OPTIONS FOR ROBUST AIRFOIL OPTIMIZATION UNDER UNCERTAINTY

Sharon L. Padula*
NASA Langley Research Center
Hampton, VA 23681

and

Dr. Wu Li†
Old Dominion University
Norfolk, VA 23529

Abstract

A robust optimization method is developed to overcome point-optimization at the sampled design points. This method combines the best features from several preliminary methods proposed by the authors and their colleagues. The robust airfoil shape optimization is a direct method for drag reduction over a given range of operating conditions and has three advantages: (1) it prevents severe degradation in the off-design performance by using a smart descent direction in each optimization iteration, (2) it uses a large number of spline control points as design variables yet the resulting airfoil shape does not need to be smoothed, and (3) it allows the user to make a trade-off between the level of optimization and the amount of computing time consumed. For illustration purposes, the robust optimization method is used to solve a lift-constrained drag minimization problem for a two-dimensional (2-D) airfoil in Euler flow with 20 geometric design variables.

Introduction

Multidisciplinary design optimization (MDO) is increasingly important to the aerospace community[1]. For example, Giesing and Barthelemy organized a pair of sessions at the 7th AIAA Multidisciplinary Analysis and Optimization Symposium addressing the uses of MDO in industry. The industry representatives provided many examples of successful MDO applications but also produced a list of new methods that are required by industry. This paper addresses one of those emerging areas: robust design or optimization under uncertainty. In particular, this research evaluates optimization methods for producing solutions that are insensitive to variability in input parameters.

The need for robust design methods appears in many contexts. During the preliminary design process, the exact value of input parameters is not known. It may be possible to make an educated guess or provide bounds for these unknown parameters, but they are not deterministic quantities. Faced with uncertain parameters, traditional optimization techniques tend to "over-optimize." Like the single-point optimization represented by the solid line in Fig. 1, these techniques produce solutions that perform well at the design point but have poor off-design characteristics. Industry favors designs that have room to grow[2]. Industry requires designs, such as the robust optimization represented by the dashed line in Fig. 1, that can be adapted to new missions or new business climates.

The present research seeks optimization methods that are robust in the sense that they produce solutions insensitive to changes in the input parameters. Furthermore, the methods must be able to find solutions by using a moderate number of high-fidelity disciplinary analyses. This second requirement acknowledges the fact that disciplinary analyses (e.g., computational fluid dynamics (CFD)) can be computationally expensive; an optimization method that requires thousands of function evaluations has limited usefulness in the current design environment.

The investigation of robust optimization is driven by several needs. The first need is for a conservative approach that achieves a robust design in the neighborhood of a baseline configuration. This approach is useful to the design team that has a good baseline design developed by traditional methods. The second need is for an exploratory approach that uncovers unsuspected trade-offs in the design space.

* Senior research scientist, Multidisciplinary Optimization Branch, MS 159, Associate Fellow, AIAA.

† Professor, Department of Mathematics and Statistics.

This approach can be used to discover new design ideas for further study.

This paper will use an airfoil optimization problem to illustrate and compare several options for optimization with uncertainty. These options were implemented in the same computer code with the same nonlinear programming techniques and they were tested and compared using a single computer architecture and baseline design. These conditions facilitate fair comparisons in terms of convergence properties, computational efficiency and final solutions.

Airfoil Optimization Problem

The concept of robust optimization is demonstrated using a 2-D airfoil shape optimization problem. Hicks and Vanderplaats studied a simplified version of this airfoil problem in 1977.[3] They observed that minimizing drag at a single design point had unintended consequences at nearby off-design points. For example, consider Fig. 2a, reproduced from Ref. 3. Hicks and Vanderplaats demonstrate that a direct optimization approach that minimizes drag at one Mach number (e.g., $M = 0.75$) actually increases drag at nearby Mach numbers (e.g., $M = 0.70$). The authors also present evidence showing that an inverse optimization approach that adjusts the airfoil shape to match some ideal pressure distribution will display similar point-optimization behavior. To make this point, the authors provide numerous pressure graphs such as Fig. 2b. A comparison of the solid and dashed curves of pressure (C_p) as a function of chord length (*x/c)* indicates that the optimization procedure produces an improved airfoil for $M = 0.75$. In Fig. 2b, the solid curve has a steep slope and a large change in C_p near $x/c = 0.7$. The dashed curve has a more gradual slope and a smaller change in C_p. Ref. 3 also contains pressure graphs for this optimized airfoil at off-design Mach numbers. These additional graphs show undesirable characteristics such as multiple shocks. The conclusion is that changes in airfoil shape which are advantageous at one Mach number cause poor performance at other Mach numbers. This result is especially apparent for supercritical airfoil designs because the relationship between wave drag and free flow velocity is quite nonlinear for high subsonic design Mach numbers.

Drela discusses this point-optimization behavior.[4] He suggests a multi-point optimization approach for lift-constrained drag minimization such as:

$$\min_{\alpha_i, d \in D} \sum_{i=1}^{n} w_i C_d(d, \alpha_i, M_i) \qquad (1)$$

subject to

$$C_l(d, \alpha_i, M_i) \geq C_l^* \quad \text{for } 1 \leq i \leq n, \qquad (2)$$

where C_l* is the required lift, w_i are arbitrary weights, d is a set of m airfoil geometric design variables, C_l is the lift coefficient and C_d is the drag coefficient. Both C_l and C_d are functions of angle-of-attack α and free stream Mach number M. Drela, Hicks and Vanderplaats, Reuther et a*l*. and other authors have proposed variations of multi-point optimization.[3-5] Most of these studies conclude that the utility of the solution to Eq. (1) depends upon the choice of the weights, w_i, and the number and spacing of the multiple design points, M_i. Unfortunately, there is no guarantee that sensible selection of weights or design points will net desirable airfoil shapes.[3,6] Moreover, several authors suggest that if the number of design points n is much smaller than the number of design variables (i.e., $n<<m$), the resulting airfoil will be misshapen.[3,7,8]

In ICASE reports by Huyse, and by Li et al., the shortcomings of Eq. (1) are exposed and new robust optimization algorithms are proposed.[7,8] The goal of robust optimization is to find the airfoil shape that minimizes the expected value of the drag coefficient over a range of free stream Mach numbers, $M_i \in [M_{\min}, M_{\max}]$. Fig. 3 illustrates the difference between multi-point and robust optimization.[9] The multi-point optimization reduces C_d at several specific M_i while a robust optimization reduces C_d over a range of Mach numbers. The robust optimization problem is stated:

$$\min_{d \in D} \int_M C_d(d, \alpha, M) f_M(M) dM \qquad (3)$$

subject to

$$C_l(d, \alpha, M) = C_l^* \quad \text{for all } M, \qquad (4)$$

where α is the angle of attack that satisfies Eq. (4) for each M, and f_M is the probability density function. For the present research, f_M is a uniform distribution. See Ref. 7 for a full discussion of the importance of f_M and for examples using nonuniform distributions.

The lift-constrained 2-D airfoil optimization problem stated in Eqs. (3) and (4) is used to test robust optimization approaches. A typical airfoil is shown in Fig. 4. Twenty bounded geometric design variables, d_j, are the spline coefficients used to create a wide variety of smooth 2-D airfoil shapes. The leading edge and trailing edge points are fixed, but the top and bottom surfaces can change as long as the lift constraint given in Eq. (4) is satisfied.

Two attractive options for solving the 2-D airfoil problem are presented in recent papers by Huyse[7] and by Li et al.[8] The present research uses the Euler inviscid CFD analysis code[10] and the 2-D unstructured grid used in Refs. 7 and 8. The contribution of the present study is to compare the available options in a systematic way.

General Solution Strategy

In this section, a general solution strategy is proposed and the methods of Refs. 7 and 8 are associated with some tuning parameters in the solution strategy. In this way, a continuum of possible methods is described and a plan for testing and characterizing those methods is proposed.

The general strategy consists of the following steps:

1. Select n Mach numbers.
2. Find the smallest feasible angle of attack for each Mach number.
3. Calculate objective, constraints, and their gradients.
4. Formulate a linear subproblem.
5. Adjust the trust region step size to achieve predetermined improvement.
6. Solve the linear subproblem and update values of design variables.
7. Decide whether to terminate or iterate from step 1.

The cases compared in this paper differ from each other in steps 1, 4, 5 and 6.

Step 1 in the general strategy can vary in the number of Mach points selected and in the method of selection. Reference 7 favors a random selection of Mach points in order to avoid the point-optimization phenomena pictured in Figs. 1-3. Reference 8 favors a fixed set of Mach points because the convergence properties associated with randomly selected points are difficult to anticipate. The Mach points M_i are selected from a range [0.7, 0.8] for all results presented in this paper. The nominal set of points is {0.70, 0.733, 0.766, 0.80}. Random points are generated using a random perturbation not to exceed 0.04. The random set of points is tested so that no two points are too close together. The range $M_i \in [0.7, 0.8]$ is selected because it contains typical cruise Mach numbers for commercial transports and because it exaggerates the difference between the methods to be tested.

Step 2 involves a line search for the smallest value of α which satisfies the lift constraint at each of the selected Mach points. This task is simplified because the relationship between C_l and α is quite linear. On the first iteration, a line search is always necessary. After that, the values of α are available from the previous iteration and are used to initialize the search. If the Mach points are fixed, a line search is often unnecessary. If the Mach points are randomly selected, then a few extra CFD analyses will be needed for the line search. Ref. 8 contains further details about the line search.

Step 3 is the calculation of the expected value objective function and involves an integral over Mach number. This integral can be approximated by use of the trapezoid rule, hermite polynomials or extra Mach points as illustrated in Fig. 5. In general, the trapezoid rule is used to estimate the objective function value, and the other estimates of the integral are used in the postprocessing assessments of the methods. Obviously, using extra points gives a more accurate estimate, however the computational expense can be prohibitive. Another way to improve the estimate is to use randomly selected points. If the design points are reselected at the start of each iteration, then the optimization algorithm will not be able to exploit the integration error to improve the performance at the design points at the expense of performance at the off-design points (recall Fig. 3).

Step 4 involves formulating the linear subproblem. Of all the possible solutions to the linear subproblem, we seek the one that minimizes the objective while making the smallest change in the design vector in a least norm sense. Also, auxiliary constraints are imposed to avoid potential off-design performance degradation due to inaccuracy in numerical estimates of the expected-value integral. The linear subproblem can be summarized as:

$$\min_{\Delta d, \Delta\alpha} \sum_{i=1}^{n} w_i C_{d,i}^{new} \qquad (5)$$

subject to

$$C_{d,i}^{new} = C_{d,i} + \left\langle \frac{\partial C_{d,i}}{\partial d}, \Delta d \right\rangle + \frac{\partial C_{d,i}}{\partial \alpha_i} \Delta\alpha_i \qquad (6)$$

$$C_{d,i}^{new} \leq (1 - \gamma_{\min}) C_{d,i} \qquad (7)$$

$$C_{l,i} + \left\langle \frac{\partial C_{l,i}}{\partial d}, \Delta d \right\rangle + \frac{\partial C_{l,i}}{\partial \alpha_i} \Delta\alpha_i = C_l^* \qquad (8)$$

for $1 \leq i \leq n$

and subject to

$$-\delta_j \leq \Delta d_j \leq \delta_j \qquad \text{for } 1 \leq j \leq m \qquad (9)$$

where the notation < a, b > indicates the dot product of vectors a and b, $C_{d,i}$ is the current value of the drag coefficient evaluated at the i^{th} design point, and $C_{d,i}^{new}$ is the linear prediction to the value of $C_{d,i}$ after the optimization step. Bounds δ_j on the change in a design variable Δd_j can be either proportional to the initial value or uniform.

This linear subproblem can be used for either multi-point optimization or robust optimization. Robust optimization is accomplished by including an extra constraint (i.e., Eq. (7)) and by choosing the weights w_i based on a numerical integration formula to approximate the integral in Eq. (3). Robust optimization seeks a smart descent direction which reduces each individual $C_{d,i}$ and the expected value.

Step 5, the determination of trust region step size, is crucial and requires the most explanation (see Ref. 8 for complete details). The size of each optimization step is controlled by two tuning parameters, $\gamma_{\min}$ and γ_{obj}. The first parameter occurs in Eq. (7) and specifies the desired decrease in C_d at each design point. The second parameter occurs in a similar equation,

$$obj^{new} = \left(1 - \gamma_{obj}\right) obj \tag{10}$$

where γ_{obj} is the desired decrease in the objective, obj is the current objective value and obj^{new} is the linear prediction of the new objective following the optimization step. Since the computational cost of solving the approximate optimization subproblem is inexpensive compared with the cost of CFD analysis, the subproblem can be solved with several different values of trust region size δ and then the final trust region size can be adjusted to satisfy Eq. (10). In this paper, γ_{obj} is always a small positive number and the optimization subproblem always attempts to find an obj^{new} that is smaller than obj. On the other hand, $\gamma_{\min}$ can be a large negative number meaning that drag at some M_i can increase as long as the expected value of drag decreases. Setting $\gamma_{\min}$ to a large negative number makes this subproblem resemble a traditional multi-point optimization since Eq. (7) will never be an active constraint. The details of the optimization subproblem and the adjustment of δ and value of $\gamma_{\min}$ are the major differences between the methods in Refs. 7 and 8.

Step 6 involves the solution of the approximate optimization subproblem and the update of the design variable values. Refs. 7 and 8 differ in this step because the authors use different linear subproblems and different optimization software, which makes it very difficult to compare the two methods based on the results in those reports. The current paper uses a quadratic programming solver to find the least norm solution of the linear program given in Eqs. (5)–(9).

Ideally, step 7 uses a termination criteria based on the comparison between the predicted decrease and the actual decrease delivered by step 6. For the purpose of comparing options, the authors feel it is better to compare after a fixed number of iterations for each method. Therefore, step 7 is not implemented.

The algorithm described in steps 1-7 is very similar to sequential linear programming, however, it has three key unconventional elements. These elements are employed to find a smart search direction that takes into account the fact that only a limited amount of information (i.e., the drag at a few design points) is given for solving the robust optimization problem. The unconventional elements are (1) a predetermined rate of linearized drag reduction to ensure a proper trust region size, (2) the auxiliary drag constraints to ensure simultaneous drag reduction at all the design points, and (3) the least norm solution of the linear programming subproblem for an efficient trade-off between the shape modification and the performance improvement. The combination of (1) and (3) yields smooth airfoils during optimization and prevents severe off-design performance degradation. The auxiliary drag constraints force the optimizer to find a local solution with a flatter drag profile over the Mach range.

Results

The general solution strategy facilitates testing of many robust airfoil optimization options. For this paper, a large number of cases were tested. Table 1 provides a summary of cases referred to in the present paper.

Case #	γ_{obj}, %	$\gamma_{\min}$, %	# of M_i	Spacing of M_i
1	3	-1000	4	fixed
2	3	1	4	fixed
3	3	2	4	fixed
4	3	3	4	fixed
5	3	-1000	4	random
6	3	1	4	random
7	3	2	4	random
8	3	3	4	random
9	4	-1000	4	fixed
10	4	2	4	fixed
11	4	4	4	fixed
12	3	-1000	21	fixed

Table 1. Summary of options tested.

Note that 4 design points are used to estimate the area under the curve, except for Case 12 which uses 21. Sometimes the design points are evenly spaced and sometimes they are randomly spaced. All plots of drag versus Mach number contain 24 points regardless of the number or the spacing of M_i used during the optimization process. Fig. 6a is an example of a drag profile plot with 24 points. These results have been postprocessed, which is necessary to compare solutions and to prove that the drag is reduced over the whole interval.

The results of these tests can be presented in many ways. For example, Fig. 6a compares the drag profiles of C_d versus M for Cases 4, 9, 10, and 11; Fig. 6b and Fig. 6c compare the airfoil shapes for these same cases. Each of these optimization cases ran for 30 iterations. Notice that Cases 9 and 10 seem to have converged to a similar solution while Cases 4 and 11 are quite different in shape and in drag profiles. As shown in Table 1, Cases 4 and 11 have $\gamma_{\text{min}} = \gamma_{\text{obj}}$. Thus, we postulate that in Cases 4 and 11, the need to reduce drag at each Mach number results in a different convergence history than in Cases 9 and 10, where the need to reduce expected value is emphasized.

Another way to look at these results is shown in Fig. 7 in which drag profiles are plotted every ten optimization steps for Case 4. Note that convergence is very smooth and the expected value of drag for the 50th step is obviously better than the expected value of drag for the 10th step.

Several conclusions can be drawn from Figs. 6 and 7. One conclusion is that the user has control of the robust optimization strategy. The user can be very conservative and set $\gamma_{\text{min}} = \gamma_{\text{obj}} = 0.03$ and get gradual improvement with each iteration by making small changes to a good baseline design. Alternatively, the user can be more aggressive and request a larger improvement with each iteration. This choice will cause the trust region step size to expand and will increase the possibility that the linear subproblem is not a good approximation to the true robust optimization problem. However, it will also increase the possibility of rapid convergence to a novel airfoil shape. Thus, the user can make a trade-off between the number of function evaluations and the level of optimization. The second conclusion is that all choices yield airfoil shapes that are smooth and airfoils that reduce drag over the whole range of Mach numbers.

A third way to study the results is shown in Fig. 8 where the expected value history is plotted for Cases 2 and 6. Fig. 8 presents a traditional convergence history of objective function versus iteration number. The estimates of expected value are calculated with a more accurate hermite interpolation for the integrand in Eq. (3); that is, the area under each curve is estimated using four values of C_{d} and four derivatives of C_{d} with respect to M_{i}. The plot indicates that each optimization strategy makes steady progress toward the goal of reducing drag over a range of Mach numbers. The progress made in Case 6 (Fig. 8b) is similar to that for Case 2 (Fig. 8a). As seen in Table 1, the two cases are similar except that Case 6 uses randomly chosen Mach numbers. Fig. 8c compares the random Mach numbers used in Case 6 with the fixed values used in Case 2. Since the set of Mach numbers changes from iteration to iteration, the estimates of expected value in Case 6 may be a little high in one iteration and a little low in the next. The estimates of expected value in Case 2 are not perfect either, but they are more consistent from iteration to iteration.

This study contained four cases of randomly selected Mach numbers and four similar cases with fixed Mach numbers (see Table 1). All cases successfully reduced the expected value of drag. For example, Fig. 9 compares the drag profiles of Case 6 with Case 2 after 30 iterations. As indicated in Fig. 9, the random and fixed cases differ very little. However, as was seen in Fig. 6, cases with $\gamma_{\text{min}} = \gamma_{\text{obj}}$ do differ somewhat from other cases where $\gamma_{min} < 0.0$.

All cases were set to run for 50 iterations, and all converged to a final expected value between 0.0003 and 0.0004. Most cases reached a value less than 0.0004 by iteration 30. A few of the cases show an odd convergence history between iteration 30 and 50. We believe that this is attributable to the well-known "over-shooting" behavior of sequential linear programming. Near the solution, the gradient of the objective approaches zero and the effects of tiny errors are magnified. Consequently, the optimization steps tend to zigzag around the true solution rather than hitting it exactly. To avoid the effects of this zigzagging and exaggerate the differences between cases, the results were compared at iteration 30. However, this anomalous behavior near the end of the optimization process reinforces our appreciation for the excellent convergence behavior everywhere else (see e.g., Fig. 8a). Obviously, most of the analysis and gradient predictions provided by the CFD code[10] are exceptionally reliable, which contributes to the success of the robust optimization procedure.

A final comparison can be made between airfoil shapes and their effect on the flow field as in Figs. 10 and 11. Fig. 10 shows the original NACA-0012 baseline airfoil compared with an optimized shape from Case 4 after 40 iterations. The vertical axis is exaggerated to show that the optimized shape is thinner and more cambered than the baseline. This smooth optimized airfoil shape is quite different from the shapes reported in previous papers (e.g., Refs. 7 and 8).

Fig. 11 compares the effect of the optimized airfoil shapes in Case 4 on the flow field. Fig. 11a shows the flow field at $M = 0.8$ after 10 iterations and Fig. 11b shows the flow field at the same Mach number after 40 iterations. A noticeable shock boundary can be seen in each of the figures. Notice that the shock is much weaker in Fig. 11b (there are no red contours indicating high local Mach number). Fig. 11c compares pressure on the upper surface of the airfoil for Case 4; dark blue squares indicate C_p after 10 iterations and light yellow triangles indicate C_p after 40 iterations. Clearly, the change in C_p is less after 40 iterations, another indication that the shock strength is reduced. The results presented in Figs. 10 and 11 are typical of the results seen for all cases. Our experience with the general solution strategy and various choices of tuning parameters is that all choices yield reasonable airfoil shapes and consistent drag reduction.

Since this is essentially a multi-objective problem, there is not one right answer. Rather, there are many good solutions and our test cases have revealed several. To demonstrate that the solutions are good ones, we compare them with the traditional multi-point solutions. For example, Case 12 uses 21 evenly spaced design points and sets $\gamma_{\min}$ to a large negative number. Thus, Case 12 resembles a multi-point optimization problem where the number of Mach numbers is larger than the number of design variables[8]. Such an approach automatically reduces the area under the drag curve by reducing drag at a large number of individual design points (recall Fig. 5c).

Fig. 12 compares the drag profiles for Case 12 after 10 and 50 optimization iterations with the drag profile for Case 1 after 50 iterations. Notice that the drag profiles for Cases 1 and 12 are very similar. Yet, Case 1 uses 4 design points and Case 12 uses 21 design points. Thus, Case 12 requires as much computational effort in 10 iterations as Case 1 requires in 50 iterations. Fig. 12 provides convincing evidence that the robust optimization methods presented in this paper produce good solutions in an efficient manner.

The robust optimization strategy has been demonstrated for an optimization problem with one uncertain parameter, cruise Mach number. The authors believe that this strategy can be extended to problems with several uncertain parameters. The challenge is to evaluate the expected value integral with a limited number of expensive disciplinary analyses. One approach is to screen the candidate uncertain parameters and fix the values of those parameters with a less significant effect on the objective and constraints. Another approach is to use an efficient sampling scheme such as Latin hypercubes. These approaches should be successful as long as the multiple uncertain parameters are not correlated.

Conclusions

This paper uses a 2-D airfoil optimization problem to illustrate and compare several options for optimization with uncertainty. These options are implemented in the same computer code with the same nonlinear programming techniques. All options proved to be viable for the test problem. Since the initial airfoil is a poor choice for the Mach range specified, all options significantly improved the design. Some options seek a robust design in the neighborhood of a baseline configuration, while others seek unexploited trade-offs in the design space. Thus, the test cases converged to several candidate airfoil designs depending on the optimization options selected.

This study illustrates several important lessons about robust optimization. First, robust optimization uses a multi-objective formulation and thus provides not one right answer but rather lots of viable candidate designs. Second, the key to successful robust optimization is the choice of a smart descent direction and an adjustable trust region size; the other options can affect the convergence rate and the convergence path. Third, a successful airfoil optimization must include dependable CFD analysis and grid generation codes and accurate predictions of gradients of the objective and constraints.

It is concluded that robust optimization is an important tool for multidisciplinary design. It is needed when some of the design parameters (e.g., operating conditions) are inherently variable, or when some of the design specifications are uncertain (e.g., maximum payload), or when some of the cost drivers are subject to change in the future (e.g., fuel prices). The importance of robust optimization increases when these uncertain parameters have a strong nonlinear effect on the objective function and constraints. An interesting example of robust optimization is provided in this application to airfoil shape optimization. In this problem, cruise Mach number is the uncertain operating condition with nonlinear effects on lift and drag of the airfoil.

References

1. Giesing, J.; and Barthelemy, J.-F. M.: "Summary of Industry MDO Applications and Needs," AIAA Paper 98-4737, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, Sept. 2–4, 1998.

2. Young, J.; Anderson, R.; and Yurkovich, R.: "A Description of the F/A-18E/F Design and Design Process," AIAA Paper 98-4701, 7th AIAA Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, Sept. 2–4, 1998.

3. Hicks, R. M.; and Vanderplaats, G. N.: "Application of Numerical Optimization to the Design of Supercritical Airfoils Without Drag-Creep," SAE Paper 770440, Business Aircraft Meeting, Wichita, Kansas, March 29– April 1, 1977.

4. Drela, M.: "Pros and Cons of Airfoil Optimization," in Frontiers of Computational Fluid Dynamics 1998, edited by D. A. Caughey and M. M. Hafez, World Scientific, 1998, pp. 363-381.

5. Reuther, James A.; Jameson, Anthony; Alonso, Juan J.; Rimlinger, Mark J.; and Saunders, David: "Constrained Multipoint Aerodynamic Shape Optimization Using Adjoint Formulation and Parallel Computers, Part 1 & 2," AIAA Journal of Aircraft, 36(1), 1999, pp. 51–74.

6. Messac, Achille: "From Dubious Construction of Objective Functions to the Application of Physical Programming," AIAA Journal, 38(1), 2000, pp. 155–163.

7. Huyse, Luc: "Free-Form Airfoil Shape Optimization Under Uncertainty Using Maximum Expected Value and Second-Order Second-Moment Strategies," NASA/CR-2001-211020 (Also ICASE Report No. 2001-18).

8. Li, Wu; Huyse, Luc; and Padula, Sharon: "Robust Airfoil Optimization to Achieve Consistent Drag Reduction Over a Mach Range," NASA/CR-2001-211042 (Also ICASE Report No. 2001-22).

9. Huyse, Luc; Padula, Sharon L.; Lewis, R. Michael; and Li, Wu: "A Probabilistic Approach to Free-Form Airfoil Shape Optimization Under Uncertainty," AIAA Journal, to appear in 2002.

10. Anderson, W. K.; and Bonhaus, D. L.: "Airfoil Design on Unstructured Grids for Turbulent Flows," AIAA Journal, 37(2), 1999, pp. 185–191.

Figures

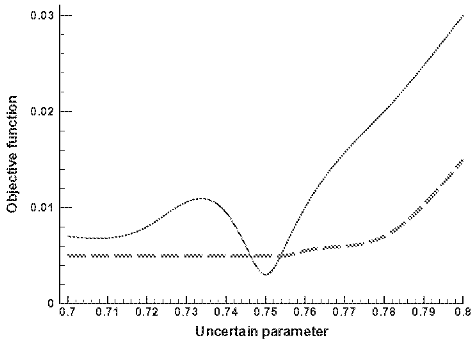
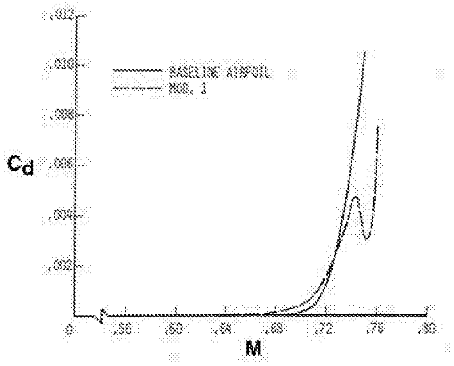
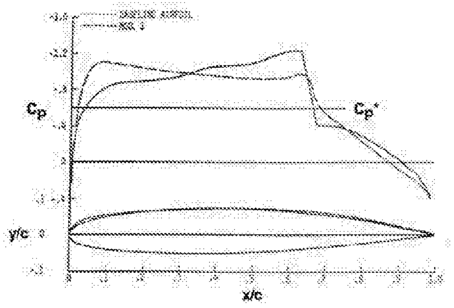


Fig. 1. Notional comparison of single-point optimization (solid line) and robust optimization (dashed line).



2a. Drag profiles for baseline and modified airfoil.



2b. Inviscid drag minimization results; $M = 0.75$, $\alpha = 1$ degree.

Fig. 2. Comparison of baseline airfoil with modified airfoil (MOD. 1) (from Hicks and Vanderplaats 1977).

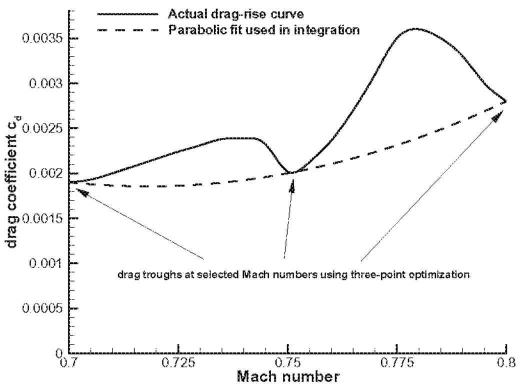


Fig. 3. Typical airfoil shape optimization results show weakness of multi-point optimization compared with robust optimization (from Huyse et. al. 2002).

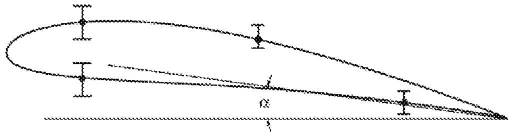
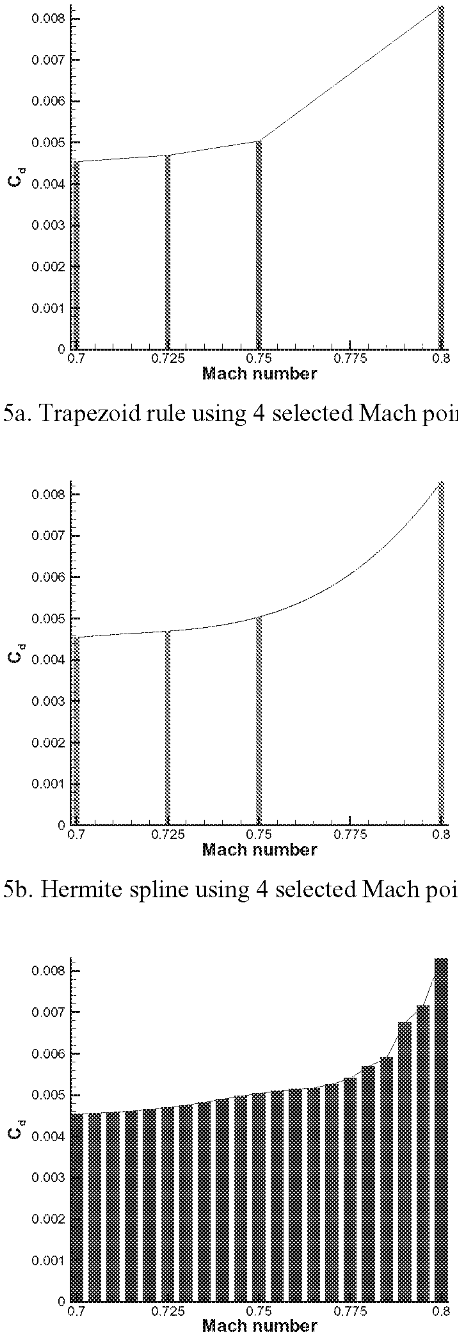


Fig. 4. Airfoil showing angle of attack and typical bounds on spline control points.

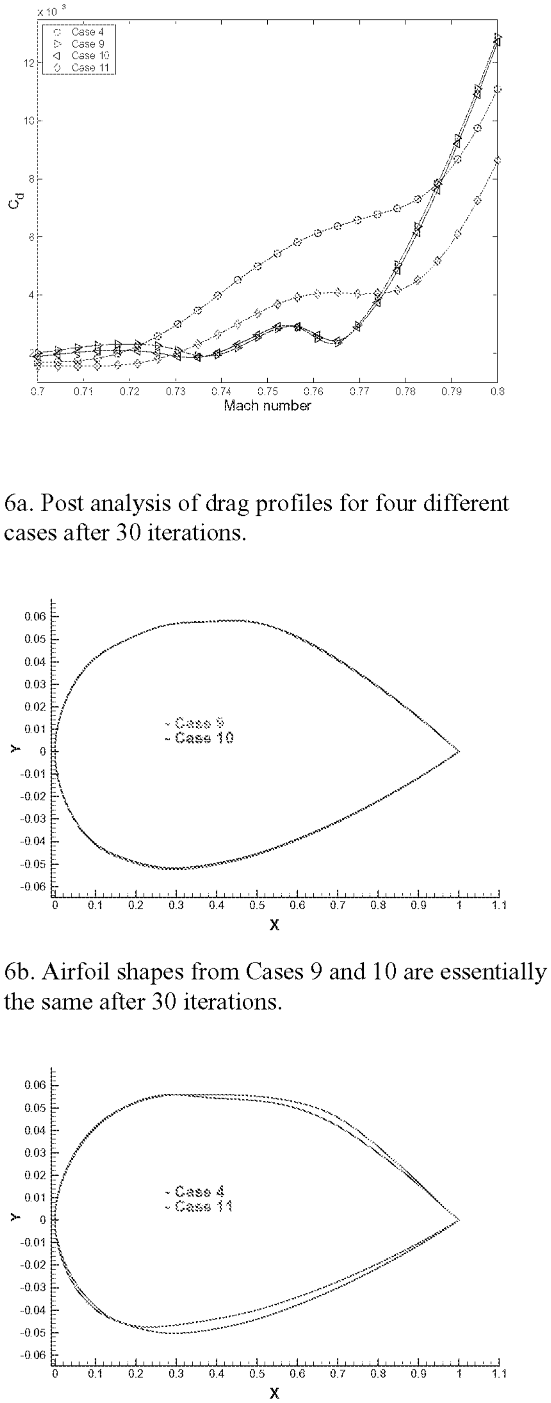


5a. Trapezoid rule using 4 selected Mach points.

5b. Hermite spline using 4 selected Mach points.

5c. Trapezoid rule using 24 Mach points.

Fig. 5. Evaluation of integral needed to calculate expected value of drag coefficient.



6a. Post analysis of drag profiles for four different cases after 30 iterations.

6b. Airfoil shapes from Cases 9 and 10 are essentially the same after 30 iterations.

6c. Airfoil shapes from Cases 4 and 11 after 30 iterations.

Fig. 6. Comparison of solutions for Cases 4, 9, 10, and 11.

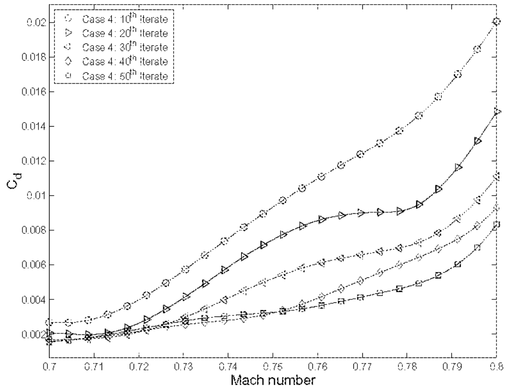
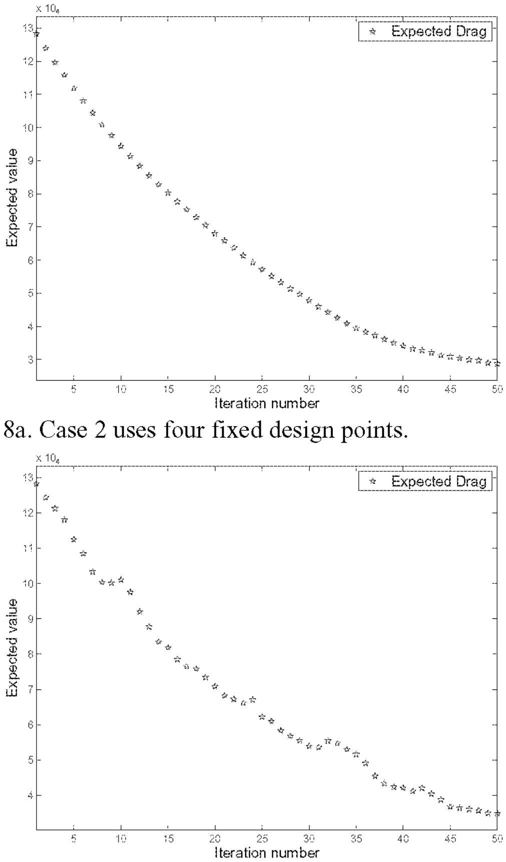
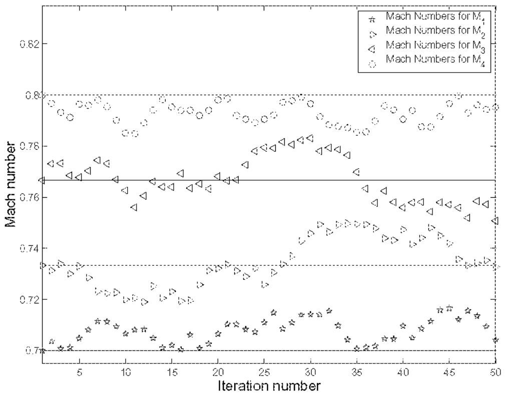


Fig. 7. Optimization convergence history of drag profiles for Case 4.



8a. Case 2 uses four fixed design points.

8b. Case 6 uses four random design points.



8c. Fixed (solid line) and random design points for each iteration.

Fig. 8. Convergence history showing decrease in the expected value of C_d with iteration number.

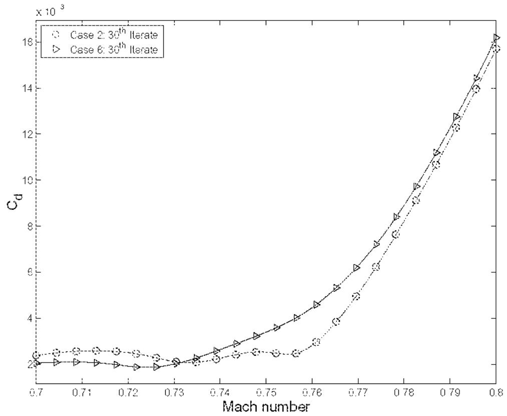


Fig. 9. Comparison of drag profiles for Cases 2 and 6 after 30 iterations.

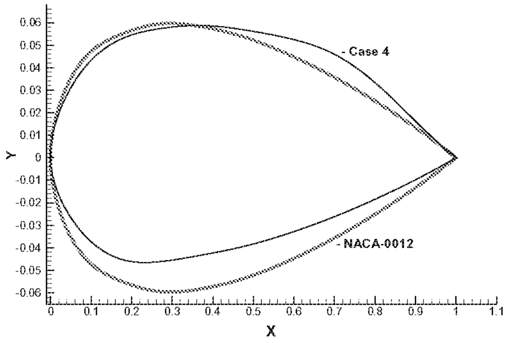
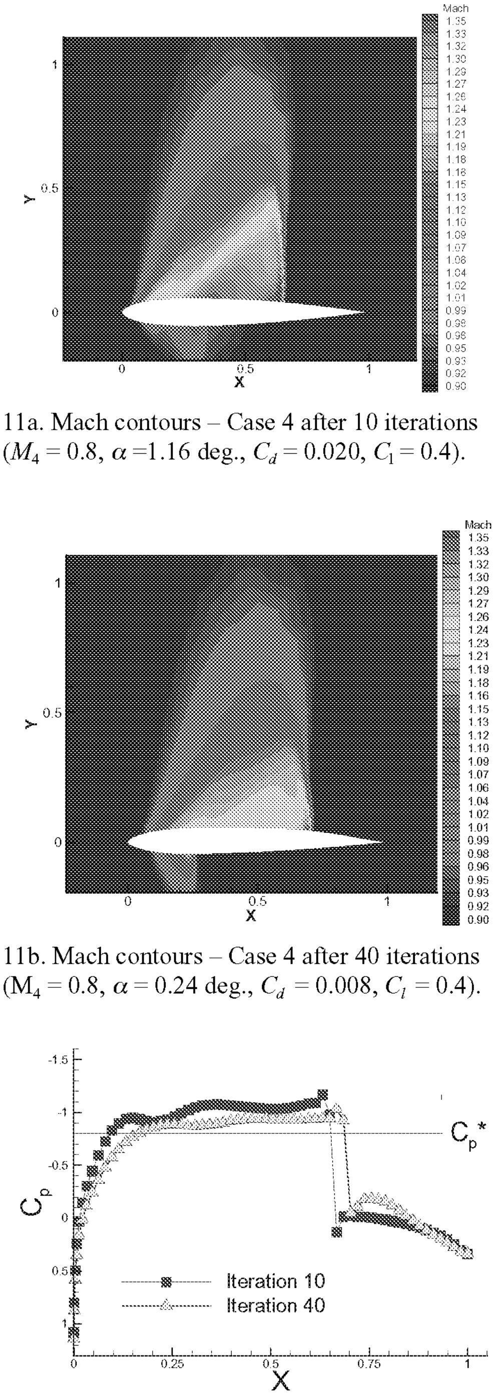


Fig. 10. Comparison of baseline and optimized airfoil shapes.



11a. Mach contours – Case 4 after 10 iterations ($M_4 = 0.8$, α =1.16 deg., $C_d = 0.020$, $C_l = 0.4$).

11b. Mach contours – Case 4 after 40 iterations ($M_4 = 0.8$, $\alpha = 0.24$ deg., $C_d = 0.008$, $C_l = 0.4$).

11c. Pressure curves after 10 and 40 iterations ($M_4 = 0.8$, $C_l = 0.400$).

Fig. 11. Comparison of local Mach contours and C_p curves for Case 4.

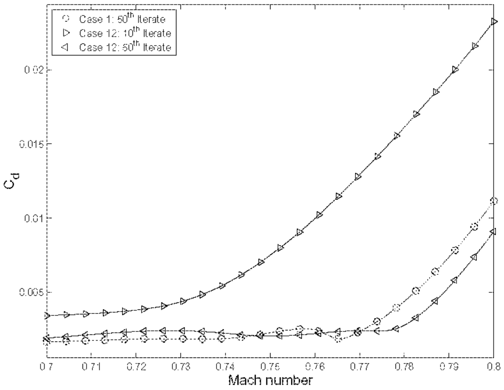


Fig. 12. Comparison of Case 1 (4 design points) and Case 12 (21 design points).