10.2514/6.2014-2996

AIAA AVIATION Forum
16-20 June 2014, Atlanta, GA
32nd AIAA Applied Aerodynamics Conference

Multifidelity Airfoil Shape Optimization Using Adaptive Meshing

Derek J. Dalle[*] and Krzysztof J. Fidkowski[†]
University of Michigan, Ann Arbor, MI 48109

Gradient-based aerodynamic shape optimization is investigated using adaptive mesh refinement. Instead of using a mesh with a fixed cell spacing and total mesh size, the mesh is adapted to each design iteration. In the early stages of optimization, the objective function is evaluated only on the coarse mesh, and progressively more mesh adaptations are performed as the design approaches optimality. This approach has several features that can reduce optimization time. First, the multifidelity aspect allows the early portion of the optimization to be computed on meshes that require much reduced computational resources. Second, output-based adaptive meshing allows the flow calculations to target areas of the flow that most affect the objective function, which reduces total mesh size for a given level of accuracy. Finally, output error estimates reduce the chance of over-optimizing on a coarse mesh, or under-optimizing on a fine one. In this work, the framework is tested on several airfoil optimization test cases, and it is compared in computational expense to other approaches, which include a traditional fixed-mesh optimization, optimization using a constant level of fidelity, and optimization using uniform refinement for the multifidelity aspect. The multifidelity schemes require additional setup time, and a framework is introduced to reduce this burden.

I. Introduction

Design optimization, especially for problems with a large number of design variables, requires many evaluations of a particular objective function, which measures the quantity that is sought to be minimized. Even gradient-based optimization, which typically requires fewer function evaluations for large problems [1], often requires hundreds or more function evaluations. For problems with constraints that must be evaluated separately from the objective function, the resources required are even greater. Because the optimization process requires many iterations, the highest-fidelity modeling tools are usually impractical to use.

Optimization of an aerodynamic shape is vital for aircraft design, yet challenging because of the potentially-complex flows possible at the flight regimes of typical interest. Efficient algorithms for automated shape optimization have thus been of interest for several decades. In particular, gradient-based shape optimization algorithms [2–7] offer efficiency benefits for high-dimensional design parametrizations.

Even with gradient-based methods, the computational costs of high-fidelity aerodynamic optimizations still remain high because of the expense of each high-fidelity simulation. To reduce this cost, *multifidelity* optimization algorithms [8–14] have been devised that employ cheaper, lower-fidelity, models to accelerate the design optimization. Lower-fidelity models could be generated from reduced-physics, from coarser meshes, or from formal model reduction procedures. The associated algorithms rely on the ability of the lower-fidelity model(s) to yield somewhat-accurate information useful for optimization, at a fraction of the cost of the high-fidelity models.

Using lower-fidelity models, or surrogates, to accelerate high-fidelity optimization entails theoretical and practical considerations that have been the subject of numerous previous studies. Proposed frameworks have addressed topics such as management of multiple models [15], choice of model and conditions of convergence [8, 9], and mappings between variable parametrizations [16]. These frameworks have shown speedups of up to one order of magnitude in computational time relative to high-fidelity optimization.

[*] Postdoctoral Research Fellow, Department of Aerospace Engineering, AIAA Member, dalle@umich.edu
[†] Assistant Professor, Department of Aerospace Engineering, AIAA Senior Member, kfid@umich.edu

Multifidelity optimization algorithms typically assume that there exists a "truth" high-fidelity model, and often the coarser models or surrogates are derived from this truth model. A natural question is then: how does one generate the truth model for an optimal design that is not known? For example, computational fluid dynamics simulations rely on a mesh of the flow domain, but a given mesh, even if generated according to best-practices, does not guarantee engineering accuracy that is necessarily worthy of a "truth" designation [17]. Here, adaptively-tailored meshes [18] can help by providing resolution where it is needed, together with error bars on outputs of interest. Furthermore, since the geometry changes during shape optimization, and since geometry affects the flow-field, which affects the optimal mesh distribution, adaptive meshes are a natural choice in the optimization context.

The present work is related to that of Nemec and Aftosmis [19], who also use output-based mesh adaptation for shape optimization. The present work extends the methodology to include constrained optimization (rather than target-seeking) and a different metric for increasing the fidelity. Rather than treating the multifidelity problem as a sequence of optimizations in which the final result from a lower fidelity is used as the initial design of a higher-fidelity optimization, an error estimate is used to decide when to increase the fidelity. This approach seeks to prevent wasting resources needed to solve low-fidelity optimization problems to convergence. Another benefit that is potentially even more important is that our proposed algorithm resists failure in cases where the low-fidelity optimum may be far from the high-fidelity optimum. In such cases where the two models have very different optima, the criteria for increasing the fidelity is likely to be reached before the low-fidelity method is able to guide the algorithm toward the errant low-fidelity optimum.

The combination of output-based error estimation and gradient-based optimization is natural as both methods rely on output adjoint solutions. We use an in-house code in which a discrete adjoint serves this dual purpose, and which has already been used for, separately, adaptive mesh generation [20–23] and design optimization [24]. We note that the multifidelity approach used in this work is classified as hierarchical [16] because the fidelity increases (order increase and mesh refinement) but the fundamental model (e.g. Navier-Stokes equations) remains fixed.

The outline for the remainder of this paper is as follows. Section II describes the optimization problems considered. Section III presents the optimization approach, and Section IV reviews the numerical discretization. We present results in Section V, and we conclude in Section VI.

II. Test Problem Description

The design space considered for this problem is the modified NACA 4-digit series [25]. This series of airfoils has five degrees of freedom as opposed to the usual three for the unmodified NACA 4-digit series. An example is NACA the 2310-65. Each digit has an effect that is explained in Table 1.

Table 1. Explanation of digits for the NACA 2310-65 airfoil.

Digit	Effect
2	maximum camber, in hundredths of chord
3	location of maximum camber, in tenths of chord
10	maximum thickness, in hundredths of chord
6	leading edge roundness index
5	location of maximum thickness, in tenths of chord

The decision variables (**x**) are scaled so that each of the five is on the order of unity. They are listed in Table 2. In addition, the maximum thickness is represented by t, but this is held fixed during optimization.

A NACA XXXX-64 airfoil is almost equivalent to the unmodified 4-series, although the thickness profile will be slightly different for the last 60% of the chord. In this work, the design space is generalized slightly such that each of the five parameters in Table 1 is a continuous variable. For example the location of maximum thickness can be at 41.2% of the chord, and so on.

Table 2. Description of decision/design variables.

Index, i	Symbolic value, x_i	Description
1	$10m$	maximum camber, scaled
2	x_m	location of maximum camber
3	$R/10$	nose radius index, scaled
4	x_t	location of maximum thickness
5	10α	angle of attack, scaled

The mean camber line is the same for the common NACA 4-digit series.

$$z_c = \begin{cases} \frac{(2x_m - x)mx}{x_m^2} & x \leq x_m \\ \frac{(x+1-2x_m)(1-x)m}{(1-x_m)^2} & x > x_m \end{cases} \tag{1}$$

The equation for the airfoil thickness is more complicated, in part because it is affected by more design variables and in part because the original definition at times is based on tabulated values decided in the 1930s. The first step, based on the process described in [25], is to calculate the slope at the trailing edge. When the modified 4-digit series was designed, these values were carefully selected based on the location of maximum thickness (x_t) to avoid inflection points in the airfoil surface. The reference values selected are shown in Table 3.

Table 3. Trailing edge slopes as a function of location of maximum thickness for NACA modified 4-digit series.

Location of maximum thickness (x_t)	Trailing edge slope (d_1)
0.2	0.2
0.3	0.234
0.4	0.315
0.5	0.465
0.6	0.7

Unfortunately, these values do not follow an analytical function, and we use cubic interpolation in this work to determine the reference trailing edge slope for intermediate values. Additionally, these airfoils are not defined for values of x_t outside of the range from 0.2 to 0.6.

Much like the unmodified NACA 4-digit series, the official definition has a trailing edge thickness of 0.002. To close this trailing edge for convenience to the CFD analysis, the reference slope d_1 is corrected to

$$c_1 = d_1 + \frac{6d_0}{1 - x_t}, \tag{2}$$

where d_0=0.002. Two other coefficients are defined for use in calculating the thickness profile of the aft portion of the airfoil:

$$c_2 = \frac{0.3 - 2(1 - x_t)c_1}{(1 - x_t)^2}, \tag{3}$$

$$c_3 = \frac{(1 - x_t)c_1 - 0.2}{(1 - x_t)^3}. \tag{4}$$

Then the thickness profile aft of the location of maximum thickness is

$$\tau(x) = \frac{t}{0.2}((1 - x)c_1 + (1 - x)^2 c_2 + (1 - x)^3 c_3), \qquad x \geq x_t. \tag{5}$$

To ensure smoothness at the point of maximum thickness, we calculate the curvature there, which is

$$\kappa = \frac{2(1-x_t)d_1 - 0.588}{(1-x_t)^2}. \tag{6}$$

A value for the leading-edge radius is based on a linear scaling of its value from the unmodified NACA 4-digit series[a].

$$a_0 = 0.2969\frac{R}{6}. \tag{7}$$

To get the remainder of the forward thickness profile, the remaining equations are applied.

$$\begin{aligned}
b_0 &= a_0\sqrt{x_t} \tag{8}\\
a_1 &= 0.5\kappa x_t + (0.3 - 1.875 b_0)/x_t \tag{9}\\
a_2 &= -\kappa + (1.25 b_0 - 0.3)/x_t^2 \tag{10}\\
a_3 &= 0.5\kappa/x_t + (0.1 - 0.375 b_0)/x_t^3 \tag{11}
\end{aligned}$$

Finally, the forward thickness profile is

$$\tau(x) = \frac{t}{0.2}(a_0\sqrt{(}x) + a_1 x + a_2 x^2 + a_3 x^3), \qquad x \le x_t. \tag{12}$$

The airfoil upper and lower surfaces are constructed by adding or subtracting, respectively, half of the thickness to the mean camber line.

The model problem considered here is to select the airfoil design and an angle of attack to minimize drag subject to a fixed lift coefficient and airfoil thickness. Therefore there are five decision variables (maximum camber, location of camber, leading edge roundness, location of maximum thickness, and angle of attack), and a single equality constraint (calculated lift coefficient minus target lift coefficient).

Symbolically, for a given thickness t and target lift coefficient $\bar{c}_\ell$, the optimization problem can be written

$$\begin{aligned}
F(t, \bar{c}_\ell) &= \min_{\mathbf{x}\in\mathbb{R}^5} f(\mathbf{x}, t) \tag{13}\\
\text{s.t.} \quad & h(\mathbf{x}, t, \bar{c}_\ell) = 0 \tag{14}
\end{aligned}$$

where $f(\mathbf{x}, t)$ calculates the drag coefficient, and

$$h(\mathbf{x}, t, \bar{c}_\ell) = c_\ell(\mathbf{x}, t) - \bar{c}_\ell. \tag{15}$$

In the remainder of this description, the dependence on thickness and target lift coefficient variables is dropped.

A. Inviscid Transonic Case

The first test problem considered is an inviscid transonic case with a freestream Mach number of 0.8. The target lift coefficient is $\bar{c}_\ell$=1.0, and the thickness-to-chord ratio is t=0.12. Because of the high lift coefficient and Mach number, this case involves a strong shock, and the positioning of this shock is one of the major drivers of the drag optimization. Although transonic cases at such a high lift coefficient are somewhat rare in the literature, high transonic lift is important for combat aircraft that must exhibit high maneuverability in the transonic flight regime [26]. However, three-dimensional considerations are important drivers for high-angle-of-attack transonic flight, and this is study only considers an introduction to the true shape optimization problem. We argue that the methods introduced in this paper make high-fidelity shape optimization of the true high-lift transonic wing feasible, as the computational cost of more direct high-fidelity

[a] In the original modified 4-digit series, a value of R=9 was taken as a special case with twice the expected roundness (i.e. half the curvature). This work does not duplicate that effect, but values of R greater than 10 are allowed.

optimization would be prohibitively expensive. The problem is restricted to the simpler case so that the costs of various optimization approaches can be compared.

The fidelity levels of this problem include an increasing solution approximation order, in the context of a variable-order finite element method, followed by hanging-node spatial mesh adaptation. A description of the flow calculation for each value of the fidelity parameter η is presented in Table 4.

Table 4. Description of fidelity levels indexed by the parameter η for the inviscid transonic shape optimization.

Parameter value (η)	Approximation Order	Number of Mesh Adaptations
0	0	0
1	1	0
2	2	0
3	2	1
n	2	$n-2$

B. Viscous Airfoil Optimization

The second optimization problem considers an airfoil in a freestream Mach number of 0.5 with a target lift coefficient of $\bar{c}_\ell$=0.2. The thickness-to-chord ratio remains at 0.12, and the Reynolds number is 5×10^3. Because of the presence of a boundary layer (even though it is quite thick for this low-Reynolds-number case), solving the low-order cases on the chosen initial coarse mesh is not possible for this problem. Instead, all solutions are calculated using a $p = 2$ (third-order) method, and the fidelity parameter is equal to the number of mesh adaptations, as shown in Table 5.

Table 5. Description of fidelity levels indexed by the parameter η for the viscous shape optimization.

Parameter value (η)	Approximation Order	Number of Mesh Adaptations
0	2	0
1	2	1
2	2	2
n	2	n

III. Optimization Approach

For the most part, a standard BFGS [27–30] algorithm is applied, with a sequential quadratic programming (SQP) approach [31] so that the equality constraint may be handled easily. However, some modifications are made to this standard approach to accommodate the multiple levels of fidelity.

During the line search, an evaluation is accepted based on a merit function ϕ. The sequential quadratic programming technique gives a proposed next design $\mathbf{x}_k + \mathbf{p}_k$, and this defines the search direction $\mathbf{p}_k$. If the point suggested by SQP does not represent a decrease in the merit function, a smaller step is taken in the same direction. Thus the line search becomes a matter of finding a value of the scalar α (usually between 0 and 1) such that $\mathbf{x}_k + \alpha\mathbf{p}_k$ gives a decrease in the merit function.

$$\phi_k(\alpha) = f(\mathbf{x}_k + \alpha\mathbf{p}_k) + \mu_k \sum_{i=1}^{m} |h_i(\mathbf{x}_k + \alpha\mathbf{p}_k)|, \qquad \mu_k = \max_i |\lambda_{k,i}|. \tag{16}$$

The constraints h_i for $i = 1, \ldots, m$ are each associated with a Lagrange multiplier λ_i. The value $\lambda_{k,i}$ denotes the value of this Lagrange multiplier calculated during the SQP process at the beginning of the iteration. Selecting the parameter μ_k at least as large as the largest of these Lagrange multipliers ensures that the SQP direction $\mathbf{p}_k$ is a descent direction of the merit function.

A. Fidelity Increases

There are many options for when to increase (or decrease) the fidelity parameter η. The simple and practical option proposed here is to estimate an error at each fidelity level and to increase fidelity whenever the expected decrease in the objective function is smaller than that error.

Suppose at iteration k_0 that our method has been decided to increase the fidelity,

$$\eta_{k_0} \leftarrow \eta_{k_0} + 1. \tag{17}$$

One way to estimate the error, which is implemented in the present study, is to evaluate the objective function and constraints at *one level higher of fidelity* and define the following measures of error:

$$\sigma_f = f(\mathbf{x}_k, \eta_k + 1) - f(\mathbf{x}_k, \eta_k), \qquad\qquad \sigma_h = h(\mathbf{x}_k, \eta_k + 1) - h(\mathbf{x}_k, \eta_k). \tag{18}$$

As an alternative to this extra evaluation, one could use an output-based error estimate in this step, and this is our plan for future work. In this work, the error is added to the constraint evaluation at each iteration, so that the constraint becomes

$$h(\mathbf{x}, \eta_k) + \sigma_h = 0. \tag{19}$$

The benefit of doing this is that the early portion of the optimization, which is performed at lower fidelity, does not get trapped driving inaccurate constraints to zero.

Fidelity increases are then made whenever both the expected decrease and the previous decrease in the merit function are below a tolerance according to

$$\max\left\{-\Delta\phi_{k-1}, -\Delta\phi_k, -\widetilde{\Delta\phi}_{k+1}\right\} \leq |\sigma_f| + \mu_k \|h_k\| \|\sigma_h\|, \tag{20}$$

where ϕ is the merit function used during the line search, and $\widetilde{\Delta\phi}_{k+1}$ is an estimate of the merit function decrease during the next iteration. The estimated function decrease is approximated very roughly using the slope of the merit function:

$$\widetilde{\Delta\phi}_{k+1} = \frac{\phi'_{k+1}(0)}{1 + \mu_{k+1}} \tag{21}$$

This simple approach works fairly well, and does not get "stuck" in low-fidelity solutions because it moves to the next fidelity level if consecutive line searches fail.

B. Geometry and Mesh Deformation

For each new value of the design variables $\mathbf{x}$, a new mesh must be obtained. Instead of creating an entirely new mesh, we use a two-step process to deform a baseline mesh (either from the previous iteration or from an original mesh of a NACA 0012-64) so that it conforms to the new airfoil. After the geometry of the new airfoil is computed using the formulas described in Sec. II, the first step is to snap all airfoil boundary nodes in the mesh to the updated geometry. This is done based on an arc-length parametrization of the two airfoils' (that is, the old one that is being deformed and the target design that is having a mesh snapped to it) upper and lower surfaces. This method prevents movement of the leading edge and distortion that can occur when the two airfoils differ significantly in total arc length.

Next, the remaining nodes in the mesh are moved using a radial basis function technique. From the first step, each mesh node that is on the boundary of the original airfoil has an assigned change in both x- and y-coordinates. These shifts are then interpolated onto the remaining nodes using a radial basis function network, which is a common method for interpolation on unstructured data sets [32]. The result is a shift in both coordinates for each node in the mesh, and the other structure (for example connectivity and number of elements) remains unchanged.

This technique has been applied in several scenarios in this work. In the baseline multifidelity routine, a fixed coarse mesh is deformed to fit to the modified geometry, and the mesh adaptations occur starting from that deformed coarse mesh. In other cases, the deformation is applied to uniform fine meshes or meshes that have been adapted one or more times. In all cases, a consistent mesh without surface bumps or negative volumes was produced.

IV. Discretization and Adaptation

We discretize the compressible Navier-Stokes equations using a discontinuous Galerkin method on unstructured, quadrilateral, body-fitted meshes. We approximate the state using order p polynomials on each element of a tessellation of the domain. We use the Roe Riemann solver [33] for convective fluxes and the second form of Bassi and Rebay (BR2) [34] for the viscous treatment. Details on the spatial discretization can be found in previous work [20, 21]. We write the final discrete form as

$$\mathbf{R}_H(\mathbf{U}_H, \mathbf{x}) \quad = \quad \mathbf{0}, \tag{22}$$

where $\mathbf{R}_H$ is the residual vector, the components of which are nonlinear functions of the state vector $\mathbf{U}_H$ and the geometry parameters $\mathbf{x}$. The subscript H denotes the fidelity of the discretization and encompasses the approximation order and mesh refinement level.

For a scalar output $J_H(\mathbf{U}_H, \mathbf{x})$, we solve a linear system to determine the associated discrete adjoint solution, $\mathbf{\Psi}_H$, which represents the sensitivity of that output to residual source perturbations [35]. This linear system is given by the following equation

$$\left(\frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H}\right)^T \mathbf{\Psi}_H + \left(\frac{\partial J}{\partial \mathbf{U}_H}\right)^T \quad = \quad \mathbf{0}. \tag{23}$$

An example of an adjoint solution is shown in Fig. 1. Note how the regions of interest in the flow solution, shown in Fig. 1a, are notably different from the regions of interest in the adjoint solution shown in Fig. 1b. Cells can be selected for refinement for either reason, as the definition of the output error estimate shows in the following paragraphs.

Since $\mathbf{\Psi}_H$ gives us the linear sensitivity of J to residual perturbations, we use it to efficiently calculate the sensitivities of J to the parameters in $\mathbf{x}$,

$$\frac{\mathrm{d}J_H}{\mathrm{d}\mathbf{x}} = \mathbf{\Psi}_H^T \frac{\partial \mathbf{R}_H}{\partial \mathbf{x}} + \frac{\partial J_H}{\partial \mathbf{x}}. \tag{24}$$

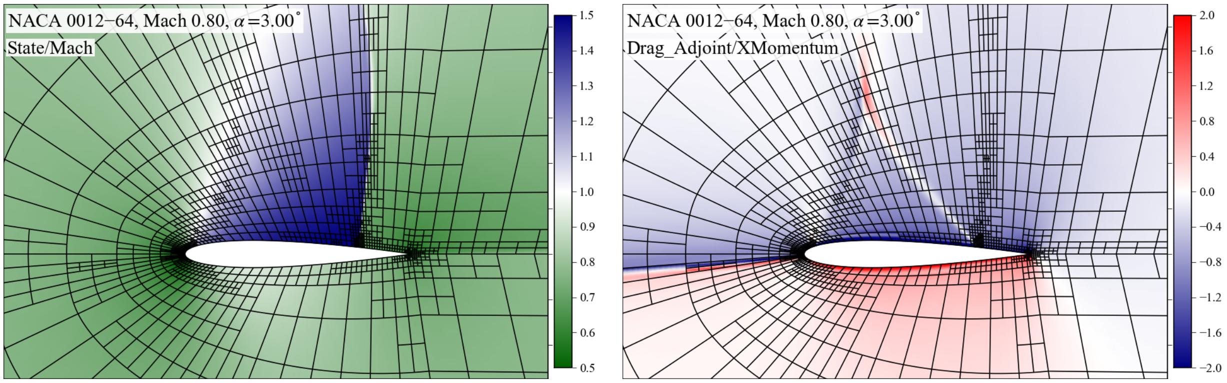
Note, these are total sensitivities that account for the state changes that must occur per Eq. 22 when $\mathbf{x}$ changes. We use these sensitivities for gradient-based optimization.

We also use the adjoint to drive an output-based strategy for spatially adapting the computational mesh with the goal of reducing numerical errors. We approximate the numerical error in J_H by comparing it, “hypothetically”, to an output computed on a finer discretization, denoted by subscript h, which in this work is the same mesh but with the approximation order incremented to $p + 1$. The comparison is hypothetical because we do not actually evaluate the state or output on the finer discretization space. Instead, we only evaluate the fine space adjoint and residual obtained after injecting the coarse space solution to the fine space. The error estimate is then given by an adjoint-weighted residual,

$$\text{output error } = \delta J \approx \mathbf{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H), \tag{25}$$

where $\mathbf{U}_h^H$ is the coarse solution injected to the fine space. The fine-space adjoint, $\mathbf{\Psi}_h$, is obtained approximately by applying several (five) iterations of an inexpensive iterative smoother to the injected coarse-space adjoint.

When a fidelity increase is desired, the error estimate in Eq. (25) is localized to mesh elements, and a fraction of the elements with the largest error estimate is chosen for isotropic refinement using a hanging-node refinement strategy [22]. In this case, the fraction is fixed at 10% of the existing cells for each refinement. Other useful schemes for deciding how many cells to refine have been discussed by Nemec et al. [36]. A schematic of the mesh adaptation process is shown in Fig. 2.



a) State solution, Mach number **b) Drag adjoint, x-momentum component**

Figure 1. Adjoint example.

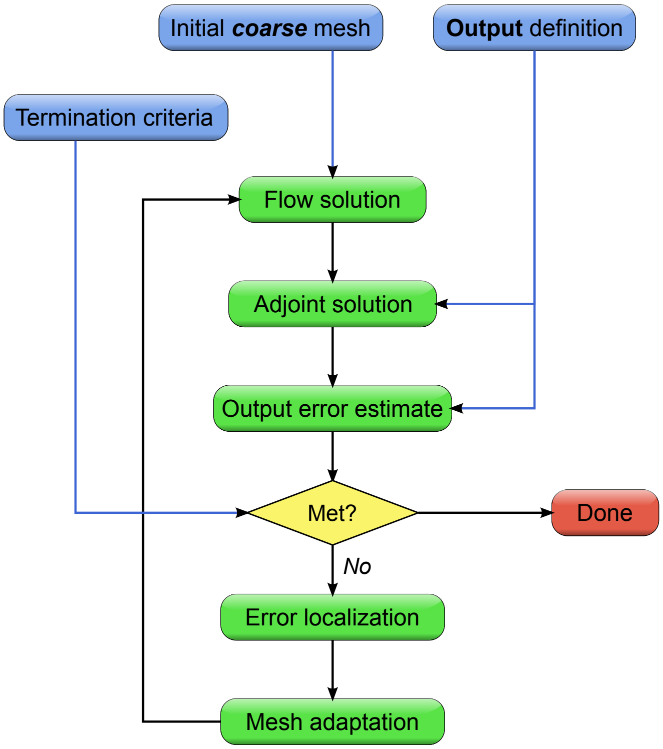


Figure 2. Graphical representation of mesh adaptation process.

V. Results

Some results of the baseline strategy as described in Sections II and III are shown in this section. The method is applied to the transonic test problem in Sec. V.A and to the viscous test problem in Sec. V.B. In Sec. C, we return to the transonic problem, but compare the results of the proposed multifidelity optimization method with several alternative approaches.

A. Inviscid Transonic Airfoil

Fig. 3 shows the results for the transonic inviscid test problem. In particular, it shows the Mach number at interesting points in the optimization. The initial evaluation, which is a NACA 1412-64 airfoil at 3° angle of attack, is shown in Fig. 3a, and Figs. 3b–3e show the last results before each fidelity increase.

The plots are constructed so that green represents subsonic flow and purple represents supersonic flow. Flow close to the local sound speed is shown in white. Because of the relatively high lift, a shock is only present on the upper surface even in the initial condition. As the fidelity increases, the resolution of this shock increases dramatically. In Fig. 3b, it is not clear from the coarse mesh and low solution order that a shock is even present. Without refining the mesh, increasing the solution order to 2, as in Fig. 3c, makes the shock notably better resolved. As the mesh is refined, the shock, the leading edge, and the trailing edge are the main targets for refinement, although other areas are refined, too. However, portions of the shock that are not close to the airfoil surface receive very little refinement, even though the local gradients are high. The cause for this is that the drag is not sensitive to those parts of the shock that are not close to the surface, and this highlights an advantage of output-based adaptation versus other adaptation methods.

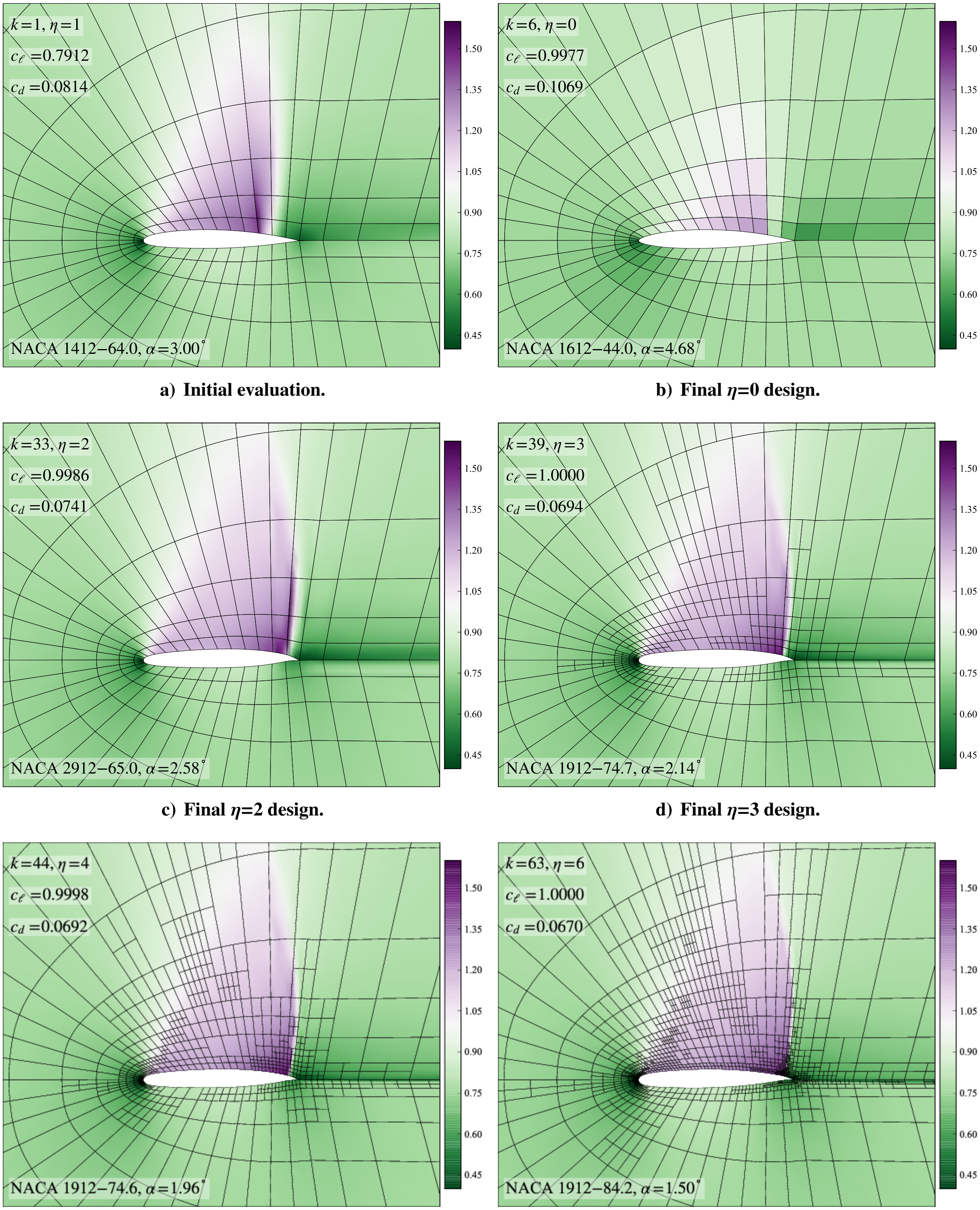
This advantage is most significant when the output value used to drive adaptation the same as or close to the optimization objective function. In this case, drag is the output used to indicate where mesh refinement should take place, and this is not quite a perfect match for the optimization problem because it is also important to estimate the lift accurately in order to satisfy the constraint. Work is currently ongoing to improve adaptive indicators that address this effect of equality constraints. In most cases, even if the aerodynamic analysis is a part of a larger optimization problem in which the overall objective function is not directly tied to lift, drag, or the moment on a wing or airfoil section, it is likely that the important output from the aerodynamic analysis is a function of these three parameters. Likely exceptions include heat transfer, aerodynamic derivatives (such as sensitivity of lift to angle of attack), and driving an aeroelastic analysis. The last example is particularly challenging because the output of interest is the pressure profile along the entire surface, and as such it is not a scalar.

Returning to the results of Fig. 3, an interesting feature is that the lowest-fidelity solution (in Fig. 3b) does not appear to be "between" the initial and final solutions. In particular, it has a very sharp nose, sharper than in the initial condition, whereas the later designs have a nose with approximately the same bluntness as the initial condition. What is not apparent from the figures is that the η=0 solution does a very poor job of estimating the lift. The correction in Eq. (19) helps somewhat, but at best it means that the lift estimate cannot be trusted. The real benefit of this method is that the optimizer does not drive the η=0 solution to its optimum. This could be far from the true optimum indeed, and that could result in the optimizer finding a local minimum once the fidelity is increased. Naturally, this method does not guarantee that the changing fidelity does not result in finding a local optimum that would otherwise be avoided, but it appears to lessen the probability of that occurrence.

As for the numerical values of the results, there is a significant reduction in drag from an initial value of about 0.10 to nearly 0.07 (the value in Fig. 3a should not count because the lift is much lower). The majority of this decrease occurs with η=1, and sequentially smaller decreases are found after that. The higher-fidelity solutions are essentially refining the lower-fidelity results, although substantial computational effort is still spent on the later stages, which can be seen from the iteration numbers (k) in Fig. 3. It may be possible to reduce the number of evaluations required at the higher fidelities. Correcting slight errors in the lift constraint immediately after a fidelity increase appears to be one cause for the number of evaluations, and slight errors in the gradients may also be a factor.

B. Viscous, Subsonic Airfoil

Figure 4 shows the results from the viscous sample problem using the methods of Sections II and III. In this case, there is no locally supersonic flow, so a different color scheme is used. White represents regions of the flow with a Mach number close to the freestream value of 0.5. Regions in blue represent flow with a local



a) Initial evaluation.

b) Final η=0 design.

c) Final η=2 design.

d) Final η=3 design.

e) Final η=4 design.

f) Final design.

Figure 3. Plots of local Mach number for inviscid transonic high-lift case. The conditions are M_∞=0.8, c_ℓ=1.0. Green regions show subsonic flow, and purple regions show supersonic flow.

Mach number below this value, while red represents higher-speed flow.

The results for this low-Reynolds-number case are unusual in at least two respects. First, the optimizer selected a very sharp nose, which is not typical for subsonic airfoil sections. Running a few slight variations of this problem, such as changing the initial condition, did not always lead to a sharp nose, although the final drag value was consistently very close to the value shown in Fig. 4e.

The second peculiar aspect is the negative camber. Optimizing an airfoil for a single positive-lift value usually results in positive camber, which allows the angle of attack to be decreased while still meeting the lift condition. For this case, which has a fairly high Mach number but a very low Reynolds number, combined with a significant thickness (t=0.12), the negative camber was a consistent result. One test that we ran to investigate the robustness of the counter-intuitive design was to repeat the optimization with the lift constraint reversed to $c_\ell = -0.2$. The result was an airfoil with a positive camber, which was almost a perfect opposite to the design shown in Fig. 4e.

Because Fig. 4 shows solutions to a subsonic problem, there is no shock, which results in fewer obvious places for the solution to adapt. The primary areas of the mesh that get adapted are regions of the flow near the airfoil surface, along the near portion of the wake, and the areas above, below, and upstream of the leading edge. Unlike the transonic test problem, there is a fairly consistent trend from the initial condition to the optimized solution. The sharpening of the nose mostly occurs during the η=2 phase (between Figs. 4c and 4d).

The drag reduction for this problem (about 7%) is much less than what it was for the transonic problem. At the same time, fewer iterations were spent at the higher fidelities. In the later phases, there is a fidelity increase every three iterations, which is a hard-coded minimum. This suggests that for easier problems, the fidelity increases rapidly once a certain level is reached. The possibility exists that this will also be true for more challenging problems, although the necessary level of fidelity may be higher for problems with shocks such as the transonic test case shown in Fig. 3.

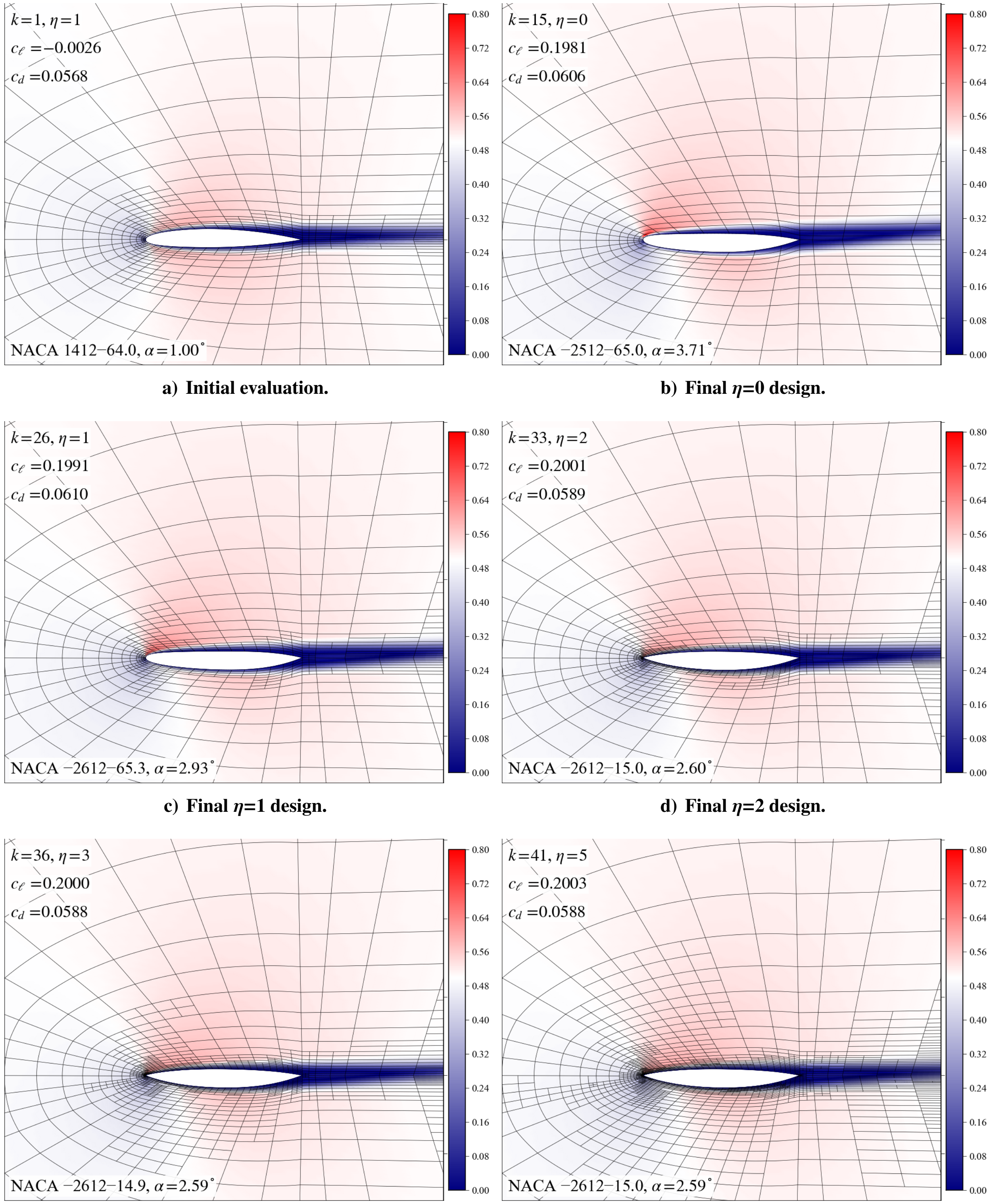
If the lower-fidelity solutions can be used as initial conditions for higher-fidelity flow solutions efficiently, the benefits to multifidelity optimization could be even greater. Such a method was tried for the present study, but the results were ambiguous, and the time savings was either unimpressive or slightly negative in the attempts tried. The reason appears to be that deforming the mesh while leaving the states (density, momentum, energy) unaltered detracts somewhat from the intuitive benefit of starting a flow analysis from a state that's near the solution. However, the trials were far from exhaustive, and it remains a potential way to go rapidly from a level of fidelity sufficient for optimization (which is determined automatically during the iterations) to a higher level once the optimization is nearly complete. Such a technique, if successful, will increase the efficiency over classical high-fidelity optimization by an even greater factor.

C. Comparison with Other Optimization Strategies

The purpose of this section is to begin to determine the benefits of each of the two main aspects to the proposed optimization method: varying fidelity and adaptive mesh refinement. This results in four optimization methods, which comes from turning on or off both of these aspects. In addition, we tested the effect of restarting high-fidelity iterations with the results from previous lower-fidelity iterations. The results, though preliminary since we consider only one test problem, suggest that there is significant savings due to both adaptive mesh refinement and varying fidelity.

Figure 5 shows the result of an optimization using a uniformly refined mesh, which can be compared to Fig. 3e. The flow solution is quite similar, but the mesh has significantly more elements. The Mach number profiles from the remaining methods are all very similar to those of either Fig. 3 or Fig. 5, so snapshots are not shown for those optimizations.

Table 6 shows two primary indicators of the five methods considered: final value of the objective function and total computational time. All results were run using 32 processors and on the same machine. Method 1 corresponds to the results from Sec. A and Fig. 3. For method 2, a small change was made. Instead using a uniform freestream flow on the coarse mesh as the initial condition, previous results are used.



a) Initial evaluation.

b) Final η=0 design.

c) Final η=1 design.

d) Final η=2 design.

e) Final η=3 design.

f) Final design.

Figure 4. Plots of local Mach number for viscous case. The conditions are M_∞=0.5, c_ℓ=0.2. Note the negative camber selected by the optimizer.

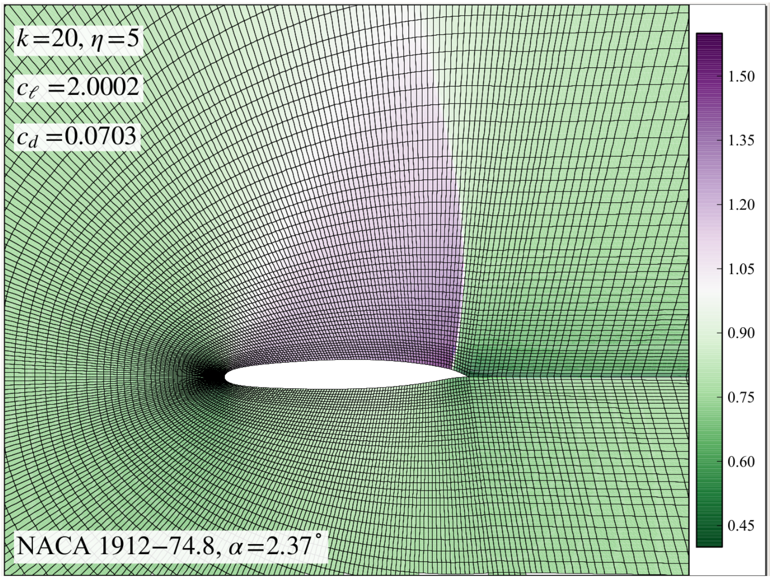


Figure 5. Final result of fixed-fidelity optimization using a uniformly refined grid.

Table 6. Comparison of results for various schemes

Number	Method	Wall time	Final c_d
1	Adaptive method, baseline	3606 s	0.0670
2	Adaptive method with restart	4043 s	0.0668
3	Fixed fidelity, $\eta = 6$	4.8 hr	0.0780
4	Uniformly refined mesh, multifidelity	15.2 hr	0.0660
5	Uniformly refined mesh, fixed fidelity	20.5 hr	0.0703

Specifically, at iteration k, which has a fidelity of η, the CFD is initialized using the result from iteration $k-1$ at fidelity $\eta-1$. This choice makes each function evaluation quicker by initializing the analysis with a state close to the solution, but by using the previous $\eta-1$ solution (instead of taking the final solution from iteration k), there is still one round of adaptive mesh refinement to adapt to small changes in the flow from one iteration to the next. Obviously there are several different choices for initial conditions that can be used, and this is just one possibility that we guessed would provide speedup without compromising accuracy. However, this method actually took somewhat longer to complete the optimization. The explanation appears to be that although most evaluations took less time to compute, the number of evaluations is a far more important factor in total time for the optimization. Since the initial condition changes from iteration to iteration, so too does the objective function, albeit very subtly. This results in a small error in the gradient, which in this case increased the average number of evaluations per line search. Because the residuals are near machine zero as a result of our Newton solver, the subtle change in the objective function is primarily due to the mesh deformation stage. If a way to overcome these slight inconsistencies in the objective function can be found, using a restart strategy such as this one may become beneficial. In the meantime, the results indicate that the increased setup effort required to use a restart strategy is unlikely to be justified by future computational time savings.

Method 3 simply removes the multifidelity aspect of the optimization but retains the adaptive mesh refinement. This took about 5 times longer to compute than the baseline multifidelity strategy, but it also failed to fully converge, which explains the higher drag. Instead, the algorithm terminated when the decrease was below a tolerance for four consecutive iterations. Removing the adaptive mesh refinement, as in methods 4 and 5, increased the computational time more dramatically. The speedup due to adaptive mesh refinement is essentially the same as the speedup for an individual function evaluation, and the iteration and evaluation counts were similar whether adaptive mesh refinement was used or not.

VI. Conclusions

The feasibility was demonstrated for a multifidelity shape optimization method using CFD with adaptive mesh refinement. The method applies to problems in which an integer list of fidelities can be constructed, which in this case was essentially the number of refinements to the mesh. Determination of when to increase the fidelity to the next level in this list is made automatically by the optimizer. To test the method, we applied it to two airfoil shape optimization problems using a small number of design variables.

When compared against traditional fixed-fidelity optimization on a uniform fine mesh, we observed a speedup factor of about 20 for the transonic airfoil test problem. Although one must be careful to generalize this result to other optimization problems, the present result is promising and justifies applying the techniques to more problems in the future. In addition, as the benefits of adaptive methods generally increase with increasing accuracy demands, there is reason to believe that the speedup will increase with the highest level of fidelity that is sought.

References

[1] Martins, J. R. R. A. and Lambe, A., "Multidisciplinary Design Optimization: Survey of Architectures," *AIAA Journal*, 2013.

[2] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233–260.

[3] Anderson, W. K. and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," Tech. Rep. Report 97-9, ICASE, 1997.

[4] Elliott, J. and Peraire, J., "Practical Three-Dimensional Aerodynamic Design and Optimization Using Unstructured Meshes," *AIAA Journal*, Vol. 35, No. 9, 1997, pp. 1479–1485.

[5] Nielsen, E. and Anderson, W., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419.

[6] Mohammadi, B. and Pironneau, O., "Mesh Adaption and Automatic Differentiation in a CAD-free Framework for Optimal Shape Design," *International Journal for Numerical Methods in Fluids*, Vol. 30, 1999, pp. 127–136.

[7] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, 2005, pp. 33–62.

[8] Eldred, M. and Dunlavy, D., "Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models," *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, number AIAA-2006-7117, Portsmouth, VA*, Vol. 199, 2006.

[9] Robinson, T., Eldred, M., Willcox, K., and Haimes, R., "Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping," *AIAA Journal*, Vol. 46, No. 11, 2008, pp. 2814–2822.

[10] Alonso, S. C. C. J., Kroo, I. M., and Wintzer, M., "Multifidelity Design Optimization of Low-Boom Supersonic Jets," *Journal of Aircraft*, Vol. 45, 2008, pp. 106–118.

[11] Alonso, S. C. C. . J. and Kroo, I. M., "Two-Level Multifidelity Design Optimization Studies for Supersonic Jets," *Journal of Aircraft*, Vol. 46, 2009, pp. 776–790.

[12] Geiselhart, K. A., Ozoroski, L. P., Fenbert, J. W., Shields, E. W., and Li, W., "Integration of Multifidelity Multidisciplinary Computer Codes for Design and Analysis of Supersonic Aircraft," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011, AIAA Paper 2011-465.

[13] March, A. and Willcox, K., "Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives," *AIAA Journal*, Vol. 50, No. 5, 2012, pp. 1079–1089.

[14] March, A. and Willcox, K., "Multifidelity Approach for Parallel Multidisciplinary Optimization," *12th AIAA Aviation Technology, Integration, and Operations Conference and 14th AIAA/ISSMO*, 2012.

[15] Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., and Newman, P. A., “Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models,” *Journal of Aircraft*, Vol. 38, No. 6, 2001, pp. 1093–1101.

[16] Robinson, T. D., Eldred, M. S., Willcox, K. E., and Haimes, R., “Strategies for Multifidelity Optimization with Variable Dimensional Hierarchical Models,” *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamcs, and Materials Conference*, 2006, AIAA Paper 2006-1819.

[17] Mavriplis, D. J., Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Levy, D., and Murayama, M., “Grid Quality and Resolution Issues from the Drag Prediction Workshop Series,” AIAA Paper 2008-930, 2008.

[18] Fidkowski, K. J. and Darmofal, D. L., “Review of output-based error estimation and mesh adaptation in computational fluid dynamics,” *AIAA J.*, Vol. 49, No. 4, 2011, pp. 673–694.

[19] Nemec, M. and Aftosmis, M. J., “Output Error Estimates and Mesh Refinement in Aerodynamic Shape Optimization,” *51st AIAA Aerospace Sciences Meeting*, 2013, AIAA Paper 2013-0865.

[20] 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.

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

[22] Ceze, M. A. and Fidkowski, K. J., “An anisotropic hp-adaptation framework for functional prediction,” *American Institute of Aeronautics and Astronautics Journal*, Vol. 51, 2013, pp. 492–509.

[23] Kast, S. M. and Fidkowski, K. J., “Output-based Mesh Adaptation for High Order Navier-Stokes Simulations on Deformable Domains,” *Journal of Computational Physics*, 2013, In Press.

[24] van Schorjenstein Lantman, M. P. and Fidkowski, K. J., “Adjoint-Based Optimization of Flapping Kinematics in Viscous Flows,” *21st AIAA Computaional Fluid Dynamics Conference*, 2013, AIAA Paper 2013-2848.

[25] Ladson, C. L., Cuyler W. Brooks, J., Hill, A. S., and Sproles, D. W., “Computer Program To Opbtain Ordinates for NACA Airfoils,” Tech. rep., NASA, 1996.

[26] Whitford, R., *Design for Air Combat*, Janes, 1st ed., 1987.

[27] Broyden, C. G., “The convergence of a class of double-rank minimization algorithms,” *Numerical optimization: Theoretical and practical aspects*, Vol. 6, 1970, pp. 76–90.

[28] Goldfarb, D., “A Family of Variable Metric UUpdate Derived by Variational Means,” *Mathematics of Computation*, Vol. 24, 1970, pp. 23–26.

[29] Fletcher, R., “A New Approach to Variable Metric Algorihms,” *Computer Journal*, Vol. 13, 1970, pp. 317–322.

[30] Shanno, D. F., “Conditioning of quasi-Newton methods for function minimization,” *Mathematics of Computation*, Vol. 24, 1970, pp. 647–656.

[31] Byrd, R. H., Tapia, R. A., and Zhang, Y., “An SQP augmented Lagrangian BFGS algorithm for constrained optimization,” *SIAM Journal on Optimization*, Vol. 2, 1992, pp. 210–241.

[32] Broomhead, D. S. and Lowe, D., “Multivariable functional interpolation and adaptive networks,” *Complex Systems*, Vol. 2, 1988, pp. 321–355.

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

[34] Bassi, F. and Rebay, S., “GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations,” *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by K. Cockburn and Shu, Springer, Berlin, 2000, pp. 197–208.

[35] Fidkowski, K. J. and Darmofal, D. L., “Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics,” *American Institute of Aeronautics and Astronautics Journal*, Vol. 49, No. 4, 2011, pp. 673–694.

[36] Nemec, M., Aftosmis, M. J., and Wintzer, M., “Adjoint-Based Adaptive Mesh Refinement for Complex Geometries,” *46th Aerospace Sciences Meeting & Exhibit*, 2008, AIAA Paper 2008-0725.