Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

AIRFOIL OPTIMIZATION WITH EFFICIENT GRADIENT CALCULATIONS

Thomas Sorensen
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Room 37-350
Cambridge, MA 02139

1 ABSTRACT

The viscous airfoil design/analysis code XFOIL was extended to allow optimization using conformal mapping coefficients as design variables. The optimization technique employed was the Steepest Descent method applied to a Penalty Function. The gradients of the aerodynamic variables with respect to the design variables were cheaply calculated as by-products of XFOIL's integral boundary layer Newton solver. The speed of the optimization process was further increased by updating the Newton system boundary layer variables after each optimization step using the available gradient information. Two examples are presented.

2 INTRODUCTION

Airfoil design can be broken into two schools of thought. The more recent of the two involves the use of inverse design methods whereby the airfoil geometry is generated to match a specified pressure distribution. The drawback is in determining what makes a good pressure distribution. Many examples of inverse design techniques exist in the literature [1, 2, 3]. The older design practice uses trial and error geometry guessing. Each new geometry is evaluated using an airfoil analysis method and is compared to previous designs. This is continued until an acceptable design is iteratively converged upon. This is a time consuming process, but, it does lend itself to numerical optimization techniques. Many methods have been tried for inviscid airfoils, several examples of which are given by Vanderplaats [4, 5]. Optimization can be computationally intensive, so to be a viable design tool the optimization method employed must be efficient. Optimization efficiency can be increased by the use of gradient information but calculation of this information adds to the computational burden. One method of obtaining the gradient information is to perform finite difference calculations, however, this can be extremely expensive.

The object of the present research was to modify an existing 2D airfoil design/analysis code to calculate gradient information during the analysis procedure, with a minimum of excess work, such that this information can be used in an optimization process. The optimizer written for the design code was simple and robust, but not necessarily the most efficient since the emphasis was on developing the ingredients for the optimization: design variables and gradient information. The code used was Drela's XFOIL code [6]. XFOIL has several design routines, and includes both viscous and inviscid analysis routines. Principles from both the design and viscous analysis routines were combined to allow viscous optimizations.

The outline for the remainder of this paper is to first present the governing equations, the choice of design variables, and how these variables allow efficient gradient calculations. These same gradients can also be used to further speed the optimization process which will be presented next. Two design examples will be given at the end.

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences
(ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

3 ANALYSIS

3.1 Governing Equations

The optimization scheme utilized in XFOIL was an iterative 'Steepest Descent'-type. In order to use this technique the Objective Function and constraints were combined into a Penalty Function such that the constrained airfoil optimization problem is converted into an unconstrained problem. A constrained airfoil optimization problem can be stated in Penalty Function form as

$$Minimize : P(\mathbf{x}) = F(\mathbf{x}) + \frac{1}{2}\sum_{j=1}^{m} K_j \left(g_j(\mathbf{x})\right)^2 , \tag{1}$$

where,

$$g_j(\mathbf{x}) \geq 0 \qquad for \;\; j = 1, m \tag{2}$$

are the constraints that the airfoil is subject to, and

$$K_j = \begin{cases} 0 & g_j(\mathbf{x}) \geq 0 \\ \kappa & g_j(\mathbf{x}) < 0 \end{cases} , \tag{3}$$

are the switches that turn the constraints on and off. The cost parameter, κ, is a large positive quantity used to control the influence of the constraint on the optimization process [5]. The Objective Function, $F(\mathbf{x})$, is the function that the optimizer will drive to the lowest possible value, subject to the stated constraints, using the design variables $\mathbf{x}$. For airfoil optimization the Objective Function could be simply the drag coefficient or a combination of several airfoil characteristics such as the negative of the range parameter, $-MC_l/C_d$.

3.2 Design Variables

The unit circle in the ζ-plane can be mapped to an airfoil in the z-plane by the transformation [3]

$$\frac{\partial z}{\partial \zeta} = (1 - \frac{1}{\zeta})^{(1-\epsilon_{te})} \exp\left\{\sum_{n=0}^{\infty}(A_n + iB_n)\zeta^{-n}\right\}, \qquad n = 0, 1, 2, \cdots \tag{4}$$

where, $\pi\epsilon_{te}$ is the trailing edge angle. The design variables employed in XFOIL's optimizer are a finite number of the real and imaginary parts of the complex coefficients of Eq. 4:

$$\mathbf{x} = \{A_2,\ A_3, \cdots A_{N_A},\ \ B_2,\ B_3, \cdots B_{N_B}\}^T . \tag{5}$$

Using the above notation, there are a total of $(N_A - 2) + (N_B - 2)$ design variables. Each design variable corresponds to a single design mode such that the optimal airfoil is constructed by a sum of these design modes. A particular convenience of these design variables is that the A_n's control the thickness distribution of the airfoil and the B_n's the camber distribution. Due to this distinction the A_n's and B_n's will be referred to, respectively, as the symmetric modes and the anti-symmetric modes. The first 3 symmetric and anti-symmetric design modes are shown in Fig. 1. The solid lines for the symmetric modes indicate the airfoil surface for one value of A_n. The dashed lines show how the surface (i.e. the thickness) changes as another value of A_n is used. For the anti-symmetric modes, the lines are not the airfoil surface, but the camber lines. The first usable design modes are A_2 and B_2 since A_0, A_1, B_0, and B_1 are constrained by Lighthill's constraints [2] and therefore are not available as design variables.

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

The A_n and B_n coefficients completely control the airfoil geometry with the exception of the trailing edge angle and gap. For a typical airfoil only the first twenty or so C_n's are required to define the airfoil. The value of the design variables for a DAE11 airfoil are plotted in Fig. 2 as an indication of their magnitudes for a typical airfoil. The higher frequency modes quickly become unimportant. In both cases, only approximately the first 15 modes are important. The DAE11 geometry is shown in Fig. 3 for reference. The higher modes, however, become important for airfoils with small leading edge radii.

3.3 Aerodynamic Quantities

For optimization efficiency it is imperative that gradient information be calculated and calculated cheaply. The gradient information will also prove useful in making XFOIL's viscous analysis procedure run faster as will be shown shortly.

In its unmodified configuration XFOIL solves a viscous flow around an airfoil by constructing 3 linearized boundary layer (BL) equations at each airfoil and wake node (N airfoil nodes, N_w wake nodes) and solving the resulting system using a Newton solver. For a viscous airfoil analysis all aerodynamic quantities of interest are functions of the five BL variables: C_τ, θ, $m \equiv u_e\delta^*$, u_e, and δ^*. In this text C_τ will represent two quantities: in laminar regions it will be the amplitude of the most-amplified Tollmien-Schlichting wave, and in turbulent regions it will be the maximum shear coefficient. The Newton system only solves for three of these variables, C_τ, θ, and m, since u_e and δ^* are related to the first three variables. For more details of XFOIL, see Drela [6].

To calculate the required BL variable gradients, consider the Newton System used in XFOIL

$$[J]\{\delta\} = -\{R\}. \tag{6}$$

This equation is a block matrix equation where the i^{th}-row, j^{th}-column block of the Jacobian Matrix is

$$[J_{i,j}] = \begin{bmatrix} \frac{\partial f_i}{\partial C_{\tau_j}} & \frac{\partial f_i}{\partial \theta_j} & \frac{\partial f_i}{\partial m_j} \\ \frac{\partial g_i}{\partial C_{\tau_j}} & \frac{\partial g_i}{\partial \theta_j} & \frac{\partial g_i}{\partial m_j} \\ \frac{\partial h_i}{\partial C_{\tau_j}} & \frac{\partial h_i}{\partial \theta_j} & \frac{\partial h_i}{\partial m_j} \end{bmatrix}. \tag{7}$$

The corresponding i^{th}-row block of the vectors are

$$\{\delta_i\} = \begin{Bmatrix} \delta C_{\tau_i} \\ \delta\theta_i \\ \delta m_i \end{Bmatrix}, \qquad \{R_i\} = \begin{Bmatrix} f_i \\ g_i \\ h_i \end{Bmatrix}. \tag{8}$$

Many of the terms in the Jacobian Matrix are zero, but the detailed structure is not important here.

Equation (6) is constructed using 3 BL equations at each node all with the functional form

$$R_i = R_i(C_{\tau_{i-1}},\ C_{\tau_i},\ \theta_{i-1},\ \theta_i,\ m_1,\ m_2, \cdots,\ m_{N+Nw}), \tag{9}$$

where, R_i can be f_i, g_i, or h_i and the subscripts indicate which node is being considered. The edge velocity, u_e, is composed of an inviscid and a viscous source contribution,

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

$$u_{e_k} = q_k + \sum_j d_{kj} m_j, \tag{10}$$

where, the inviscid part q_k depends on the airfoil geometry and hence A_n and B_n. The mass defect, m, therefore also depends on A_n and B_n, and so does the viscous residual R_i in Eq. (9). Consequently, a new Newton system is obtained in the form

$$[J \mid A] \left\{ \frac{\delta}{\Delta} \right\} = - \{R\}. \tag{11}$$

The i^{th}-row block of the Jacobian addition, $[A]$, is

$$[A_i] = \begin{bmatrix} \frac{\partial f_i}{\partial A_2} & \frac{\partial f_i}{\partial A_3} & \cdots & \frac{\partial f_i}{\partial A_{N_A}} & \frac{\partial f_i}{\partial B_2} & \frac{\partial f_i}{\partial B_3} & \cdots & \frac{\partial f_i}{\partial B_{N_B}} \\ \frac{\partial g_i}{\partial A_2} & \frac{\partial g_i}{\partial A_3} & \cdots & \frac{\partial g_i}{\partial A_{N_A}} & \frac{\partial g_i}{\partial B_2} & \frac{\partial g_i}{\partial B_3} & \cdots & \frac{\partial g_i}{\partial B_{N_B}} \\ \frac{\partial h_i}{\partial A_2} & \frac{\partial h_i}{\partial A_3} & \cdots & \frac{\partial h_i}{\partial A_{N_A}} & \frac{\partial h_i}{\partial B_2} & \frac{\partial h_i}{\partial B_3} & \cdots & \frac{\partial h_i}{\partial B_{N_B}} \end{bmatrix}. \tag{12}$$

The added vector term contains the changes in the design variables

$$\{\Delta\} = \left\{ \Delta A_2, \ \Delta A_3, \ \cdots \ \Delta A_{N_A}, \quad \Delta B_2, \ \Delta B_3, \ \cdots \ \Delta B_{N_B} \right\}^T, \tag{13}$$

where, $\Delta(\ )$ implies a change in the design variables between the current optimization step and the next optimization step. The modified Jacobian matrix, $[J \mid A]$, is no longer square, but during normal viscous calculations the geometry is fixed and thus the ΔA_n and ΔB_n's are known (i.e. they are zero). Therefore, rewriting Eq. (11) with all knowns on the right hand side and then pre-multiplying both sides by $[J]^{-1}$ the system reduces to

$$\{\delta\} = - [J]^{-1} \{R\} + [D] \{\Delta\}, \tag{14}$$

where,

$$[D] = - [J]^{-1} [A]. \tag{15}$$

The viscous solution is obtained when the residual, $\{R\}$, is zero. Thus, at convergence Eq. (14) will have the same form as a first order Taylor series expansion of the 3 BL equations in terms of the design variables. For example, the Taylor expansion for C_τ, θ, and m at the i^{th} node is

$$\begin{Bmatrix} \delta C_{\tau_i} \\ \delta \theta_i \\ \delta m_i \end{Bmatrix} = \sum_{n=2}^{N_A} \Delta A_n \begin{Bmatrix} \frac{\partial C_{\tau_i}}{\partial A_n} \\ \frac{\partial \theta_i}{\partial A_n} \\ \frac{\partial m_i}{\partial A_n} \end{Bmatrix} + \sum_{n=2}^{N_B} \Delta B_n \begin{Bmatrix} \frac{\partial C_{\tau_i}}{\partial B_n} \\ \frac{\partial \theta_i}{\partial B_n} \\ \frac{\partial m_i}{\partial B_n} \end{Bmatrix}. \tag{16}$$

The Taylor coefficients are the BL variable derivatives being sought and after close examination it can be seen that they are the columns of $[D]$. For example, the i^{th}-row block of $[D]$ is

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

$$[D_i] = \begin{bmatrix} \frac{\partial C_{\tau_i}}{\partial A_2} & \frac{\partial C_{\tau_i}}{\partial A_3} & \cdots & \frac{\partial C_{\tau_i}}{\partial A_{N_A}} & \frac{\partial C_{\tau_i}}{\partial B_2} & \frac{\partial C_{\tau_i}}{\partial B_3} & \cdots & \frac{\partial C_{\tau_i}}{\partial B_{N_B}} \\ \frac{\partial \theta_i}{\partial A_2} & \frac{\partial \theta_i}{\partial A_3} & \cdots & \frac{\partial \theta_i}{\partial A_{N_A}} & \frac{\partial \theta_i}{\partial B_2} & \frac{\partial \theta_i}{\partial B_3} & \cdots & \frac{\partial \theta_i}{\partial B_{N_B}} \\ \frac{\partial m_i}{\partial A_2} & \frac{\partial m_i}{\partial A_3} & \cdots & \frac{\partial m_i}{\partial A_{N_A}} & \frac{\partial m_i}{\partial B_2} & \frac{\partial m_i}{\partial B_3} & \cdots & \frac{\partial m_i}{\partial B_{N_B}} \end{bmatrix}. \tag{17}$$

The elements of this matrix are found not by carrying out the matrix multiplication as indicated in Eq. (15) but by solving the original Newton system with the columns of $[A]$ added as extra right hand sides. Since a direct matrix solver is used, very little extra work is needed to calculate the required sensitivities. In addition, the extra right hand sides only have to be included *after* convergence of the system, not *every* time the system is solved.

The above derivation presents a scheme to compute the BL variable gradients if the gradients of the BL equations, Eqs. (9), are known (i.e. if the terms of $[A]$ are known). The terms in $[A]$ are found by use of the chain rule and are included here without derivation

$$\frac{\partial R_i}{\partial A_n} = \left(\frac{\partial R_i}{\partial q_{i-1}}\right)\left(\frac{\partial q_{i-1}}{\partial A_n}\right) + \left(\frac{\partial R_i}{\partial q_i}\right)\left(\frac{\partial q_i}{\partial A_n}\right), \tag{18}$$

where,

$$\frac{\partial R_i}{\partial q_{i-1}} = \frac{\partial R_i}{\partial u_{e_{i-1}}} - \frac{\partial R_i}{\partial \delta^*_{i-1}} \frac{m_{i-1}}{u^2_{e,-1}}, \tag{19}$$

is found using Eq. (10) and the definition of the mass defect, $m = u_e\delta^*$. Similarly for the B_n derivatives. In the above four equations R_i can be f_i, g_i, or h_i. At node i the derivatives depend only on the information at that node and the upstream node $i-1$. All the terms in Eq. (19) are already available once XFOIL constructs the Newton system. Further details of the above equations can be found in the author's Master's Thesis [7].

The only remaining unknown sensitivities in Eq. (18) are the derivatives of q. These can be calculated analytically from the expression for q obtained after the complex potential is mapped from the circle-plane to the airfoil-plane. At any point, ζ, in the circle-plane, the physical speed is

$$q = \exp\left\{\Re\left[\ln\left(\left(1-\frac{1}{\zeta}\right)^{\epsilon_{te}}\left(e^{-i\alpha} + e^{i\alpha}\zeta^{-1}\right)\right) - \sum_{n=0}^{\infty}(A_n + iB_n)\zeta^{-n}\right]\right\}. \tag{20}$$

The derivatives of this equation are remarkably easy and cheap to compute:

$$\frac{\partial q}{\partial A_n} = -q\Re\left(\frac{1}{\zeta^n}\right), \tag{21}$$

$$\frac{\partial q}{\partial B_n} = +q\Im\left(\frac{1}{\zeta^n}\right). \tag{22}$$

3.4 Geometry Gradient

Now, all aerodynamic variables that depend on the flow solution have been differentiated, and only one further piece of gradient information is necessary; the geometry sensitivity. This can be found analytically using the integrated form of Eq. (4), however, in practice there is a complication. The difficulty arises due to the need for the geometry gradient for the unit chord

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

airfoil. Equation (4), when integrated, does not produce a unit chord airfoil and therefore its gradient will not be for a unit chord. The geometry is subsequently normalized, however this is not completely satisfactory for the gradient due to movement of the leading edge. This is not a concern for symmetric airfoils and is a relatively small effect for cambered airfoils. Therefore, the movement of the leading edge point was ignored in calculations for the gradient of z.

3.5 Updating BL Variables

The Newton system of XFOIL uses the BL variables of the previous solution as the starting point of the new solution, therefore, the speed of the optimization can be increased by simply approximating the BL variables of the new airfoil. This can be done by adding the following perturbations to the BL variables at the old optimization step at those nodes not affected by the transition point:

$$\{\delta\} = [D]\{\Delta\}. \tag{23}$$

The ΔA_n's and ΔB_n's in the $\{\Delta\}$ vector of Eq. (23) are the changes in the design variables between the current and new optimization steps, and are calculated from Steepest Descent Equation. The remaining two perturbations, δu_e and $\delta\delta^*$, can be found using

$$\delta u_e = \sum_{n=2}^{N_A} \frac{\partial u_e}{\partial A_n}\Delta A_n + \sum_{n=2}^{N_B} \frac{\partial u_e}{\partial B_n}\Delta B_n, \tag{24}$$

and

$$\delta\delta^* = \sum_{n=2}^{N_A} \frac{\partial \delta^*}{\partial A_n}\Delta A_n + \sum_{n=2}^{N_B} \frac{\partial \delta^*}{\partial B_n}\Delta B_n. \tag{25}$$

For a reasonable optimization step size this linear extrapolation will give a good approximation to the new BL variables. Thus, the Newton system constructed during the analysis of the new design point will converge faster than if no updating were done since it will have a better initial condition.

Movement of the upper and lower surface transition points from one panel to another will cause such severe changes in the BL variables that this linear extrapolation will not work near the transition points. If not considered separately, the poor transition point approximations would be enough to negate the gains in efficiency promised by the updating. The new location of the transition points is approximated and then the BL variables at each panel the transition points have passed over are 'fudged' . This 'fudging' process will only affect the rate at which the Newton system converges, it will not affect the converged solution. For C_τ, θ, and u_e the approximation across the transition point shift is a linear extrapolation from the previous two approximated points, i.e.

$$C_{\tau_i} = 2C_{\tau_{i-1}} - C_{\tau_{i-2}}, \tag{26}$$

where i is a BL node the transition point has passed over. The equations for θ and u_e are similar. For the remaining two BL variables, m and δ^*, it was found to be a better approximation is to set $m_i = m_{i-1}$ and $\delta^*_i = \delta^*_{i-1}$. All that remains to be able to use these transition point approximations is to determine how far the transition point has shifted. This is done using

$$\delta x_{tran} = \frac{\partial x_{tran}}{\partial C_\tau}\delta C_\tau + \frac{\partial x_{tran}}{\partial \theta}\delta\theta + \frac{\partial x_{tran}}{\partial \delta^*}\delta\delta^* + \frac{\partial x_{tran}}{\partial u_e}\delta u_e. \tag{27}$$

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences
(ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

All the derivative terms in the above are already calculated in XFOIL to construct the Newton system, so the derivation is complete.

The convergence histories for a simple test case with and without updating the BL variables are shown in Fig. 4. The number of iterations for the Newton solver to convergence is plotted versus the optimization step number. The amount of time saved is not extensive, but the low cost of updating makes it worthwhile. As the optimization continues the savings will be smaller since the step sizes are small.

4 RESULTS

The two examples presented in this section were run on a DecStation 5000. These examples were chosen to show the various properties of XFOIL's optimizer, they are not designed to be realistic design problems.

5 Example 1 - C_d minimization, $M = 0$, $\alpha = 0°$

The first test case was designed as a simple example to build faith in the optimization code. A NACA 0015 airfoil was used as the seed airfoil with C_d used as the Objective Function. The only constraint was to keep the angle of attack constant at 0°. The Reynolds Number based on the chord was 10^6. The two design variables used were A_2 and A_3. Using only two design variables will allow a pictorial representation of the optimization path to be constructed.

Figure 5 portrays the optimization space for this test case. The contours are of constant C_d and a local minimum is located in the upper left corner. The seed airfoil is located out of the picture in the lower right corner and the path taken by the optimizer is marked by the crosses. Convergence took 24 iterations and approximately 12 minutes. Figure 5 clearly shows the larger step sizes in the first five steps, i.e. in the region of large slope. The step directions are perpendicular to the contours, as they should be, where the gradients are large. As the optimum is neared the step directions start to parallel the contours. This is due to the approximations made in the gradient calculations. This is not a detriment since the exact mathematical optimum is relatively unimportant.

From Fig. 6 it is obvious that the largest drag reductions are produced in the first few iterations. This is a recurrent observation. Figure 7 compares the optimal airfoil to the seed airfoil. Because only two design modes were utilized, the possible change in the airfoil is small. However, large changes were made in C_d by modifying the airfoil such that the transition points were moved further aft.

5.1 Example 2 - C_d minimization, $M = 0$, $C_l = 0.5$

The second example optimized the C_d of an airfoil using 7 symmetric and 5 anti-symmetric design modes. The seed airfoil was an NACA 3412 and was constrained for a constant lift coefficient and a minimum allowed thickness at 95% of the chord. This constraint was necessary to prevent negative thickness airfoils. The cost parameter and the Reynolds number were $\kappa = 100$ and $Re = 5 \times 10^6$.

This example was stopped after a viscous Newton system was unconverged at the 38^{th} optimization iteration. The Penalty Function is shown in Fig. 8. The drag reduction slows slightly after 20 iterations but is definitely still headed down when the optimizer was stopped. The optimizer was restarted using the last airfoil generated before the Newton system failed as

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

the new seed airfoil. Optimization convergence was achieved after an additional 15 iterations. The optimization required approximately 30 minutes. The drag was further lowered from $C_d = 0.00389$ to $C_d = 0.00380$. The reason for the unconverged Newton system is unexplained but it does not invalidate the results of the optimizer.

The pressure plots of the seed and optimized airfoils are shown in Figs. 9 and 10, respectively. The dashed lines in the C_p curves are the inviscid solutions and the solid lines the viscous solutions. The waviness apparent in the C_p curve of the optimized airfoil is due to the fact that higher design modes were not used during the optimization.

Modification of an airfoil design code to use mapping coefficients as the design variables was successfully implemented. Gradient information was calculated within the analysis portion of the code with a minimum of extra effort. The gradient information was shown to be accurate

When used in the proper way, the XFOIL optimizer can become a valuable design tool. The optimizer should not be used as a 'black box' to create perfect airfoils but as a designer's tool that will free the designer to become more creative and productive by reducing the time spent in iterative design modifications. The 'optimal' airfoils obtained should be used to give the designer ideas for what characteristics the real airfoil should have.

There were also several areas in which the XFOIL optimizer did not live up to expectations. The first is the limited number of design variables that could be utilized. It was found that the optimizer should be restricted to $N_A \leq 12$ and $N_B \leq 12$ because the higher mode derivatives became inaccurate. This does not allow the generation of completely general airfoils with the chosen design variables. This is a disappointment, however the cheap gradient calculations made possible by using the mapping coefficients as design variables make up for this deficiency. Another disappointment was the temperamental nature of XFOIL's Viscous Newton solver. This does not destroy the promise of the optimizer it only enforces that some care needs to be exercised when using the optimizer.

Another area for future research is the development of design variables that can also control the trailing edge angle and gap, and if possible, be completely general.

6 ACKNOWLEDGMENTS

This research was supported by MIT's Department of Aeronautics and Astronautics' Fellowship and the National Science Foundation's PYI Program.

7 NOMENCLATURE

F	Objective function
$\mathbf{x}$	General design variables
g_j	Constraints
m	Number of constraints
A_n	XFOIL thickness design variables (symmetric)
B_n	XFOIL camber design variables (anti-symmetric)
N_A	Last symmetric design mode used in optimization
N_B	Last anti-symmetric design mode used in optimization
$[J]$	Newton system Jacobian matrix
$[A]$	Addition to Jacobian matrix
$\{\delta\}$	Newton system unknown vector
$\{\Delta\}$	Addition to unknown vector

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

$\{R\}$	Residual vector
$[D]$	Aerodynamic variables derivative matrix
C_l	Coefficient of lift
C_d	Coefficient of drag
M	Mach number
Re	Reynolds number based on airfoil chord
N	Number of airfoil nodes
N_w	Number of wake nodes
$f_i,\ g_i,\ h_i$	Node i boundary layer equations
$C_\tau,\ \theta,\ m,\ d_{kj},\ u_e,\ \delta^*$	Boundary layer variables
x_{tran}	Transition point location
ϵ_{te}	Trailing edge angle parameter
α	Angle of attack
q	Inviscid surface speed
$\zeta = re^{i\omega}$	Complex circle-plane coordinate
Δ	Difference operator
$\delta(\ )$	Newton system perturbation
$\Re(\ )$	Real part of the quantity in the parenthesis
$\Im(\ )$	Imaginary part of the quantity in the parenthesis

References

[1] R. Eppler and D. M. Somers. A computer program for the design and analysis of low-speed airfoils. NASA TM 80210, Aug 1980.

[2] M. S. Selig and M. D. Maughmer. A multi-point inverse airfoil design method based on conformal mapping. In *29th Aerospace Sciences Meeting*, Reno, Nevada, Jan 1991.

[3] J. L. Van Ingen. A program for airfoil section design utilizing computer graphics. In *AGARD-VKI Short Course on High Reynolds Number Subsonic Aerodynamics*, AGARD LS-37-70, April 1969.

[4] G. N. Vanderplaats. Efficient algorithm for numerical airfoil optimization. *Journal of Aircraft*, 16(12), Dec 1979.

[5] G. N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design: with Applications*. McGraw-Hill, New York, 1984.

[6] M. Drela. XFOIL: An analysis and design system for low Reynolds number airfoils. In T.J. Mueller, editor, *Low Reynolds Number Aerodynamics*. Springer-Verlag, Jun 1989. Lecture Notes in Engineering, No. 54.

[7] T.M. Sorensen. Viscous airfoil optimization using conformal mapping coefficients as design variables. Master's thesis, Massachusetts Institute of Technology, Jun 1989.

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

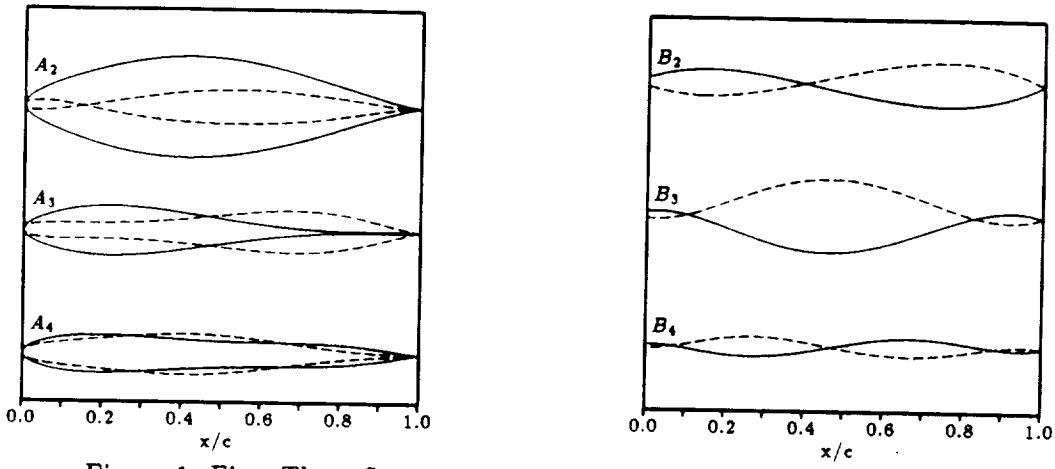

Figure 1: First Three Symmetric and Anti-Symmetric Design Modes

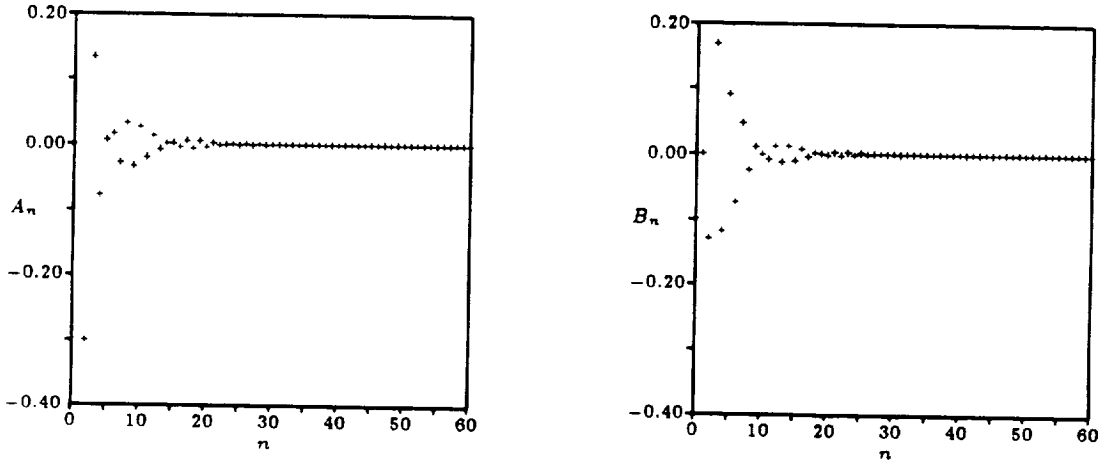

Figure 2: A_n and B_n Distributions for a DAE11 Airfoil

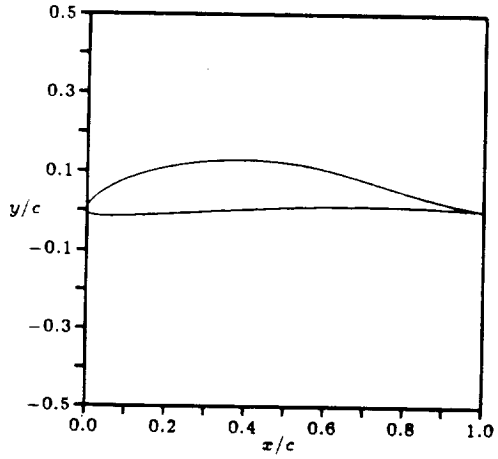

Figure 3: DAE11 Airfoil Geometry

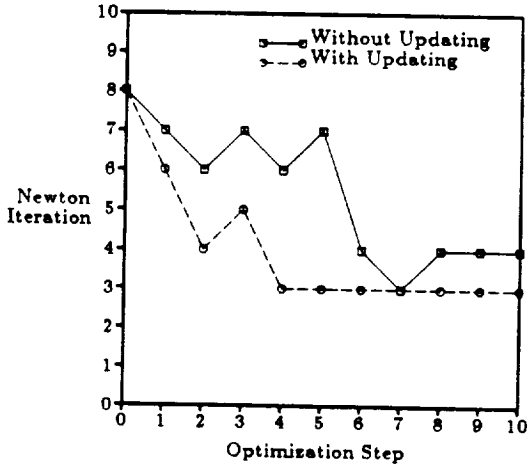

Figure 4: Convergence History With and Without Updating

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

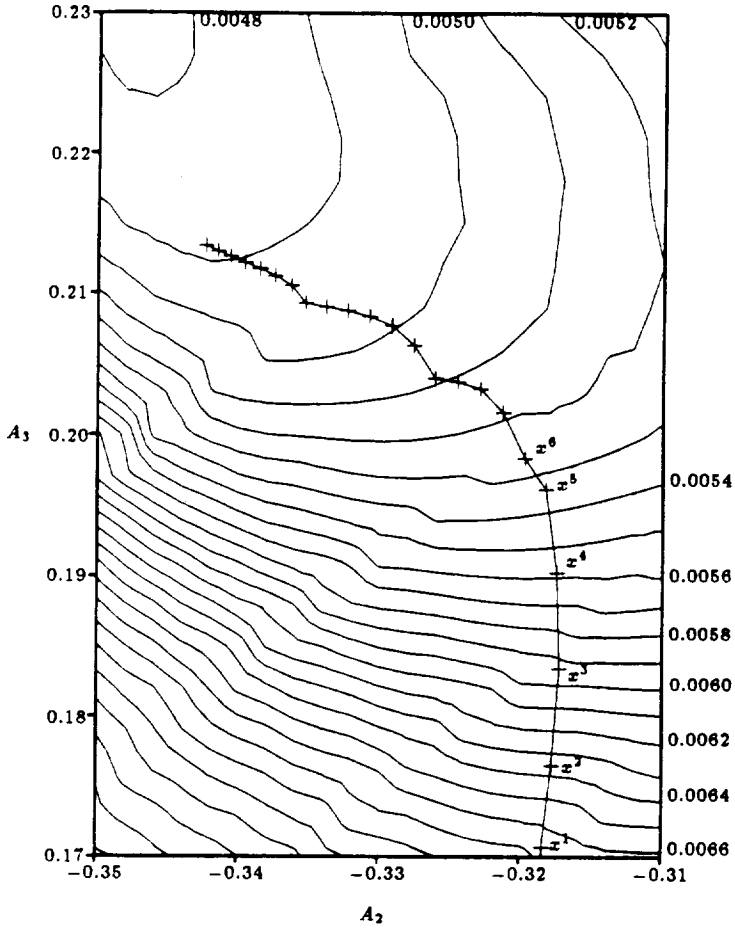

Figure 5: Example 1 - Optimization Path

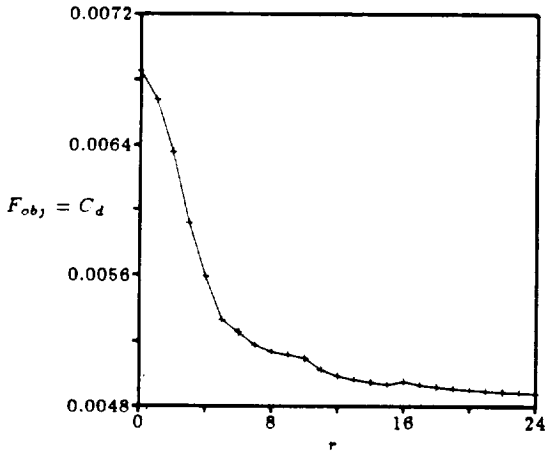

Figure 6: Example 1 - Optimization History

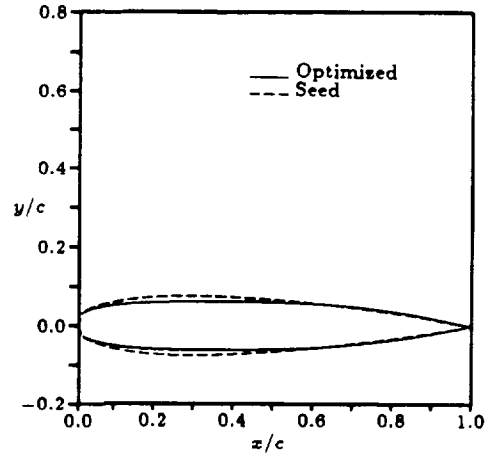

Figure 7: Example 1 - Airfoil Comparisons

Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences
(ICIDES-III), Editor: G.S. Dulikravich, Washington D.C., October 23-25, 1991.

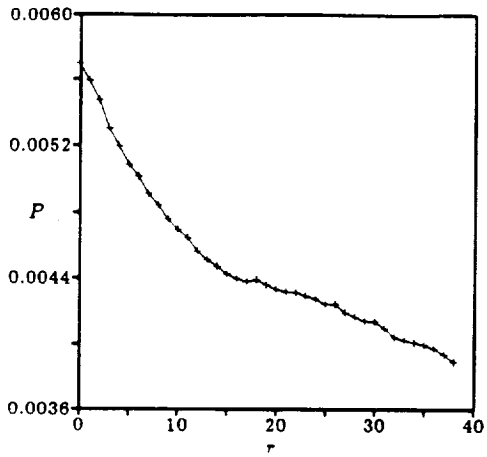

Figure 8: Example 2 - Optimization History

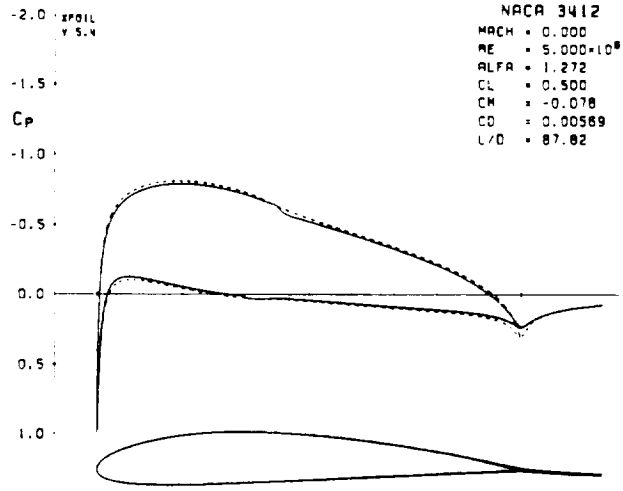

Figure 9: Example 2 - NACA 3412 C_p Plot

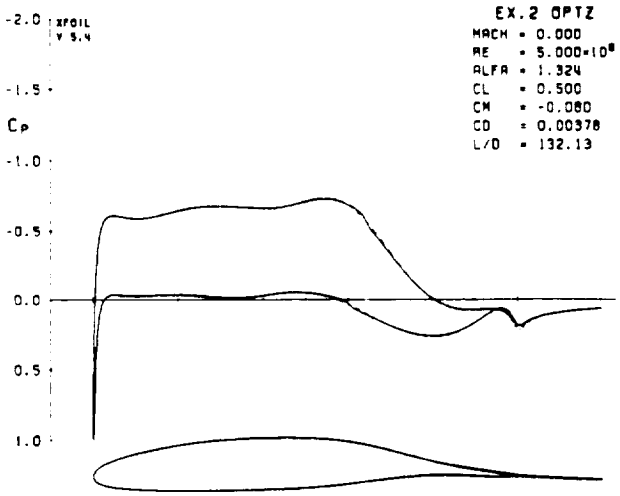

Figure 10: Example 2 - Optimized Airfoil C_p Plot