Output-Based Mesh Adaptation for Variable-Fidelity Multipoint Aerodynamic Optimization

Guodong Chen* and Krzysztof J. Fidkowski†

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

In aerodynamic design, vehicle performance is often of interest over a range of operating conditions. This can be achieved by multipoint optimization, in which the design is evaluated under various flight conditions, generally with a fixed computational mesh in simulation-based optimization. However, flow fields at various operating conditions can have very different features, or even different physics. In order to maintain high-fidelity of the simulations, a mesh must be used that is sufficiently fine to capture all of the important features for different design points. Alternatively, separate meshes can be carefully generated for each operating condition, but this can be non-trivial and time-consuming as the number of design points increases. Furthermore, even if the initial mesh can predict the outputs accurately on the original design, the accuracy during the optimization is not controlled, and the error in the outputs can increase as the shape changes. In this paper, we address these issues by using meshes adapted via adjoint-based error estimates in the optimization to actively control the discretization error at each design point. The method is capable of dealing with optimization problems containing output constraints, in which the mesh is adapted to predict both objective and constraint outputs with appropriate accuracy. A multi-fidelity optimization framework is developed by taking advantage of variable fidelity offered by adaptive meshes. The objective function at each point is first evaluated on the same initial coarse mesh, which is then subsequently adapted for each design point individually as the shape optimization proceeds. The effort to setup the optimization can be substantially reduced since the initial mesh can be fairly coarse and easy to generate. As the shape approaches the optimal design, the mesh at each design point becomes finer, in regions necessary for that particular operating condition. The variable-fidelity framework saves computational resources by reducing the mesh size at early stages of optimization, when the design is far from optimal and most of the shape changes happen. The optimization at each fidelity terminates once the change of the objective is smaller than the tolerance (estimated error) at the current-fidelity mesh, reducing unnecessary effort in optimizing on low-fidelity meshes. We demonstrate the accuracy and efficiency of our proposed method on a fixed-lift drag minimization problem of a transonic airfoil. We expect the framework to be even more important for more complex configurations, dramatic shape changes, or high-accuracy requirements.

I. Introduction

Over the past decades, Computational Fluid Dynamics (CFD) has benefited greatly from the increasing power of high-performance computers, such that nowadays, CFD is an integral part of aerodynamic design. The capacity of design evaluation at almost arbitrary test conditions, as well as cheaper cost compared to wind tunnel tests, have made CFD an indispensable tool in the aerodynamic design process. Successful use of CFD in practical design problems requires both accurate simulations for a given configuration and efficient optimization methods to improve design configurations. Gradient-free methods such as genetic algorithms may be made robust for non-smooth or non-convex problems [1], but they are generally not as efficient as gradient-based methods, especially for problems with a large number of design parameters. Specifically, gradient-based algorithms converge to the optimum with fewer evaluations of the objective function and lower

*Graduate Research Assistant, AIAA Student Member

†Associate Professor, AIAA Senior Member

cost, even when taking into account the gradient calculation. Moreover, with the development of adjoint-based sensitivity analysis [2–6], the computational cost of gradient-based optimization has been dramatically reduced.

Ideally, the design is expected to retain favorable performance over a wide range of operating conditions. Optimization at one specific cruise condition can lead to mediocre performance on the overall mission profile, and/or poor performance at off-design conditions. Therefore, aerodynamic optimization in practice must take into account various flight conditions in both the objective and constraints [7–10], making the setup for high-fidelity aerodynamic optimization challenging. In order to achieve high accuracy for each design point, the single mesh used for all points has to be able to capture all the important flow features over a wide range of operating conditions. The designer, either expert in meshing or not, cannot reliably generate a mesh appropriate for all cruise conditions. This situation can be even worse when the geometry is complex or as the number of design points increases. Also, the resulting mesh may be quite fine, making the computational cost needed for high-fidelity multipoint aerodynamic optimization prohibitive in practice. On the other hand, numerical errors are typically only investigated via grid convergence studies for the initial and final designs, before and after the optimization, which can potentially lead to inaccurate or spurious optima [11, 12]. These are the problems that we tackle in the present work.

In order to aid practical multipoint aerodynamic design and to reduce the optimization cost, automated adapted meshes are introduced into the optimization process. The designer only needs to provide a relatively coarse background mesh to start the optimization run. Then, the computational mesh is adapted individually at necessary regions based on the output error estimates, with active control of numerical error at various operating conditions. The idea to combine output error estimation and gradient-based optimization is natural, as both methods require output adjoint solutions. After showing many success in a wide range of aerospace computational applications [13–19], output-based error estimation and mesh adaptation methods have been demonstrated in several single-point aerodynamic shape optimization problems [11, 12, 20–24], however, no attention has been paid to more complicated multipoint optimization problems.

The present work proposes a variable-fidelity implementation of multipoint aerodynamic optimization, integrating output-based error estimation and mesh adaptation with a gradient-based algorithm to actively control the numerical error during optimization. We adopt the error estimation and mesh adaptation strategy developed in our previous work [12], taking the errors in the constraint outputs into account during the error estimation and mesh adaptation, since the errors in the constraints can indirectly affect the calculation of the objectives [25]. Two unstructured mesh adaptation methods are considered in this paper: (a) mesh adaptation with Hessian-based anisotropy, and (b) mesh optimization via error sampling and synthesis (MOESS). In addition to the time saving on the optimization setup, this method also has the potential to reduce the run-time computational cost of the optimization. As the physics of various operating conditions can differ substantially, the meshes required to accurately predict the outputs also differ. With the proposed method, meshes for the points whose physics are relatively simple (e.g. low speed, laminar flow) can be coarse, whereas substantial mesh refinement can be added to those points governed by more complex physics (e.g. shocks, turbulent flows). A variable-fidelity optimization framework is built into the proposed method, taking advantage of the variable fidelity offered by the adaptive meshes. The variable-fidelity framework reduces the computational cost when the shape is far from the optimum, thus avoiding over-refining on an undesired configuration. On the other hand, the error estimation prevents optimization directions from being polluted by discretization errors and over-optimization on a coarse mesh.

The remainder of this paper proceeds as follows. We describe the general aerodynamic optimization problem in Section II and the discontinuous Galerkin discretization in Section III. Details of the error estimation and mesh adaptation are given in Section IV. Section V presents the coupling of gradient-based optimization with error estimation and mesh adaptation. The primary results are shown in Section VI, and Section VII concludes the present work and discusses potential future work.

II. Problem Formulation

II.A. Multipoint Aerodynamic Optimization

In general, the aerodynamic shape optimization problem can be stated as a search for the design variables $\mathbf{x}$ that minimize a given objective function J:

$$\begin{aligned} &\min_{\mathbf{x}} \ J(\mathbf{U},\mathbf{x}) \\ \text{s.t.} \quad &\mathbf{R}^{\text{e}}(\mathbf{U},\mathbf{x}) = \mathbf{0} \\ &\mathbf{R}^{\text{ie}}(\mathbf{U},\mathbf{x}) \geq \mathbf{0} \end{aligned} \tag{1}$$

where J represents a scalar objective function, always defined by the aerodynamic output, for example lift or drag or a combination of these for multi-objective optimization. $\mathbf{U}$ denotes the flow variables, and $\mathbf{R}^{\text{e}}$ and $\mathbf{R}^{\text{ie}}$ are the equality and inequality constraints, respectively. The flow variables $\mathbf{U}$ are solved within a feasible design space Ω to satisfy the flow equations, often the Euler or Navier-Stokes equations. In discretized form, these consist of a set of nonlinear equations that implicitly define $\mathbf{U}$ as a function of $\mathbf{x}$.

$$\mathbf{R}(\mathbf{U},\mathbf{x}) = \mathbf{0}, \quad \forall\ \mathbf{x} \in \Omega \tag{2}$$

The weighted-sum method is used here for multipoint optimization problems,

$$J_m = \boldsymbol{\omega}^T \mathbf{J} = \sum_{i=1}^{N_m} \omega_i J_i = \sum_{i=1}^{N_m} J_{m,i} \tag{3}$$

where $\mathbf{J}$ is the vector containing the objectives at N_m design points (typically various Mach numbers), $\boldsymbol{\omega}$ is the weights vector specified by the designer, and J_m is the composite objective used in the multipoint optimization problem, with weighted objective component $J_{m,i}$ at each design point.

II.B. Adjoint and Design Equations

Inactive inequality constraints $\mathbf{R}^{\text{ie}}_{\text{ia}}(\mathbf{U},\mathbf{x})$, do not affect the optimization explicitly, while the active ones $\mathbf{R}^{\text{ie}}_{\text{a}} = \mathbf{0}$ behave as equality constraints. In general, the inequality constraints can also be transformed into equality constraints with non-negative slack variables [26]. For easier illustration, we only consider the active inequality constraints and equality constraints, put together into one vector of dimension N_t as trim constraints, $(\mathbf{R}^{\text{trim}})^T = [(\mathbf{R}^{\text{e}})^T\ (\mathbf{R}^{\text{ie}}_{\text{a}})^T] \in \mathbb{R}^{N_t}$,

$$\mathbf{R}^{\text{trim}}(\mathbf{U},\mathbf{x}) = \mathbf{J}^{\text{trim}}(\mathbf{U},\mathbf{x}) - \bar{\mathbf{J}}^{\text{trim}} = \mathbf{0} \tag{4}$$

where $\bar{\mathbf{J}}^{\text{trim}}$ is a set of target trim outputs, for example, the target lift in a lift-constrained problem. In multipoint optimization problems, each operating condition can have the same or different trim constraints, $\mathbf{J}^{\text{trim}}$ is a vector concatenated with the trim outputs at different operating conditions $\mathbf{J}^{\text{trim}}_i \in \mathbb{R}^{N^i_t}$, $\sum_i^{N_m} N^i_t = N_t$. In order to distinguish the trim outputs from the objective output, we denote the latter by $J^{\text{adapt}}_m = \boldsymbol{\omega}^T \mathbf{J}^{\text{adapt}}$, as the objective output is the direct target of adaptation.

The adjoint-based optimization is equivalent to searching for the stationary point of the Lagrangian function, which augments the flow equations with additional constraints,

$$\mathcal{L} = \sum_{i=1}^{N_m} \omega_i J^{\text{adapt}}_i(\mathbf{U}_i,\mathbf{x}) + \sum_{i=1}^{N_m} \boldsymbol{\lambda_i}^T \mathbf{R}_i(\mathbf{U}_i,\mathbf{x}) + \sum_{i=1}^{N_m} \boldsymbol{\mu_i}^T \mathbf{R}^{\text{trim}}_i(\mathbf{U}_i,\mathbf{x}) \tag{5}$$

where $\mathbf{R}_i(\mathbf{U}_i,\mathbf{x})$ are the flow equations at each design point, $\boldsymbol{\lambda}_i, \boldsymbol{\mu}_i$ are the Lagrange multipliers associated with the flow equations and the trim constraints, respectively. The first-order optimality conditions are given

by setting the partial derivatives of $\mathcal{L}$ to zero,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \sum_{i=1}^{N_m} \omega_i \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{x}} + \sum_{i=1}^{N_m} \boldsymbol{\lambda}_i^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{x}} + \sum_{i=1}^{N_m} \boldsymbol{\mu}_i^T \frac{\partial \mathbf{R}_i^{\text{trim}}}{\partial \mathbf{x}} = \mathbf{0} \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}_i} = \omega_i \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{U}_i} + \boldsymbol{\lambda}_i^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_i} + \boldsymbol{\mu}_i^T \frac{\partial \mathbf{R}_i^{\text{trim}}}{\partial \mathbf{U}_i} = \mathbf{0} \qquad i = 1, ..., N_m \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}_i} = \mathbf{R}_i(\mathbf{U}_i, \mathbf{x}) = \mathbf{0} \qquad i = 1, ..., N_m \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_i} = \mathbf{R}_i^{\text{trim}}(\mathbf{U}_i, \mathbf{x}) = \mathbf{0} \qquad i = 1, ..., N_m \tag{9}$$

As we solve the flow equations for a given design $\mathbf{x}$ each time at every design point, Eqn. 8 is always satisfied during the optimization. The trim constraints can be handled by either the flow solver or the optimizer; the former is used currently. A set of design variables is dedicated to satisfying the trim constraints, denoted as trim variables $\mathbf{x}_t$, $\dim(\mathbf{x}_t) = \dim(\mathbf{J}^{\text{trim}})$. Hence Eqn. 9 is satisfied by the variation of the trim variables, so that Eqn. 6 breaks down into,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_s} = \sum_{i=1}^{N_m} \omega_i \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{x}_s} + \sum_{i=1}^{N_m} \boldsymbol{\lambda}_i^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{x}_s} + \sum_{i=1}^{N_m} \boldsymbol{\mu}_i^T \frac{\partial \mathbf{R}_i^{\text{trim}}}{\partial \mathbf{x}_s} = \mathbf{0} \tag{10}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_t} = \sum_{i=1}^{N_m} \omega_i \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{x}_t} + \sum_{i=1}^{N_m} \boldsymbol{\lambda}_i^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{x}_t} + \sum_{i=1}^{N_m} \boldsymbol{\mu}_i^T \frac{\partial \mathbf{R}_i^{\text{trim}}}{\partial \mathbf{x}_t} = \mathbf{0} \tag{11}$$

where $\mathbf{x}_s$ is the set of active design parameters in the optimization. We can now choose $\boldsymbol{\lambda}_i$ and $\boldsymbol{\mu}_i$ at each point such that Eqn. 7 and Eqn. 11 are enforced after each flow solve,

$$\begin{aligned} \boldsymbol{\lambda}_i^T &= -\left(\omega_i \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{U}_i} + \boldsymbol{\mu}_i^T \frac{\partial \mathbf{R}_i^{\text{trim}}}{\partial \mathbf{U}_i}\right) \frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_i}^{-1} = (\omega_i \boldsymbol{\Psi}_i^{\text{adapt}} + \boldsymbol{\Psi}_i^{\text{trim}} \boldsymbol{\mu}_i)^T \\ \boldsymbol{\mu}^T &= [\boldsymbol{\mu}_1^T, ..., \boldsymbol{\mu}_{N_m}^T] = -\left(\sum_{i=1}^{N_m} \omega_i \frac{dJ_i^{\text{adapt}}}{d\mathbf{x}_t}\right) \left(\frac{d\mathbf{J}^{\text{trim}}}{d\mathbf{x}_t}\right)^{-1} \end{aligned} \tag{12}$$

There may exist a set of trim variables which can eliminate the coupling of trim constraints among different design points ($d\mathbf{J}^{\text{trim}}/d\mathbf{x}_t$ is diagonal), then $\boldsymbol{\mu}_i$ only depends on the i^{th} design point. In lift-constrained optimization problems, angle of attack is such a choice. Eqn. 12 gives a coupled adjoint variable $\boldsymbol{\lambda}_i$ that incorporates the adjoints of both the objective and the trim outputs at each design point, $\boldsymbol{\Psi}_i^{\text{adapt}}$ and $\boldsymbol{\Psi}_i^{\text{trim}}$, which satisfy

$$\frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_i}^T \boldsymbol{\Psi}_i^{\text{adapt}} + \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{U}_i}^T = \mathbf{0}, \qquad \frac{\partial \mathbf{R}_i}{\partial \mathbf{U}_i}^T \boldsymbol{\Psi}_i^{\text{trim}} + \frac{\partial \mathbf{J}_i^{\text{trim}}}{\partial \mathbf{U}_i}^T = \mathbf{0} \tag{13}$$

In Eqn. 12, $d(\cdot)/d\mathbf{x}_t$ terms are defined as,

$$\begin{aligned} \frac{dJ_i^{\text{adapt}}}{d\mathbf{x}_t} &= \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{x}_t} + \left(\boldsymbol{\Psi}_i^{\text{adapt}}\right)^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{x}_t} \\ \frac{d\mathbf{J}^{\text{trim}}}{d\mathbf{x}_t} &= \frac{\partial \mathbf{J}^{\text{trim}}}{\partial \mathbf{x}_t} + \left(\boldsymbol{\Psi}_i^{\text{trim}}\right)^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{x}_t} \end{aligned} \tag{14}$$

the total derivative symbols are used in a sense that the outputs only depend on the trim variables $\mathbf{x}_t$ when the active design parameters $\mathbf{x}_s$ are held constant, i.e., the total derivative is defined in a sub-problem varying $\mathbf{x}_t$ to satisfy the constraints given fixed $\mathbf{x}_s$. For our optimization problem, $\mathbf{x}_t$ depends on $\mathbf{x}_s$, the total derivatives should only be defined with respect to the active design parameters $\mathbf{x}_s$.

With the specific choice of $\boldsymbol{\lambda}_i$ and $\boldsymbol{\mu}_i$ in Eqn. 12, we can evaluate the objective gradients, i.e., the total derivatives in optimization, with respect to the active design variables $\mathbf{x}_s$ via the Lagrangian function,

starting with Eqn. 10,

$$\begin{aligned}
\frac{dJ_m^{\text{adapt}}}{d\mathbf{x}_s} &= \frac{\partial \mathcal{L}}{\partial \mathbf{x}_s} = \sum_{i=1}^{N_m} \omega_i \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{x}_s} + \sum_{i=1}^{N_m} \boldsymbol{\lambda}_i^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{x}_s} + \sum_{i=1}^{N_m} \boldsymbol{\mu}_i^T \frac{\partial \mathbf{R}_i^{\text{trim}}}{\partial \mathbf{x}_s} \\
&= \sum_{i=1}^{N_m} \omega_i \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{x}_s} + \sum_{i=1}^{N_m} \omega_i (\boldsymbol{\Psi}_i^{\text{adapt}})^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{x}_s} + \sum_{i=1}^{N_m} \boldsymbol{\mu}_i^T \left[\frac{\partial \mathbf{R}_i^{\text{trim}}}{\partial \mathbf{x}_s} + (\boldsymbol{\Psi}_i^{\text{trim}})^T \frac{\partial \mathbf{R}_i}{\partial \mathbf{x}_s} \right] \\
&= \sum_{i=1}^{N_m} \omega_i \frac{dJ_i^{\text{adapt}}}{d\mathbf{x}_s} + \sum_{i=1}^{N_m} \boldsymbol{\mu}_i^T \frac{d\mathbf{J}_i^{\text{trim}}}{d\mathbf{x}_s} \\
&= \sum_{i=1}^{N_m} \left(\omega_i \frac{dJ_i^{\text{adapt}}}{d\mathbf{x}_s} + \boldsymbol{\mu}_i^T \frac{d\mathbf{J}_i^{\text{trim}}}{d\mathbf{x}_s} \right)
\end{aligned} \tag{15}$$

Now the optimization problem has been reduced to finding an optimal design $\mathbf{x}_s$ that drives to zero the gradients in Eqn. 15. However, in a practical calculation, on a finite-dimensional space, discretization errors appear in both the flow equations and the adjoint equations, so that infinite-dimensional optimality cannot be guaranteed even when the finite-dimensional optimality condition is satisfied. The present work focuses on controlling the error in the optimization problem via error estimation and mesh adaptation.

III. Discretization

Evaluation of the objective function at each optimization step relies on a flow simulation, in this work over an airfoil. The governing equations for the fluid flow are compressible Navier-Stokes,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) + \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0}, \tag{16}$$

where $\mathbf{u}$ is the flow state vector at a particular spatial location, $\vec{\mathbf{F}}$ is the sum of the inviscid and viscous flux vectors, and $\mathbf{S}$ is a source term required when modeling turbulence. When running Reynolds-averaged turbulent cases, we use the Spalart-Allmaras one-equation model, with a negative turbulent-viscosity modification [27]. We discretize Eqn. 16 with the discontinuous Galerkin (DG) finite-element method, which is suitable for high-order accuracy and *hp*-refinement [28–30]. However, the framework proposed in this work can be applied to other discretizations supporting output-based error estimation and mesh adaptation.

The DG spatial discretization of Eqn. 16 yields a system of discrete algebraic equations in the form of Eqn. 2,

$$\mathbf{R}_H(\mathbf{U}_H, \mathbf{x}) = \mathbf{0}, \tag{17}$$

where $\mathbf{R}_H$ is the residual vector, a nonlinear function of the discrete state vector $\mathbf{U}_H$ and the design variables $\mathbf{x}$. The subscript H refers to fidelity of the approximation/test space with respect to the approximation order and mesh refinement.

IV. Error Estimation and Mesh Adaptation

IV.A. Adjoint-based Error Estimation

In practice it is not possible to obtain the true numerical error for an output, whereas the difference between a coarse space and fine space solution serves as an acceptable surrogate,

$$\text{output error: } \delta J \equiv J_H(\mathbf{U}_H) - J_h(\mathbf{U}_h). \tag{18}$$

In this expression, J represents the output of interest, and the subscripts h and H denote the fine and coarse spaces, respectively. In the present work, the fine space is achieved by increasing the elements' approximation order p, to $p+1$. We do not solve the nonlinear fine-space flow problem for the error prediction, and instead we use the linear fine-space adjoint solution, $\boldsymbol{\Psi}_h$, defined as the sensitivity of the output to the residuals. The adjoint weights the residual perturbation to produce an output perturbation [31],

$$\begin{aligned}
\delta J &\approx J_H(\mathbf{U}_H) - J_h(\mathbf{U}_h) = J_h(\mathbf{U}_h^H) - J_h(\mathbf{U}_h) \\
&= -\boldsymbol{\Psi}_h^T [\mathbf{R}_h(\mathbf{U}_h^H) - \mathbf{R}_h(\mathbf{U}_h)] = -\boldsymbol{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H),
\end{aligned} \tag{19}$$

where $\mathbf{U}_h$ is the (hypothetical) exact solution on the fine space, $\mathbf{U}_h^H$ is the state injected into the fine space from the coarse one, which generally will not give a zero fine space residual, $\mathbf{R}_h(\mathbf{U}_h^H) \neq \mathbf{R}_h(\mathbf{U}_h) = \mathbf{0}$. The derivation of Eqn. 19 originates from the small perturbation assumption, and is valid for outputs whose definition does not change between the coarse and fine spaces, $J_H(\mathbf{U}_H) = J_h(\mathbf{U}_h^H)$.

IV.B. Output Error Estimation Under Trim Constraints

Normally, the error estimation is applied only to the output in which we are most interested, i.e. the objective. However, our optimization problem requires the simultaneous solution of flow equations and trim constraints. The numerical error of the trim outputs may indirectly affect the calculation of the objective. To take this effect into account, the coupled adjoint should be used for the error estimates.

Consider a given design $\mathbf{x}_s$, the error in the objective now comes from both the inexact solution $\mathbf{U}_h^H$ and inexact trim constraints satisfaction (inexact trim variables $\mathbf{x}_{t,H}$). We can estimate the error in the composite objective, with the linearization given by Eqn. 7 and Eqn. 11,

$$\begin{aligned}
\delta J_m^{\text{adapt}} &= \sum_{i=1}^{N_m} \omega_i \left(\frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{U}_i} \delta \mathbf{U}_i + \frac{\partial J_i^{\text{adapt}}}{\partial \mathbf{x}_t} \delta \mathbf{x}_t \right) \\
&= -\sum_{i=1}^{N_m} (\boldsymbol{\lambda}_{h,i}^T \delta \mathbf{R}_{h,i} + \boldsymbol{\mu}_{h,i}^T \delta \mathbf{R}_{h,i}^{\text{trim}}) \\
&= -\sum_{i=1}^{N_m} \left[\boldsymbol{\lambda}_{h,i}^T \mathbf{R}_{h,i}(\mathbf{U}_{h,i}^H, \mathbf{x}_{t,H}) + \boldsymbol{\mu}_{h,i}^T \mathbf{R}_{h,i}^{\text{trim}}(\mathbf{U}_{h,i}^H, \mathbf{x}_{t,H}) \right]
\end{aligned} \tag{20}$$

For the second term in Eqn. 20, we can expand the trim residual as

$$\begin{aligned}
\mathbf{R}_{h,i}^{\text{trim}}(\mathbf{U}_{h,i}^H, \mathbf{x}_{t,H}) &= \mathbf{J}_{h,i}^{\text{trim}}(\mathbf{U}_{h,i}^H, \mathbf{x}_{t,H}) - \bar{\mathbf{J}}_i^{\text{trim}} \\
&= [\mathbf{J}_{H,i}^{\text{trim}}(\mathbf{U}_{H,i}, \mathbf{x}_{t,H}) - \bar{\mathbf{J}}_i^{\text{trim}}] + [\mathbf{J}_{h,i}^{\text{trim}}(\mathbf{U}_{h,i}^H, \mathbf{x}_{t,H}) - \mathbf{J}_{H,i}^{\text{trim}}(\mathbf{U}_{H,i}, \mathbf{x}_{t,H})]
\end{aligned} \tag{21}$$

The first term above is automatically driven to zero because of the trimming on the coarse space. For the second term, again, if the definition of the trim outputs does not depend on the approximation space, then $\mathbf{J}_{h,i}^{\text{trim}}(\mathbf{U}_{h,i}^H, \mathbf{x}_{t,H}) = \mathbf{J}_{H,i}^{\text{trim}}(\mathbf{U}_{H,i}, \mathbf{x}_{t,H})$. Hence, the second term in Eqn. 20 is often negligible, resulting a simpler form for the error of the composite objective,

$$\begin{aligned}
\delta J_m^{\text{adapt}} &= -\sum_{i=1}^{N_m} \boldsymbol{\lambda}_{h,i}^T \mathbf{R}_{h,i}(\mathbf{U}_{h,i}^H, \mathbf{x}_{t,H}) \\
&= -\sum_{i=1}^{N_m} (\omega_i \boldsymbol{\Psi}_{h,i}^{\text{adapt}} + \boldsymbol{\Psi}_{h,i}^{\text{trim}} \boldsymbol{\mu}_{h,i})^T \mathbf{R}_{h,i}(\mathbf{U}_{h,i}^H, \mathbf{x}_{t,H}) \\
&= \sum_{i=1}^{N_m} (\omega_i \delta J_i^{\text{adapt}} + \boldsymbol{\mu}_{h,i}^T \delta \mathbf{J}_i^{\text{trim}})
\end{aligned} \tag{22}$$

IV.C. Mesh Adaptation

If we would like to use the same mesh for all of the design points, then Eqn. 22 can be directly used to localize the error to each element and serves as the indicator for mesh adaptation. However, this can be inefficient when the flow features change significantly at different operating conditions, e.g., from subsonic to supersonic regimes. To achieve certain accuracy in such cases, the mesh should be adapted in the areas important for all of the design points, and hence unnecessary computational effort is added to each flow solve if using a single mesh.

In present work, we allow different meshes for different design points. The objective error in Eqn. 22 is first localized to each design point as $\delta J_{m,i}^{\text{adapt}} = \omega_i \delta J_i^{\text{adapt}} + \boldsymbol{\mu}_{h,i}^T \delta \mathbf{J}_i^{\text{trim}}$. Then a common approach for obtaining an error indicator is to take the absolute value of the elemental error contribution. When the trim outputs are involved, we have another decision to make: whether we allow cancellation between the objective

and trim output error estimates. In this work we do not let such cancellation happen when calculating the error indicator, so that the final error indicator on element k at flight condition i, $\eta_{i,k}$ is given by

$$\eta_{i,k} = \omega_i |\delta J_{i,k}^{\text{adapt}}| + |\boldsymbol{\mu}_i^T||\delta \mathbf{J}_{i,k}^{\text{trim}}| = \omega_i \eta_{i,k}^{\text{adapt}} + |\boldsymbol{\mu}_i^T|\boldsymbol{\eta}_{i,k}^{\text{trim}}, \tag{23}$$

where $\eta_{i,k}^{\text{adapt}}$ and $\boldsymbol{\eta}_{i,k}^{\text{trim}}$ are the non-negative error indicators for objective output and trim outputs respectively, at the i^{th} flight condition.

At each operating condition, we can naively refine a certain fraction of the elements which have the largest adaptation indicator. However, such a strategy cannot detect strong directional features, such as shocks or boundary layers in flow problems. In order to efficiently capture the anisotropic flow features, we have to take the solution anisotropy into account in the mesh adaptation. Two anisotropic adaptation techniques are considered here: (a) mesh adaptation with Hessian-based anisotropy detection [a], and (b) mesh optimization via error sampling and synthesis (MOESS). Both methods rely on metric-based global remeshing, in which the mesh information, including the desired element sizes and stretching direction, is encoded with a continuous Riemannian metric field. The metric field is determined by the elemental error indicator given by Eqn. 23, together with local anisotropy information. The main difference between Hessian-based mesh adaptation and MOESS is how the anisotropy is specified. Hessian-based mesh adaptation uses the Hessian matrix of a scalar, usually the Mach number, to obtain the solution anisotropy, while MOESS extracts the anisotropy through the local error sampling and synthesis process.

IV.C.1. Error Allocation for Hessian-based Mesh Adaptation

A general, though not always optimal, goal in mesh adaptation is to reduce and equally distribute the objective error [32]. Hence, given an error tolerance $\mathcal{E}$ of the composite objective J_m^{adapt}, we first equally distribute the allowable error to each design point.

$$\max(\delta J_{m,i}^{\text{adapt}}) \leq \mathcal{E}_i \equiv \frac{\mathcal{E}}{N_m}. \tag{24}$$

We note that such a split may not be optimal if the design points represent vastly different conditions, in which case an optimal distribution of error need not be uniform; e.g. more error could be allowed for design points with a high cost of reducing the error. A more sophisticated approach of equidistributing the marginal error to cost ratio among the points is a topic for future work.

Given the allocation in Eqn. 24, the meshes can be adapted individually at each design point, with maximum allowable error $\mathcal{E}_i$. At each design point, the local adaptation indicator is obtained using Eqn. 23. In Hessian-based mesh adaptation, the error indicator determines the desired element size in the new adapted mesh, while the local Mach Hessian matrix controls the element stretching. The Hessian-based adaptation we used in this paper follows a fixed-growth refinement strategy. At each adaptation iteration, the mesh size, i.e., the number of elements, increases with a fixed factor, and a new mesh is generated with the desired metric field implied by the error indicator and Mach Hessian information [33, 34]. Given a target error tolerance $\mathcal{E}_i$, the mesh is refined until the error estimate drops below the tolerance.

IV.C.2. Cost Allocation for MOESS

For standalone MOESS, the mesh size and stretching are optimized with a fixed total cost, usually measured by the system degrees of freedom, DOF. Instead of equally distributing the cost among different flight conditions, the total cost can be divided in a way such that the objective errors $\delta J_{m,i}^{\text{adapt}}$ are equally distributed – again, a more sophisticated approach would be to equidistribute the marginal error to cost ratio, and this is a topic for future work. Here, an a priori error-cost model is assumed,

$$|\delta J_{m,i}^{\text{adapt}}| \propto C_i^{-(2p+1)/d}, \tag{25}$$

where C_i is the cost at i^{th} flight condition, d is the spatial dimension, p is the solution approximation order. The ideal super-convergence rate $2p+1$ is used to prevent too aggressive cost redistribution, though lower

[a] In this work, the solution (Mach number) Hessian is only used to provide anisotropy information. The error indicator, which governs the size of the elements, is still output-based [12].

convergence rates should be expected for under-resolved meshes. For one adaptive iteration, the changes in the error due to cost redistribution are

$$\delta J_{m,i}^{\text{adapt}} = \delta J_{m,i}^{\text{adapt},0} \left(\frac{C_i}{C_i^0} \right)^{-(2p+1)/d}, \tag{26}$$

where the superscript 0 indicates values in the unadapted meshes. With the requirements of error equidistribution, the desired cost ratios f_i among different design points are,

$$f_i = \frac{C_i}{C_1} = \frac{C_i^0}{C_1^0} \left[\frac{\delta J_{m,i}^{\text{adapt},0}}{\delta J_{m,1}^{\text{adapt},0}} \right]^{\frac{1}{(2p+1)/d}}, \quad i = 1, 2, ..., N_m \tag{27}$$

Given a fixed total cost C of the multipoint mesh optimization, the cost at each design point is then redistributed as,

$$C_i = \frac{f_i}{\sum_{j=1}^{N_m} f_j} C. \tag{28}$$

In the implementation, the cost ratio is limited to a certain range to avoid too much change in the cost. With the cost redistributed at each design point, the mesh optimization then follows an iterative process to equidistribute the marginal error to cost ratio of local refinements [35]. More details about the mesh adaptation strategy in the multi-fidelity optimization are given in the next section.

V. Optimization Approach

V.A. Objective and Constraints

For demonstration, two-dimensional airfoil shape optimization problems are considered in this work. In particular, the problem considered here is to search for an optimal design (including the airfoil shape and the angles of attack) to minimize the overall drag under a range of flight conditions, subject to fixed lift coefficients and a minimum airfoil volume. The objective is the weighted sum of the drag coefficients at different design points, and the corresponding constraints are

$$R_i^e(\mathbf{U}_i, \mathbf{x}) = c_{\ell,i}(\mathbf{U}_i, \mathbf{x}) - c_{\ell,i}^* = 0, \tag{29}$$

$$R^{ie}(\mathbf{U}_i, \mathbf{x}) = A(\mathbf{x}) - A_{\min} \geq 0, \tag{30}$$

where A and $A_{\min}$ represent the current and minimum volumes of the airfoil, and $c_{\ell,i}$ and $c_{\ell,i}^*$ denote the current and target lift coefficients at each design point.

The lift constraints are enforced by a trimming process during each flow solve. The angle of attack α at each design point is chosen as the trim variable, $\mathbf{x}_t = [\alpha_1, ..., \alpha_{N_m}]$, to decouple the trim conditions at various points. Then the lift constraint at each design point is enforced individually by finding the trimming angle of attack through a Newton-Raphson iteration. The inequality constraint (independent of the flow states, assumed to be measured exactly) is handled by the optimizer.

V.B. Airfoil Parameterization and Mesh Deformation

The airfoil shape is parameterized using the Hicks-Henne basis functions [36], taking a baseline airfoil and creating a new airfoil shape by adding a linear combination of "bump" functions to its upper and lower surfaces,

$$z = z_{\text{base}} + \sum_{i=1}^{n_s} a_i \phi_i(x), \tag{31}$$

where the scalar x denotes in this case the position along the airfoil chord rather than the design parameter vector, and z is the vertical coordinate of the upper or lower airfoil surface. The summation term in Eqn. 31 is a linear combination of the bump basis functions $\phi_i(x)$ with different coefficients a_i, which constitute the active design parameters for the optimization problem, $\mathbf{x}_s = [a_1, a_2, ..., a_{n_s}] \in \mathbb{R}^{n_s}$. A set of optimized Hicks-Henne basis functions [37] is used here. Table 1 summarizes the multipoint aerodynamic optimization problem considered in this paper.

Table 1. Multipoint aerodynamic shape optimization problem

	Function/Variable	Description	Quantity
Minimize	$\sum_{i=1}^{N_m} \omega_i c_{d,i}$	Weighted drag coefficients sum	1
With respect to	$\mathbf{x}_s$	Hicks-Henne basis function coefficients	n_s
	$\mathbf{x}_t$	Angles of attack	N_m
Subject to	$c_{\ell,i} - c^*_{\ell,i} = 0$	Lift constraints	N_m
	$A - A_{\min} \geq 0$	Volume constraint	1

At each iteration in the optimization, the objective function needs to be re-evaluated, which requires a flow solution on the updated geometry, and hence a new mesh must be obtained every time. Regeneration of a mesh, especially for a complex geometry or with high resolution, could be time-consuming and non-trivial. Thus, an efficient way to update the computational mesh is needed, and in this work, we use an inverse distance weight interpolation [38] to deform the mesh.

V.C. Optimization Algorithm

Sequential Least Squares Programming (SLSQP) [39] with Broyden-Fletcher-Goldfarb-Shanno (BFGS) [40] type Hessian approximation is used in this work. The weak Wolfe condition is used to terminate the line search, ensuring a sufficient decrease at each optimization step. The gradients of the objective function are calculated by the adjoint method, per Eqn. 15, and the objective is evaluated with the numerical solution of Eqn. 16 based on the discretization given in Section III.

Instead of optimizing on a mesh with fixed resolution, which would always require the highest fidelity for accurate calculations, the mesh is progressively refined as the optimization proceeds, resulting in a variable-fidelity optimization. Rather than performing optimization and mesh adaptation sequentially, one after another, an interactive framework is introduced. Two possible ways to incorporate the mesh adaptation and design optimization are considered here: optimization-driven adaptation and adaptation-driven optimization. In the former approach, the optimization tolerance at each fidelity is prescribed by the user. Starting with a loose optimization tolerance, the total maximum allowable error, set to be equal to the optimization tolerance, is first divided into each design point by Eqn. 24. The objective function at each design point is then evaluated on the same coarse initial mesh, and the error estimation and mesh adaptation are performed individually to control the numerical error to be below the error tolerance at current fidelity. The allowable numerical error decreases as the optimization fidelity increases. For the latter approach, several mesh levels (degrees of freedom) are defined before the optimization. Starting with a low total cost, the degrees of freedom are redistributed according to Eqn. 28. Then the mesh is optimized at each design point to achieve the best accuracy. Once the objective change or the gradient norm is smaller than the objective error estimate, the optimization terminates at the current cost level and the fidelity increases through mesh optimization with a higher cost. This approach is designed for MOESS to optimize the mesh for a given cost at each fidelity. We refer to these methods as error-based or cost-based, depending on the information specified.

Compared with traditional optimization methodology with an *a priori* mesh, unnecessarily fine meshes at the early stages of shape optimization are avoided in the proposed variable-fidelity framework. The problem setup time is significantly reduced with easier mesh generation. Moreover, the elements that introduce most of the error may differ a lot for different shape configurations during the optimization. Both approaches reduce the chance of over-refining elements that are not relatively important for the final design, which is necessary if the adaptation mechanics do not allow for coarsening. On the other hand, the coupling between error tolerance and optimization tolerance at each fidelity actively controls the optimization at each step to avoid unnecessary convergence at low fidelity. Finally, the new framework can effectively prevent over-refining on an unintended shape, or over-optimizing on a coarse mesh.

The proposed optimization frameworks with error estimation and mesh adaptation are summarized in Algorithm 1 and Algorithm 2, using error-based and cost-based approaches, respectively. Optimization tolerance levels and cost levels are specified by the user, driving the mesh adaptation to actively control the numerical errors. In this paper, we assume the error estimation is sufficiently accurate to represent the "true" numerical error, which may be inappropriate when the adjoint is not well-resolved or when the problem is highly nonlinear. In practice, a safety factor η can be used to ensure the numerical error to be always below

the optimization tolerance; $\eta = 1$ is adopted in this paper.

Algorithm 1: Optimization with error estimation and mesh adaptation (error-based)

input : initial design $\mathbf{x}_0$, initial coarse mesh $\mathcal{T}_h$, optimization tolerance levels $\tau_0, \tau_1, ..., \tau_n$, safety factor $\eta \leq 1$
output: adapted meshes at each design point $\mathcal{T}_{h,i}$
optimized design $\mathbf{x}^*$ with controlled objective error $\mathcal{E}(J^{\text{adapt}}_{m,h}) \leq \tau_n$

for $l = 0, 1, ..., n$ **do**
  set total error tolerance as $\mathcal{E}_l \leftarrow \eta\tau_l$
  equidistribute error tolerance among various design points, $\mathcal{E}_{l,i} = \mathcal{E}_l / N_m$
  while *not converged* **do** ▷ optimization algorithm
    for $i = 1, ..., N_m$ **do**
      while $\delta J^{adapt}_{m,i} > \mathcal{E}_{l,i}$ **do**
        trim (update $\mathbf{x}_{t,l}$) and adapt the mesh $\mathcal{T}_{h,i}$ ▷ trimming process
      end
      compute objective function $J^{\text{adapt}}_{h,i}(\mathbf{x}_{s,l})$
    end
    update the composite objective $J^{\text{adapt}}_{h,m} = \sum_{i=1}^{N_m} \omega_i J^{\text{adapt}}_{h,i}$
    calculate composite objective gradient $dJ^{\text{adapt}}_{h,m}/d\mathbf{x}_{s,l}$ and update the design $\mathbf{x}_{s,l}$ ▷ line search
  end
  finish optimization at level l, $\mathbf{x}_{l+1} = \mathbf{x}_l$
end

Algorithm 2: Optimization with error estimation and mesh adaptation (cost-based)

input : initial design $\mathbf{x}_0$, initial coarse mesh $\mathcal{T}_h$, cost levels $C_0, C_1, ..., C_n$, safety factor $\eta \geq 1$
output: optimized mesh at each design point $\mathcal{T}_{h,i}$ with total cost C_n
optimized design $\mathbf{x}^*$ with optimal accuracy at given total cost C_n

Initialize the cost among various design points, $C_{0,i} = C_0 / N_m$
for $l = 0, 1, ..., n$ **do**
  while *not converged* **do** ▷ optimization algorithm
    for $i = 1, ..., N_m$ **do**
      trim (update $\mathbf{x}_{t,l}$) and optimize the mesh $\mathcal{T}_{h,i}$ at fixed cost $C_{l,i}$ ▷ trimming process
      compute objective function $J^{\text{adapt}}_{h,i}(\mathbf{x}_{s,l})$ and its error estimate $\delta J^{\text{adapt}}_{m,i}(\mathbf{x}_{s,l})$
    end
    update the composite objective $J^{\text{adapt}}_{h,m} = \sum_{i=1}^{N_m} \omega_i J^{\text{adapt}}_{h,i}$
    set optimization tolerance $\tau_l = \eta \sum_{i=1}^{N_m} \delta J^{\text{adapt}}_{m,i}$
    calculate objective gradient $dJ^{\text{adapt}}_{h,m}/d\mathbf{x}_{s,l}$ and update the design $\mathbf{x}_{s,l}$ ▷ line search
    redistribute the computational cost among different points $C_{l,i} = \frac{f_i}{\sum_{j=1}^{N_m}} C_l$
  end
  finish optimization at level l, $\mathbf{x}_{l+1} = \mathbf{x}_l$
end

VI. Results

As a simple demonstration of the proposed optimization framework, we consider two-dimensional airfoil shape optimization problems in transonic flow regimes, over a range of flight conditions. The goal of the optimization is to search for an optimal shape and angles of attack to minimize the drag coefficients, subject to fixed lift trim conditions and minimum volume constraint. We only consider the discretization errors in the drag and lift calculations, and the airfoil volume measurements are assumed to be exact. Furthermore,

the trimming constraints tolerance is always set to be small enough to make sure the sensitivity calculation in Eqn. 15 is accurate. The airfoil geometry is parameterized with 16 Hicks-Henne basis functions, and the design parameter vector includes both the shape parameters and the angle of attack at each design point. Unstructured triangular meshes and DG $p = 2$ approximation are used for the discretization. The airfoil boundary is represented by cubic curved mesh elements. We first test our proposed methods in a two-point, inviscid transonic airfoil optimization problem, following which a more practical turbulent case including three flight conditions is considered.

VI.A. Multipoint Inviscid Transonic Airfoil Optimization

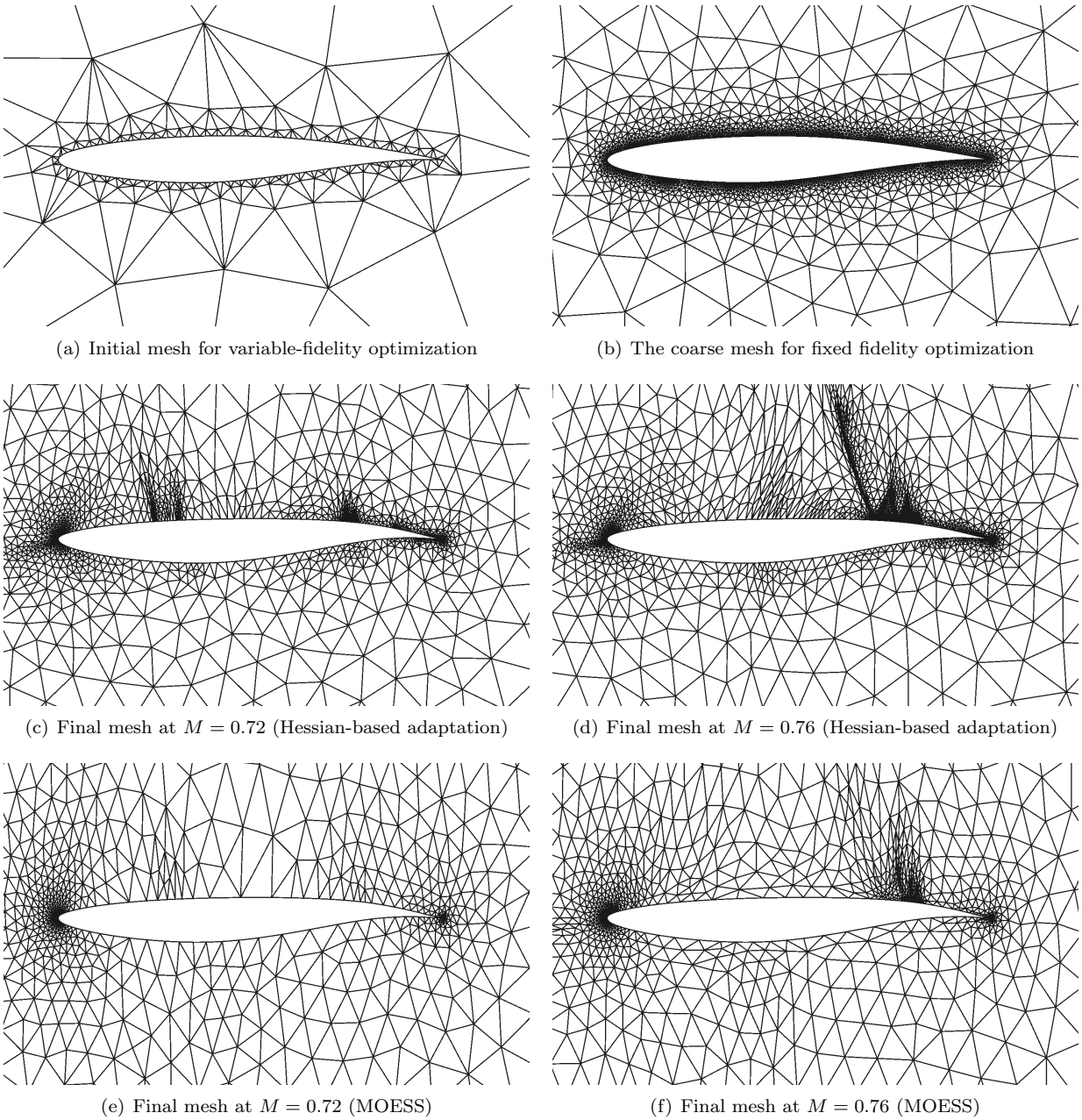
In this test case, the two-point optimization starts with a Royal Aircraft Establishment (RAE) 2822 airfoil at Mach numbers of $M_1 = 0.72$ and $M_2 = 0.76$, and seeks an optimal shape and angles of attack to minimize the weighted drag coefficients (equal weights are used, $\omega_1 = \omega_2 = 0.5$) subject to the fixed lift trim constraints $c^*_{\ell,1} = c^*_{\ell,2} = 0.75$ and the minimum volume as 95% of the initial value.

Under the lift trim condition, flow around the original RAE 2822 airfoil features a strong shock on the upper surface, the location and strength of which vary depending on the operating conditions (e.g. Mach number). Without any prior knowledge about the flow fields around the airfoil under Mach numbers of 0.72 and 0.76, a fairly fine mesh with specific refinement around the airfoil is generally used in optimization. Effort can be put in generating meshes suitable for capturing the shocks effectively, either based on experience or output-based error estimates. However, this only helps the analysis on the original shape; if the shock moves or its strength reduces as the optimization proceeds, this resolution is wasted. Particularly, in this case, we expect that the shape would be modified during the optimization such that the shock strength is significantly weakened. Any substantial refinement on the initial shock location will thus not effectively increase the accuracy but instead add significant computational cost to the optimization.

We start the optimization with the same initial mesh consisting of 393 triangular elements, as shown in Figure 1(a). For the shape optimization with error estimation and mesh adaptation, two different anisotropic mesh adaptation mechanics are used: Hessian-based adaptation, and mesh optimization through error sampling and synthesis (MOESS). The former uses an error-based multifidelity optimization framework, while the latter adopts a cost-based one, as described in Algorithm 1 and Algorithm 2, respectively. The optimization with Hessian-based mesh adaptation is performed using user-specified multiple error levels with an ultimate tolerance of 0.02 drag counts, i.e., 2×10^{-6}. On the other hand, the optimization using MOESS starts at a fairly low cost level, and degrees of freedom are added once the optimization converges at the current cost level. The optimization stops when the total objective error (setting to be the optimization tolerance) is below 0.02 drag counts. To compare with traditional methods, we also run fixed-fidelity optimization on two fixed meshes (the coarse one has comparable DOF as the finest meshes used in the variable-fidelity optimization while the fine mesh has double the cost). The optimization tolerances are also set to be 0.02 drag counts. The meshes used in these different optimization strategies are summarized in Figure 1(b)–1(f). Only the coarse mesh used in the fixed-fidelity optimization is shown for conciseness, as the finer one has more elements but similar uniform refinement around the airfoil boundary.

The objective convergence history and mesh evolution are shown in Figure 2. From the convergence plot Figure 2(a), we see that the estimated discretization error of the objective is always above the optimization tolerance during the fixed-fidelity (fixed-mesh) optimization. On the coarse fixed mesh, the discretization error is comparable to or even larger than the objective value. Although the objective error decreases as a finer fixed mesh is used, it is still fairly large compared to the optimization tolerance. In these cases, the optimizer may work on the numerical error instead of the physics to minimize the drag, leading to inaccurate designs. On the other hand, discretization error is always controlled to be below the optimization tolerance, or the optimization tolerance is adjusted to be equal to the discretization error in the proposed methods. Moreover, the variable-fidelity optimizations with different adaptation techniques both converge faster at the highest fidelity by virtue of better starting shapes obtained from the lower fidelity. Significant computational resources can be saved with fast, low-fidelity optimizations.

We can also observe in the mesh evolution plot in Figure 2(b), that the mesh sizes required to achieve similar accuracy on different operating conditions are different. For both mesh adaptation mechanics, the final mesh size for Mach number of 0.72 is less than the one used for Mach number of 0.76. Mesh adaptation prevents unnecessarily fine meshes from being used for relatively simple operating conditions. Furthermore, we can see that at the low optimization levels, optimizations with both adaptation methods have similar cost, while optimization with MOESS achieves better accuracy so that the shape obtained at lower fidelities



(a) Initial mesh for variable-fidelity optimization

(b) The coarse mesh for fixed fidelity optimization

(c) Final mesh at $M = 0.72$ (Hessian-based adaptation)

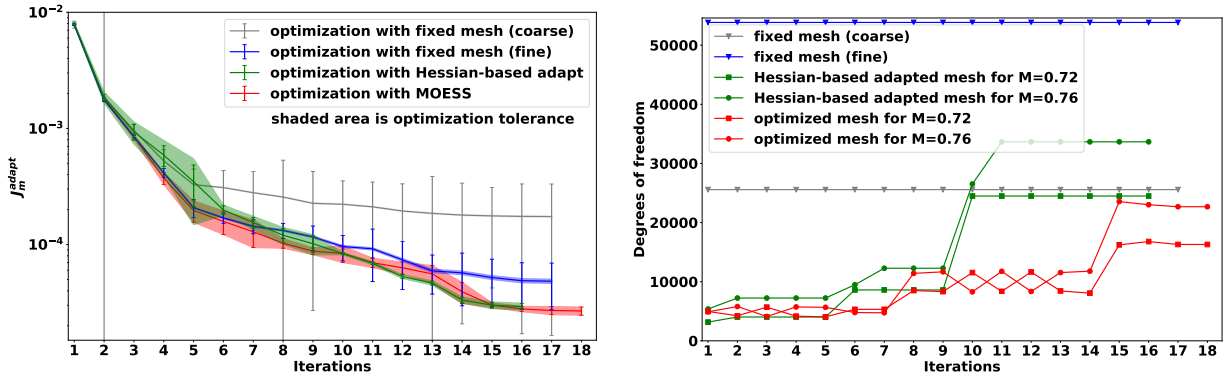
(d) Final mesh at $M = 0.76$ (Hessian-based adaptation)

(e) Final mesh at $M = 0.72$ (MOESS)

(f) Final mesh at $M = 0.76$ (MOESS)

Figure 1. Meshes for variable-fidelity and fixed-fidelity optimization (inviscid, transonic)

is better, resulting in faster convergence at the highest fidelity. At the highest fidelity, MOESS needs less cost to achieve similar accuracy compared to Hessian-based adaptation. These accuracy improvements and cost savings are achieved by redistributing the degrees of freedom among various flight conditions and optimizing the mesh at each of them. The computational cost saving is reflected in Table 2: the proposed method with Hessian-based adaptation achieves around 7 times speedup compared to fixed-fidelity optimization with the fine mesh, while optimization with MOESS attains more than 15 times speedup.



(a) Objective convergence history

(b) Mesh size evolution

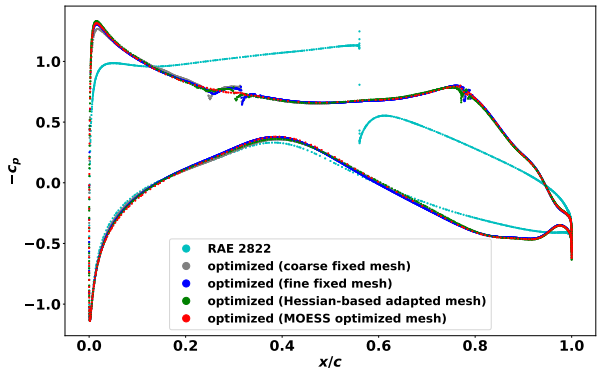
Figure 2. Convergence history and mesh size evolution for different methods (inviscid, transonic)

Table 2. Computational cost comparison (inviscid, transonic)

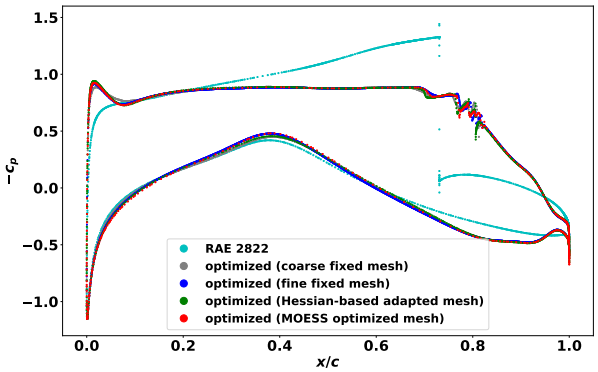
	Optimization level	Optimization tol (Drag count)	Wall time (core*hours)
Fixed fidelity (coarse)	L3	0.02	164.67
Fixed fidelity (fine)	L3	0.02	572.67
Variable-fidelity (Hessian-based)	L1	2.00	3.20
	L2	0.20	9.73
	L3	0.02	75.33
Variable-fidelity (MOESS)	L1	$\delta J_m^{\text{adapt}}$	4.13
	L2		13.6
	L3		13.87

The initial and optimized airfoils are compared in Figure 3. The final objective values are also collected in Table 3; the corresponding "true" objective values are obtained on adapted meshes on the optimized designs to reduce the discretization error to be one order of magnitude smaller than the optimization tolerance. All of the optimization strategies flatten the upper surface near the forward section, while curving and increasing the thickness via the lower surface. The curvature reduction on the top surface smooths the flow acceleration region to weaken the shock. The thickened lower surface and curved aft section are required to maintain the lift and area constraints. Therefore, the strong shock is significantly reduced at both operating conditions, as shown in the pressure distributions in Figure 3(a) and Figure 3(b). In the optimization runs with mesh adaptation, areas around the leading and trailing edges are significantly refined, and many elements are dedicated to the upper surface when discontinuities appear. However, in the optimization with fixed meshes, elements are not efficiently distributed, and areas where flow discontinuities occur are not refined, which causes high objective error as seen in the convergence plot shown in Figure 2(a). When the numerical error is too high, for example on the coarse fixed mesh, the optimization on that mesh converges to a noticeably different design compared to designs obtained from other optimizations, as shown in Figure 3(c). Note particularly differences on the upper surface, to which the drag is most sensitive. As a result, the "true" objective value for the optimized design on the coarse fixed mesh is much larger compared to designs obtained on other meshes. In the optimization with the fine fixed mesh, the discretization error is still high, but the optimization is able to get a similar design compared to the designs produced by optimizations with discretization error control as shown in Figure 3(c). Although the "true" objective value is also close to the objective values of our proposed methods with mesh adaptation (the difference among these three methods

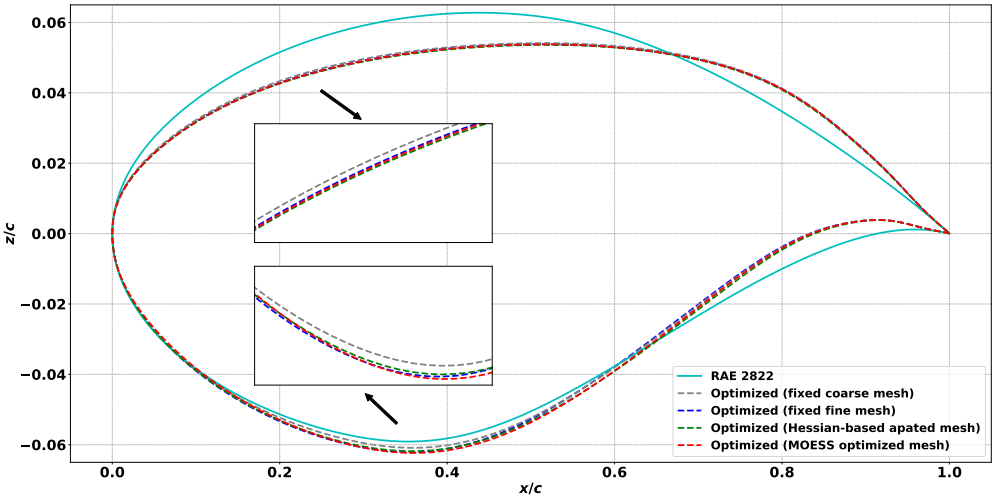
is below or comparable to the optimization tolerance, which means that the optimization on these meshes converges correctly), the final objective value reported on the fixed mesh is far from accurate for practical design and the cost is extremely high compared to our proposed methods, which is shown in both Table 2 and Table 3. The two proposed methods, with either Hessian-based adaptation or MOESS, are able to obtain a reasonable design, and the associated error estimation is also accurate enough to provide confidence in the final design and computed output quantities.



(a) Pressure distribution at $M = 0.72$



(b) Pressure distribution at $M = 0.76$



(c) Initial and optimized shapes

Figure 3. Pressure distribution for the initial and optimized designs (inviscid, transonic)

Table 3. Optimization results on different meshes (inviscid, transonic)

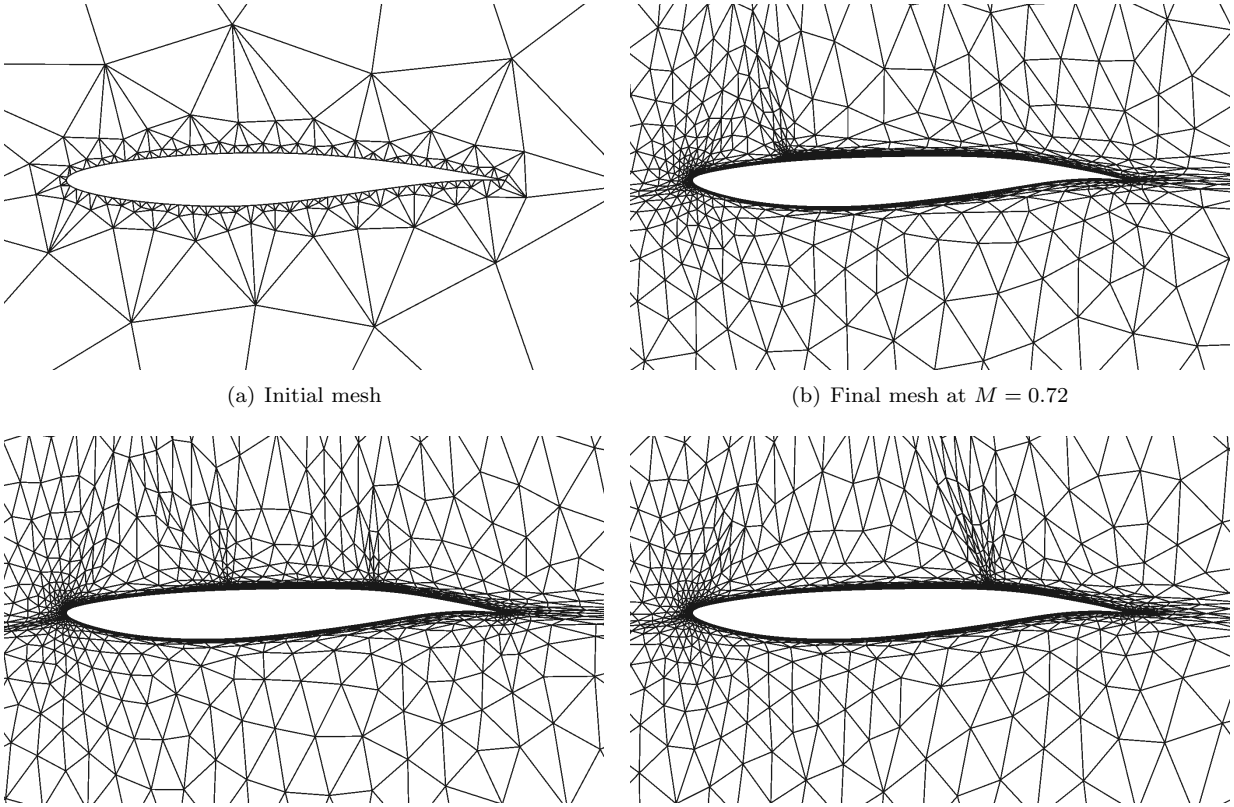
	Total final DOF	J_m^{adapt}	J_m^{adapt} ("true")
Fixed mesh (coarse)	51144	$1.742 \times 10^{-4} \pm 1.577 \times 10^{-4}$	3.973×10^{-5}
Fixed mesh (fine)	107688	$4.833 \times 10^{-5} \pm 2.090 \times 10^{-5}$	2.778×10^{-5}
Hessian-based adapt	58140	$2.927 \times 10^{-5} \pm 1.727 \times 10^{-6}$	2.765×10^{-5}
MOESS	38976	$2.679 \times 10^{-5} \pm 1.458 \times 10^{-6}$	2.533×10^{-5}

VI.B. Multipoint Turbulent Transonic Airfoil Optimization

Another problem considered in this paper is a more sophisticated fully-turbulent case. We set up a three-point optimization problem at Reynolds number of $Re = 6.5 \times 10^6$, with design Mach numbers as $M_1 = 0.72$, $M_2 = 0.74$ and $M_3 = 0.76$. The optimization again starts with the RAE 2822 airfoil, seeks an optimal shape

and angles of attack to minimize the composite drag coefficients (equal weights are used, $\omega_1 = \omega_2 = \omega_3 = 1/3$) subject to fixed lift constraints $c^*_{\ell,1} = c^*_{\ell,2} = c^*_{\ell,3} = 0.75$ and the same volume constraints used in Section VI.A.

For turbulent flow simulation at high Reynolds number, one of the key flow features is the thin boundary layer. Due to the linear velocity profile in the viscous sub-layer, Hessian-based mesh adaptation is usually inefficient since the Mach number Hessian is close to zero. Therefore, only MOESS with cost-based variable-fidelity optimization is used in this case. In the mesh optimization for turbulent airfoils, many degrees of freedom are put into the boundary layer with highly-anisotropic elements around the airfoil boundary. Once the boundary layer is properly resolved, the discretization error drops quickly. Then, more degrees of freedom are added to the upper surface, where a shock is present. The starting and final meshes at each flight condition are compared in Figure 4.



(a) Initial mesh

(b) Final mesh at $M = 0.72$

(c) Final mesh at $M = 0.74$

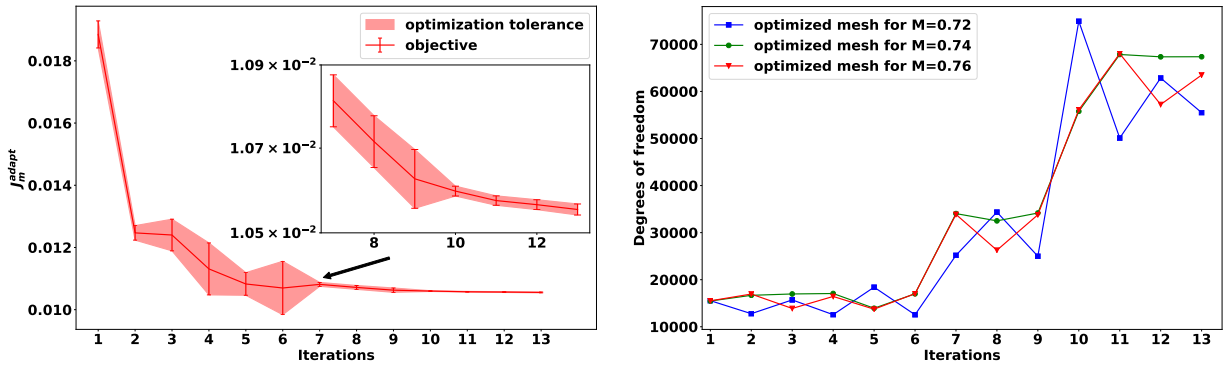
(d) Final mesh at $M = 0.76$

Figure 4. Initial mesh and final meshes during the optimization (turbulent, transonic)

The objective and mesh size are collected at each optimization step as shown in Figure 5. In the convergence plot, we can see that the composite objective errors are close even for different designs during the optimization, as the total degrees of freedom are optimally distributed among different flight conditions, and the meshes are optimized individually at each flight condition. Although some oscillations on the objective error are observed at the lowest optimization cost, mainly due to large shape changes and not well-resolved adjoint solutions. As the shape is optimized and the meshes are refined, the objective error also reduces and remains stable at the medium and highest cost levels, i.e., both the shape and meshes converge to the optimum.

Figure 6(a)–6(c) shows the initial and final pressure distribution for each design point, the corresponding final airfoil shape is shown in Figure 6(e). Similar to the inviscid case, the upper surface of the airfoil is flattened, meanwhile more curvature is added to the lower surface aft section. As we can see in the pressure distribution plots, the strong shock at each design point is either significantly reduced or the location of the shock is moved, resulting a huge reduction on the composite drag coefficient shown in Figure 5(a). Figure 6(d) shows the values of the drag coefficients under a range of Mach numbers with fixed lift trim condition, $c^*_l = 0.75$, for both the original RAE 2822 airfoil and the optimized design. The new design achieves significantly lower drag values for Mach numbers above 0.72, while the performance on lower cruise

Mach numbers is only slightly sacrificed as these points are not included in the optimization.



(a) Objective convergence history

(b) Mesh size evolution

Figure 5. Convergence history and mesh size evolution for different methods (turbulent, transonic)

VII. Conclusion

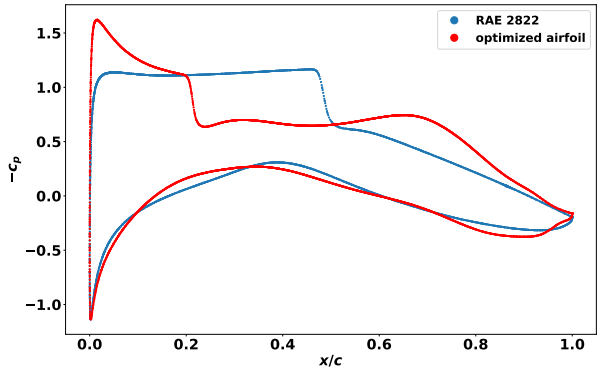
In practical aerodynamic design processes, the optimization problem has to be posed such that a range of operating conditions, including off-design points, are considered in the objective as well as the constraints. To ensure the convergence to the "true" optimal design, the numerical error at each design point has to be carefully controlled. As the flow conditions involved can vary dramatically, *a priori* meshes appropriate for all the design points can be hard to generate and are generally not sufficient for the requirements of high-fidelity optimization.

In this work, we presented a variable-fidelity framework that integrates output-based error estimation and mesh adaptation with a gradient-based algorithm for multipoint aerodynamic shape optimization problems. The proposed framework can considerably facilitate the optimization setup and accelerate the design process. The designer only needs to input an initial mesh, which can be fairly coarse and easy to generate. The mesh adaptation (fidelity increase) is then tightly coupled with the optimization algorithm either with an error-based or a cost-based strategy. The variable-fidelity optimization framework driven by mesh adaptation is capable of preventing over-optimizing and over-refining, as shown in the test cases. Design optimization with mesh optimization via error sampling and synthesis (MOESS) is shown to be more efficient and effective by optimized computational cost distribution among various flight conditions and optimized meshes at each point. This benefit can become more significant when higher fidelity is required, or when more highly anisotropic physics govern the system.

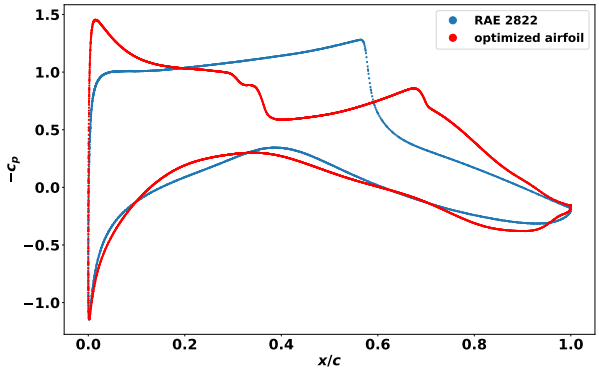
With more judicious considerations of the objective functions and constraints, and additional parameters, the new method can provide realistic configurations in practical design scenarios. Also, the computational cost allocation adopted in this work is based on error equidistribution, which can be inefficient when different convergence rates occur among the cases considered, or when any design point stalls the objective error convergence. More appropriate ways to distribute the computational costs can be developed. Furthermore, only mesh adaptation (h-adaptation) is considered here to control the discretization error. More efficient adaptation mechanics such as approximation order increment (p-adaptation), and combinations (hp-adaptation) can also be applied to the proposed methods in the future.

Acknowledgments

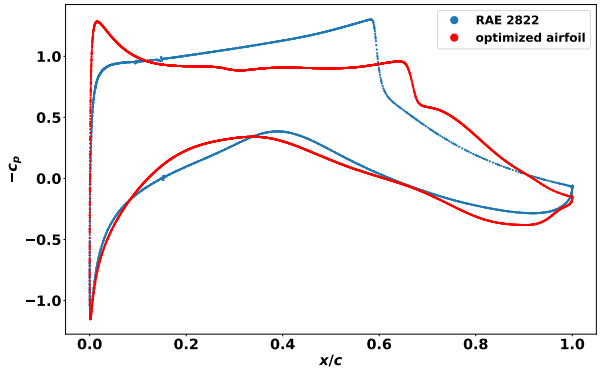
The authors acknowledge the support of the Boeing Company, with technical monitor Dr. Mori Mani, and the Department of Energy under grant DE-FG02-13ER26146/DE-SC0010341.



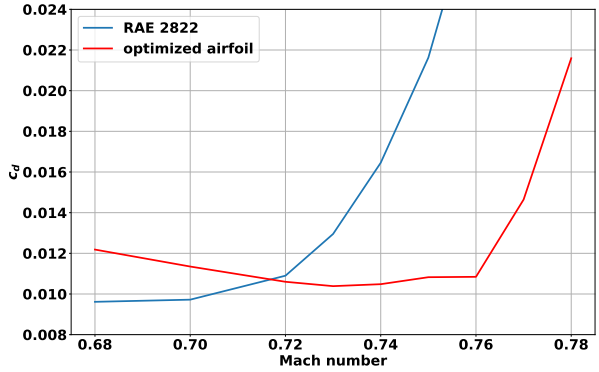
(a) Pressure distribution at $M = 0.72$



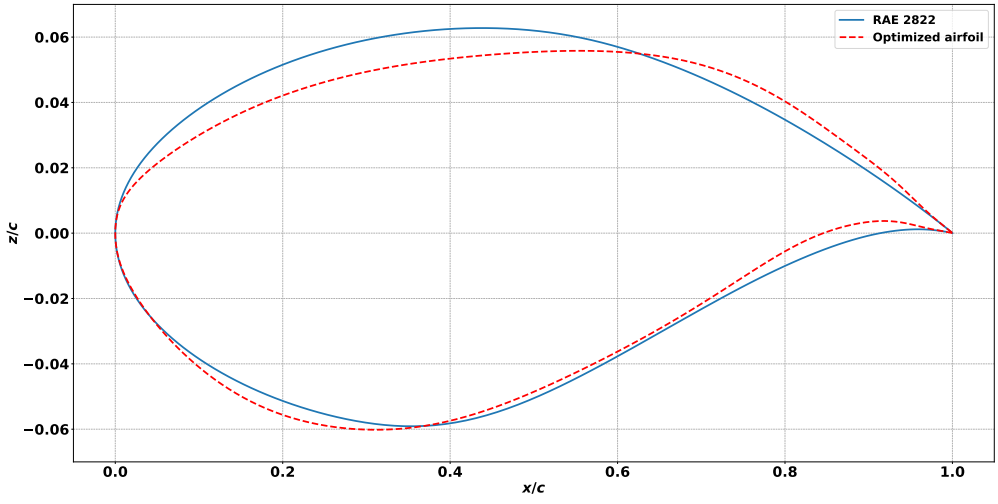
(b) Pressure distribution at $M = 0.74$



(c) Pressure distribution at $M = 0.76$



(d) Drag coefficients at $c_l = c_l^* = 0.75$



(e) Initial and optimized shapes

Figure 6. Pressure distribution for the initial and optimized designs

References

[1] Hajela, P., "Genetic search - An approach to the nonconvex optimization problem," *AIAA Journal*, Vol. 28, No. 7, 1990, pp. 1205–1210. doi:10.2514/3.25195, URL https://doi.org/10.2514/3.25195.

[2] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi:10.1007/bf01061285, URL https://doi.org/10.1007/bf01061285.

[3] Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D., "Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation," *34th Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, 1996, p. 0094. doi:10.2514/6.1996-94, URL https://doi.org/10.2514/6.1996-94.

[4] Anderson, W., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," *Computers & Fluids*, Vol. 28, No. 4-5, 1999, pp. 443–480. doi:10.1016/s0045-7930(98)00041-3, URL https://doi.org/10.1016/s0045-7930(98)00041-3.

[5] Giles, M. B., and Pierce, N. A., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, No. 3/4, 2000, pp. 393–415. doi:10.1023/a:1011430410075, URL https://doi.org/10.1023/a:1011430410075.

[6] Mader, C. A., Martins, J. R. R. A., Alonso, J. J., and Weide, E. V. D., "ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers," *AIAA Journal*, Vol. 46, No. 4, 2008, pp. 863–873. doi:10.2514/1.29123, URL https://doi.org/10.2514/1.29123.

[7] Cliff, S. E., Reuther, J. J., Saunders, D. A., and Hicks, R. M., "Single-Point and Multipoint Aerodynamic Shape Optimization of High-Speed Civil Transport," *Journal of Aircraft*, Vol. 38, No. 6, 2001, pp. 997–1005. doi:10.2514/2.2886, URL https://doi.org/10.2514/2.2886.

[8] Nemec, M., Zingg, D. W., and Pulliam, T. H., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065. doi:10.2514/1.10415, URL https://doi.org/10.2514/1.10415.

[9] Zingg, D. W., and Elias, S., "Aerodynamic Optimization Under a Range of Operating Conditions," *AIAA Journal*, Vol. 44, No. 11, 2006, pp. 2787–2792. doi:10.2514/1.23658, URL https://doi.org/10.2514/1.23658.

[10] Kenway, G. K. W., and Martins, J. R. R. A., "Multipoint Aerodynamic Shape Optimization Investigations of the Common Research Model Wing," *AIAA Journal*, Vol. 54, No. 1, 2016, pp. 113–128. doi:10.2514/1.j054154, URL https://doi.org/10.2514/1.j054154.

[11] Hicken, J., and Alonso, J., "PDE-constrained optimization with error estimation and control," *Journal of Computational Physics*, Vol. 263, 2014, pp. 136–150. doi:10.1016/j.jcp.2013.12.050, URL https://doi.org/10.1016/j.jcp.2013.12.050.

[12] Chen, G., and Fidkowski, K. J., "Discretization error control for constrained aerodynamic shape optimization," *Journal of Computational Physics*, Vol. 387, 2019, pp. 163–185. doi:10.1016/j.jcp.2019.02.038, URL https://doi.org/10.1016/j.jcp.2019.02.038.

[13] Becker, R., and Rannacher, R., "An optimal control approach to a posteriori error estimation in finite element methods," *Acta Numerica*, Vol. 10, 2001. doi:10.1017/s0962492901000010, URL https://doi.org/10.1017/s0962492901000010.

[14] 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. doi:10.1006/jcph.2002.7206, URL https://doi.org/10.1006/jcph.2002.7206.

[15] Venditti, D. A., and Darmofal, D. L., "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows," *Journal of Computational Physics*, Vol. 176, No. 1, 2002, pp. 40–69. doi:10.1006/jcph.2001.6967, URL https://doi.org/10.1006/jcph.2001.6967.

[16] Park, M. A., "Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation," *AIAA Journal*, Vol. 42, No. 9, 2004, pp. 1854–1862. doi:10.2514/1.10051, URL https://doi.org/10.2514/1.10051.

[17] Kast, S. M., and Fidkowski, K. J., "Output-based mesh adaptation for high order Navier–Stokes simulations on deformable domains," *Journal of Computational Physics*, Vol. 252, 2013, pp. 468–494. doi:10.1016/j.jcp.2013.06.007, URL https://doi.org/10.1016/j.jcp.2013.06.007.

[18] Ceze, M., and Fidkowski, K. J., "Drag Prediction Using Adaptive Discontinuous Finite Elements," *Journal of Aircraft*, Vol. 51, No. 4, 2014, pp. 1284–1294. doi:10.2514/1.c032622, URL https://doi.org/10.2514/1.c032622.

[19] Shimizu, Y. S., and Fidkowski, K. J., "Output-Based Error Estimation for Chaotic Flows Using Reduced-Order Modeling," *2018 AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2018, p. 0826. doi:10.2514/6.2018-0826, URL https://doi.org/10.2514/6.2018-0826.

[20] Lu, J., "An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2005. Available: http://hdl.handle.net/1721.1/34134,.

[21] Nemec, M., and Aftosmis, M., "Output Error Estimates and Mesh Refinement in Aerodynamic Shape Optimization," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, 2013, p. 0865. doi:10.2514/6.2013-865, URL https://doi.org/10.2514/6.2013-865.

[22] Li, D., and Hartmann, R., "Adjoint-based airfoil optimization with discretization error control," *International Journal for Numerical Methods in Fluids*, Vol. 77, No. 1, 2015, pp. 1–17. doi:10.1002/fld.3971, URL https://doi.org/10.1002/fld.3971.

[23] Chen, G., and Fidkowski, K., "Airfoil Shape Optimization Using Output-Based Adapted Meshes," *23rd AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2017, p. 3102. doi:10.2514/6.2017-3102, URL https://doi.org/10.2514/6.2017-3102.

[24] Wang, K., Yu, S., Wang, Z., Feng, R., and Liu, T., "Adjoint-based airfoil optimization with adaptive isogeometric discontinuous Galerkin method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 344, 2019, pp. 602–625. doi:10.1016/j.cma.2018.10.033, URL https://doi.org/10.1016/j.cma.2018.10.033.

[25] Rothacker, B. A., Ceze, M., and Fidkowski, K., "Adjoint-Based Error Estimation and Mesh Adaptation for Problems with Output Constraints," *32nd AIAA Applied Aerodynamics Conference*, American Institute of Aeronautics and Astronautics, 2014, p. 2576. doi:10.2514/6.2014-2576, URL https://doi.org/10.2514/6.2014-2576.

[26] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004. doi:10.1017/cbo9780511804441, URL https://doi.org/10.1017/cbo9780511804441.

[27] Allmaras, S., Johnson, F., and Spalart, P., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," Seventh International Conference on Computational Fluid Dynamics (ICCFD7) 1902, 2012.

[28] Bassi, F., and Rebay, S., "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 131, No. 2, 1997, pp. 267–279. doi:10.1006/jcph.1996.5572, URL https://doi.org/10.1006/jcph.1996.5572.

[29] Cockburn, B., and Shu, C.-W., "Runge–Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261. doi:10.1023/a:1012873910884, URL https://doi.org/10.1023/a:1012873910884.

[30] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations," *Journal of Computational Physics*, Vol. 207, No. 1, 2005, pp. 92–113. doi:10.1016/j.jcp.2005.01.005, URL https://doi.org/10.1016/j.jcp.2005.01.005.

[31] Fidkowski, K. J., and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694. doi:10.2514/1.j050073, URL https://doi.org/10.2514/1.j050073.

[32] Zienkiewicz, O. C., and Zhu, J. Z., "Adaptivity and Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 4, 1991, pp. 783–810. doi:10.1002/nme.1620320409, URL https://doi.org/10.1002/nme.1620320409.

[33] 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. doi:10.1016/s0021-9991(03)00074-3, URL https://doi.org/10.1016/s0021-9991(03)00074-3.

[34] Ceze, M., and Fidkowski, K. J., "Anisotropic hp-Adaptation Framework for Functional Prediction," *AIAA Journal*, Vol. 51, No. 2, 2013, pp. 492–509. doi:10.2514/1.j051845, URL https://doi.org/10.2514/1.j051845.

[35] Yano, M., "An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2005. Available: http://hdl.handle.net/1721.1/76090,.

[36] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412. doi:10.2514/3.58379, URL https://doi.org/10.2514/3.58379.

[37] Wu, H.-Y., Yang, S., Liu, F., and Tsai, H.-M., "Comparisons of Three Geometric Representations of Airfoils for Aerodynamic Optimization," *16th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2003, p. 4095. doi:10.2514/6.2003-4095, URL https://doi.org/10.2514/6.2003-4095.

[38] Luke, E., Collins, E., and Blades, E., "A fast mesh deformation method using explicit interpolation," *Journal of Computational Physics*, Vol. 231, No. 2, 2012, pp. 586–601. doi:10.1016/j.jcp.2011.09.021, URL https://doi.org/10.1016/j.jcp.2011.09.021.

[39] Kraft, D., "A software package for sequential quadratic programming," Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Center–Institute for Flight Mechanics, Köln, Germany, 1988.

[40] Broyden, C. G., "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation*, Vol. 19, No. 92, 1965, pp. 577–577. doi:10.1090/s0025-5718-1965-0198670-6, URL https://doi.org/10.1090/s0025-5718-1965-0198670-6.