



# Airfoil Shape Optimization Using Output-Based Adapted Meshes

Guodong Chen\* and Krzysztof J. Fidkowski†

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

In this paper, gradient-based aerodynamic shape optimization with output constraints is implemented using adaptive meshes updated via adjoint-based error estimates. All the constraints, including geometry and trim conditions, are handled simultaneously in the optimization. The trim constraints may involve outputs that are not directly targeted for optimization, and hence also not for error estimation and mesh adaptation. However, numerical errors in these outputs often indirectly affect the calculation of the objective. The method adopted in this work takes this effect into account, so that the mesh is adapted to predict both objective outputs and constraint outputs with appropriate accuracy. In other words, the entire optimization problem is targeted for adaptation. Instead of optimizing with a single fixed mesh resolution, the objective function is first evaluated on a relatively coarse mesh, which is subsequently adapted as the shape optimization proceeds. As the shape approaches the optimal design, the mesh becomes finer, in necessary regions, leading to a multi-fidelity optimization process. The multi-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. The use of output-based error estimates prevents over-optimizing on a coarse mesh, or over-refining on an undesired shape. The proposed framework is demonstrated on several optimization problems, starting with a NACA 0012 airfoil. In each case, the shape is optimized to minimize the drag given a target lift and a minimum airfoil area. The accuracy and efficiency of the proposed method are investigated through comparisons with existing optimization techniques. We expect the framework to be even more important for more complex configurations, dramatic shape changes, or high-accuracy requirements.

## I. Introduction

With the increasing power of modern computers, Computational Fluid Dynamics (CFD) has become common in aerospace design. CFD simulations reduce the need for expensive wind tunnel tests and offer almost arbitrary test conditions and configurations. They are an attractive tool in aerospace design, especially for optimization problems involving a large number of design parameters. Successful use of CFD in practical design problems requires both accurate simulations for a given configuration and efficient optimization methods to improve design configurations.

The design process combines CFD analysis with numerical optimization methods, minimizing the *objective function* based on the CFD output, e.g. drag, lift, or moment. For the optimization algorithms, both gradient-based and gradient-free methods can be used. Gradient-free methods like genetic algorithms<sup>1-3</sup> and neural networks<sup>4,5</sup> may be made robust for non-smooth or non-convex problems,<sup>6</sup> 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.<sup>7</sup>

Aerodynamic shape optimization problems are often parameterized with a large number of design variables and, with the desire for high accuracy, heavily depend on expensive high-fidelity CFD analysis,<sup>7</sup> so

\*Graduate Research Assistant, AIAA Member

†Associate Professor, AIAA Senior Member

that gradient-based methods are extensively used in aerospace design. Gradients with respect to each design parameter are needed at every optimization iteration, which requires an accurate and efficient method for estimating the sensitivities. Methods such as finite differencing, complex-step derivative approximation,<sup>8</sup> and algorithmic differentiation<sup>9</sup> depend on the number of design parameters, while the adjoint method<sup>7,10–13</sup> is largely independent of the number of design parameters.

Even when using gradient-based algorithms with efficient sensitivity analysis, high-fidelity aerodynamic shape optimization still remains computationally taxing because each evaluation of the objective function requires an expensive high-fidelity CFD simulation. Although high-fidelity CFD simulations are now routinely carried out in aerodynamic analysis, the computational resources needed for high-fidelity design preclude its wide-spread use. This is the problem that we tackle in the present work. In order to reduce the optimization cost, adjoint-based output error estimation and mesh adaptation are introduced into the optimization process. The idea to combine output error estimation and gradient-based optimization is natural, as both methods require output adjoint solutions. Even though output-based error estimation and mesh adaptation methods have been demonstrated in a wide range of aerospace computational simulations,<sup>14–20</sup> their application to aerodynamic shape optimization problems has not received much attention. Nemic and Aftosmis<sup>21</sup> incorporated gradient-based optimization with output-based error estimates and mesh refinement, and Dalle and Fidkowski<sup>22</sup> proposed a multi-fidelity optimization framework built on this idea. Both of these approaches used progressive optimization with mesh adaptation solely based on the error of the objective output, without considering the effects of constraint errors on the objective. Li and Hartmann<sup>23</sup> introduced a multi-target adaptation algorithm in which mesh adaptation targets the objective output and the constraint outputs equally on a fixed fidelity.

The present work proposes a new variable-fidelity implementation and modified mesh adaptation strategy. The error estimation not only includes the objective output, but also accounts for the errors in the constraint outputs, which may indirectly affect the objective function.<sup>24</sup> 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. The primary results are shown in Section VI, and Section VII concludes the present work and discusses potential future work.

## II. Optimization Problem

### II.A. Problem Formulation

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  $J$ :

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

where  $J$  represents a scalar objective function, always defined by the output, for example lift or drag or a combination of these for multi-objective optimization.<sup>25</sup>  $\mathbf{U}$  denotes the flow variables, and  $\mathbf{R}^e$  and  $\mathbf{R}^{ie}$  are the equality and inequality constraints, respectively. The flow variables  $\mathbf{U}$  are solved within a feasible design space  $\Omega$  to satisfy the flow equations, often the Euler or Navier-Stokes equations. In discretized form, these consist of a set of nonlinear equations,

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

which implicitly defines  $\mathbf{U}$  as a function of  $\mathbf{x}$ .

## II.B. Adjoint and Design Equations

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

$$\mathbf{R}^{\text{trim}}(\mathbf{U}, \mathbf{x}) = \mathbf{J}^{\text{trim}}(\mathbf{U}, \mathbf{x}) - \bar{\mathbf{J}}^{\text{trim}} = \mathbf{0} \quad (3)$$

where  $\bar{\mathbf{J}}^{\text{trim}}$  is a set of target trim outputs, for example, the target lift in a lift-constrained problem. In order to distinguish the trim outputs from the objective output, we denote the latter by  $J^{\text{adapt}}$ , as the objective output is the 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,

$$\mathcal{L}(\mathbf{U}, \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = J^{\text{adapt}}(\mathbf{U}, \mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{R}(\mathbf{U}, \mathbf{x}) + \boldsymbol{\mu}^T \mathbf{R}^{\text{trim}}(\mathbf{U}, \mathbf{x}) \quad (4)$$

where  $\boldsymbol{\lambda}, \boldsymbol{\mu}$  are the Lagrange multipliers associated with the flow equations and the trim constraints, respectively. The first-order optimality conditions are given by setting the derivatives of  $\mathcal{L}$  to zero,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial J^{\text{adapt}}}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \boldsymbol{\mu}^T \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{x}} = \mathbf{0} \quad (5)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = \frac{\partial J^{\text{adapt}}}{\partial \mathbf{U}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{U}} + \boldsymbol{\mu}^T \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{U}} = \mathbf{0} \quad (6)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{R}(\mathbf{U}, \mathbf{x}) = \mathbf{0} \quad (7)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = \mathbf{R}^{\text{trim}}(\mathbf{U}, \mathbf{x}) = \mathbf{0} \quad (8)$$

As we solve the flow equations for a given design  $\mathbf{x}$  each time, Eqn. 7 is always satisfied during the optimization. We can choose  $\boldsymbol{\lambda}$  such that Eqn. 6 is enforced after each flow solve,

$$\boldsymbol{\lambda}^T = - \left( \frac{\partial J^{\text{adapt}}}{\partial \mathbf{U}} + \boldsymbol{\mu}^T \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{U}} \right) \frac{\partial \mathbf{R}}{\partial \mathbf{U}}^{-1} = (\boldsymbol{\Psi}^{\text{adapt}} + \boldsymbol{\Psi}^{\text{trim}} \boldsymbol{\mu})^T \quad (9)$$

Eqn. 9 gives a coupled adjoint variable  $\boldsymbol{\lambda}$  incorporating the adjoints of both the objective and the trim outputs,  $\boldsymbol{\Psi}^{\text{adapt}}$  and  $\boldsymbol{\Psi}^{\text{trim}}$ , which satisfy

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T \boldsymbol{\Psi}^{\text{adapt}} + \frac{\partial J^{\text{adapt}}}{\partial \mathbf{U}}^T = \mathbf{0}, \quad \frac{\partial \mathbf{R}}{\partial \mathbf{U}}^T \boldsymbol{\Psi}^{\text{trim}} + \frac{\partial \mathbf{J}^{\text{trim}}}{\partial \mathbf{U}}^T = \mathbf{0} \quad (10)$$

With this specific choice of  $\boldsymbol{\lambda}$ , we can evaluate the gradient of the Lagrangian function with respect to the design variables, starting with Eqn. 5,

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{x}} &= \frac{\partial J^{\text{adapt}}}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \boldsymbol{\mu}^T \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{x}} \\ &= \frac{\partial J^{\text{adapt}}}{\partial \mathbf{x}} + (\boldsymbol{\Psi}^{\text{adapt}})^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \boldsymbol{\mu}^T \left[ \frac{\partial \mathbf{R}^{\text{trim}}}{\partial \mathbf{x}} + (\boldsymbol{\Psi}^{\text{trim}})^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}} \right] \\ &= \frac{dJ^{\text{adapt}}}{d\mathbf{x}} + \boldsymbol{\mu}^T \frac{d\mathbf{J}^{\text{trim}}}{d\mathbf{x}} \end{aligned} \quad (11)$$

Now the optimization problem has been reduced to finding an optimal design  $\mathbf{x}$  and the corresponding Lagrange multipliers  $\boldsymbol{\mu}$  satisfying,

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{x}} &= \frac{dJ^{\text{adapt}}}{d\mathbf{x}} + \boldsymbol{\mu}^T \frac{d\mathbf{J}^{\text{trim}}}{d\mathbf{x}} = \mathbf{0} \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} &= \mathbf{R}^{\text{trim}} = \mathbf{0} \end{aligned} \quad (12)$$

However, since in a practical calculation, on a finite-dimensional space, the discretization error appears in both the flow equations and the adjoint equations, optimality cannot be guaranteed even when Eqn. 12 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 over the airfoil. The governing equations for the fluid flow are compressible Navier-Stokes,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) + \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0} \quad (13)$$

where  $\mathbf{u}$  is the flow state vector at a particular spatial location,  $\vec{\mathbf{F}}$  is the total inviscid and viscous flux vector, 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. We discretize Eqn. 13 with the discontinuous Galerkin (DG) finite-element method, which is suitable for high-order accuracy and  $hp$ -refinement.<sup>27-29</sup> 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. 13 yields a system of discrete algebraic equations in the form of Eqn. 2,

$$\mathbf{R}_H(\mathbf{U}_H, \mathbf{x}) = \mathbf{0} \quad (14)$$

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,

$$\text{output error: } \delta J \equiv J_H(\mathbf{U}_H) - J_h(\mathbf{U}_h) \quad (15)$$

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,  $\Psi_h$ , defined as the sensitivity of the output to the residuals. The adjoint weights the residual perturbation to produce an output perturbation,

$$\begin{aligned} \delta J &\approx J_H(\mathbf{U}_H) - J_h(\mathbf{U}_h) = J_h(\mathbf{U}_h^H) - J_h(\mathbf{U}_h) \\ &= -\Psi_h^T [\mathbf{R}_h(\mathbf{U}_h^H) - \mathbf{R}_h(\mathbf{U}_h)] = -\Psi_h^T \mathbf{R}_h(\mathbf{U}_h^H) \end{aligned} \quad (16)$$

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. 16 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 for Optimization Problems

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  $\mathbf{x}$ , and suppose that the error of the objective only comes from the inexact solution  $\mathbf{U}_h^H$ . We can estimate the error in the objective with the linearization given by Eqn. 6,

$$\begin{aligned} \delta J^{\text{adapt}} &= -\lambda_h^T \delta \mathbf{R}_h - \mu_h^T \delta \mathbf{R}_h^{\text{trim}} = -\lambda_h^T \mathbf{R}_h(\mathbf{U}_h^H) - \mu_h^T \delta \mathbf{R}_h^{\text{trim}} \\ &= -(\Psi_h^{\text{adapt}} + \Psi_h^{\text{trim}} \mu_h)^T \mathbf{R}_h(\mathbf{U}_h^H) - \mu_h^T (\mathbf{J}_h^{\text{trim}}(\mathbf{U}_h^H) - \mathbf{J}_h^{\text{trim}}(\mathbf{U}_h)) \\ &= \delta J^{\text{adapt}} - \mu_h^T (\Psi_h^{\text{trim}})^T \mathbf{R}_h(\mathbf{U}_h^H) - \mu_h^T \delta \mathbf{J}^{\text{trim}} \\ &= \delta J^{\text{adapt}} \end{aligned} \quad (17)$$

This is consistent with the previous analysis without the trim conditions, since we keep the same design between the coarse and fine spaces, and because we assume that the error only comes from the inexact solution  $\mathbf{U}_h^H$ . In general, however, we need to deal with both the objective error and the constraints error. The problem becomes worse if we have high accuracy in the objective while little confidence in the constraint outputs, or vice versa. If we run simulations in the fine space and the coarse space, with the same constraint outputs, we will get different designs. This error may come from the deviation of both the design parameters and the flow states, and separate error estimation and mesh adaptation for the objective and trim outputs may be inefficient.

If we consider the optimal design on the coarse space  $(\mathbf{U}_H, \mathbf{x}_H)$  and the fine space  $(\mathbf{U}_h, \mathbf{x}_h)$ , since the optimality conditions Eqn. 5 and Eqn. 6 both hold now, the error comes from the inexact solution  $\mathbf{U}_h^H$  as well as the deficient design  $\mathbf{x}_H$ ,

$$\begin{aligned} \delta J_{\text{opt}}^{\text{adapt}} &= -\lambda_h^T \delta \mathbf{R}_h - \boldsymbol{\mu}_h^T \delta \mathbf{R}_h^{\text{trim}} = -\lambda_h^T \mathbf{R}_h(\mathbf{U}_h^H, \mathbf{x}_H) - \boldsymbol{\mu}_h^T \mathbf{R}_h^{\text{trim}}(\mathbf{U}_h^H, \mathbf{x}_H) \\ &= -(\boldsymbol{\Psi}_h^{\text{adapt}} + \boldsymbol{\Psi}_h^{\text{trim}} \boldsymbol{\mu}_h)^T \mathbf{R}_h(\mathbf{U}_h^H, \mathbf{x}_H) - \boldsymbol{\mu}_h^T (\mathbf{J}_h^{\text{trim}}(\mathbf{U}_h^H, \mathbf{x}_H) - \bar{\mathbf{J}}^{\text{trim}}) \\ &= \delta J^{\text{adapt}}(\mathbf{x}_H) + \boldsymbol{\mu}_h^T \delta \mathbf{J}^{\text{trim}}(\mathbf{x}_H) - \boldsymbol{\mu}_h^T (\mathbf{J}_h^{\text{trim}}(\mathbf{U}_h^H, \mathbf{x}_H) - \bar{\mathbf{J}}^{\text{trim}}) \end{aligned} \quad (18)$$

Since the definition of the outputs is often the same on the coarse and fine spaces, we have that

$$\mathbf{J}_h^{\text{trim}}(\mathbf{U}_h^H, \mathbf{x}_H) = \mathbf{J}_H^{\text{trim}}(\mathbf{U}_H, \mathbf{x}_H) = \bar{\mathbf{J}}^{\text{trim}} = \mathbf{J}_h^{\text{trim}}(\mathbf{U}_h, \mathbf{x}_h) \quad (19)$$

Hence, the last term in Eqn. 18 is often negligible for the optimal design, resulting a simpler form for the error of the optimality condition,

$$\delta J_{\text{opt}}^{\text{adapt}} = -(\boldsymbol{\Psi}_h^{\text{adapt}} + \boldsymbol{\Psi}_h^{\text{trim}} \boldsymbol{\mu}_h)^T \mathbf{R}_h(\mathbf{U}_h^H, \mathbf{x}_H) = \delta J^{\text{adapt}}(\mathbf{x}_H) + \boldsymbol{\mu}_h^T \delta \mathbf{J}^{\text{trim}}(\mathbf{x}_H) \quad (20)$$

The error without the subscript opt is the output error without the trim constraints. Eqn. 20 gives a prediction of optimality error on the coarse space due to the spatial discretization. With the advantages of adjoint-based error estimation, we avoid the expensive solves of both the the optimal design  $\mathbf{x}_h$  and flow states  $\mathbf{U}_h$ , i.e. the whole optimization process on the fine space. However, the estimation requires the fine-space adjoints  $\boldsymbol{\Psi}_h$  as well as the fine-space Lagrange multipliers  $\boldsymbol{\mu}_h$ . In our implementation, the fine-space adjoints  $\boldsymbol{\Psi}_h$  are approximated by reconstructing the coarse-space adjoints  $\boldsymbol{\Psi}_H$ ,<sup>19</sup> while the Lagrange multipliers are extracted from the optimizer on the coarse space.

Eqn. 20 provides the error estimate of the optimality condition, which does not always hold during the optimization process. Thus, it is neither the error of the objective nor the error of the constraints when the design is away from the optimality condition. However, it couples the objective error and constraints error, giving a better error level of the whole optimization problem, so it is expected to serve as a better adaptation indicator for the optimization or constraint problems. Moreover, in the multi-fidelity optimization framework, when most of the mesh adaptation happens after a successful optimization on the current fidelity, Eqn. 20 works better than the objective error without constraints.

#### IV.C. Mesh Adaptation

The error estimate is localized to each element and serves as the indicator for mesh adaptation. A common approach for obtaining an error indicator is to take the absolute value of the elemental error contribution, the residual vector inside each element dotted with the discrete adjoint vector. When the trim outputs are involved, we have another decision to make: whether we would 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  $k$  is given by

$$\eta_k = \eta_k^{\text{adapt}} + |\boldsymbol{\mu}^T| \boldsymbol{\eta}_k^{\text{trim}} \quad (21)$$

where  $\eta_k^{\text{adapt}}$  is the objective output error indicator, and  $\boldsymbol{\eta}_k^{\text{trim}}$  is a vector of trim output error indicators. More details about the mesh adaptation strategy in the multi-fidelity optimization are given in the next section.

## V. Optimization Approach

### V.A. Optimization Algorithm

To be consistent with the analysis in the adjoint and design equations in Section II.B, the optimization algorithm should use the Lagrange multipliers. Sequential Least Squares Programming (SLSQP)<sup>30</sup> with Broyden-Fletcher-Goldfarb-Shanno (BFGS)<sup>31</sup> type Hessian approximation is used in this work. Both the equality and inequality constraints are handled by the optimizer. The weak Wolfe condition is used to terminate the line search, ensuring a sufficient decrease at each optimization step. The Lagrange multipliers associated with the trim constraints are extracted after each optimization step as a surrogate for the fine-space multipliers. The gradients of the objective function are calculated by the adjoint method, per Eqn. 12, and the objective is evaluated with the numerical solution of Eqn. 13 based on the discretization given in Section III.

### V.B. 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 angle of attack) to minimize the drag subject to a fixed lift coefficient and a minimum airfoil area. The objective is the drag coefficient of the airfoil, and the corresponding constraints are

$$R^e(\mathbf{U}, \mathbf{x}) = c_\ell(\mathbf{U}, \mathbf{x}) - c_\ell^* = 0 \quad (22)$$

$$R^{ie}(\mathbf{U}, \mathbf{x}) = A(\mathbf{x}) - A_{\min} \geq 0 \quad (23)$$

where  $A$  and  $A_{\min}$  represent the current and minimum areas of the airfoil, and  $c_\ell$  and  $c_\ell^*$  denote the current and target lift coefficients.

Since we allow the infeasible region of the trim constraints, the angle of attack  $\alpha$  is also included in the design variables. Sometimes the angle of attack can serve solely as a trim variable to enforce the fixed lift coefficient, then the trim conditions are satisfied all the time, and the error estimation in Eqn. 20 is valid for each optimization step.

### V.C. Airfoil Parameterization

The airfoil shape is parameterized using the deformation method proposed by Hicks and Henne,<sup>32</sup> taking a baseline airfoil and creating a new airfoil shape by adding a linear combination of “bump” functions to its upper and lower surfaces,

$$z = z_{\text{base}} + \sum_{i=0}^n a_i \phi_i(x) \quad (24)$$

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. 24 is a linear combination of the bump basis function  $\phi_i(x)$  with different coefficients  $a_i$ , which constitute the design parameters for the optimization problem along with the angle of attack:  $\mathbf{x} = [\alpha, a_1 \ a_2 \ \dots \ a_n]^T$ . One widely used basis set is the sine functions:<sup>33</sup>

$$\begin{aligned} \phi_i(x) &= \sin^{t_i}(\pi x^{m_i}) \\ m_i &= \ln(0.5) / \ln(x_{M_i}) \end{aligned} \quad (25)$$

where  $x_{M_i}$  is the preselected location of the maxima for each basis function and  $t_i$  controls the width of the bump function. We adopt the optimized  $x_{M_i}$  values suggested by Wu et al.<sup>34</sup> for fixed  $t_i = 4$ ,

$$x_{M_i} = \frac{1}{2} \left[ 1 - \cos \left( \frac{i\pi}{n+1} \right) \right], \quad i = 1, \dots, n \quad (26)$$

### V.D. Mesh Movement

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 radial basis function (RBF) interpolation to deform the mesh.

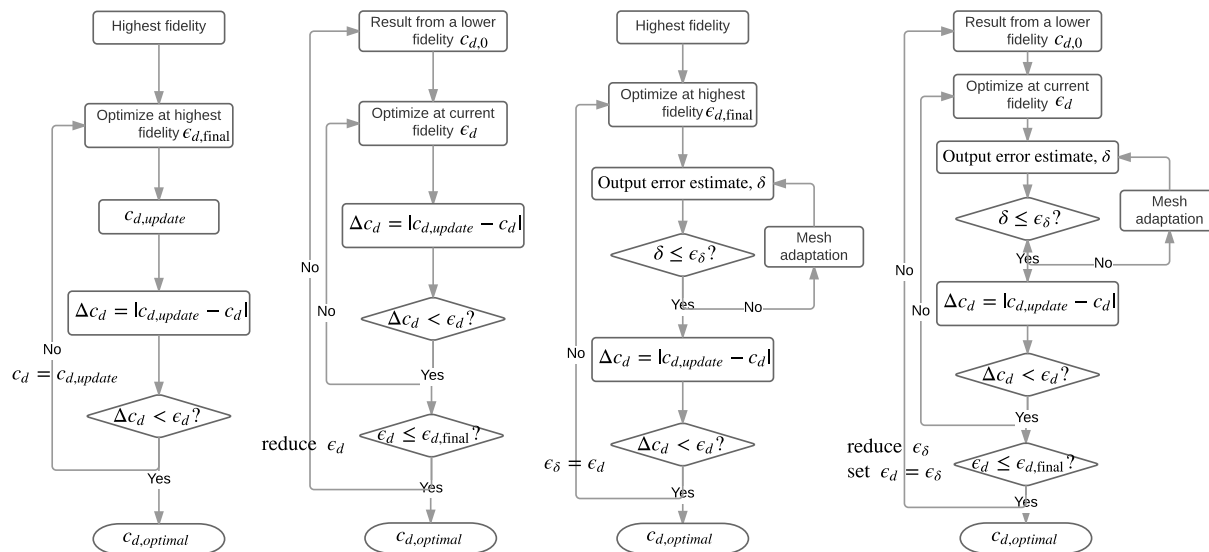
A radial basis function is a real-valued function that depends only on the distance from the origin, or some specified point  $\vec{c}$ , such that  $\phi(\vec{x}) = \phi(\|\vec{x}\|)$  or  $\phi(\vec{x}, \vec{c}) = \phi(\|\vec{x} - \vec{c}\|)$ . Assume that we are given the displacements  $\mathbf{d}(\mathbf{x}) = [\vec{d}_1, \vec{d}_2, \dots, \vec{d}_N]^T$  of the airfoil boundary nodes  $\mathbf{x} = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N]^T$ , which are called “centers” in RBF interpolation. Then we can use the sum of the RBFs and a polynomial  $\vec{p}(\vec{x})$  to interpolate the original displacement function,

$$\vec{d}(\vec{x}) \approx \tilde{d}(\vec{x}) = \sum_{i=1}^{N_b} \vec{r}_i \phi(\|\vec{x} - \vec{x}_i\|) + \vec{p}(\vec{x}) \quad (27)$$

where  $\vec{r}_i$  is the coefficient for the  $i^{\text{th}}$  RBF, and  $N_b$  is the number of boundary nodes. With the interpolation, we can construct the mesh movement just with the information of the boundary nodes. The requirement for solving  $\vec{r}_i$  is a linear system whose size is  $\mathcal{O}(N_b)$ , which is small compared to the number of nodes in the entire mesh. Moreover, the connectivity of the mesh is not required for the interpolation. More details can be found in the previous work of Jakobsson and Amoignon<sup>35</sup> and Boer et al.<sup>36</sup>

### V.E. Mesh Adaptation in a Multi-Fidelity Setting

The output error estimated with Eqn. 20 is localized to each element using Eqn. 21: the contribution of each element to the total error is taken as the error indicator for mesh adaptation. 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 multi-fidelity optimization. Rather than performing optimization and mesh adaptation sequentially, one after another, an interactive framework is introduced. The objective function is first evaluated on a relatively coarse mesh. Then the optimization on that mesh is driven by the mesh adaptation tolerance (maximum allowed estimated error). Once the objective change or the gradient norm is below the current resolution, the optimization terminates on the current mesh and the fidelity increases through mesh adaptation. As the shape approaches the optimal design, the mesh is adapted more and more frequently, since the shape changes become smaller and smaller. A comparison of the traditional and proposed optimization strategies is shown in Fig. 1.



**Figure 1. Flow-charts of different optimization strategies: fixed fidelity without error estimation (left); multi-fidelity without error estimation (middle left); fixed fidelity with error estimation (middle right) and multi-fidelity with error estimation (right).**

Compared with the fixed-fidelity optimization, unnecessarily fine meshes at the early stages of shape optimization are avoided in the proposed multi-fidelity framework. Moreover, the elements that introduce most of the error may differ a lot for different shape configurations during the optimization. This approach

reduces the chance of over-refining elements that are not relatively important for the next optimization step, which is important if the adaptation mechanics do not allow for coarsening. Compared with the multi-fidelity optimization without error estimation, the error tolerance serves as an optimization tolerance at each fidelity, actively controlling the optimization at each step and avoiding the waste of low-fidelity convergence. Finally, the new framework can effectively prevent over-optimizing on a coarse mesh, or over-refining on an unintended shape.

## VI. Results

As simple tests for the proposed optimization framework, we consider a laminar flow case, an inviscid transonic case, and a low-speed turbulent case for a starting NACA 0012 airfoil geometry, seeking an optimal shape and incidence angle to minimize the drag coefficient subject to the fixed lift trim condition and the minimum volume constraint. The airfoil is parametrized with 10 Hicks-Henne basis functions, and cubic curved mesh elements are used to represent the boundary.

### VI.A. Laminar, Subsonic Airfoil

A subsonic laminar flow case is first studied. The initial angle of attack is zero with a free-stream Mach number  $M_\infty = 0.5$  and Reynolds number  $Re = 5000$ . To investigate the effectiveness of the proposed framework, two different methods are also carried out in the optimization problem: fixed (highest) fidelity optimization without error estimation, and multi-fidelity optimization with error estimation and mesh adaptation only on the objective ( $c_d$ ).

All of the optimization runs start from the same initial mesh consisting of 533 triangular elements. For the fixed-fidelity optimization, the mesh is first adapted to meet the objective tolerance, and then no more mesh adaptation occurs during the optimization. In contrast, in the other two methods, the mesh is adapted during the optimization. The initial symmetric airfoil should produce zero lift at zero incidence, and thus the initial condition is further from feasible if a higher target lift is specified, which means higher error of the outputs may appear during the optimization. Therefore, a very low lift constraint (nearly feasible) and a relatively high target lift (infeasible) are both tested for this case.

#### VI.A.1. $c_l^* = 0.02$

The target lift is set to be  $c_l^* = 0.02$  for this case, and the optimization tolerance is  $5 \times 10^{-5}$  in the drag coefficient. We should expect the final meshes to be comparable in size for all of the methods because of the same ultimate tolerance for the optimization. The meshes during the optimization, however, may be quite different. The fixed fidelity mesh and the meshes at the same intermediate optimization fidelity for the multi-fidelity methods are shown in Figure 2. These meshes show that multi-fidelity optimization significantly reduces the mesh size and computational resources during the early optimization iterations. The drag adjoints and lift adjoints are also shown with the meshes for two multi-fidelity optimizations. Though the shape is not the same, the flow features are similar. Most of the mesh adaptation happens at the leading edge since the drag adjoints are the largest (and least resolved) near the airfoil nose, while the proposed framework, Figure 2(d), adapts more on the trailing edge, which is also important for accurate lift prediction.

The objective convergence history and mesh size evolution are shown in Figure 3. From the convergence plot, we see that the discretization error of the objective increases as the shape changes for the fixed-fidelity optimization without mesh adaptation. The objective error may be above the optimization tolerance during the optimization even though we start with a fairly fine mesh, while discretization error is always controlled to be below the optimization tolerance via mesh adaptation in the other two methods. Furthermore, the fixed-fidelity optimization requires the most iterations on the highest fidelity (the number of objective evaluations would be even more because of the line search between each major iteration). The proposed method gives us the fewest iterations on the finest mesh since both the objective and the constraints are more accurate at the lower fidelity compared to the optimization that only adapts on the objective.

The final designs obtained by all of these three methods are shown in Figure 4. The final designs are almost the same, as the target lift is not far away from the original value. The objective error of fixed fidelity optimization is only slightly higher than the tolerance, and hence we can still achieve an acceptable optimal design.



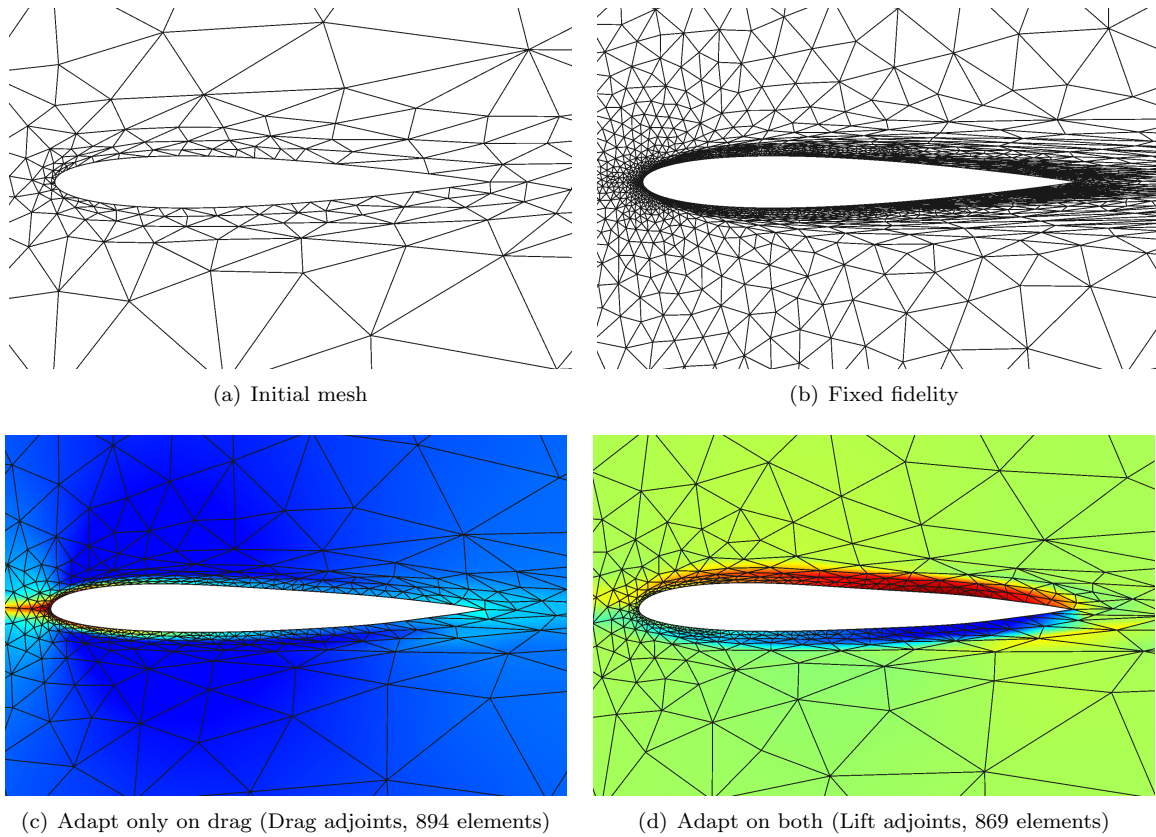


Figure 2. Initial mesh and intermediate meshes for different optimization strategies (low lift, laminar)

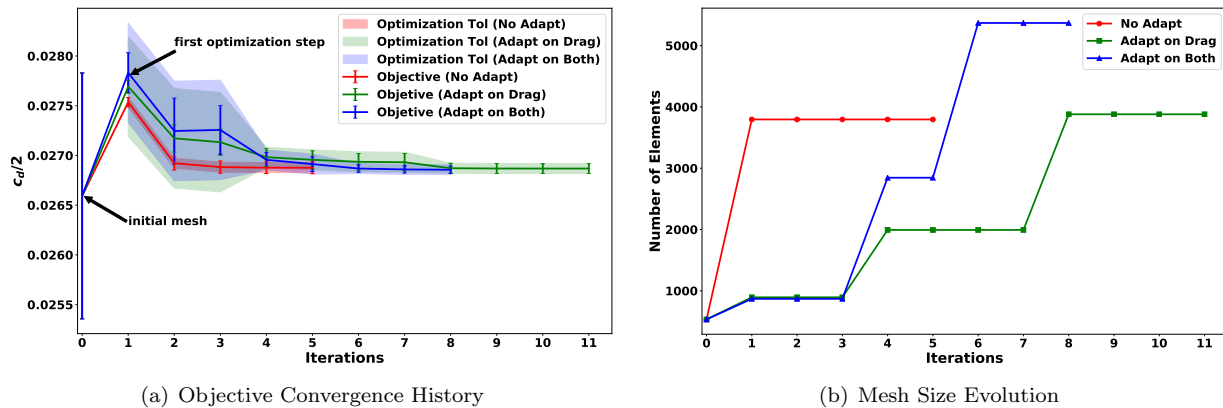


Figure 3. Convergence history and mesh size evolution for different methods (low lift, laminar)

### VI.A.2. $c_l^* = 0.1$

For this case we specify a higher lift coefficient,  $c_l^* = 0.1$ , which is further from the original design. Again, the meshes of different methods during the optimizations are shown in Figure 5. This time the two adaptation strategies differ more: although Figure 5(c) has more global elements, it does not adapt much on the upper surface near the trailing edge since it's not important for the drag calculation. This area does get adapted in the proposed method, as shown in Figure 5(d).

The objective convergence and mesh evolution are given in Figure 6. The fixed-fidelity optimization without mesh adaptation exhibits a higher error this time, and it converges to a different design compared to the results of the other two methods with error control. Therefore, if the original shape is far from optimal

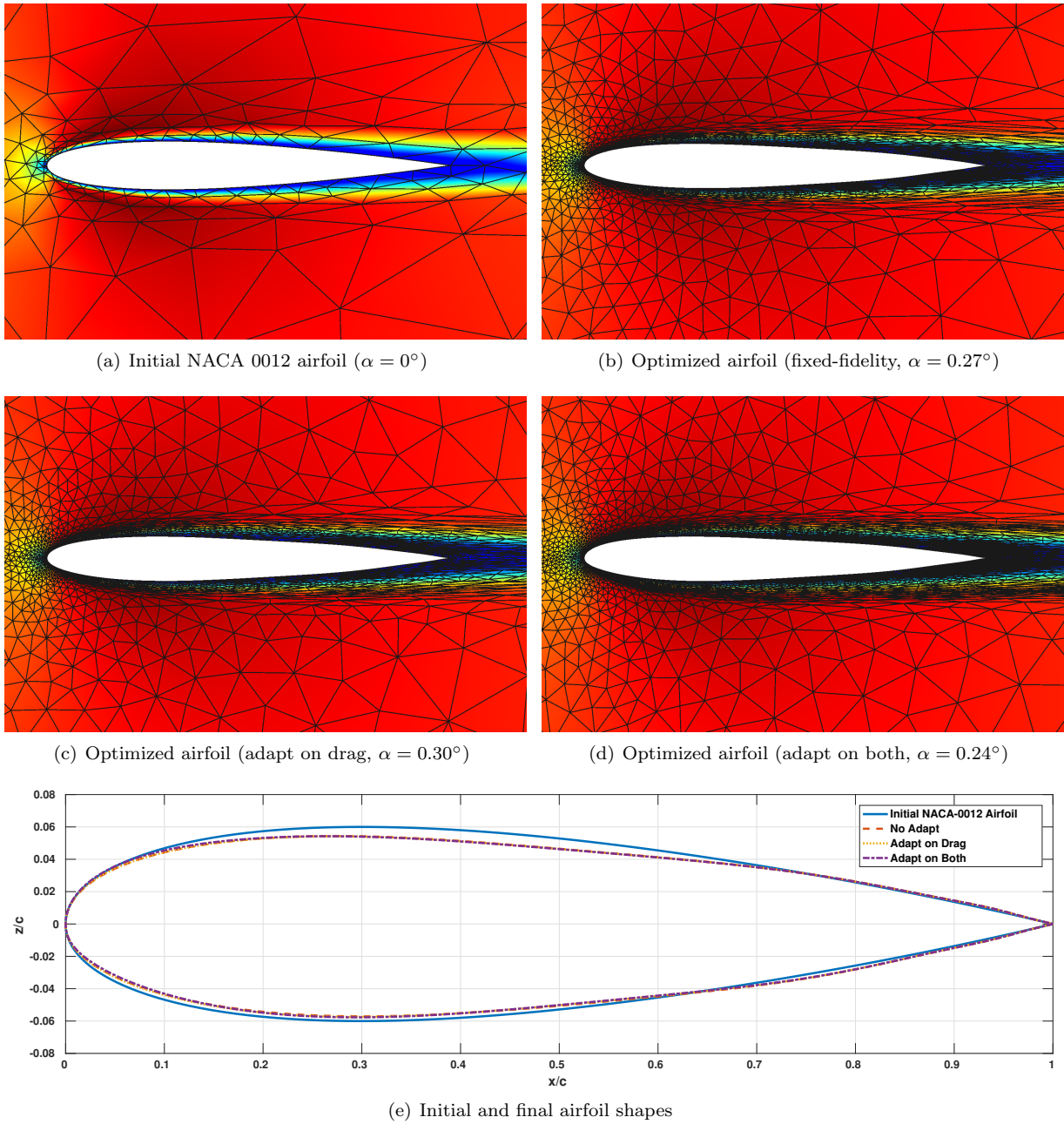


Figure 4. Local Mach number ( $0 \sim 0.6$ ) for the initial and final designs for different methods (low lift, laminar)

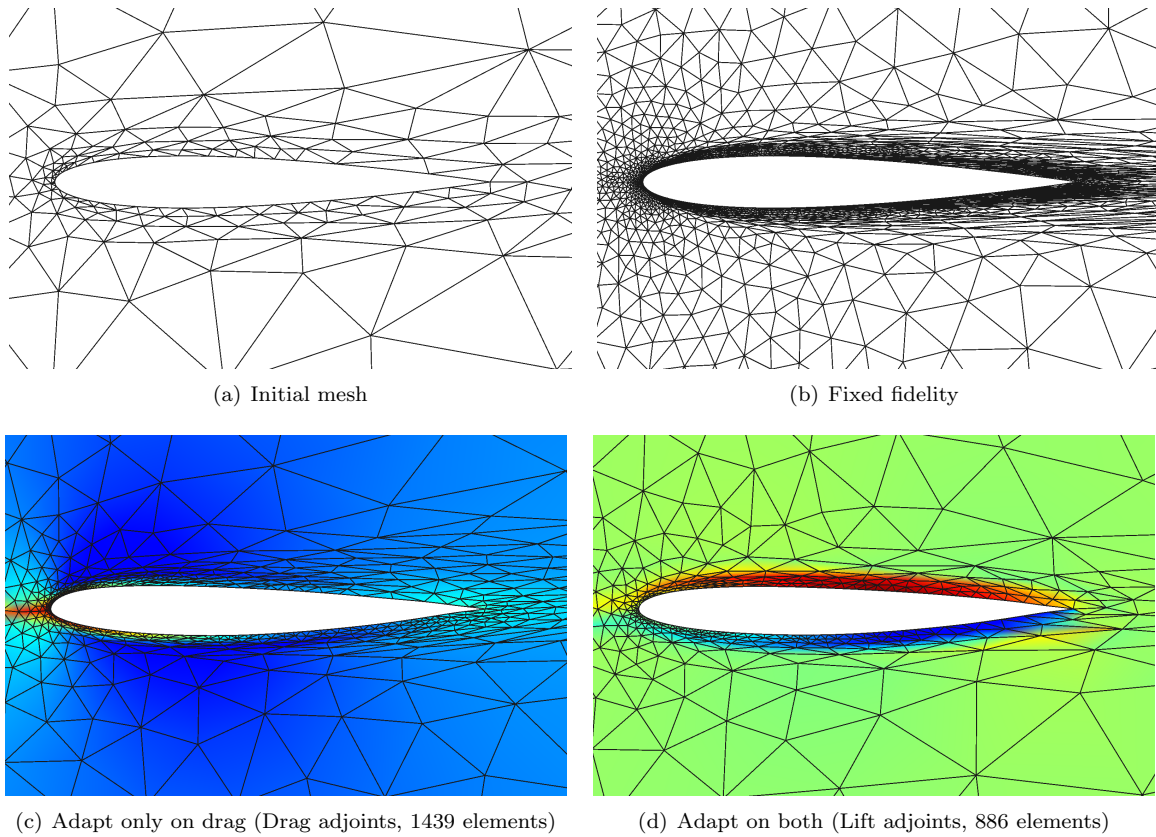


Figure 5. Initial mesh and intermediate meshes for different optimization strategies (high lift, laminar)

in the optimization, it's necessary to do the error estimation and mesh adaptation even though the starting mesh is fairly fine. Again, the proposed framework needs the fewest iterations on the highest fidelity, though the mesh is largest since a higher accuracy on the constraints is also achieved. If we require even higher fidelity, we would expect the proposed framework to converge even faster on the finest mesh, where most of the computational resources are spent.

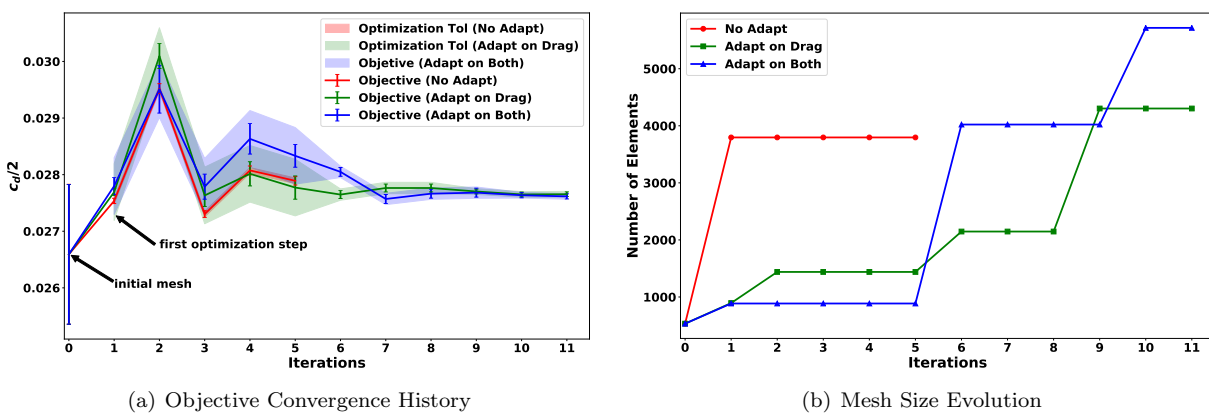


Figure 6. Convergence history and mesh size evolution for different methods (high lift, laminar)

The final designs achieved by all these three methods are given in Figure 7. We can clearly see in Figure 7(e) that the final shape provided by the fixed-fidelity optimization without mesh adaptation deviates from the results of the optimization with error controlled by mesh adaptation.

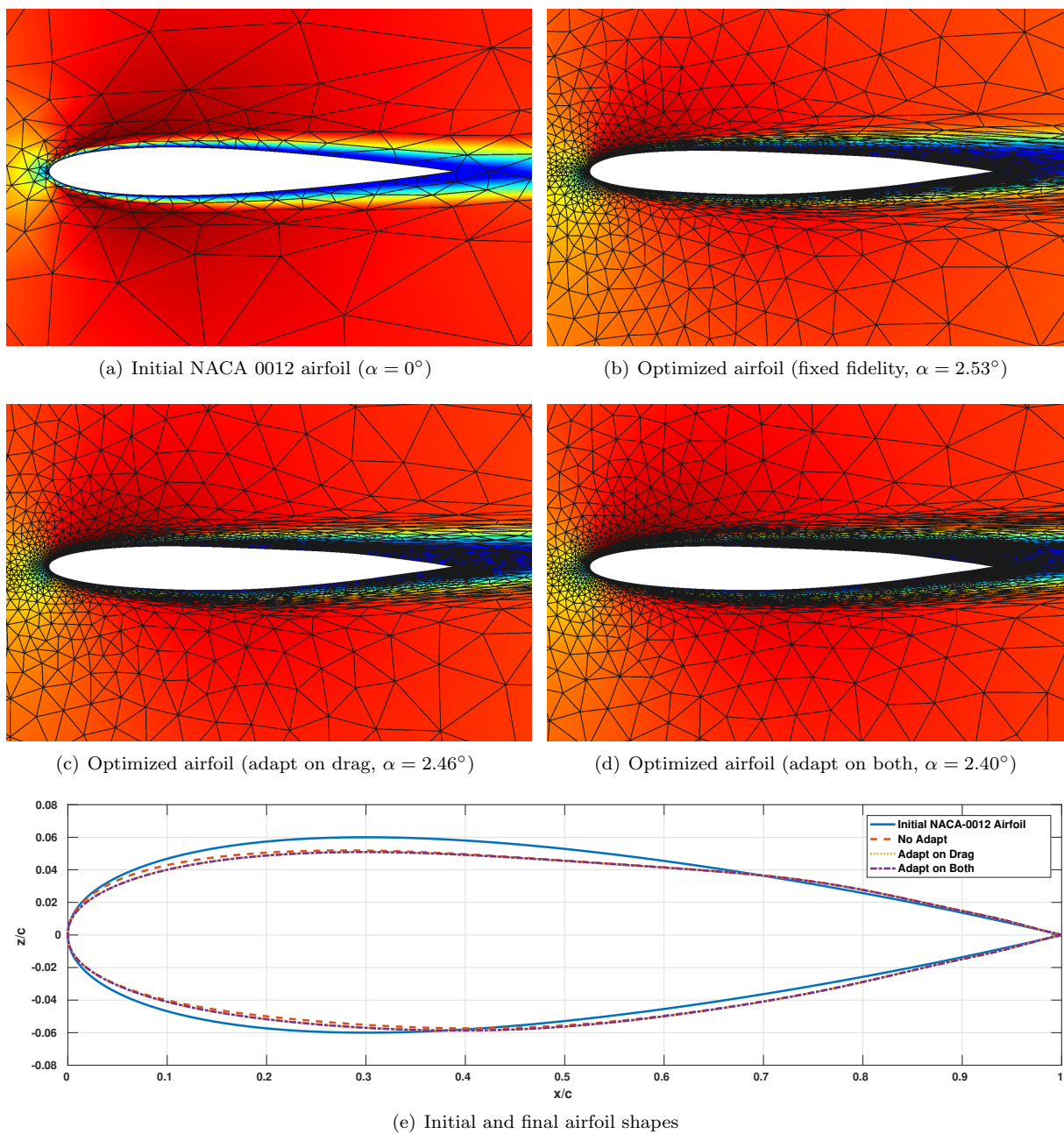


Figure 7. Local Mach Number ( $0 \sim 0.6$ ) for the initial and final designs for different methods (high lift, laminar)

## VI.B. Inviscid, Transonic Airfoil

We apply the new method to an optimization problem based on an inviscid transonic flow at  $M_\infty = 0.8$  around the NACA 0012 airfoil. The initial angle of attack is  $\alpha = 1.25^\circ$ , and the goal is to minimize the drag with a target lift of  $c_l^* = 0.4$ . Although the transonic flow around the original NACA 0012 airfoil features a strong shock on the upper surface near the trailing edge, we expect that the shape would be modified during the optimization such that the shock strength is weakened or the shock completely removed. The flow features and the outputs of interest are highly related to the location and strength of the shock, which may change a lot during the optimization. Thus, error estimation and mesh adaptation become more crucial in this case. We start with the same initial mesh used in the laminar case. The optimization history is shown in Figure 8.

As shown in the convergence plot, most of the drag reduction happens at the lowest fidelity, where the



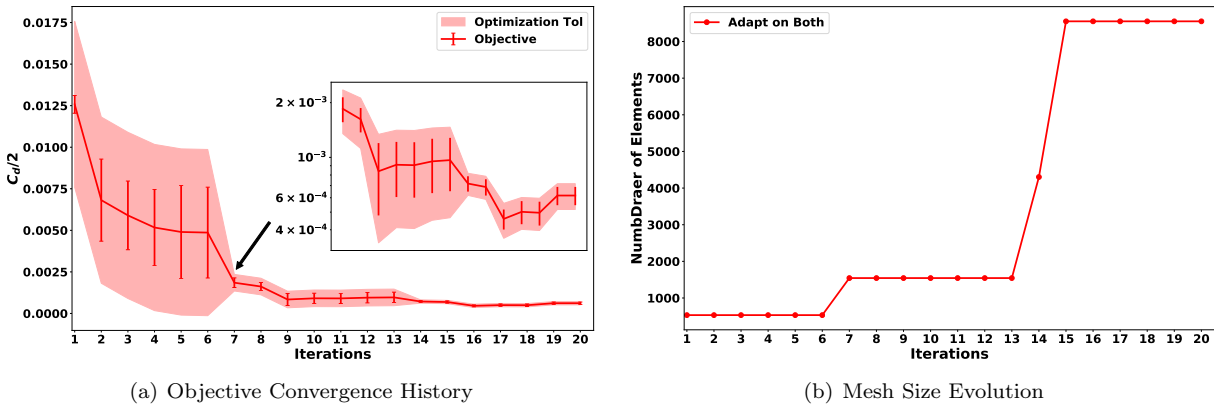


Figure 8. Convergence history and mesh size evolution for the proposed method (inviscid, transonic)

flow solve is very cheap, though quite inaccurate. Although the starting point of the high-fidelity is close to the optimal, we still need more iterations compared to the laminar case mainly because the outputs (both the objective and constraints) are very sensitive to the shape change in the transonic regime. The ultimate optimization tolerance is  $1 \times 10^{-4}$ , and we obtain a final drag coefficient of  $c_{d,\text{opt}}/2 = 6.170 \times 10^{-4}$  compared with the initial value,  $c_{d,0}/2 = 1.167 \times 10^{-2}$  (this number is obtained by a refined mesh on the original NACA 0012 airfoil with the same tolerance as the optimization, not the first point in the convergence plot). The total drag is therefore reduced by 95%. The modified and original shapes are compared in Figure 9. The final design approaches a flattened upper surface and a higher aft camber, resembling a super-critical airfoil. The initial strong shock is significantly weakened on the optimized airfoil. We can still see a discontinuity in the flow field, though the drag could be further reduced if more basis functions were used to parametrize the airfoil shape and even higher fidelity flow calculations were performed.

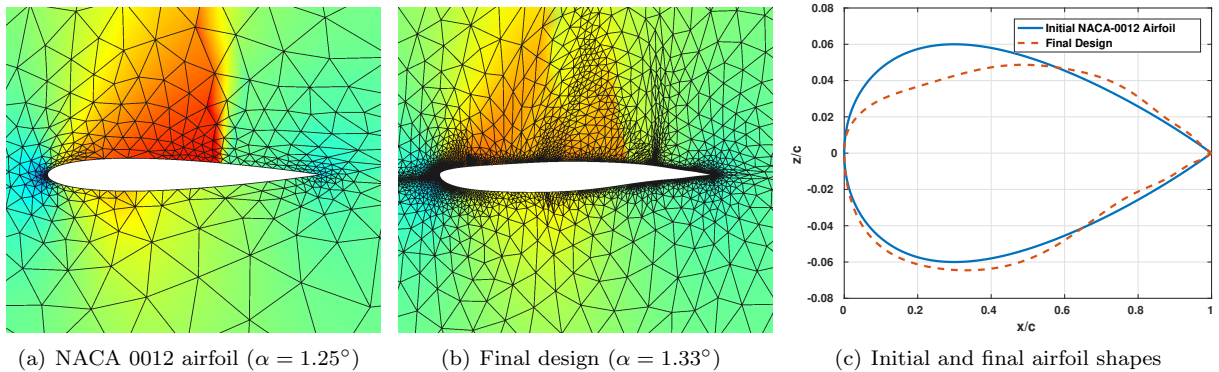


Figure 9. Local Mach Number ( $0 \sim 1.6$ ) for the initial and final designs (inviscid, transonic)

### VI.C. Turbulent, Low-speed Airfoil

The final problem considered in this paper is a turbulent, low-speed flow around the NACA 0012 airfoil. The initial angle of attack is  $6^\circ$  with a freestream Mach number of  $M_\infty = 0.15$  and Reynolds number  $Re = 10^6$ , the target lift is set to be  $c_l^* = 0.6$ . The starting mesh for the turbulent case consists of 1448 elements, which is finer than the initial mesh used for the laminar and transonic runs.

The mesh size and objective are collected at each optimization step as shown in Figure 10. For the turbulent case, most of the adaptation focuses on resolving the boundary layer. Therefore, many adaptation iterations may happen at low fidelity. Once the boundary layer is properly resolved, the error drops down very quickly, and less adaptation is required. In this case, we can see that a lot of adaptation happens at low fidelity, and we can also see some coarsening on the final fidelity. Since the flow is always attached and less error is produced during the shape changes, we can redistribute the mesh or even coarsen it while

still meeting the tolerance. However, for higher-speed or higher-incidence cases, the boundary layer changes more as the shape changes, more complicated flow features like shocks or separation may happen during the optimization, and more mesh adaptation would be required at higher fidelity.

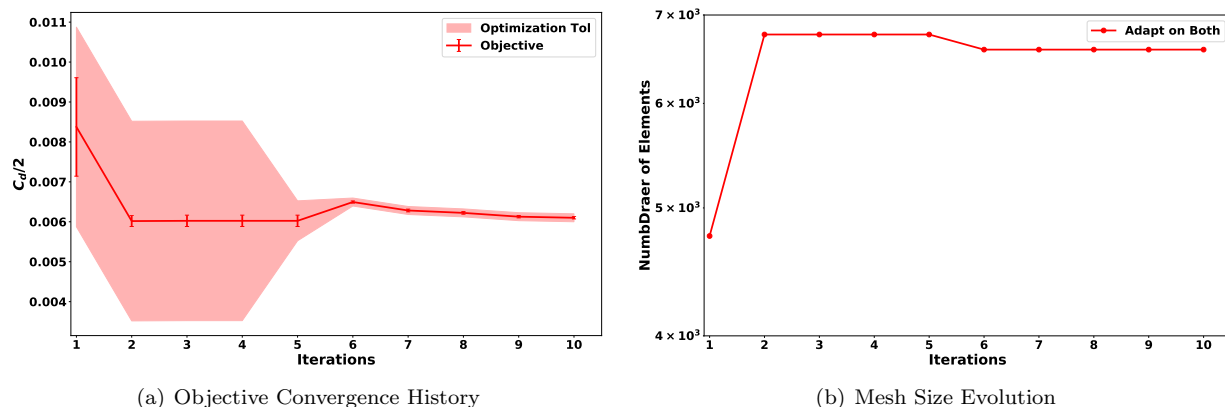


Figure 10. Convergence history and mesh size evolution for the proposed method (low-speed, turbulent)

The meshes used in the optimization and the optimized design are summarized in Figure 11. The optimization flattens the lower surface while curving the upper surface, which differs from the transonic case. From the contour plots, we see that the modified design has a larger and smoother acceleration region on the upper surface, which is favorable for higher lift generation. Thus, a lower angle of attack is achieved with the same lift constraints. The flow near the trailing edge also exhibits less separation in the final design, and, therefore, the drag drops for the optimized shape.

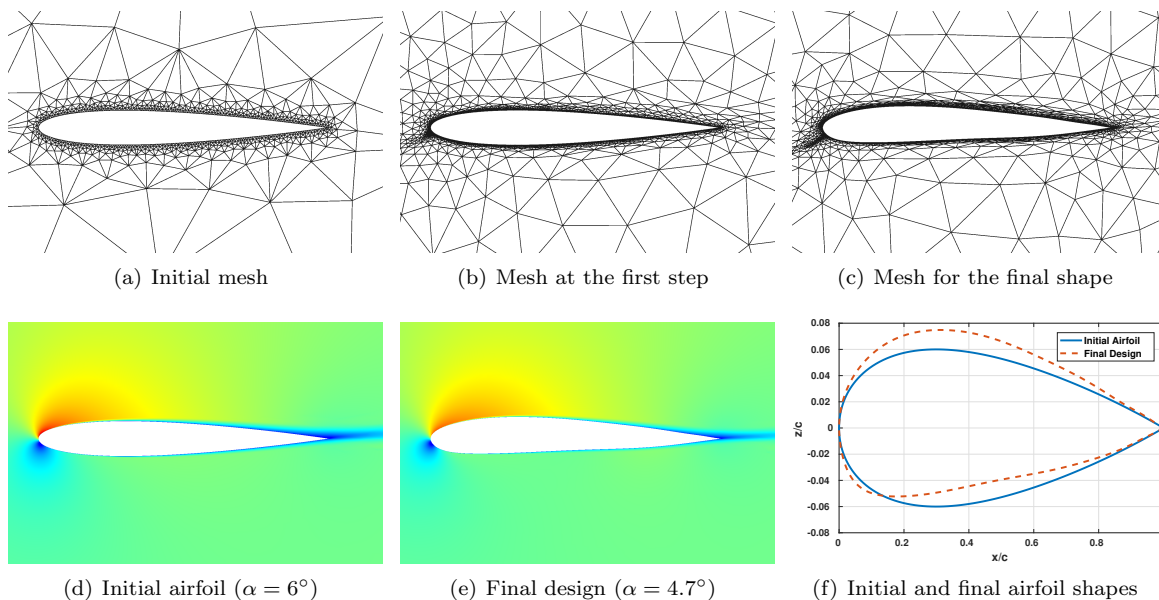


Figure 11. Meshes and Local Mach Number ( $0 \sim 0.3$ ) for the initial and final designs (low-speed, turbulent)

## VII. Conclusion

Most aerodynamic shape optimization methods work with the discretized governing flow equations. Thus, numerical error should be carefully controlled to ensure convergence to the “true” optimal design at a prescribed fidelity. Without properly controlling this error, the optimizer may arrive at a sub-optimal design with inaccurate information provided by the flow and the gradient solver as shown in the test cases.

In this work, we presented a framework that integrates output-based error estimation and mesh adap-

tation with a traditional gradient-based algorithm for airfoil shape optimization with trim constraints. The multi-fidelity optimization approach consists of progressive refinement of the computational mesh and is capable of preventing over-optimizing and over-refining. The mesh adaptation (fidelity increase) is determined by the absolute change or the projected gradient norm of the objective function. A coupled adjoint is also introduced, offering a way to include the trimming error into the objective error estimation. Compared to fixed-fidelity optimization without mesh adaptation and to multi-fidelity optimization that only adapts on the objective, the proposed method provides improved efficiency and better convergence at the highest fidelity, as demonstrated by the test cases. This benefit becomes more significant when higher fidelity is required.

With more judicious considerations of the objective functions and constraints, and additional parameters, the new method can provide realistic configurations in practical design scenarios. The fidelity increase is presently driven by an adaptation tolerance that decreases by a fixed factor each time or that is specified by the user. However, for more practical problems, without a priori knowledge of the objective convergence, an improved and automated fidelity increase strategy should be developed to fulfill the potential of the present optimization framework to increase the accuracy and efficiency for aerodynamic shape optimization problems.

## 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.

## References

- <sup>1</sup>Hajela, P., "Genetic search-an approach to the nonconvex optimization problem," *AIAA journal*, Vol. 28, No. 7, 1990, pp. 1205–1210.
- <sup>2</sup>Obayashi, S. and Takanashi, S., "Genetic optimization of target pressure distributions for inverse design methods," *AIAA journal*, Vol. 34, No. 5, 1996, pp. 881–886.
- <sup>3</sup>Jones, B. R., Crossley, W. A., and Lyrintzis, A. S., "Aerodynamic and aeroacoustic optimization of rotorcraft airfoils via a parallel genetic algorithm," *Journal of Aircraft*, Vol. 37, No. 6, 2000, pp. 1088–1096.
- <sup>4</sup>Pierret, S. and Van den Braembussche, R., "Turbomachinery blade design using a Navier-Stokes solver and artificial neural network," *ASME 1998 International Gas Turbine and Aeroengine Congress and Exhibition*, American Society of Mechanical Engineers, 1998, pp. V001T01A002–V001T01A002.
- <sup>5</sup>Rai, M. M. and Madavan, N. K., "Aerodynamic design using neural networks," *AIAA journal*, Vol. 38, No. 1, 2000, pp. 173–182.
- <sup>6</sup>Hajela, P., "Nongradient methods in multidisciplinary design optimization-status and potential," *Journal of aircraft*, Vol. 36, No. 1, 1999, pp. 255–265.
- <sup>7</sup>Martins, J. R., Alonso, J. J., and Reuther, J. J., "A coupled-adjoint sensitivity analysis method for high-fidelity aerostuctural design," *Optimization and Engineering*, Vol. 6, No. 1, 2005, pp. 33–62.
- <sup>8</sup>Martins, J. R., Sturdza, P., and Alonso, J. J., "The complex-step derivative approximation," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 29, No. 3, 2003, pp. 245–262.
- <sup>9</sup>Griewank, A. and Walther, A., *Evaluating derivatives: principles and techniques of algorithmic differentiation*, Siam, 2008.
- <sup>10</sup>Jameson, A., "Aerodynamic design via control theory," *Journal of scientific computing*, Vol. 3, No. 3, 1988, pp. 233–260.
- <sup>11</sup>Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D., "Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation," *AIAA paper*, Vol. 94, 1996.
- <sup>12</sup>Anderson, W. K. and Venkatakrishnan, V., "Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation," *Computers & Fluids*, Vol. 28, No. 4, 1999, pp. 443–480.
- <sup>13</sup>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.
- <sup>14</sup>Venditti, D. A. and Darmofal, D. L., "Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46.
- <sup>15</sup>Park, M. A., "Adjoint-based, three-dimensional error prediction and grid adaptation," *AIAA journal*, Vol. 42, No. 9, 2004, pp. 1854–1862.
- <sup>16</sup>Nemec, M. and Aftosmis, M. J., "Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes," *AIAA Paper*, Vol. 4187, 2007, pp. 2007.
- <sup>17</sup>Nemec, M., Aftosmis, M. J., and Wintzer, M., "Adjoint-based adaptive mesh refinement for complex geometries," *AIAA Paper*, Vol. 725, 2008, pp. 2008.
- <sup>18</sup>Fidkowski, K. J. and Roe, P. L., "An entropy adjoint approach to mesh refinement," *SIAM Journal on Scientific Computing*, Vol. 32, No. 3, 2010, pp. 1261–1287.

- <sup>19</sup>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.
- <sup>20</sup>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.
- <sup>21</sup>Nemec, M. and Aftosmis, M. J., “Output error estimates and mesh refinement in aerodynamic shape optimization,” *AIAA Paper 2013*, Vol. 865, 2013.
- <sup>22</sup>Dalle, D. J. and Fidkowski, K. J., “Multifidelity Airfoil Shape Optimization Using Adaptive Meshing,” *32nd AIAA Applied Aerodynamics Conference*, 2014, p. 2996.
- <sup>23</sup>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.
- <sup>24</sup>Rothacker, B. A., Ceze, M., and Fidkowski, K. J., “Adjoint-Based Error Estimation and Mesh Adaptation for Problems with Output Constraints,” *32nd AIAA Applied Aerodynamics Conference*, 2014, p. 2576.
- <sup>25</sup>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.
- <sup>26</sup>Boyd, S. and Vandenberghe, L., *Convex optimization*, Cambridge university press, 2004.
- <sup>27</sup>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.
- <sup>28</sup>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.
- <sup>29</sup>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.
- <sup>30</sup>Kraft, D., “A software package for sequential quadratic programming,” *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
- <sup>31</sup>Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J., “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on Mathematical Software (TOMS)*, Vol. 23, No. 4, 1997, pp. 550–560.
- <sup>32</sup>Hicks, R. M. and Henne, P. A., “Wing design by numerical optimization,” *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.
- <sup>33</sup>Masters, D. A., Taylor, N. J., Rendall, T., Allen, C. B., and Poole, D. J., “Review of aerofoil parameterisation methods for aerodynamic shape optimisation,” *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 0761.
- <sup>34</sup>Wu, H.-Y., Yang, S., Liu, F., and Tsai, H.-M., “Comparison of three geometric representations of airfoils for aerodynamic optimization,” *AIAA Paper*, Citeseer, 2003.
- <sup>35</sup>Jakobsson, S. and Amoignon, O., “Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization,” *Computers & Fluids*, Vol. 36, No. 6, 2007, pp. 1119–1136.
- <sup>36</sup>De Boer, A., Van der Schoot, M., and Bijl, H., “Mesh deformation based on radial basis function interpolation,” *Computers & structures*, Vol. 85, No. 11, 2007, pp. 784–795.