

# Robust airfoil optimization to achieve drag reduction over a range of Mach numbers

W. Li, L. Huyse, and S. Padula

**Abstract** An airfoil shape optimization method that reduces drag over a range of free stream Mach numbers is sought. We show that one acceptable choice is a weighted multipoint optimization method using more design points than there are free-design variables. Alternate methods that use far fewer design points are explored. A new method called profile optimization is developed and analyzed. This method has three main advantages: (a) it prevents severe degradation at the off-design points by using a smart descent direction, (b) there is no random airfoil shape distortion for any iterate it generates, and (c) it is not sensitive to the number of design points. For illustration purposes, we use the profile optimization method to solve a lift-constrained drag minimization problem for 2-D airfoil in Euler flow with 20 free-design variables. A comparison with other airfoil optimization methods is also included.

---

**Key words** robust optimization, airfoil shape optimization, consistent drag reduction, lift-constrained drag minimization, adaptive weight adjustment

## 1 Introduction

Optimization of aerospace vehicles or aircraft wings is based on a mathematical model of the physical reality. The design variables are parameters in the mathematical model and changes in these design variables result in new

---

*Received September 3, 2001*

*Revised manuscript received November 29, 2001*

W. Li<sup>1</sup>, L. Huyse<sup>2</sup>, and S. Padula<sup>3</sup>

<sup>1</sup> Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, USA

e-mail: wli@odu.edu

<sup>2</sup> ICASE, NASA Langley Research Center, Hampton, VA 23681, USA

e-mail: Luc.Huyse@swri.org

<sup>3</sup> Multidisciplinary Optimization Branch, NASA Langley Research Center, Hampton, VA 23681, USA

e-mail: S.L.PADULA@LARC.NASA.GOV

physical objects. Aerodynamic optimization is a process of finding a set of design variables that corresponds to a new physical object with better aerodynamic and structural properties.

For airfoil shape optimization, design variables are parameters that define the airfoil shape. One accepted practice for modelling airfoil shapes is to use empirical algebraic expressions that are based on the knowledge of aerodynamic properties of airfoils. There are two advantages to such an airfoil model.

- (a) Each set of design variables (such as maximum thickness, camber, radius of leading edge, etc.) generates an airfoil with aerodynamic properties that are well understood by experts.
- (b) The number of design variables is small, thus the corresponding optimization problem is computationally affordable.

A drawback to such an airfoil model is that the optimization process is merely selecting a desirable airfoil shape among the predetermined shapes given by the specific airfoil model. The usefulness of the optimal airfoil is essentially determined by the usefulness of the airfoil model.

To achieve truly innovative airfoil designs, it is therefore desirable to consider “free-form” shape optimization, which allows “truly optimum” designs to be computed (Drela 1998, Section 1). Here “free-form” means geometric shapes represented by a linear combination of general basis functions (such as splines or sinusoidal basis functions). However, there are several challenges associated with “free-form” airfoil shape optimization. Drela (1998, Subsection 5.1) pointed out that if presented with sufficient design model resolution, an optimizer will readily (and annoyingly) manipulate and exploit the flow at the smallest significant physical scales present that tends to produce improved performance only near the sampled operating conditions. The point-optimized airfoil often shows a possibly severe degradation in the off-design performance and optimized aerodynamic shapes are usually “noisy” and usually require a posteriori smoothing.

The main objective of this paper is to develop a robust airfoil optimization scheme for achieving consistent

drag reduction over a given Mach range with only a few design points. This scheme (called the profile optimization method) has the following three main advantages: (a) it prevents severe degradation in the off-design performance by using a common descent direction for the drag at all design points in each optimization iteration, (b) there is no random airfoil shape distortion for any iterate it generates, and (c) it is not sensitive to the number of design points.

The term “robustness” has been used to mean a variety of things. For optimization under uncertainty, we can distinguish the following meanings and goals of “robust optimization”.

1. Identify designs that minimize the variability of the performance under uncertain operating conditions. This is the objective of Taguchi methods (Fowlkes *et al.* 1995), which are most practical when the expected value of the performance can be adjusted at negligible cost.
2. Mitigate the detrimental effects of the worst-case performance. This is the objective of minimax strategies, which can be used to find a design with the optimal worst-case performance (Ben-Tal and Nemirovski 1997).
3. Provide the best overall performance of a system by maximizing the expected value of its utility (Huysse and Lewis 2001).
4. Achieve consistent improvements of the performance over a given range of uncertainty parameters. This is the main objective of this work.

In this paper we use the uncertainty of the flight speed to show how the profile optimization method can be used for optimization under uncertainty.

The paper is organized as follows. In Section 2, we give a brief introduction to the airfoil optimization problem. Section 3 is devoted to Drela’s hypothesis on the necessity of using at least  $\mathcal{O}(m)$  design points for multipoint airfoil shape optimization, where  $m$  is the number of free-design variables. The main result is a mathematical argument that provides some insight on why the multipoint airfoil shape optimization needs at least  $(m + 1)$  design points. We present two airfoil shape optimization formulations in Section 4 and prove the mathematical equivalence between these two formulations under finite sampling in Section 5. The discrete optimization formulations provide accurate approximations of their continuous versions only when a sufficiently large number of design points are used, which makes them less attractive as practical robust optimization tools due to high computational costs associated with a large number of function/gradient evaluations in each optimization iteration. The profile optimization method is introduced in Section 6 as a robust optimization tool that works with only a few design points. Numerical simulation results are given in Section 7 and final conclusions are drawn in Section 8.

## 2

### Airfoil shape optimization

Recently, there has been significant progress in airfoil shape optimization (see Anderson and Bonhaus 1999; Anderson and Venkatakrishnan 1997; Drela 1998; Nielsen and Anderson 1998, and references therein). These papers demonstrate impressive shape optimization using high-fidelity CFD codes, reliable grid generation, and numerically efficient sensitivity calculations. Equally impressive progress has been made in optimization of 3-D wings (Elliott and Peraire 1997, 1998; Reuther *et al.* 1999; Nielsen and Anderson 2001) and in coupled structural-aerodynamic optimization (Gumbert *et al.* 2001). However, with a few exceptions such as (Drela 1998) and (Reuther *et al.* 1999), these aerodynamic shape optimization projects all find optimal shapes based on one fixed operating condition. In this paper, we study a simplified 2-D airfoil shape optimization problem using a low-fidelity Euler flow solver, but we include uncertainty in the operating conditions for the airfoil shape optimization.

Airfoil shape optimization is a PDE-constrained optimization problem. A general mathematical framework for multipoint airfoil optimization can be described as follows (Drela 1998):

$$\min_{D, \alpha_i} \sum_{i=1}^r w_i c_d(D, \alpha_i, M_i) \quad (1)$$

subject to

$$c_l(D, \alpha_i, M_i) \geq c_l^* \quad \text{for } 1 \leq i \leq r \quad (2)$$

and  $D \in \mathcal{F}$ , where  $w_i$ ’s are positive weights,  $c_l^*$  is the minimal lift value,  $\mathcal{F}$  is a given feasible set for geometric design variables,  $M_i$ ’s are the Mach numbers,  $\alpha_i$ ’s are the angles of attack, and  $c_d$  (or  $c_l$ ) is the drag (or lift) coefficient. The feasible set  $\mathcal{F}$  mainly depends on the airfoil model. In this paper, we use splines for airfoil shape modelling: the components of  $D$  are the control points for a spline curve and the feasible set  $\mathcal{F} = \mathbb{R}^m$ , where  $m$  is the number of geometric design variables.

Drela (1998) studied the behavior of the optimization solutions of (1) in two-dimensional viscous flow when the number of free-design variables is relatively large. He concluded that increasing the number of geometric design variables requires a corresponding increase in the number of design conditions (Mach numbers) used in the multipoint optimization problem (1). He also suggested that it is necessary to have  $r = \mathcal{O}(m)$ , where  $m$  is the number of free-design variables, to achieve a smooth airfoil geometry. Other notable conclusions made by Drela (1998) are as follows.

- Near-continuous sampling of the operating space (i.e., in the range of Mach numbers) may be required in the

theoretical limit of a general airfoil design problem with a very large number of degree of freedom (for geometric variables) — a very expensive proposition.

- The most suitable operating points to be actually sampled in multipoint airfoil optimization (i.e.,  $M_1, M_2, \dots, M_r$ ) are not apparent a priori. From limited experience, sampling somewhat beyond the expected operating range appears to be best.
- The point weights (i.e.,  $w_1, w_2, \dots, w_r$ ) used in multipoint airfoil optimization are arbitrary, and their appropriate values can not be easily estimated without prior experience.
- Optimized aerodynamic shapes are usually “noisy” and require a posteriori smoothing.

### 3

#### The critical number of design points for multipoint airfoil optimization

In this section, we try to validate Drela’s hypothesis on the necessity of having the number of design points proportional to the number of design variables. Specifically, we give a mathematical argument to explain why the number of design points must be greater than the number of free-design variables in order to avoid point-optimization at the design points. By point-optimization, we mean that drag is minimized at the  $r$  design points but rises unexpectedly at some of the off-design points.

In (1), the angle of attack,  $\alpha_i$ , is predominantly determined by the constraint on the lift coefficient at  $M_i$ . Therefore, it is theoretically possible to eliminate the constraint and consider the following unconstrained reformulation of (1):

$$\min_D \sum_{i=1}^r w_i f(D, M_i), \quad (3)$$

where  $f$  is a function related to the drag coefficient.

Let  $\hat{D}$  be an optimal solution of (3). Then

$$\sum_{i=1}^r w_i \frac{\partial f}{\partial D}(\hat{D}, M_i) = 0, \quad (4)$$

where  $\frac{\partial f}{\partial D}$  denotes the gradient of  $f$  with respect to  $D$ .

If  $r \leq m$ , then  $(r - 1) < m$ . So we can find a nonzero vector  $\Delta D$  in the orthogonal complement of the subspace of  $\mathbb{R}^m$  generated by the following  $(r - 1)$  vectors:

$$\frac{\partial f}{\partial D}(\hat{D}, M_2), \dots, \frac{\partial f}{\partial D}(\hat{D}, M_r).$$

By (4) and  $w_1 > 0$ ,  $\Delta D$  must also be orthogonal to  $\frac{\partial f}{\partial D}(\hat{D}, M_1)$ . Therefore, we obtain a nonzero vector  $\Delta D$  such that

$$\left\langle \frac{\partial f}{\partial D}(\hat{D}, M_i), \Delta D \right\rangle = 0 \quad \text{for } 1 \leq i \leq r, \quad (5)$$

where  $\langle u, v \rangle$  denotes the dot product of vectors  $u$  and  $v$ .

Let  $M_{r+1}$  be any Mach number such that

$$\left\langle \frac{\partial f}{\partial D}(\hat{D}, M_{r+1}), \Delta D \right\rangle \neq 0. \quad (6)$$

Let  $\{\hat{w}_1, \dots, \hat{w}_{r+1}\}$  be a set of arbitrary positive weights. By Taylor expansion, we get

$$\begin{aligned} \sum_{i=1}^{r+1} \hat{w}_i f(\hat{D} + t\Delta D, M_i) &= \sum_{i=1}^{r+1} \hat{w}_i f(\hat{D}, M_i) + \\ t \sum_{i=1}^{r+1} \hat{w}_i \left\langle \frac{\partial f}{\partial D}(\hat{D}, M_i), \Delta D \right\rangle &+ \mathcal{O}(t^2). \end{aligned} \quad (7)$$

It follows from (7) and (5) that

$$\begin{aligned} \sum_{i=1}^{r+1} \hat{w}_i f(\hat{D} + t\Delta D, M_i) &= \sum_{i=1}^{r+1} \hat{w}_i f(\hat{D}, M_i) + \\ t \hat{w}_{r+1} \left\langle \frac{\partial f}{\partial D}(\hat{D}, M_{r+1}), \Delta D \right\rangle &+ \mathcal{O}(t^2). \end{aligned} \quad (8)$$

By (6), we can choose  $t$  such that  $|t|$  is sufficiently small and

$$t \left\langle \frac{\partial f}{\partial D}(\hat{D}, M_{r+1}), \Delta D \right\rangle < 0.$$

Then (8) implies

$$\sum_{i=1}^{r+1} \hat{w}_i f(\hat{D} + t\Delta D, M_i) < \sum_{i=1}^{r+1} \hat{w}_i f(\hat{D}, M_i).$$

That is, after adding one more design point,  $\hat{D}$  is not an optimal solution no matter how the weights are selected.

The above argument shows that if the number of design points is less than one plus the number of free-design variables, then a large improvement at a new design point can be accomplished with only a small deterioration in the performance at the old design points. This means that it is attractive to use more design points for the multipoint optimization method.

We can also use the above argument to explain why point-optimization is likely to occur if the number of design points is no more than the number of free-design variables. For example, suppose that  $(\hat{D} + t\Delta D)$  given in the above argument is the current design vector. Then the multipoint optimization method finds the optimal solution  $\hat{D}$  by reducing the drag at the design points  $M_1, \dots, M_r$  marginally, while unintentionally increasing the drag at off-design points (such as  $M_{r+1}$ ) significantly. In other words, if the number of design points is no more than the number of free-design variables, then it is possible to have a first-order increase of the value of  $f$  at some off-design Mach number  $M_{r+1}$  (in order of  $|t|$ ) and only second-order reduction of the values of  $f$  at  $M_i$ 's ( $1 \leq i \leq r$ ) (in order of  $t^2$ ). This could result in an optimal airfoil with point-optimization behavior: low drag at the design points, but severe degradation of the performance at off-design points.

## 4

## Robust airfoil optimization formulations

For engineering design problems, an optimal design is usually obtained under some explicit/implicit assumptions. This leads to a design that works well under ideal operating conditions but may perform inadequately under non-ideal (i.e., off-design) conditions. The problem is that the optimal design does not consider the uncertainty or variability of some parameters/data that will affect the actual performance of the design in a real-world situation. Therefore, it is necessary to include uncertainties in a practical design optimization process. In this paper, we assume that  $M$  is the only uncertain parameter and  $[M_{\min}, M_{\max}]$  is the range of Mach numbers considered.

Let  $p(M)$  be the probability density function of the uncertain parameter  $M$ . Then a stochastic programming formulation for airfoil optimization under uncertainty can be described as follows (Huysse and Lewis 2001):

$$\min_{D, \alpha(\cdot)} \int_{M_{\min}}^{M_{\max}} c_d(D, M, \alpha(M)) p(M) dM \quad (9)$$

subject to

$$c_l(D, M, \alpha(M)) \geq c_l^* \quad \text{for } M_{\min} \leq M \leq M_{\max} \quad (10)$$

and  $D \in \mathcal{F}$ , where  $\mathcal{F}$  is the feasible geometric design space and  $c_l^*$  is the minimal lift value. This corresponds to the third definition of ‘‘robust optimization’’ given in the introduction.

On the other hand, one can also use the following minimax optimization formulation for robust optimization under uncertainty (Ben-Tal and Nemirovski 1997):

$$\min_{D, \alpha(\cdot)} \max_{M_{\min} \leq M \leq M_{\max}} \rho(M) c_d(D, M, \alpha(M)) \quad (11)$$

subject to the constraints given in (10). Here  $\rho(M) > 0$  is a positive weighting function of  $M$ . This corresponds to the second definition of ‘‘robust optimization’’ given in the introduction.

The constraints on the lift can be eliminated by choosing an appropriate value for the angle of attack corresponding to each  $M$ . In fact, Elliott and Peraire (1997) suggested to incorporate a means for adjusting the angle of attack to satisfy the lift constraint into the flow analysis algorithm. One problem with this elimination approach is that the angle of attack becomes a function of  $M$  and  $D$ , and the derivatives of  $\alpha$  with respect to  $D$  have to be computed in order to get the derivatives of  $c_l$  and  $c_d$  with respect to  $D$ . In this study, we retain the lift constraint to make our methods applicable to general constrained aerodynamic optimization problems (with constraints on the pitching moment and the leading edge radius, etc.).

## 5

## Approximations to robust optimization formulations

Since we cannot compute  $c_l$  and  $c_d$  for all  $M$  in the Mach range  $[M_{\min}, M_{\max}]$ , computationally tractable approximations of (9) and (11) must be used. The simplest approximation scheme is to replace  $[M_{\min}, M_{\max}]$  by a finite subset of  $[M_{\min}, M_{\max}]$ , say  $\{M_1, M_2, \dots, M_r\}$ . Then (9) is reduced to the multipoint optimization problem (1) (Drela 1998), but the weights  $w_i$  depend on the probability density function  $p(M)$  and a chosen integration scheme. Similarly, the minimax formulation (11) can be discretized as follows:

$$\min_{D, \alpha_1, \dots, \alpha_r} \max_{1 \leq i \leq r} \rho_i c_d(D, \alpha_i, M_i), \quad (12)$$

subject to the constraints given in (2). Here  $\rho_i > 0$  is determined by  $\rho(M)$  and  $M_i$ .

It seems that (1) and (12) are completely different. However, under the strict complementarity condition (i.e., Lagrange multipliers are nonzero for all active constraints), (1) is mathematically equivalent to (12). Here we give a proof of the mathematical equivalence between (1) and (12).

Let  $(\hat{D}, \hat{\alpha}_1, \dots, \hat{\alpha}_r)$  be a stationary solution of (1). For simplicity, assume that  $\hat{D}$  is in the interior of  $\mathcal{F}$  and the equality holds for lift constraints. Then, under the strict complementarity condition, the following first-order optimality condition (or the KKT-condition) for (1) holds:

$$\begin{aligned} \sum_{i=1}^r w_i \frac{\partial c_d}{\partial D}(\hat{D}, \hat{\alpha}_i, M_i) - \sum_{i=1}^r \lambda_i \frac{\partial c_l}{\partial D}(\hat{D}, \hat{\alpha}_i, M_i) &= 0, \\ w_i \frac{\partial c_d}{\partial \alpha_i}(\hat{D}, \hat{\alpha}_i, M_i) &= \lambda_i \frac{\partial c_l}{\partial \alpha_i}(\hat{D}, \hat{\alpha}_i, M_i) \quad \text{for } 1 \leq i \leq r, \end{aligned}$$

$$\lambda_i > 0, \quad c_l(\hat{D}, \hat{\alpha}_i, M_i) = c_l^* \quad \text{for } 1 \leq i \leq r. \quad (13)$$

Define

$$\hat{\gamma} := \sum_{i=1}^r w_i c_d(\hat{D}, \hat{\alpha}_i, M_i),$$

$$\rho_i := \frac{\hat{\gamma}}{c_d(\hat{D}, \hat{\alpha}_i, M_i)} > 0 \quad \text{for } 1 \leq i \leq r,$$

$$\theta_i := \frac{w_i}{\rho_i} \quad \text{for } 1 \leq i \leq r.$$

Then (13) implies the following conditions:

$$\sum_{i=1}^r \theta_i \rho_i \frac{\partial c_d}{\partial D}(\hat{D}, \hat{\alpha}_i, M_i) - \sum_{i=1}^r \lambda_i \frac{\partial c_l}{\partial D}(\hat{D}, \hat{\alpha}_i, M_i) = 0,$$

$$\theta_i \rho_i \frac{\partial c_d}{\partial \alpha_i}(\hat{D}, \hat{\alpha}_i, M_i) = \lambda_i \frac{\partial c_l}{\partial \alpha_i}(\hat{D}, \hat{\alpha}_i, M_i) \quad \text{for } 1 \leq i \leq r,$$

$$\lambda_i > 0, \quad c_l(\hat{D}, \hat{\alpha}_i, M_i) = c_l^* \quad \text{for } 1 \leq i \leq r,$$

$$\rho_i c_d(\hat{D}, \hat{\alpha}_i, M_i) = \hat{\gamma} \quad \text{for } 1 \leq i \leq r,$$

$$\theta_i > 0 \quad \text{for } 1 \leq i \leq r, \quad \text{and} \quad \sum_{i=1}^r \theta_i = 1. \quad (14)$$

By (14),  $(\hat{\gamma}, \hat{D}, \hat{\alpha}_1, \dots, \hat{\alpha}_r)$  satisfies the KKT-condition of the following optimization problem:

$$\min_{D, \alpha_i, \gamma} \quad \gamma \quad \text{subject to} \quad c_l(D, \alpha_i, M_i) \geq c_l^*$$

and

$$c_d(D, \alpha_i, M_i) \leq \frac{\gamma}{\rho_i} \quad \text{for } 1 \leq i \leq r. \quad (15)$$

Since (15) is mathematically equivalent to (12),  $(\hat{D}, \hat{\alpha}_1, \dots, \hat{\alpha}_r)$  is also a stationary point of (12).

On the other hand, assume that  $(\hat{D}, \hat{\alpha}_1, \dots, \hat{\alpha}_r)$  is a stationary point of (12), the equality holds in (2), and  $\rho_i c_d(\hat{D}, \hat{\alpha}_i, M_i) = \hat{\gamma}$  for  $1 \leq i \leq r$ , where

$$\hat{\gamma} := \max_{1 \leq i \leq r} \rho_i c_d(\hat{D}, \hat{\alpha}_i, M_i).$$

Then  $(\hat{\gamma}, \hat{D}, \hat{\alpha}_1, \dots, \hat{\alpha}_r)$  is a stationary solution of (15). If we further assume that the strict complementarity condition holds, then the KKT-condition (14) for (15) holds. It follows from (14) that (13) holds for  $w_i = \rho_i \theta_i$ . Therefore,  $(\hat{D}, \hat{\alpha}_1, \dots, \hat{\alpha}_r)$  is also a stationary point of (1).

*Remark 1* Note that the strict complementarity condition holds naturally for (1). In fact, if the lift is greater than its target value, then one can reduce the angle of attack so that the lift with respect to the new angle of attack is equal to the target value, but the drag with respect to the new angle of attack is reduced. This is not possible at a stationary point. Therefore, all lift constraints at a stationary point become equality constraints. Also, the derivative of the drag with respect to the angle of attack is positive for the range of the angle of attack considered herein, which implies that the Lagrange multipliers  $\lambda_i$  are positive (cf. (13) or (14)). Thus, the strict complementarity condition for (1) always holds. An implication is that it is possible to recover an optimal solution of (1) by solving (12) with appropriate choices of  $\rho_i$ 's.

However, it is a little more difficult to claim that a stationary point of (12) is also a stationary point of (1). In the proof given above, we used two additional assumptions: (a) all  $\rho_i c_d(\hat{D}, \hat{\alpha}_i, M_i)$  have the same value  $\hat{\gamma}$ , and (b) the strict complementarity condition holds for the drag constraints in (15). The first assumption is not realistic in the sense that we do not know how to choose  $\rho_i$ 's to make  $\rho_i c_d(\hat{D}, \hat{\alpha}_i, M_i)$  have the same value for an optimal solution. Adaptive adjustments of  $\rho_i$ 's are used in the

profile optimization method to ensure all  $\rho_i c_d(D, \alpha_i, M_i)$  ( $1 \leq i \leq r$ ) have the same value during the optimization iterations. The assumption (b) is a natural one, since the strict complementarity condition for the drag constraints means that every drag constraint is necessary for getting the optimal solution.

The results in (Drela 1998) indicate that the multi-point airfoil optimization formulation (1) tends to produce airfoils that have possibly severe degradation in the off-design performance, which Drela referred to as point-optimization behavior. The above equivalence analysis also implies that the same conclusion holds for the min-max formulation (12).

Huyse and Lewis (2001) concluded that the point-optimization behavior can be attributed to the discretization error between (9) and (1). The multipoint formulation (1) approximates the integral as a discrete sum: low drag is obtained at the selected Mach numbers  $M_1, \dots, M_r$ , but there is no control over, nor requirements for, the drag at the other Mach numbers. Due to the highly nonlinear nature of the PDEs for flow simulation, the optimizer is able to mold the objective function  $c_d$  to its own advantage: distinct drag troughs appear at each of the Mach numbers  $M_1, \dots, M_r$  (see also Drela 1998). To prevent the optimization algorithm from exploiting the approximation error, Huyse and Lewis (2001) suggest changing the integration points  $M_1, \dots, M_r$  during each optimization step. This effectively ensures that low drag is obtained not for just  $r$  fixed Mach numbers, but for any combination of  $r$  Mach numbers  $M_1, \dots, M_r$ . In their study, they added a random perturbation to the integration points in each optimization step but stated that “any adaptive scheme that varies the location of the integration points  $M_k$  for each optimization step will do.”

In the following two sections, we propose an alternate method called profile optimization and compare it with other airfoil optimization methods. We will show that with only far fewer design points than the number of free-design variables, the profile optimization method can adaptively adjust  $\rho_i$ 's in (12) to achieve a consistent drag reduction over the given Mach range, so there will be no degradation in the off-design performance of an optimal airfoil. We will also point out advantages and disadvantages of the profile optimization method when compared with other airfoil optimization methods.

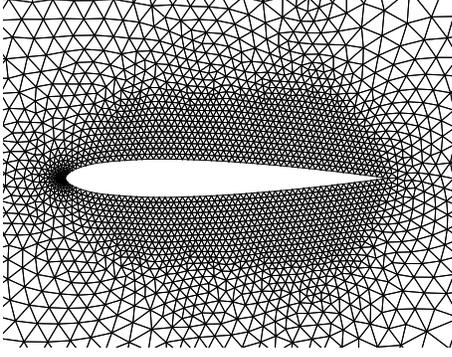
## 6

### Profile optimization method

In this section, we first give a motivation for a new shape optimization strategy based on the robust optimization model (12) and then present the profile optimization method.

Let  $\hat{D}$  be a fixed feasible design vector. We can plot the drag with respect to the Mach numbers over the interval  $[M_{\min}, M_{\max}]$  while keeping the lift at a constant

value by adjusting the angles of attack for different Mach numbers. Such a plot will be called a profile for  $\hat{D}$  (see Fig. 3(a) for example). Ideally, we want to obtain a design vector  $\hat{D}$  with a desirable profile (for example, with the drag rise occurring at a very high Mach number).



**Fig. 1** The initial grid used for solving the Euler equation by FUN2D

Suppose that a “perfect” design vector  $\bar{D}$  could be obtained by some magic process, which might be either theoretical or experimental. Let us look at the unknown process from a reverse engineering point of view. That is, even though we have no idea about how the airfoil was designed, we want to construct an optimization scheme that produces an airfoil with a similar or better profile.

Assume that the profile curve is generated by computing the drag coefficients at Mach numbers  $M_1, M_2, \dots, M_r$ . Let  $1/\rho_i$  be the value of the drag coefficient of the “perfect” airfoil at Mach number  $M_i$  with angle of attack  $\bar{\alpha}_i$ , which is chosen so that the lift coefficient is equal to  $c_l^*$ . Then the global optimal value of (12) is no more than 1. If  $(\hat{D}, \hat{\alpha}_1, \dots, \hat{\alpha}_r)$  is a global optimal solution of (12), then

$$c_d(\hat{D}, \hat{\alpha}_i, M_i) \leq \frac{1}{\rho_i} = c_d(\bar{D}, \bar{\alpha}_i, M_i) \quad \text{for } 1 \leq i \leq r.$$

Therefore, with the given analysis (i.e., using the profile of the drag coefficient at  $M_1, \dots, M_r$ ),  $\hat{D}$  is no worse than the “perfect” design vector  $\bar{D}$ .

This shows that for any given profile analysis method, it is possible to recover or improve a desirable airfoil by solving (12), if the weights are appropriately chosen. In this sense, (12) provides a very flexible tool for airfoil optimization. By experimenting with various choices of the weights  $\rho_i$ , a robust airfoil may be obtained (if such an airfoil exists). However, instead of guessing the appropriate weights in (12), we can adaptively adjust the weights during the optimization iterations. This leads to the profile optimization method.

Before presenting the profile optimization method, we give a linear programming formulation of a trust-region method for solving (12). Using linearization of the nonlinear functions in the mathematically equivalent formulation (15) at the current iteration point  $(D^k, \alpha_{1,k}, \dots, \alpha_{r,k})$ , we get the following trust-region subproblem for (15):

$$\min_{\Delta D, \Delta \alpha_i, \gamma} \quad \gamma \quad \text{subject to} \quad (16)$$

$$-\sigma_i \delta \leq \Delta D_i \leq \sigma_i \delta \quad \text{for } 1 \leq i \leq m,$$

$$-\alpha_{i,k} \leq \Delta \alpha_i \leq \alpha_{\max} - \alpha_{i,k} \quad \text{for } 1 \leq i \leq r,$$

$$c_l(D^k, \alpha_{i,k}, M_i) + \left\langle \frac{\partial c_l}{\partial D}(D^k, \alpha_{i,k}, M_i), \Delta D \right\rangle +$$

$$\frac{\partial c_l}{\partial \alpha}(D^k, \alpha_{i,k}, M_i) \Delta \alpha_i \geq c_l^* \quad \text{for } 1 \leq i \leq r,$$

$$c_d(D^k, \alpha_{i,k}, M_i) + \left\langle \frac{\partial c_d}{\partial D}(D^k, \alpha_{i,k}, M_i), \Delta D \right\rangle +$$

$$\frac{\partial c_d}{\partial \alpha}(D^k, \alpha_{i,k}, M_i) \Delta \alpha_i \leq \frac{\gamma}{\rho_i} \quad \text{for } 1 \leq i \leq r,$$

where  $\Delta D$  and  $\Delta \alpha_i$  are the increments for  $D$  and  $\alpha_i$ , respectively,  $\sigma_i$  and  $\delta$  are positive constants that define the trust region for  $\Delta D$ , and  $\alpha_{\max}$  is the maximum angle of attack allowed.

**Algorithm 1 (Profile Optimization Method)** *Suppose that  $D^0$  is a given design vector and  $M_1, M_2, \dots, M_r$  are the design points. Let  $0 < \eta < 1$  be the predicted percentage reduction rate of the drag for the trust-region method and let  $0 < \epsilon < 1$  be a parameter for termination of the algorithm. Then construct a new design vector as follows.*

**(1.0) Initialize the angles of attack:** *Find  $\alpha_{1,0}, \alpha_{2,0}, \dots, \alpha_{r,0}$  such that  $c_l(D^0, \alpha_{i,0}, M_i) = c_l^*$  for  $1 \leq i \leq r$ ; and let  $k = 0$ .*

**(1.1) Adjust weights:** *Let*

$$\rho_i := \frac{1}{c_d(D^k, \alpha_{i,k}, M_i)} \quad \text{for } 1 \leq i \leq r.$$

**(1.2) Check early termination:** *If the zero vector is an optimal solution of (16), then output  $D^k$  as an optimal solution and terminate the algorithm.*

**(1.3) Find a trust region for the predicted percentage reduction of the drag:** *Find  $\delta > 0$  such that the optimal objective function value of (16) is  $(1 - \eta)$ .*

**(1.4) Solve the trust-region subproblem:** *Find the least norm solution  $(\Delta D^k, \Delta \alpha_{1,k}, \dots, \Delta \alpha_{r,k})$  of (16).*

**(1.5) Generate the new iterate:** Let  $\alpha_{i,k+1} = \alpha_{i,k} + \Delta\alpha_{i,k}$  for  $1 \leq i \leq r$ , and  $D^{k+1} = D^k + \Delta D^k$ .

**(1.6) Check heuristic termination conditions:** If

$$\max_{1 \leq i \leq r} \rho_i c_d(D^{k+1}, \alpha_{i,k+1}, M_i) > 1 \quad \text{and} \quad (17)$$

$$\epsilon_k < \epsilon \sum_{i=1}^r c_d(D^k, \alpha_{i,k}, M_i), \quad (18)$$

where  $\epsilon_k$  is the accumulative reduction of the drag at the design points defined by

$$\epsilon_k := \sum_{i=1}^r \left( c_d(D^k, \alpha_{i,k}, M_i) - c_d(D^{k+1}, \alpha_{i,k+1}, M_i) \right),$$

then output  $D^k$  as an optimal solution and terminate the algorithm.

**(1.7) Start a new iteration:** Update  $k := k + 1$  and go back to (1.1).

*Remark 2* The adaptive adjustment of weights makes it possible to achieve a consistent drag reduction over the Mach range  $[M_{\min}, M_{\max}]$ . Note that moving along the descent direction  $\Delta D^k$  gives a simultaneous drag reduction at all design points  $M_1, \dots, M_r$  (at least if a small step is used). If the selected design points are a ‘‘fair representation’’ of all Mach numbers in  $[M_{\min}, M_{\max}]$ , then moving along  $\Delta D^k$  will not induce any hidden rise of the drag at some unselected Mach number. This leads to a robust optimization method that achieves a consistent drag reduction over the given Mach range from iteration to iteration (see the profiles of the drag coefficient for iterates generated by the profile optimization method in Fig. 4(a)).

In contrast, without adaptive weight adjustments, it is necessary to use more design points than the number of free-design variables with the multipoint optimization method in order to avoid point-optimization at selected design points (cf. Sections 3 and 5).

Besides the adaptive adjustment of weights, there are three features of the profile optimization method that are not standard for a trust-region method.

(a) First, the size of the trust region is modified to achieve a given predicted percentage reduction rate  $\eta$  of the drag. This approach is employed to avoid the following two problems, which are incurred in practical applications of a standard trust-region method: (i) an appropriate size of the trust region is unknown without prior experiences, and (ii) if each iteration takes a long time to complete, then rejecting a new iterate might be hard to accept by a designer.

For airfoil shape optimization, if the percentage reduction of  $c_d$  is too small, then the reduction could be a consequence of numerical errors and is less reliable. If the predicted percentage reduction of  $c_d$  is too large, then the size of the trust region is also relatively large and the

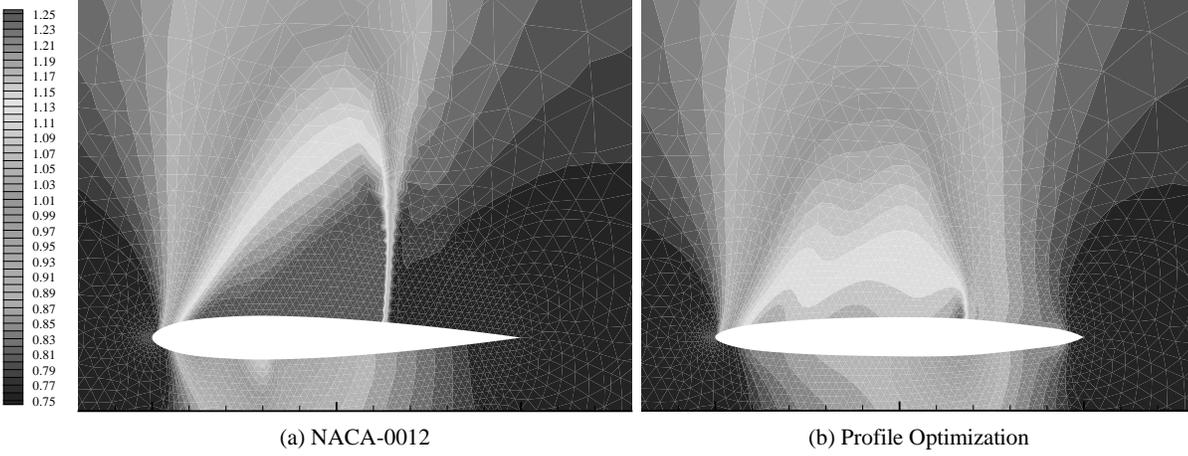
predicted reduction can not be trusted. By choosing an appropriate percentage reduction rate for each iteration, these two problems might be avoided.

Note that the choice of the predicted percentage reduction rate depends on a user’s knowledge of the accuracy of the simulation analysis tool involved, the time allowed to generate a new design, and the expected overall performance improvement. For example, if the relative numerical error for computing  $c_d$  is 0.5%, each iteration takes 3 hours, a designer has 1 day to generate a new design, and the expected improvement is 8%, then the designer could set  $\eta = 1\%$  so that each iteration will produce some meaningful improvement of the design and the final design might have about 8% improvement over the original design after 1 day (or 8 iterations).

(b) The second non-standard feature is that we use the least norm solution of the linear programming problem (16), which can be obtained by solving a quadratic perturbation of (16) (Mangasarian 1984). Our implicit assumption here is that the original airfoil is reasonable. Therefore, we do not want a new airfoil to deviate too much from the original one if unnecessary. By using the least norm solution of (16), we intend to select a new airfoil closest to the original one while achieving a predetermined amount of improvement in performance.

(c) The third non-standard feature of the profile optimization method is the termination criterion, which is based on design heuristics. The drag of the new design updated by the solution to the linearized problem (16) is reevaluated by using the CFD code. If a good descent direction for drag reduction at all design points does not produce actual drag reduction at every design point (i.e., (17) holds) and the overall percentage reduction is insignificant (i.e., (18) holds), then there is no need to continue the optimization process.

Finally, we point out that the requirement of improvement at every design point during each optimization iteration makes the profile optimization method more likely to get trapped at a local solution than the multipoint optimization method, which only requires an improvement of the weighted average of the drag at all design points. There are two possible remedies for this weakness of the profile optimization method: (a) use multiple starting points to search for a truly optimal solution (which increases its computational costs) or (b) when getting trapped, run a few iterations of a more aggressive optimization method such as the multipoint optimization method to get a new design vector and then restart the profile optimization from the new design vector.



**Fig. 2** The Mach waves (for the Mach number 0.796) around the NACA-0012 and an optimal airfoil generated by the profile optimization method

## 7

### Numerical simulation results

For the current implementation of the profile optimization method, search methods based on linear interpolation are used to find initial values of the angles of attack for  $r$  Mach numbers and the size  $\eta$  for the trust region in each iteration. The function values and gradients of  $c_l$  and  $c_d$  are computed in parallel for  $r$  Mach numbers. See Li *et al.* (2001) for implementation details.

We use the NACA-0012 airfoil as the initial point for all optimization methods discussed in this section. This choice allows us to compare new results with previously published results; however, this choice may violate the assumption that the original airfoil is reasonable, which is the basis for one optimization strategy behind the profile optimization method: a minimal shape modification should be made to achieve the required drag reduction at the design points.

A periodic spline representation of the NACA-0012 with 23 control points is used to get an initial parametric model of the airfoil:

$$x = \sum_{i=0}^{22} a_i p_i(t), \quad y = \sum_{i=0}^{22} b_i q_i(t), \quad \text{for } 0 \leq t \leq 1,$$

where  $p_i$  and  $q_i$  are spline basis functions. The shape of the airfoil can be modified by changing the values of the  $b_i$ 's. The tip and the tail of the airfoil are fixed during the optimization (i.e.,  $b_0$ ,  $b_{11}$ , and  $b_{22}$  are fixed) so that the chord length remains constant and the trailing edge comes to a sharp point. As a consequence, we have 20 free-design variables ( $b_1, \dots, b_{10}, b_{12}, \dots, b_{21}$ ) in the airfoil shape optimization.

The CFD code FUN2D is used to compute the function values and gradients of the lift and drag (Nielsen 2002). Some technical details of the flow solver (for function evaluations) and the adjoint equation solver (for gra-

dient evaluations) in FUN2D can be found in (Anderson and Bonhaus 1994) and (Nielsen and Anderson 1998), respectively.

We use Euler flow to demonstrate the usefulness of the profile optimization method as a robust airfoil optimization tool. The main reason to choose Euler flow simulation analysis is that we can complete our preliminary evaluation of the profile optimization method in a reasonable amount of time.

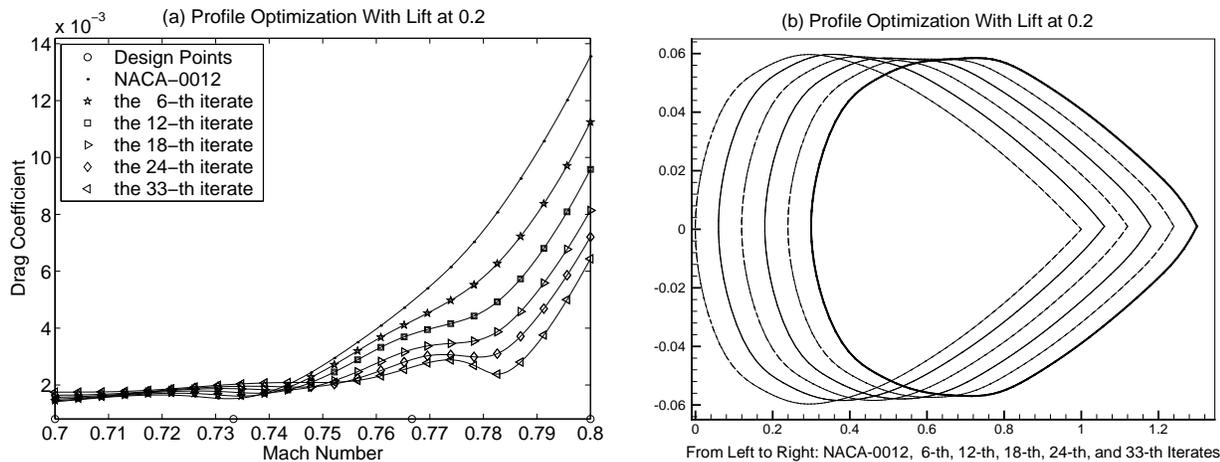
Elliott and Peraire (1997, Section 1) stated that almost any drag minimization exercise based on the Euler equations and applied to modern supercritical wings in cruise condition is doomed to failure. However, they also concluded that Euler-based optimization can be useful as a first step in the design process (Elliott and Peraire 1998, Abstract).

The initial grid for FUN2D used in this paper has 124 points around the airfoil and 32 points at the far field (at a distance of 50 chord lengths). The grid has 3060 nodes, 9025 faces, and 6120 elements. Fig. 1 shows the initial grid around the NACA-0012. See Huyse (2001) for a comparison of the numerical approximations of the lift and drag using this grid and other grids.

We compute the least norm solution of (16) by the quadratic programming solver QLD, which was first developed by Powell (1983) and then modified by Schittkowski (1986).

Our numerical simulations are designed to examine the impact of the number of design points and the target lift value on the profile optimization method. Also, we shall compare the simulation results obtained by the profile optimization method with the multipoint optimization method and the expected value optimization method. Therefore, we consider the following cases for numerical simulation:

1. the profile optimization with 4 design points equally spaced in the Mach range  $[0.7, 0.8]$ ,  $c_l^* = 0.4$ , a fixed



**Fig. 3** The two figures illustrate the behaviors of the iterates generated by the profile optimization method, with  $c_l^* = 0.2$  and 4 equally spaced design points. (a) Profiles of the drag coefficient versus the Mach number for the iterates and (b) the shapes of airfoils corresponding to the iterates. (Note: the airfoils are shifted to improve the visibility)

percentage reduction rate  $\eta = 5\%$ , and a termination parameter  $\epsilon = 0.3\%$ ;

2. the profile optimization with 3 design points equally spaced in the Mach range  $[0.7, 0.8]$ ,  $c_l^* = 0.4$ , a fixed percentage reduction rate  $\eta = 5\%$ , and a termination parameter  $\epsilon = 0.3\%$ ;
3. the profile optimization with 8 design points equally spaced in the Mach range  $[0.7, 0.8]$ ,  $c_l^* = 0.4$ , a fixed percentage reduction rate  $\eta = 5\%$ , and a termination parameter  $\epsilon = 0.3\%$ ;
4. the profile optimization with 4 design points equally spaced in the Mach range  $[0.7, 0.8]$ ,  $c_l^* = 0.2$ , a fixed percentage reduction rate  $\eta = 5\%$ , and a termination parameter  $\epsilon = 0.3\%$ ;
5. the multipoint airfoil optimization with 21 design points equally spaced in the Mach range  $[0.7, 0.8]$ ,  $w_1 = w_{21} = 1/40$ ,  $w_2 = \dots = w_{20} = 1/20$ , and  $c_l^* = 0.4$ ;
6. the multipoint airfoil optimization with 4 design points equally spaced in the Mach range  $[0.7, 0.8]$ ,  $w_1 = w_4 = 1/6$ ,  $w_2 = w_3 = 1/3$ , and  $c_l^* = 0.4$ ;
7. the expected value airfoil optimization with adaptive changes of  $w_i$  corresponding to 4 randomized integration points, and  $c_l^* = 0.4$ .

The first case is the benchmark case. The next two cases are included to examine the impact of the number of design points on the profile optimization. We use Case 4 to examine the impact of the target lift value on the profile optimization. Case 5 is included to validate the claim that a smooth airfoil with no off-design performance degradation could be found if the number of design points is greater than the number of free-design variables. The last two cases are for comparison of the profile optimization with other optimization methods. The profiles of the drag coefficient for a given  $c_l^*$  are plotted by using 24 equally spaced Mach numbers in  $[0.7, 0.8]$ .

## 7.1

### Impact of the number of design points

The profile optimization method terminates after 69, 67, and 57 iterations for 3, 4, and 8 design points, respectively. The number of design points has no significant impact on the optimal airfoils generated by the profile optimization method. All optimal airfoils have similar shapes (cf. Table 1) and only a marginal drag reduction is achieved by adding more iterations (cf. Table 2). By contrast, the number of design points makes a significant difference in the multipoint optimization results (cf. Fig. 5(a) and (Drela 1998; Huyse 2001)).

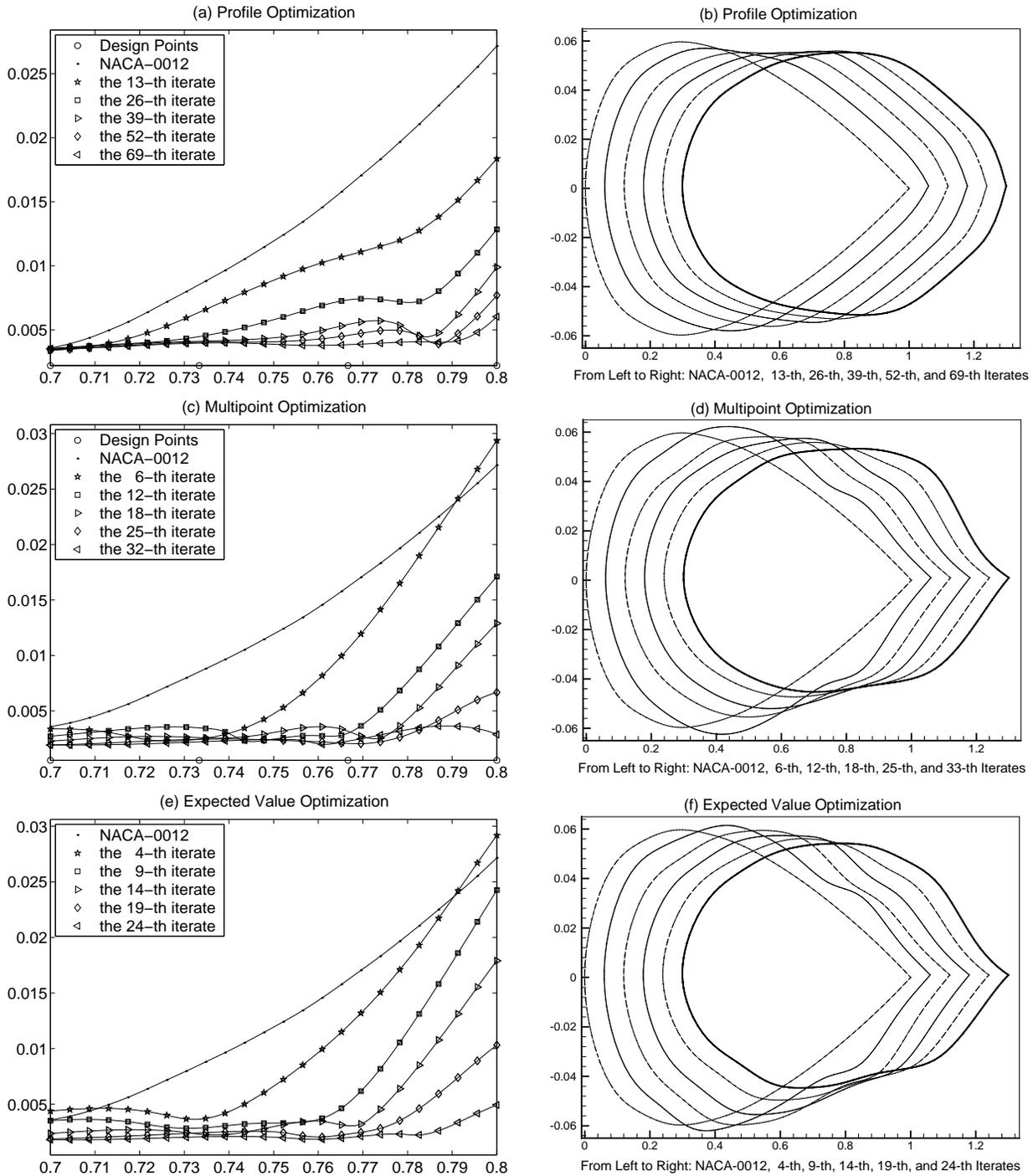
Each number in Table 1 denotes the relative difference of two design vectors:  $\|\mathbf{D}_r - \mathbf{D}_c\|/\|\mathbf{D}_r\|$ , where  $\mathbf{D}_r$  (or  $\mathbf{D}_c$ ) is either the NACA-0012 or the optimal design vector obtained by the profile optimization method with the number of design points listed in the corresponding row (or column).

**Table 1** The relative differences (in percent) of the design vectors

|           | NACA-0012 | 3 Pts | 4 Pts | 8 Pts |
|-----------|-----------|-------|-------|-------|
| NACA-0012 | –         | 24.5  | 25.7  | 23.8  |
| 3 Pts     | 24.7      | –     | 1.7   | 1.2   |
| 4 Pts     | 25.9      | 1.8   | –     | 2.6   |
| 8 Pts     | 23.9      | 1.2   | 2.6   | –     |

A typical optimal airfoil is given in Fig. 2(b). This Mach wave plot illustrates how one strong shock wave for the NACA-0012 is reduced to several weaker shock waves for the optimal airfoil.

With four design points, the profile optimization method achieves consistent drag reduction over the Mach range

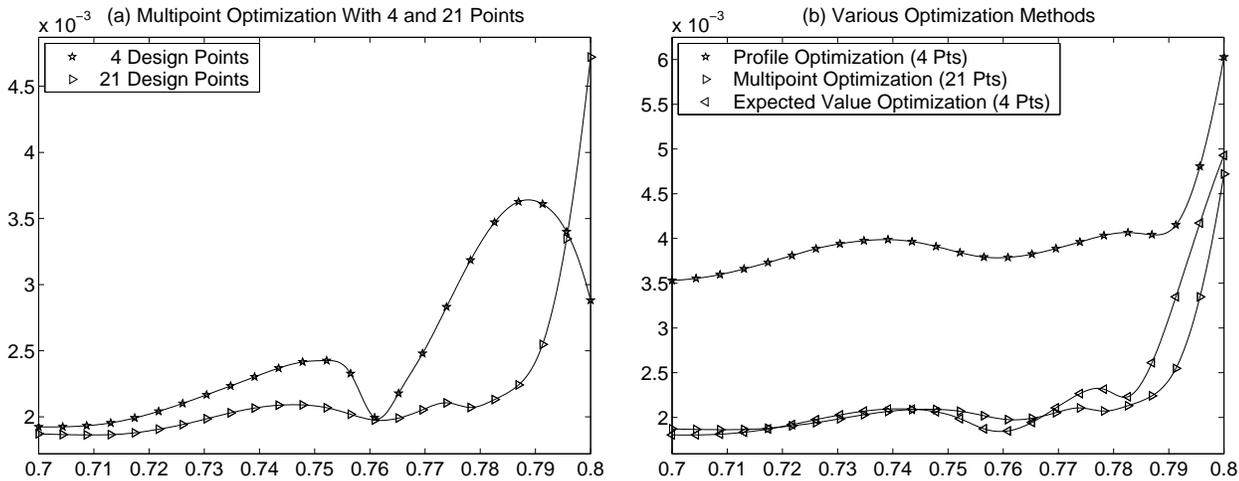


**Fig. 4** The behaviors of iterates generated by various optimization methods with  $c_t^* = 0.4$  and four design points

[0.7, 0.8] (cf. Fig. 3(a) and 4(a)). There is no random distortion of airfoil shapes during the optimization process (cf. Fig. 3(b) and 4(b)).

It is worth pointing out that the iterates generated by the profile optimization method are also independent of the number of design points. For example, the 56-th iterate generated by the profile optimization method for 4 design points differs from the 56-th iterate correspond-

ing to 3 (or 8) design points by 0.35% (or 1.2%). The profiles of the drag coefficient for optimal airfoils generated by the profile optimization with 3 and 4 design points, respectively, are almost identical. The profile of the drag coefficient for the optimal airfoil corresponding to 8 design points is very similar to the drag profile of the 57-th iterate generated by the profile optimization with 4 design points.



**Fig. 5** The profiles of the drag coefficient for various optimal airfoils

## 7.2

### Impact of the target lift value

The target lift value  $c_l^*$  has a quite significant impact on the optimal airfoil generated by the profile optimization method.

During each profile optimization iteration, we use a descent direction for the drag at all the design points, which guarantees the reduction of the drag at every design point if a sufficiently small trust region size  $\delta$  is used. However, it does not guarantee the actual reduction of the drag at every design point if the stepsize is chosen to satisfy the required rate of reduction of the linear approximation of the drag at the current iterate (cf. (1.3) of Algorithm 1). With  $c_l^* = 0.2$ , the NACA-0012 is almost optimal for low Mach numbers, which results in some minor increase of the drag around  $M = 0.73$  for the iterates generated by the profile optimization method. However, the profile optimization method manages to keep the drag as low as possible for Mach numbers between 0.7 and 0.75, while reducing the drag significantly for Mach numbers between 0.75 and 0.8 (cf. Fig. 3(a)).

Note that the drag bucket (i.e., a drop of the drag before its dramatic rise) occurs near  $M = 0.73$  for the NACA-0012 and the drag bucket for the optimal airfoil occurs near  $M = 0.78$ . Such a delay of the drag rise is desirable. The only compromise made by the optimal airfoil to the NACA-0012 is a small increase of the drag around the original drag bucket for the NACA-0012, which is quite reasonable.

For the higher target lift  $c_l^* = 0.4$ , the optimal airfoil shape (shown in Fig. 4(b)) deviates more from the NACA-0012. Also, the 33rd iterate generated by the profile optimization method for  $c_l^* = 0.4$  differs from the 33rd iterate for  $c_l^* = 0.2$  by 10%.

Note that the shape variations of the airfoils corresponding to iterates generated by the profile optimization method follow a similar trend no matter what the

target lift is: the aft end thickens while the front section narrows. This pushes the shock location towards the aft region of the airfoil.

A similar airfoil with thin front and fat tail was obtained by Elliott and Peraire (1997, Fig. 34) in a totally different context, where they used two thickness design variables and one camber design variable for a lift-constrained drag optimization with 2-D separated viscous flow and an additional area constraint. The initial airfoil used by them is also the NACA-0012. Their explanation of the merit of an airfoil with thinner front and fatter tail over the NACA-0012 is that the thickness distribution has been redistributed such that the maximum is further aft (i.e., closer to the tail), like the early natural laminar flow airfoils, this delays the start of the adverse pressure gradient to aft of that maximum thickness point.

## 7.3

### Impact of the optimization strategy

From the analysis given in Sections 3 and 5, we know that if the design points and weights are fixed as in either (1) or (12), then a severe degradation in the off-design performance is likely when the number of design points (i.e., 4) is much smaller than the number of free-design variables (i.e., 20). Figure 5(a) clearly reveals the point-optimization features of the optimal airfoil generated by the multipoint optimization method with 4 design points. Figure 4(c) suggests that the point-optimization behavior becomes more pronounced as the iteration number increases. This is consistent with our mathematical argument and the explanation given in Section 2: the multipoint optimization method is likely to trade marginal performance improvements at the design points for significant performance degradation at off-design points. However, such off-design performance degradation does not occur if the number of design points is larger than the

number of free-design variables, as shown in the drag profile of the optimal airfoil generated by the multipoint optimization method with 21 design points (cf. Fig. 5(a)). However, a total of 21 Euler flow problems (for getting the lift and drag at 21 Mach numbers) and 42 related nonlinear adjoint equations (for getting the gradients of the lift and drag at 21 Mach numbers) have to be solved in each iteration of the multipoint optimization method, which might not be practical if it takes days and many processors to solve one flow/adjoint problem for high fidelity aerodynamic optimization problems such as 3-D wing optimization (Nielsen and Anderson 2001).

The expected value optimization method (Huysse and Lewis 2001) adds a random perturbation to the integration points from iteration to iteration and adjusts the weights accordingly so that a severe degradation in the off-design performance can be avoided. Fig. 4(e) shows that this technique allows the optimizer to reduce the drag while avoiding point-optimization.

The profile optimization avoids the problem of point-optimization at the design points (cf. Fig. 3(a) and 4(a)) by using a smart descent direction to achieve consistent drag reduction over the whole Mach range.

Table 2 gives an overall sense of how each optimization method reduces the drag coefficient with  $c_l^* = 0.4$ . Let  $M_1, M_2, \dots, M_{24}$  be 24 equally spaced points in  $[0.7, 0.8]$ . For each  $M_i$ , let  $c_{d,i}$  (or  $\hat{c}_{d,i}$ ) be the drag coefficient of the NACA-0012 (or an optimal airfoil) with the lift coefficient fixed at 0.4. Then the relative reduction of the drag at each Mach number  $M_i$  is  $(c_{d,i} - \hat{c}_{d,i})/c_{d,i}$ . The numbers in ‘‘Max’’, ‘‘Min’’, and ‘‘Average’’ columns of Table 2 correspond to

$$\max_{1 \leq i \leq 24} \frac{c_{d,i} - \hat{c}_{d,i}}{c_{d,i}}, \quad \min_{1 \leq i \leq 24} \frac{c_{d,i} - \hat{c}_{d,i}}{c_{d,i}}, \quad \text{and}$$

$$\left( \sum_{i=1}^{24} c_{d,i} - \sum_{i=1}^{24} \hat{c}_{d,i} \right) / \left( \sum_{i=1}^{24} c_{d,i} \right), \quad \text{respectively.}$$

**Table 2** Relative drag reduction rates (in percent)

|                             | Max  | Min  | Average |
|-----------------------------|------|------|---------|
| Profile Opt. (3 Pts)        | 82.9 | 5.1  | 70.2    |
| Profile Opt. (4 Pts)        | 82.7 | 1.4  | 69.7    |
| Profile Opt. (8 Pts)        | 82.6 | 4.1  | 67.9    |
| Multipoint Opt. (4 Pts)     | 89.4 | 46.2 | 81.1    |
| Multipoint Opt. (21 Pts)    | 90.0 | 47.7 | 83.3    |
| Expected Value Opt. (4 Pts) | 89.4 | 49.6 | 82.6    |

Note also that the multipoint optimization method and the expected value optimization method resulted in an optimal airfoil with a drag curve that is consistently lower over the Mach range than the profile optimization method (cf. Fig. 5(b)). It seems that it should be possible for the profile optimization method to find a descent direction for a consistent drag reduction over the

whole Mach range  $[0.7, 0.8]$ , by moving toward the optimal solution given by either the multipoint optimization method or the expected value optimization method. However, due to nonlinearity of the drag coefficient (with respect to the design variables), the profile optimization method can not find such a descent direction by using the local derivative information only.

## 8 Conclusion

In this paper, we introduce a new robust optimization scheme called the profile optimization method and use airfoil optimization under uncertain flight conditions as a case study to evaluate this method. We used an Euler-based CFD code, which does not include viscous effects, to test the profile optimization method. Because of this lack of fidelity, the generated airfoils may be somewhat unrealistic.

For the multipoint optimization method, we give a mathematical argument that provides some insight on why it is necessary to use more design points than the number of free-design variables with the multipoint optimization method in order to avoid point-optimization at selected design points.

However, with only a few design points, the profile optimization method adaptively adjusts the weights in a minimax optimization formulation to find a drag reduction direction for all design conditions, which leads to a consistent drag reduction over a given range of Mach numbers from iteration to iteration.

Our numerical results demonstrate that the profile optimization method is not sensitive to the number of selected design points. With 20 free geometric design variables and as little as 4 design points, the optimal airfoil generated by the profile optimization method is smooth and has no degradation in the off-design performance.

The profile optimization method can be easily modified to solve other optimization problems under uncertainty by replacing  $c_d$  with another performance measurement function and  $M$  with other uncertain parameters.

The profile optimization method also has the potential of becoming a practical design tool for optimization under uncertainty. The use of a small number of sampled design points from the range of uncertain parameters makes the profile optimization method computationally affordable. The consistent performance improvements over the range of uncertain parameters from iteration to iteration allows a designer to stop the iterative process at any time with an improved new design.

However, by requiring drag reduction at every design point during each optimization iteration, the profile optimization method is more likely to get trapped in a local solution than the multipoint optimization method, which only requires an improvement of the weighted average of the drag at all design points. One might use multiple

starting points to search for an truly optimal solution or incorporate another more aggressive optimization strategy into the profile optimization method to get out of the trap.

For practical problems, the probability distribution function is likely to be very different from the uniform distribution used in our numerical simulation. The actual probability distribution function can be determined if appropriate historical flight data is available. The use of a different distribution function can be easily implemented in the multipoint optimization method and the expected value optimization method by using the weights corresponding to the distribution function (Huyse 2001). However, the weights in the profile optimization method are adaptively adjusted in each optimization iteration and are not related to the distribution function of the uncertainty parameter. Therefore, it is important to study variations of the profile optimization method that incorporate the probability distribution function of the uncertainty parameter(s) into the robust optimization model.

*Acknowledgements* This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the first two authors were in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681.

The authors are grateful to Perry A. Newman at the MDO Branch of NASA Langley Research Center and Manuel D. Salas at ICASE for their insightful comments on aerodynamics related to airfoil designs, and to Eric Nielsen at the Aerodynamic and Acoustic Methods Branch of NASA Langley Research Center for his expertise of FUN2D. We also extend our thanks to Michael Lewis for suggesting airfoil optimization as a case study of robust optimization methods.

## References

- Anderson, W.; Bonhaus, D. 1994: An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Comp. Fluids* **23**, 1–21
- Anderson, W.; Bonhaus, D. 1999: Airfoil optimization on unstructured grids for turbulent flows. *AIAA J.* **37**, 185–191
- Anderson, W.; Venkatakrisnan, V. 1997: Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *AIAA-97-0643*
- Ben-Tal, A.; Nemirowski, A. 1997: Robust truss topology design via semidefinite programming. *SIAM J. Optimiz.* **7**, 991–1016
- Drela, M. 1998: Pros and cons of airfoil optimization. In: Caughey, D.A.; Hafez, M.M. (eds.) *Proc. Frontiers of computational fluid dynamics 1998*. World Scientific
- Elliott, J.; Peraire, J. 1997: Aerodynamic optimization on unstructured meshes with viscous effects. *AIAA-97-1849*
- Elliott, J.; Peraire, J. 1998: Constrained, multipoint shape optimization for complex 3D configurations. *Aeron. J.* **102**, 365–376
- Fowlkes, W.Y.; Creveling, C.M. 1995: *Engineering methods for robust product design using Taguchi methods in technology and product development*. Reading, MA: Addison-Wesley
- Gumbert, G.; Hou, G.; Newman, P. 2001: Simultaneous aerodynamic analysis and design optimization (SAADO) for a 3-D flexible wing. *Proc. 39-th AIAA Aerospace Science Meeting and Exhibit* (held in Reno, Nevada), AIAA-2001-1107
- Huyse, L. 2001: Free-form airfoil shape optimization under uncertainty using maximum expected value and second-order second-moment strategies. *NASA/CR-2001-211020* or *ICASE Report No. 2001-18*
- Huyse, L.; Lewis, R. 2001: Aerodynamic shape optimization of two-dimensional airfoils under uncertain operating conditions. *NASA/CR-2001-210648* or *ICASE Report No. 2001-1*
- Li, W.; Huyse, L.; Padula, S. 2001: Robust airfoil optimization to achieve consistent drag reduction over a Mach range. *NASA/CR-2001-211042* or *ICASE Report No. 2001-22*
- Loncaric, J. 2001: ICASE, [www.icas.edu/CoralProject.html](http://www.icas.edu/CoralProject.html)
- Mangasarian, O. 1984: Normal solutions of linear programs. *Math. Prog. Study*, **22**, 206–216
- Nielsen, E. 2002: NASA Langley Research Center, [fun3d.larc.nasa.gov](http://fun3d.larc.nasa.gov)
- Nielsen, E.; Anderson, W. 1998: Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations. *AIAA-98-4809*
- Nielsen, E.; Anderson, W. 2001: Recent improvements in aerodynamic design optimization on unstructured meshes. *AIAA-2001-0596*
- Reuther, J., Jameson, A., Alonso, J., Rimlinger, M., and Saunders, D. 1999: Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers (Parts 1 and 2). *Journal of Aircraft*, **36**, 51–74.
- Powell, M. 1983: ZQPCVX, a FORTRAN subroutine for convex programming. *Report DAMTP/1983/NA17*, University of Cambridge, UK
- Schittkowski, K. 1986: QLD – A FORTRAN code for quadratic programming, user's Guide. *Mathematisches Institut, Universität Bayreuth, Germany*