = 50,000$: Drag coefficient error convergence for the tested refinement strategies.

Figure 11 presents the drag coefficient error convergence. A significant difference in the number of degrees of freedom is apparent between the isotropic and the anisotropic adaptation runs. Specifically, the difference is approximately a factor of 2.5 for the final drag adjoint adapted meshes, at a similar error level.

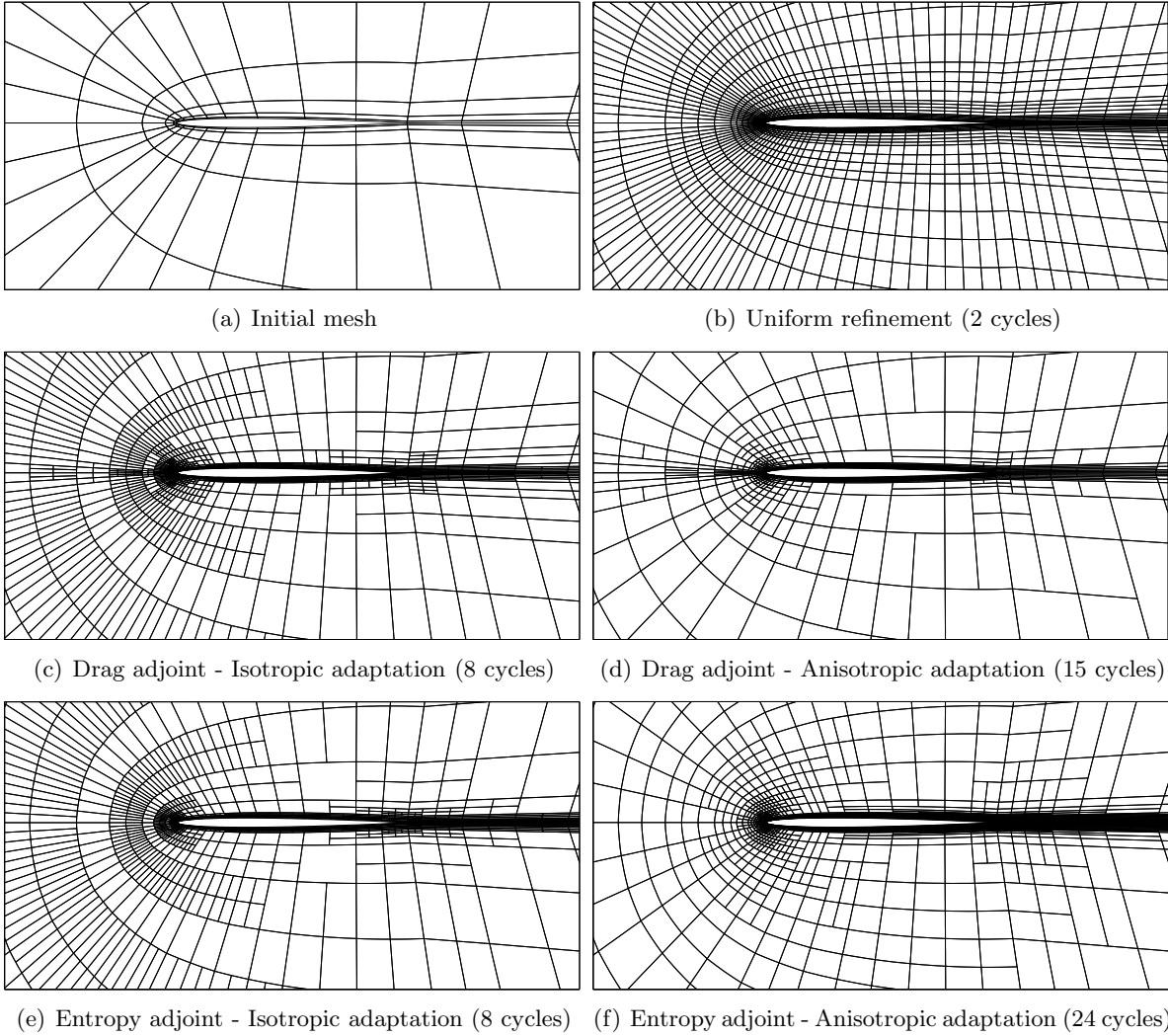


Figure 12. NACA 0004, $M = 0.4, \alpha = 0^\circ, Re = 50,000$: Adapted meshes for the tested refinement strategies.

As in the NACA 0012 case, the isotropic adaptation methods appear to waste degrees of freedom in the vicinity of the leading edge and in front of the airfoil, due to the preservation of the initial mesh anisotropy, whereas the anisotropic adaptation relieves this initial anisotropy through directional cuts. In contrast, in the boundary layer and in the wake, the direct optimization strategy acts to increase the element anisotropy. This is expected based on the thin boundary layer and wake in this example.

D. Turbulent NACA 0012, $M = 0.8$, $\alpha = 1.25^\circ$, $Re = 100,000$

The fourth example is of turbulent, transonic flow over an NACA 0012 airfoil. $p = 2$ is used for solution approximation, and the meshes consist of $q = 3$ quadrilateral elements. The initial mesh is shown in Figure 13(a).

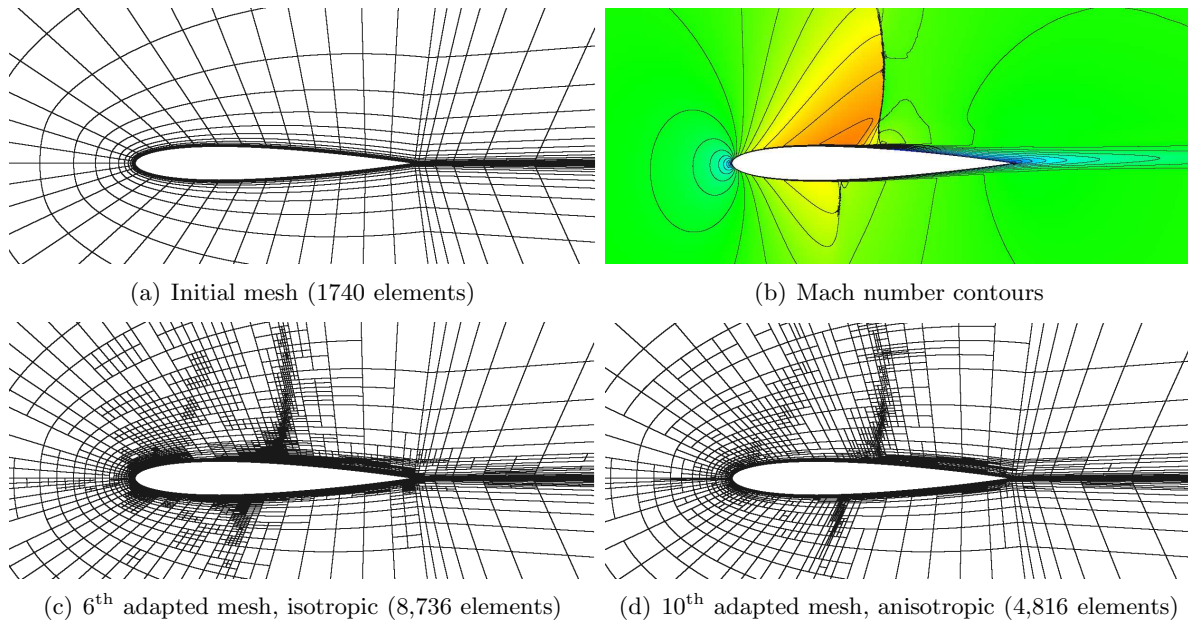


Figure 13. Turbulent NACA 0012, $M = 0.8$, $\alpha = 1.25^\circ$, $Re = 100,000$: Initial mesh, solution contours, and adapted meshes.

Convergence of the drag and lift coefficients is shown in Figure 14. The plots are similar in that the lift output converges as rapidly as the drag output for all of the adaptation schemes. Also, the drag adaptation performs well for the lift output and vice versa. The anisotropic adaptations converge much more rapidly compared to the isotropic adaptations: the outputs do not change much after 40,000 degrees of freedom with the anisotropic adaptation, while changes are still observed after nearly 100,000 degrees of freedom when using isotropic adaptation.

Two of the drag-adapted meshes are shown in Figure 13: one from isotropic adaptation after six iterations and one from anisotropic adaptation after ten iterations. Differences are evident in the boundary layer and wake, where anisotropic adaptation is more efficient. The shock also appears to be more tightly resolved in the anisotropically-adapted mesh.

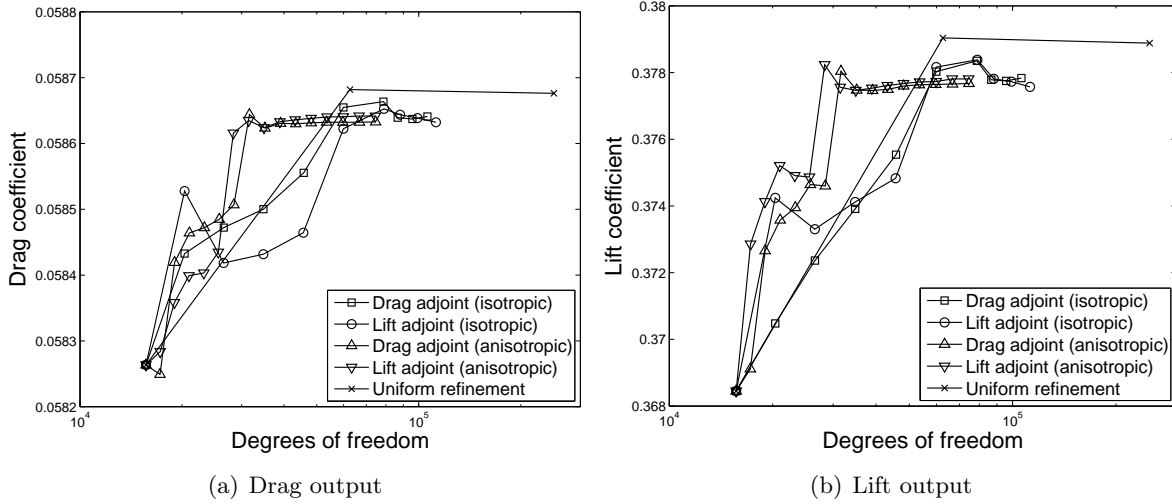


Figure 14. Turbulent NACA 0012, $M = 0.8$, $\alpha = 1.25^\circ$, $Re = 100,000$: Drag and lift convergence with degrees of freedom using isotropic and anisotropic refinement.

E. Drag Prediction Workshop Wing, $M = 0.76$, $\alpha = 0.5^\circ$, $Re = 5 \times 10^6$

The last example is a preliminary result using the wing-alone test case (Wing 1) from the third AIAA Drag Prediction Workshop.³ This case consists of turbulent, transonic flow over a tapered wing. The initial curved mesh, shown in Figure 15(a), was obtained from a finer multiblock structured linear mesh through element agglomeration. Specifically, each curved hexahedral element was obtained by merging twenty seven linear elements using a distance-based Lagrange interpolation of the nodal coordinates, resulting in cubic ($q = 3$) geometry interpolation. The Spalart-Allmaras turbulence model was used for this case, and a solution on the initial 38,912-element mesh is shown in Figure 15(b).

Due to limited computational resources, only one adaptation iteration was performed with $p = 2$ solution approximation, using the isotropic and anisotropic refinement algorithms. Drag was chosen as the output of interest. The adapted meshes are shown in Figure 15(c,d). The discrete choices optimization exhibits a propensity for isotropic refinement in this case, perhaps due to relatively optimal anisotropy distribution in the initial mesh. Nevertheless, differences exist, as seen by the element counts and by mesh differences near the leading and trailing edges.

Figure 16 shows the behavior of the drag and lift outputs for the two adaptations. Both the drag and lift coefficients compare reasonably well to the workshop submissions documented in³ – the differences from the submission means are of the same magnitude as the spreads in the submissions. Also, the change in drag coefficient is similar for the isotropic and anisotropic adaptations, with the anisotropic adaptation creating about two thirds as many elements compared to isotropic adaptation. Although these results look promising, they are preliminary and more adaptation iterations are required to draw decisive conclusions.

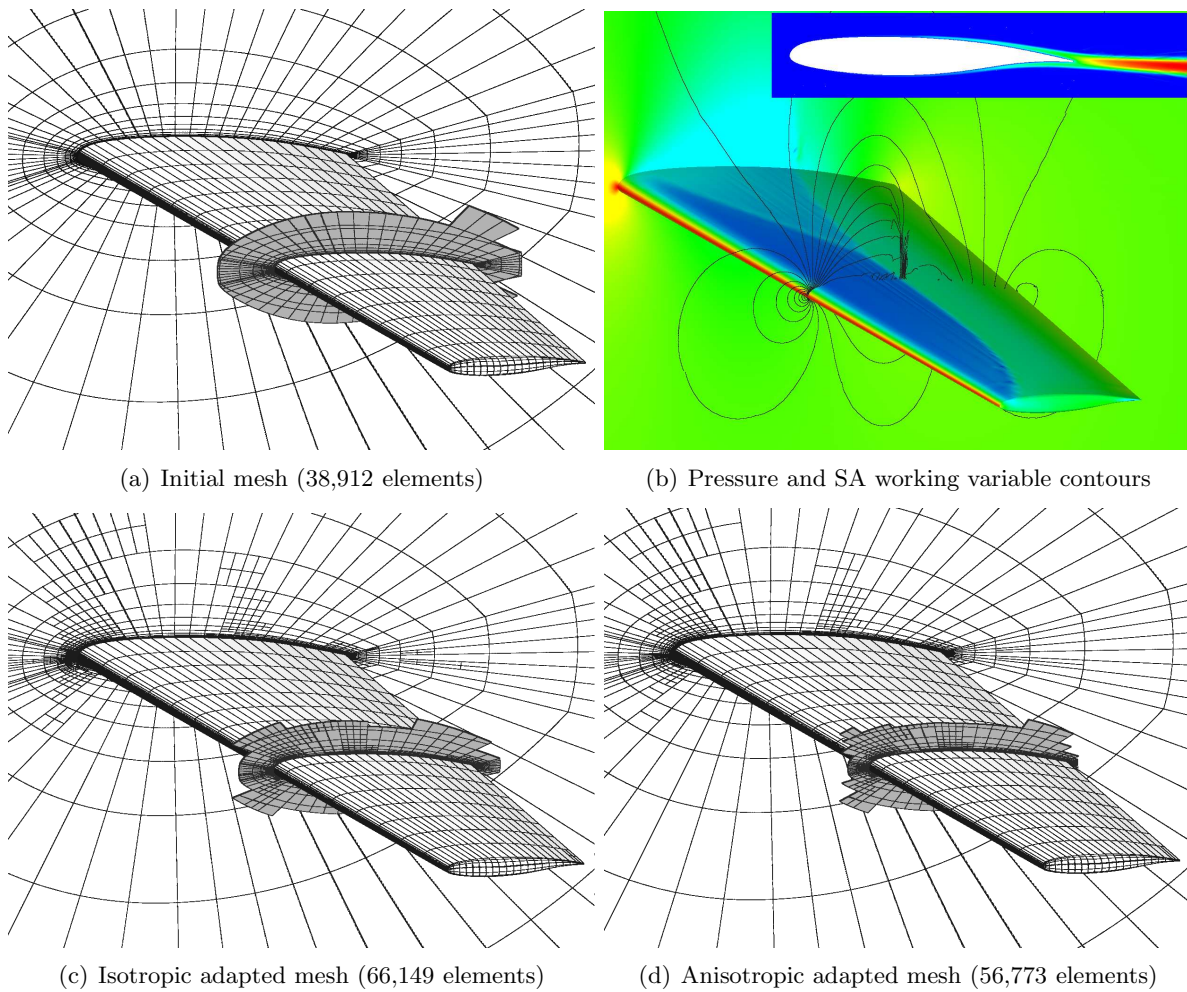


Figure 15. DPW Wing 1, $M = 0.76$, $\alpha = 0.5^\circ$, $Re = 5 \times 10^6$: Initial and adapted meshes, and solution contours.

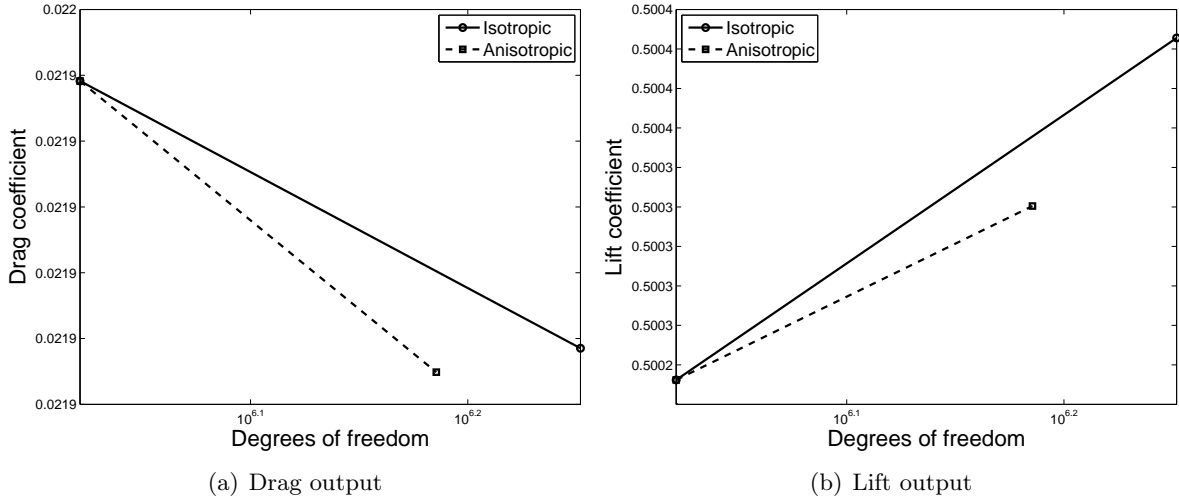


Figure 16. DPW Wing 1, $M = 0.76$, $\alpha = 0.5^\circ$, $Re = 5 \times 10^6$: Drag and lift changes for one adaptation iteration, using isotropic and anisotropic refinement.

VI. Conclusions

The present paper introduces an output-based anisotropic mesh adaptation method. The framework for choosing the refinement option is implemented in the context of adjoint-based output error estimation with hanging-node meshes. However, it can be applied to general adaptation strategies in which a discrete number of refinement options are available. This versatility is due to the fact that the method only relies on a definition of a cost function which can be tailored for a specific adaptation strategy, e.g., the unweighted residual strategy for which η_{κ_r} in Eqn. 8 could be substituted by the absolute value of the residuals in each embedded element.

The reduction in the number of degrees of freedom for the anisotropic cases presented here ranges from 2 to 3 in comparison to the purely isotropic refinement cases. This advantage was more pronounced for the drag coefficient error convergence in the laminar cases, suggesting that mesh anisotropy is more important for efficient resolution of drag in these runs. The lift coefficient did benefit from anisotropic refinement in the turbulent case considered, where the convergence curves of lift and drag were similar.

In practice, the ideal refinement strategy for a general output is not known *a priori*, so that using a non-heuristic method is expected to be most robust. In that regard, the discrete optimization approach is an attempt to eliminate the empiricism in the refinement direction decision. Some flexibility still remains in the adaptation process, specifically in the definition of the cost function $\mathcal{C}_{\kappa_H}(i)$, which can be based on *a priori* knowledge of the spatial discretization scheme used for solving the primal and adjoint problems.

Albeit some isotropy is seen in the final meshes for the cases presented here, the isotropic refinement option was typically chosen in less than 1% of the elements marked for adaptation. The reason for this is the fact that the isotropic option present much larger cost due to the number of additional elements and internal faces compared to the anisotropic counterparts. Consequently, the isotropy observed in the final meshes is mostly created by alternating the direction of refinement in subsequent cycles. This can be interpreted as a flaw of the cost function definition since it

requires more adaptation iterations to achieve a solution that could theoretically be obtained by a more appropriate formula of the cost function. However, the alternating anisotropic refinements is potentially caused by elements in the vicinity of a flagged element that were refined in different ways affecting the adaptation decision in that specific element in later cycles. In addition, the inclusion of a large-enough stencil to include more elements in each of the sub-domains would make the overall problem less tractable since a more sophisticated technique would be required for solving the sub-problems at a higher computational cost.

A limitation of the method presented in this work is the requirement of an initial quadrilateral or hexahedral mesh with the geometry represented to an accurate enough fidelity. However, many such meshes and associated meshing programs exist from the structured CFD community, and these can be leveraged to produce the starting meshes for the proposed adaptation.

The results presented in this paper suggest that the computational cost reduction improves in higher Reynolds number flows. Three-dimensional problems are expected to benefit even more from anisotropic adaptation due to the larger ratio of computational cost for isotropic versus single-cut anisotropic refinements. Preliminary results in the present work hint that this is the case, but additional refinements are needed to draw decisive conclusions. Further investigation into three-dimensional flows is the subject of ongoing work.

References

- ¹Levy, D. W., Zickuhr, T., Vassberg, J., Agrawal, S., Wahls, R. A., Pirzadeh, S., and Hensch, 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.
- ²Laffin, 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.
- ³Frink, N. T., “Test case results from the 3rd AIAA drag prediction workshop,” NASA Langley, 2007, http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3/final_results_jm.tar.gz.
- ⁴Becker, 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.
- ⁵Giles, M. and Pierce, N., “Adjoint error correction for integral outputs,” *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, Vol. 25, Springer, Berlin, 2002.
- ⁶Hartmann, 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.
- ⁷Venditti, 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.
- ⁸Nemec, M., Aftosmis, M. J., and Wintzer, M., “Adjoint-based adaptive mesh refinement for complex geometries,” AIAA Paper 2008-0725, 2008.
- ⁹Dwight, R. P., “Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation,” *Journal of Computational Physics*, Vol. 227, 2008, pp. 2845–2863.
- ¹⁰Park, M. A., “Adjoint-based, three-dimensional error prediction and grid adaptation,” AIAA Paper 2002-3286, 2002.
- ¹¹Hartmann, 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.
- ¹²Fidkowski, 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.
- ¹³Aftosmis, M. and Berger, M., “Multilevel error estimation and adaptive h-refinement for Cartesian meshes with embedded boundaries,” AIAA Paper 2002-14322, 2002.
- ¹⁴Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., “Adaptive remeshing for compressible flow computations,” *Journal of Computational Physics*, Vol. 72, 1987, pp. 449–466.

- ¹⁵Mavriplis, D. J., “Adaptive mesh generation for viscous flows using Delaunay triangulation,” *Journal of Computational Physics*, Vol. 90, 1990, pp. 271–291.
- ¹⁶Castro-Diaz, M. J., Hecht, F., Mohammadi, B., and Pironneau, O., “Anisotropic unstructured mesh adaptation for flow simulations,” *International Journal for Numerical Methods in Fluids*, Vol. 25, 1997, pp. 475–491.
- ¹⁷Habashi, W. G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M., and Vallet, M.-G., “Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles,” *International Journal for Numerical Methods in Fluids*, Vol. 32, 2000, pp. 725–744.
- ¹⁸Venditti, D. A., *Grid Adaptation for Functional Outputs of Compressible Flow Simulations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2002.
- ¹⁹Fidkowski, K. J., *A Simplex Cut-Cell Adaptive Method for High-order Discretizations of the Compressible Navier-Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2007.
- ²⁰Fidkowski, K. J. and Darmofal, D. L., “A triangular cut-cell adaptive method for high-order discretizations of the compressible navier-stokes equations,” *Journal of Computational Physics*, Vol. 225, 2007, pp. 1653–1672.
- ²¹Formaggia, L., Perotto, S., and Zunino, P., “An anisotropic a posteriori error estimate for a convection-diffusion problem,” *Computing and Visualization in Science*, Vol. 4, 2001, pp. 99–104.
- ²²Formaggia, L. and Perotto, S., “New anisotropic a priori error estimates,” *Numerische Mathematik*, Vol. 89, 2001, pp. 641–667.
- ²³Formaggia, L., Micheletti, S., and Perotto, S., “Anisotropic mesh adaptation with applications to CFD problems,” *Fifth World Congress on Computational Mechanics*, edited by H. A. Mang, F. G. Rammerstorfer, and J. Eberhardsteiner, Vienna, Austria, July 7-12 2002.
- ²⁴Schneider, R. and Jimack, P. K., “Toward anisotropic mesh adaptation based upon sensitivity of a posteriori estimates,” Tech. Rep. 2005.03, University of Leeds, School of Computing, 2005.
- ²⁵Park, M. A., *Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.
- ²⁶Houston, P., Georgoulis, E. H., and Hall, E., “Adaptivity and a posteriori error estimation for DG methods on anisotropic meshes,” *International Conference on Boundary and Interior Layers*, 2006.
- ²⁷Fidkowski, K. J. and Darmofal, D. L., “Output-based error estimation and mesh adaptation: Overview and recent results,” AIAA Paper 2009-1303, 2009.
- ²⁸Rannacher, R., “Adaptive Galerkin finite element methods for partial differential equations,” *Journal of Computational and Applied Mathematics*, Vol. 128, 2001, pp. 205–233.
- ²⁹Barth, 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.
- ³⁰Solín, P. and Demkowicz, L., “Goal-oriented hp-adaptivity for elliptic problems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 193, 2004, pp. 449–468.
- ³¹Lu, 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.
- ³²Hartmann, R., “Adjoint consistency analysis of discontinuous Galerkin discretizations,” *SIAM Journal on Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696.
- ³³Oliver, T. A., *A High-order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.
- ³⁴Venditti, 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.
- ³⁵Giles, 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.
- ³⁶Roe, P. L., “Approximate Riemann solvers, parametric vectors, and difference schemes,” *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.
- ³⁷Bassi, 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.
- ³⁸Fidkowski, K. J. and Roe, P. L., “Entropy-based mesh refinement, I: The entropy adjoint approach,” AIAA Paper 2009-3790, 2009.