INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **30**: 205–216 (1999)

RECENT PROGRESS IN AERODYNAMIC DESIGN OPTIMIZATION

R.G. MELVIN*, W.P. HUFFMAN, D.P. YOUNG, F.T. JOHNSON, C.L. HILMES AND M.B. BIETERMAN

The Boeing Company, PO Box 24346, M/S 7L-21, Seattle WA, 98124-0346, USA

SUMMARY

Recent emphasis on reduction of design cycle time and cost in the design of commercial aircraft (P.E. Rubbert, 'CFD and the changing world of airplane design', *AIAA Wright Brothers Lecture*, September, 1994) has sparked a renewed interest in design optimization in aerodynamics, structures and aeroelastics. In this paper, recent developments in the use of design optimization in aerodynamics using the TRANAIR code are considered. Globalization techniques and the extension of the methodology to multipoint design will be discussed. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: aerodynamic design optimization; multipoint design; aeroelastics

1. INTRODUCTION

Today, commercial aircraft are routinely designed using a combination of computational fluid dynamics (CFD) and wind tunnel testing. One possible design method is the use of repetitive analysis and/or wind tunnel testing to develop engineering understanding. An alternative is the inverse design method, in which the designer specifies desirable flow features, in most cases the pressure distribution on the wing surface, and asks the CFD code to compute the geometry that produces such flow features. The many cut-and-try iteration loops of repetitive analysis are now replaced by a systematic inverse CFD procedure.

There are, however, still a number of drawbacks of the inverse design approach. The first one is the difficulty of finding 'good' pressure distributions for highly three-dimensional flows. The second difficulty is the consideration of off-design performance. The inverse design method is inherently a single-point design process, although the designer usually has some knowledge of what pressure architecture of an airfoil at the cruise condition will likely give reasonable off-design characteristics.

These difficulties lead to the desire to develop an optimization methodology for aerodynamic design. A well-formulated optimization method may help to achieve quickly a good compromise between aerodynamic or cost objectives and the constraints imposed on the geometry by other disciplines, such as manufacturing and structures. It will also allow multiple critical flight conditions to be considered so that iteration with off-design considerations can be eliminated, as has been demonstrated in 2D by Drela [1]. Over the past few years, the authors have begun to develop an optimization capability in the TRANAIR code [2–5]. This

* Correspondence to: The Boeing Company, PO Box 24346, M/S 7L-21, Seattle WA, 98124-0346, USA.

CCC 0271–2091/99/100205–12$17.50

code is a two- and three-dimensional full potential flow code with directly coupled strip boundary layer, capable of handling complex geometries through solution adaptive local grid refinement.

In Section 2, an overview of the optimization methodology in TRANAIR is given. In Section 3, various globalization strategies that have been applied will be described and their effectiveness will be demonstrated with some examples. Section 4 describes how the optimization method is extended to multipoint design and shows how it can be used for airfoil design.

2. TRANAIR OPTIMIZATION METHODOLOGY

There are several aspects of the TRANAIR analysis algorithm that lend themselves to design optimization. These are the use of Newton's method, the use of solution adaptive grid refinement, the general geometry capability, and the ability to use transpiration boundary conditions to model changes in surface shape. One of the consequences of the use of Newton's method is the experience with inexactness of derivatives, which carries over almost directly to the optimization case. In addition, the use of a sensitivity formulation enables the implicit and cost effective generation of partial second-order information.

In a solution adaptive grid method for solving partial differential equations, such as the one implemented at TRANAIR, there is a series of grids, $l = 1, 2, \ldots, NG$. On each grid, the flow problem is discretized and solved. The design and optimization capability have been combined with this solution adaptive grid capability by defining a discrete simplified optimization problem on each grid in such a way that the entire process can converge to the solution of the continuous optimal control problem. This is described in detail in [4,5]. For simplicity, in the following description of optimization methods, only the discrete version of the problem that is posed on a given grid is dealt with.

The discretized non-linear state (flow) equations will be denoted by

$$F(X, u) = (F_1(X, u), F_2(X, u), \ldots, F_n(X, u))^T = 0,$$

where the state variables $X = (X_1, X_2, \ldots, X_n)^T$ consist of both the inviscid flow variables and the boundary layer variables. The design parameters $u = (u_1, u_2, \ldots, u_m)^T$ could represent the geometry shape or flow quantities, such as angle of attack or free-stream Mach number. Included in $F(X, u)$ is an inhomogeneous Neumann boundary condition $T(X, u)$, used to approximate the effect of boundary motion [4]. One simple form of this condition is $T(X, u) = \vec{W} \cdot \delta\hat{n}$, where $\vec{W}$ is the mass flux through the boundary, and $\delta\hat{n}$ is the total change in unit normal due to the design parameter u.

Now, consider the problem of minimizing a scalar objective function $I(X, u)$ subject to the constraint that $F(X, u) = 0$. It is assumed that the Jacobian matrix $\partial F_i / \partial X_k$ is invertible for the values of u of interest and the notation that

$$\partial I / \partial X = (\partial I / \partial X_1, \partial I / \partial X_2, \ldots, \partial I / \partial X_n)$$

is a row vector is used.

The necessary conditions for optimality are often formulated by introducing the Lagrange multipliers λ as independent variables [6]. The Lagrangian is then defined by

$$L(X, u, \lambda) = I(X, u) + \lambda^T F(X, u).$$

At an optimum, the Lagrange multiplier λ_i is the derivative of the value of I with respect to changes in the value of the constraint F_i, i.e. $\lambda_i = \mathrm{d}I / \mathrm{d}F_i$. Necessary conditions for an optimum are that the gradient of L be zero.

$$\frac{\partial L}{\partial u} = \frac{\partial I}{\partial u} + \lambda^T \frac{\partial F}{\partial u} = 0, \tag{1}$$

$$\frac{\partial L}{\partial X} = \frac{\partial I}{\partial X} + \lambda^T \frac{\partial F}{\partial X} = 0, \tag{2}$$

$$\frac{\partial L}{\partial \lambda} = F(X, u) = 0. \tag{3}$$

In cases where $\partial F/\partial X$ is invertible for the values of u of interest, the Lagrange multiplier can be eliminated from the formulation. This is done by first solving Equation (2) for λ. This yields

$$\lambda = -\left(\frac{\partial F}{\partial X}\right)^{-T}\left(\frac{\partial I}{\partial X}\right)^T. \tag{4}$$

If this is substituted into Equation (1), the following equivalent necessary conditions for optimality are obtained:

$$\frac{\mathrm{d}I}{\mathrm{d}u} \equiv \frac{\partial I}{\partial u} - \left(\frac{\partial I}{\partial X}\right)\left(\frac{\partial F}{\partial X}\right)^{-1}\left(\frac{\partial F}{\partial u}\right) = 0, \qquad F(X, u) = 0. \tag{5}$$

The quantity $\mathrm{d}I/\mathrm{d}u$ defined above is often called the reduced gradient. The necessary conditions can now be formulated either in terms of solving an adjoint problem (4) for λ, or directly by solving the linearized state equations using multiple right-hand-sides to compute each column of the $n \times m$ matrix

$$Q \equiv \frac{\partial X_i}{\partial u_j} = -\left(\frac{\partial F_k}{\partial X_i}\right)^{-1}\left(\frac{\partial F_k}{\partial u_j}\right). \tag{6}$$

The latter approach is the one implemented in the design and optimization version of TRANAIR. A transpiration boundary condition (an inhomogeneous Neumann boundary condition) is used to approximate $\partial F/\partial u$ as described in [4].

Now, it is shown how to derive a reduced space form of the classical Lagrange–Newton method [7]. Applying Newton's method to Equations (1)–(3), the following equations for the updates δu to u, δX to X, and $\delta\lambda$ to λ are arrived at:

$$\begin{bmatrix} L_{u,u} & L_{u,X} & F_u^T \\ L_{X,u} & L_{X,X} & F_X^T \\ F_u & F_X & 0 \end{bmatrix} \begin{bmatrix} \delta u \\ \delta X \\ \hat{\lambda} \end{bmatrix} = - \begin{bmatrix} I_u^T \\ I_X^T \\ F \end{bmatrix}, \tag{7}$$

where $\hat{\lambda} = \lambda + \delta\lambda$ and the subscript notation for partial derivatives has been used, e.g. $F_{XX} = \partial^2 F/\partial X_j\,\partial X_l$. The block entries above can be readily expanded as $L_{u,u} = I_{u,u} + \lambda^T F_{u,u}$, $L_{u,X} = I_{u,X} + \lambda^T F_{u,X}$, etc. The variables $\hat{\lambda}$ can be eliminated from this system, resulting in the following equivalent system:

$$H\delta u = -G^T, \tag{8}$$

$$F_u \delta u + F_X \delta X = -F(X, u), \tag{9}$$

where

$$H = I_{uu} - Q^T I_{Xu} - I_{uX}Q + Q^T I_{XX}Q + \lambda^T F_{uu} - Q^T\lambda^T F_{Xu} - \lambda^T F_{uX}Q + Q^T\lambda^T F_{XX}Q \tag{10}$$

and

$$G^T = (I_u - I_X F_X^{-1} F_u)^T - (I_{uX} + \lambda^T F_{uX})F_X^{-1}F - F_u^T F_X^{-T}(I_{XX} + \lambda^T F_{XX})F_X^{-1}F. \tag{11}$$

The matrix $H = \mathrm{d}^2 I/\mathrm{d}u_j\,\mathrm{d}u_i$ is often called the reduced Hessian and G is a modified form of the reduced gradient. The first term in the formula for G^T is the transpose of the reduced gradient given by Equation (5) above. The last two terms can be neglected if $F_{\bar{X}}^{-1}F$, i.e. the Newton step for the state equations, is small. If H is positive definite, solving Equation (8) is equivalent to minimizing the quadratic functional:

$$QI(\delta u) = \frac{1}{2}(\delta u)^T H(\delta u) + G(\delta u) + I_0. \tag{12}$$

It is possible to impose any inequality or equality constraints involving either X and/or u at this stage.

If superscripts are used to denote the grid number, the optimization method in TRANAIR can be described as follows:

1. Given the initial design variables u^1 and an initial guess for the state variables $\hat{X}^0$.
2. For each grid $l = 1, 2, \ldots, NG$:
   (a) Discretize the state equations (the flow equations) on grid l.
   (b) Solve this discrete problem $F^l(X^l, u^l) = 0$ approximately for X^l using initial values derived from $\hat{X}^{l-1}$.
   (c) Calculate an approximation to the reduced gradient vector G and the reduced Hessian matrix H and solve the quadratic program given by Equation (12) for δu^l. Constraints on either flow or design variables can be applied at this stage.
   (d) Update the design variables, $u^{l+1} = u^l + \epsilon^l \delta u^l$ where ϵ^l is determined by a globalization strategy. Update the flow variables by solving Equation (9) assuming $F^l = 0$, i.e. $\hat{X}^l = X^l + \epsilon^l (F_X^l)^{-1} F_u^l \delta u^l$. The globalization procedures discussed in Section 3 may give different values of $\hat{X}^l$.
   (e) Estimate the discretization error using $\hat{X}^l$.
   (f) Use the error estimate to determine grid $l+1$ and go to (a) above.
3. Reloft the geometry using the final value of the design parameters u^{NG}. Determine the new values of the design variables u. If the final transpiration $T^{NG}(X^{NG}, u^{NG})$ is not sufficiently small or if the design space was changed, go to step 1 above.

The outer loop (step 3) enables convergence to the solution of the continuous design problem with some level of artificial dissipation. This outer loop involves relofting the geometry (actually moving the surface), doing a surface discretization (called paneling), and then redesigning until the transpiration error becomes insignificant. In the authors experience, convergence of this outer loop has taken at most two relofts of the geometry. Step 2 above is an inexact Newton method [8] for solving the necessary conditions for optimality and allows the incorporation of solution adaptivity into the optimization process. Below, the various approximations used for the reduced gradient and Hessian are described. In this context, approximations to λ and Q can be substituted in the formulas derived above, with only marginal effects on global convergence [5].

Step 2(b) is not strictly needed for the solution of Equations (8) and (9) but does enable dropping the $F(X, u)$ from the right-hand-side from Equation (9) and dropping the last two terms in Equation (11) for G. For highly non-linear flow regimes, this step is often necessary to prevent divergence as discussed in the next section. Step 2(c) above requires linearization of any flow or design constraints, $c(X, u)$. Assuming that the sensitivities $\partial X_i/\partial u_j$ are known, this is done using the reduced gradient formula given in the first part of Equation (5):

$$\frac{\mathrm{d}c}{\mathrm{d}u_i} = \frac{\partial c}{\partial u_i} + \sum_{k=1}^{n} \frac{\partial c}{\partial X_k} \frac{\partial X_k}{\partial u_i}. \tag{13}$$

Approximate Hessian information can be introduced in several ways. In the case of a least-squares objective function, the normal matrix involves only the sensitivities. Using this matrix to approximate the Hessian gives rise to a linear least-squares problem. This method is called the Gauss–Newton method and details are given in [4,7]. The resulting least-squares problem is currently solved with the package LSSOL [9]. In the case of a non-least-squares objective function, the quadratic programming approach suggested above is not yet fully implemented. Instead, an approximate optimization problem is solved that assumes a linearized flow velocity $\vec{v}$. It is assumed that I depends on the flow only through $\vec{v}$ and the function J given by

$$J(u) = I\left(\vec{v}^0 + (u - u^0)\frac{\partial \vec{v}}{\partial u}\right), \tag{14}$$

where $\partial \vec{v}/\partial u$ is computed using Equation (13), is minimized. The gradient of J is the same as the gradient of I. However, the Hessian of J is given by the first four terms of Equation (10). The minimization is currently accomplished by using the optimization package NPSOL [10]. Since J is inexpensive to evaluate, gradients can be computed by finite differences. The result is that second-order information corresponding to the first four terms of Equation (10) is generated implicitly in a BFGS update strategy, for which the evaluations are very inexpensive allowing a large number of iterations for each subproblem.

The storage for the sensitivities might seem at first glance to be prohibitive for large numbers of design variables. However, almost all aerodynamic optimization problems involve objective functions and constraints that only depend on the velocity at points on the surface of the configuration. Advantage is taken of this fact by storing $\partial \vec{v}/\partial u$ only for these surface points. The computational cost of the sensitivities can be somewhat ameliorated by using parallelism and by the use of block GMRES [11].

Now the adjoint formulation [12] for optimal control is discussed and compared with the sensitivity method. The two methods arise from the observation that the matrix product given in Equation (5) for the reduced gradient can be computed in two ways. If the multipliers are computed first, the method is usually referred to as an adjoint method. Partial second-order information is easily available in the sensitivity method in the form of the first four terms of Equation (10). In the adjoint method, no second-order information is immediately available and additional adjoint solvers are required. In fact, if a full second-order method is desired, the apparent cost advantage of the adjoint method disappears. Further, with the sensitivity method, linearizations of all constraints are available immediately. Good convergence has been observed for this optimal control method; probably due to the implicit use of the second-order information available in a sensitivity method. As discussed above, this information is generated implicitly in a BFGS update strategy, for which the evaluations are very inexpensive, allowing a large number of iterations for each subproblem.

TRANAIR design usually converges to a reasonable level of the reduced gradient in five to ten optimization subproblems, each of which is less costly than a single analysis. Because of the efficient generation of sensitivities, the total CPU cost for up to 400 design variables in three space dimensions is usually less than the cost of ten analyses. This is at least two orders of magnitude less than comparable black box methods.

3. GLOBALIZATION

All Newton-based methods either for analysis or optimization require some kind of globalization method to prevent divergence in the case of a poor initial approximation to the solution. There is a plethora of such methods in the literature; many of these methods are classical, such as the Levenberg–Marquardt method. In the current optimization method, step size control is sometimes achieved through user-specified constraints that bound the change in some aerodynamic quantity. In design, as in analysis, changes in local Mach number have proven to be effective. There is some reason based on experience in analysis cases to believe that convergence would be enhanced by some type of a more automatic step size control for the step in the design variables once the direction is determined by solving the optimization subproblem.

Recently, several options have been implemented in TRANAIR to choose the step size in the design variables. These options involve two levels of inexactness and are described below.

Line search method 1

Choose ϵ^l to minimize $I(\hat{X}^l, u^l + \epsilon^l \delta u^l) + \Phi_M$, where Φ_M is a merit function and where $\hat{X}^l$ is determined by solving to some level of accuracy, the state equation $F(\hat{X}^l, u^l + \epsilon^l \delta u^l) = 0$. In this case, $X^{l+1} = \hat{X}^l$ can be set and step 2(b) can be skipped in the overall method given in the previous section.

This modified line search maintains the satisfaction of the state equations to some level of accuracy for each change in the design variables considered. A minor variation is possible by partitioning the state equations into two subsets F_1 and F_2 and the state variables into corresponding subsets X_1 and X_2 such that given X_2, one can solve F_1 for X_1. Then one could apply non-linear elimination to F_1, resulting in a somewhat modified line search. In the current case, F_1 is the boundary layer equations and X_1 the boundary layer variables, while F_2 are the inviscid equations and X_2 the inviscid variables [1,3].

Line search method 2

Choose ϵ^l to minimize $I(\hat{X}_1^l, X_2^1 + \epsilon^l \delta X_2^1, u^l + \epsilon^l \delta u^l) + \Phi_M$, where $\hat{X}_1^l$ is determining by solving to some level of accuracy, the state equation $F_1(\hat{X}_1^l, X_2^l + \epsilon^l \delta X_2^l, u^l + \epsilon^l \delta u^l) = 0$. Set $X_1^{l+1} = \hat{X}_1^l$, $X_2^{l+1} = X_2^l + \epsilon^l \delta X_2^1$ and $u^{l+1} = u^l + \epsilon^l \delta u^l$.

There are many popular choices for the merit function Φ_M, which is designed to take into account constraint violations [7,13]. Line search method 2 maintains feasibility only for F_1. As discussed below, there are 'inexact' versions of both line search methods that are probably more efficient in practice.

One of the motivations for the introduction of these globalization methods was the possibility that a design step could result in a transpiration on the next grid for which the boundary layer non-linear solution process would not converge. This did in fact happen in several cases and had to be mitigated by user-specified constraints. In this section, the results of applying these methods to two non-linear least-squares problems are discussed. The first is an inviscid pressure matching case and the second is a similar case in viscous flow.

The first test case is a pressure matching case involving a target pressure distribution obtained by analyzing an airfoil section similar to the ONERA M6 section. The initial airfoil was the NACA 0012. The flight conditions were $M_\infty = 0.75$ and $\alpha = 1.0$. The objective function is

$$I = \frac{1}{2} \sum_{k=1}^{q} W_k[(c_p)_k - (\hat{c}_p)_k]^2,$$

with the weights W_k taken to be the surface point in question and $(c_p)_k$ are pressure coefficients. Figure 1 shows the initial geometry, target geometry and final geometry resulting from the design and also the initial pressure distribution, target pressure distribution and final pressure distribution on the designed airfoil. The difference between the target and the designed airfoil is due to small errors in the transpiration approximation.

In this case, a large number of grids was used to examine asymptotic convergence issues. For this case, the grid was frozen after grid 10. The case was run with no globalization, with line search method 1, and with line search method 2. In the case of no step size control, there is a dramatic tendency to 'overshoot' the actual solution and great difficulty is encountered in converging the analysis on grid 8 due to the large change introduced by the undamped optimization step. The code eventually stops because of convergence stagnation on this grid. Line search method 2 is rather simple since there is no boundary layer in this case. The step size selection is based on a linear estimate of the change in local Mach number with a value greater than 0.4 being considered an error. This results in step sizes that are initially too conservative. Once the shock wave becomes established, it takes many grids to move it to the correct position. Line search method 1 is the most effective, finding the solution well before the grid is frozen. This is due to the better estimation of the true solution on the next grid, taking into account the optimization step.

Another globalization strategy that can prove effective is to impose constraints on the optimization subproblem that restrict quantities such as changes in local Mach number. Such constraints are effective but require user knowledge of the problem and are less flexible than the globalization strategies discussed in this section. For example, a user-imposed constraint on the change in local Mach of 0.4 would result in a convergence history similar to that reported for line search method 2 given above.

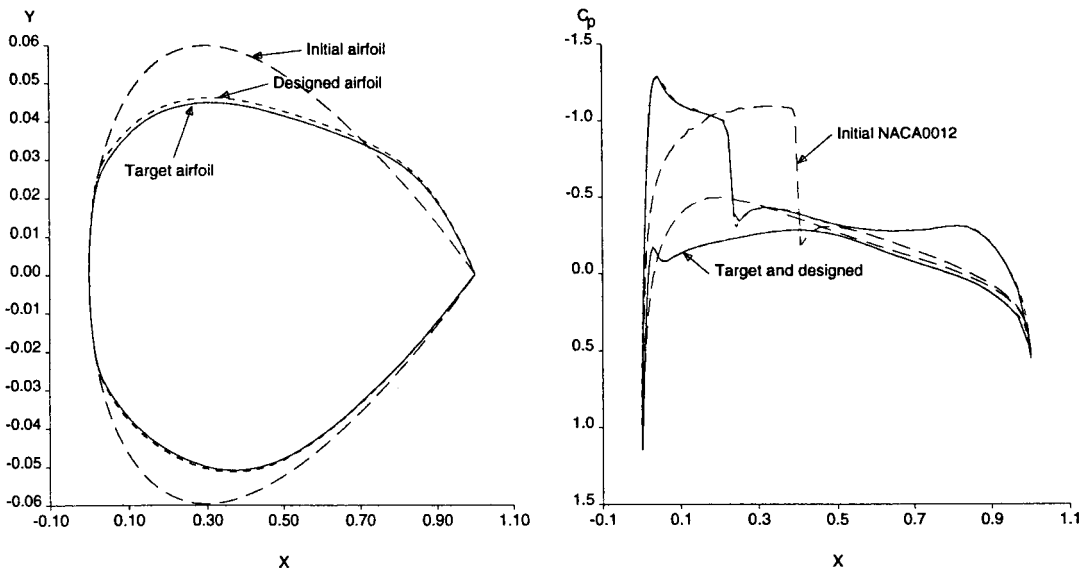

Figure 1. Geometry and solution for design least-squares case.

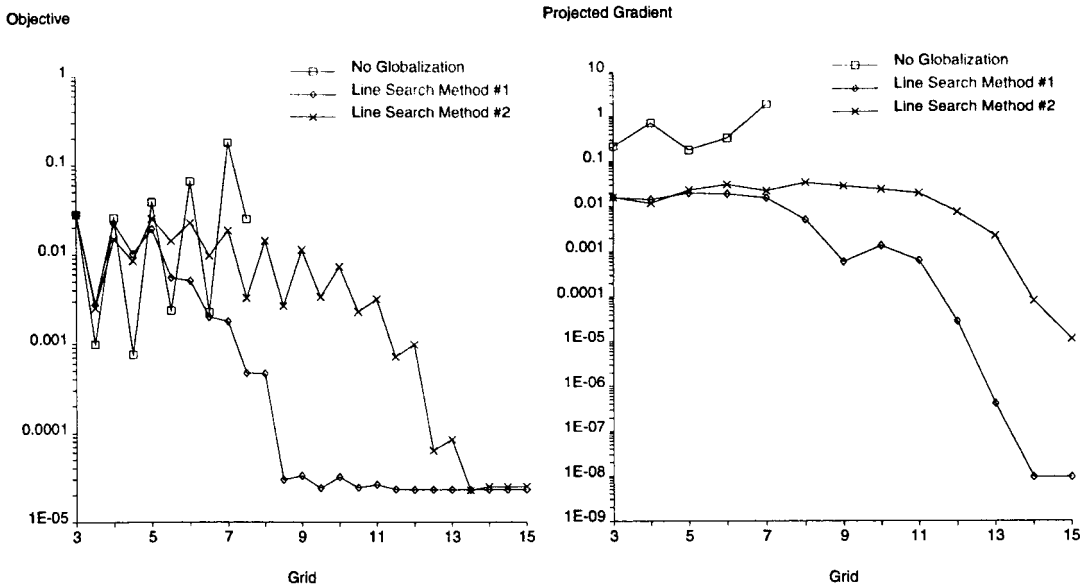

Figure 2. Convergence of design least-squares case with various globalization options.

Figure 2 shows the convergence history for this case. Both the objective function and the estimated projected gradient are shown as functions of the grid number. There are two values of the objective shown for each grid. The first results from the analysis of a fixed transpiration due to the initial values of the design variables on that grid. The second value is the best estimate of the value after the updating of the design and flow variables on that grid. Severe oscillation indicates that the estimated values of the objective are seriously inaccurate. This can be due to a design step that is beyond the range of validity of the model of the objective used in the optimization subproblem. On early grids, it can result from grid-to-grid changes in the solution due to solution adaptive grid refinement.

A second test case illustrates the globalization methods as applied to a viscous flow case formulated using the same airfoils as before. In this case, the effect of line search method 1 is more significant. With no step size control, lack of convergence occurs on grid 5. The same is true of line search method 2, even though the step sizes are reduced considerably. With line search method 1, the design converges. The convergence histories are given in Figure 3.

4. MULTIPOINT DESIGN

The TRANAIR design and optimization method described in Section 2 has been extended to optimization at multiple flow conditions. Multipoint design is desirable for at least two reasons. First, it is obvious that a single-point design has no control of the performance at other conditions (except through constraints that may be *ad hoc* or difficult to formulate) and in practice, it often leads to poor off-design performance. Second, there is hope that multipoint design will be more stable than single-point design; for example, in minimizing drag for an airfoil, there are many shock-free single-point airfoil designs.

If the objective function is drag, then the goal is to minimize the drag at several different operating conditions. Currently, the objective is taken to be a weighted linear combination of the objectives from the different flow conditions. Thus, the optimization problem is formulated as minimize $\Sigma\, W_{ip} I_{ip}$ such that $F_{ip}(X_{ip}, u) = 0$ and $c_{ip}(X_{ip}, u) = 0$ for $ip = 1, \ldots, np$, where np is the number of flow conditions in the multipoint design. Here, $F_{ip} = 0$ represents solving the flow equations for the ipth condition and $c_{ip} = 0$ represents the geometry and flow constraints imposed on the ipth condition.

The algorithm for multipoint design is very similar to that described in Section 2. Step 1 is to give each condition the initial geometry (i.e. values for the design variables). Steps 2(a) and (b) can proceed in parallel and are completely independent. In fact, different computers have been used to work on different conditions. Steps 2(c) and (d) require an optimization that must get input from each of the flow conditions. The objective function is assembled, and the constraint and sensitivity information combined to form one large optimization problem. The resulting updates to the design variables are passed back to each of the flow conditions and steps 2(e) and (f) can again proceed independently for each flow condition.

Below, the results of applying single-point and multipoint optimizations to the problem of minimizing drag for a 2D airfoil are shown. The drag for these 2D airfoils is computed by summing the wave drag and the profile drag. The wave drag is calculated by identifying the shock surface, and integrating over the surface using the shock jump conditions with the normal Mach number obtained from the upstream head of the shock. The profile drag is computed by using the Squire–Young formula.

The optimization problem is to minimize the drag on the RAE 2822 airfoil. The 'cruise' condition is taken to be free-stream Mach number, $M_\infty = 0.725$, Reynolds number, $Re = 6.5 \times 10^6$, coefficient of lift, $C_L = 0.73$. Also computed are solutions for other values of M_∞ with the same C_L and Re. Figure 4 shows the resulting drag rise curve for this airfoil.

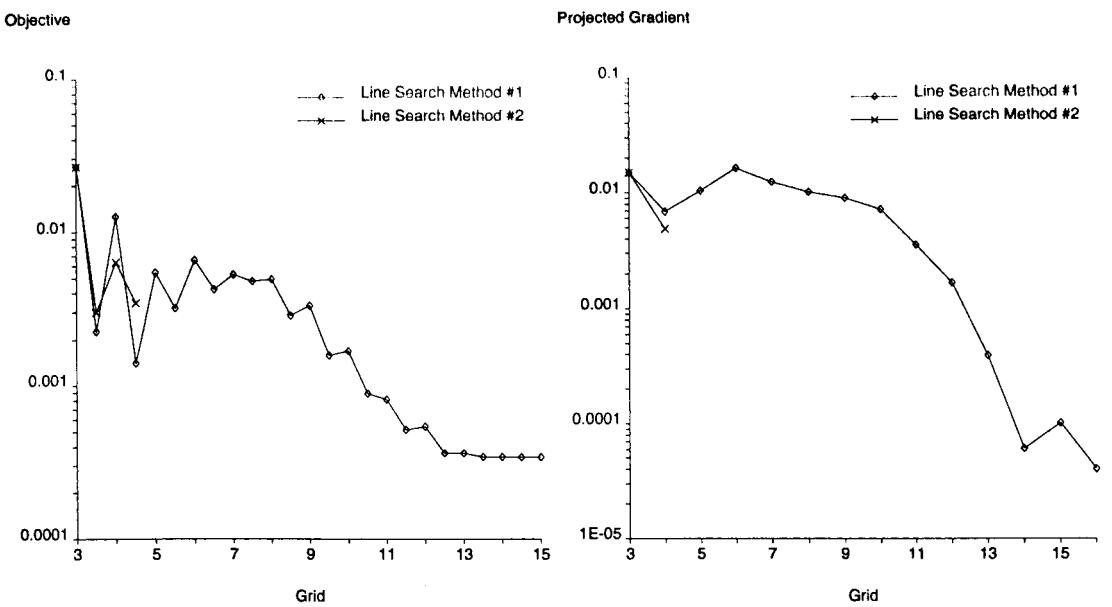

Figure 3. Convergence of design least-squares case with various globalization options for viscous design case.

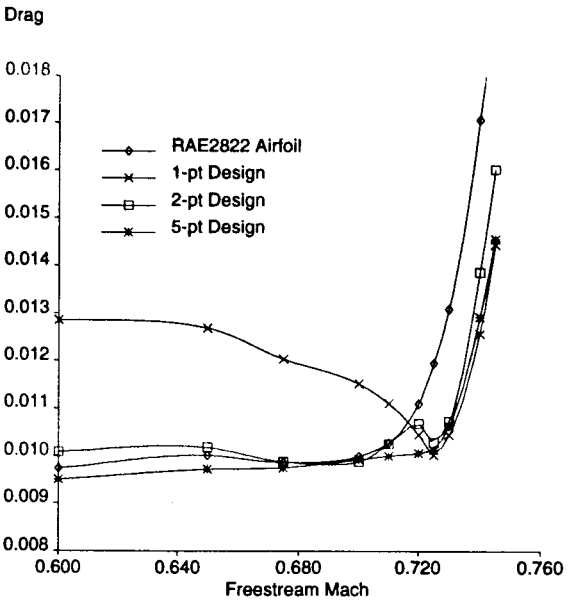

Figure 4. Drag rise curves for RAE 2822 airfoil and airfoils designed without curvature constraint.

For all the designs, the geometry was only permitted to change in camber with thickness held fixed. The airfoil is modified by a spline interpolation of changes in camber parameterized by variables defined at points every 10% of the chord of the airfoil. In addition, the angle of attack was allowed to vary in order to match the specified lift. The adaptive gridding strategies used for analysis and design were similar. Starting from a very coarse initial grid, the grid was adaptively refined eight times to a total of about 10000 grid boxes. For the design runs, an additional two steps with no change in grid were made to allow the optimization two additional steps in which to fine tune the changes in geometry. In addition to the lift constraint, the change in transpiration from grid to grid was constrained to be less than 0.02 to discourage the optimizer from taking steps that were too large, especially when the grid was coarse. This is needed because the globalization methods described above are not yet implemented in the multipoint capability. Also, the kinematic shape parameter H_k was constrained to ensure attached flow in the turbulent flow region. After the new geometry is generated, it was then re-analyzed to produce the curve shown in Figure 4. It is noted that in all cases, the design predictions of drag were in excellent agreement with those computed in the re-analysis, i.e. the transpiration error was very small. In the case of single-point design, it can be seen that while the drag is significantly reduced at the design point, the performance is much worse at the lower Mach number conditions. Next, a two-point calculation was made with $M_\infty = 0.700$ and $M_\infty = 0.725$ and the objective function chosen to minimize the sum of the two drags. This improves the performance around $M_\infty = 0.700$, but between the design points and lower than $M_\infty = 0.700$, the performance is still poor. Next, results of a five-point design with $M_\infty = 0.650$, 0.700, 0.710, 0.720, 0.725 are shown. Again, the conditions were equally weighted. In this study, the choice of weights has not been investigated. Now, Figure 4 shows a monotonic drag rise curve with significantly improved cruise performance. Figure 5 shows the original (RAE 2822) airfoil and the five-point design airfoil (labeled without curvature constraints).

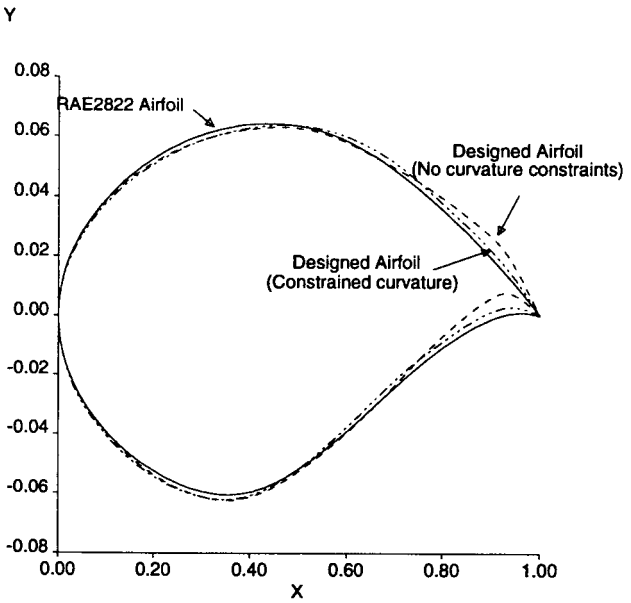

Figure 5. RAE 2822 airfoil and airfoils designed with and without curvature constraint.

The designed airfoil shows quite large curvature near the back-end and also has a small amount of reverse curvature on the upper surface. It is suspected that the large curvature may be caused by the optimizer trying to exploit a deficiency in the boundary layer code.

In an effort to remedy these problems, the designs were repeated with pointwise curvature constraints applied. The upper surface was forced to have the same sign of curvature

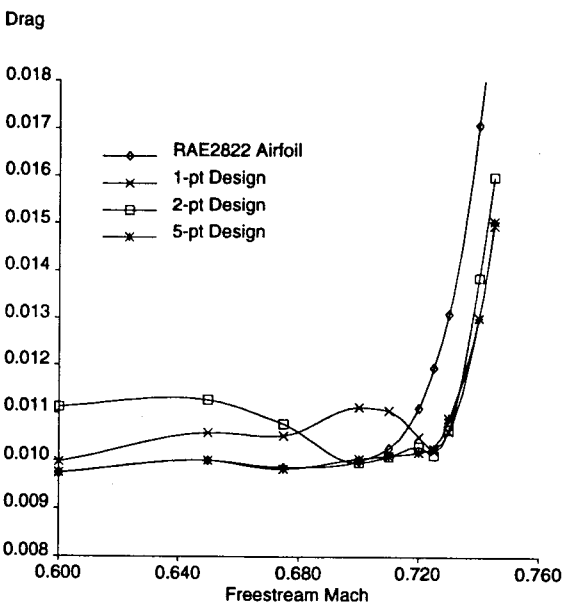

Figure 6. Drag rise curves for RAE 2822 airfoil and airfoils designed with curvature constraint.

everywhere and the curvature was bounded in the aft 50% of chord. Figure 6 shows the drag rise curves of the resulting designs.

Once again the one- and two-point designs exhibit poor off-design behavior, although in this case, the curvature constraints have reduced the movement so the behavior is not as bad as when unconstrained. Again, the five-point design produces a good overall design. In this case, the drag rise curve now resembles the original RAE 2822 for low Mach numbers, but has significantly reduced drag for higher Mach numbers. Note in Figure 5 that the constrained design produces an airfoil that has a much more acceptable shape. In this study the thickness distribution was held fixed. One could expect further improvements by, for example, allowing thickness to change but constraining the volume of the airfoil.

5. SUMMARY

The authors have improved the single-point design and optimization methodology in TRANAIR to include a range of globalization strategies and extended it to a multipoint design capability. Both enhancements improve the robustness and usefulness of the capability. They have offered a rationale for preferring the sensitivity method to the adjoint method by showing that some second-order information is naturally and inexpensively available in the sensitivity method.

REFERENCES

1. M. Drela, 'Design and optimization method for multielement airfoils', *AIAA Paper 93-0969*, 1993.
2. D.P. Young, R.G. Melvin, M.B. Bieterman, F.T. Johnson, S.S. Samant and J.E. Bussoletti, 'A locally refined rectangular grid finite element method: application to computational fluid dynamics and computational physics', *J. Comp. Phys.*, **92**, 1–66 (1991).
3. M.B. Bieterman, R.G. Melvin, F.T. Johnson, J.E. Bussoletti, D.P. Young, W.P. Huffman, C.L. Hilmes and M. Drela, 'Boundary layer coupling in a general configuration full potential code', *Tech. Rep. BCSTECH-94-032*, Boeing Computer Services, 1994.
4. W.P. Huffman, R.G. Melvin, D.P. Young, F.T. Johnson, J.E. Bussoletti, M.B. Bieterman and C.L. Hilmes, 'Practical design and optimization in computational fluid dynamics', *AIAA Paper 93-3111*, 1993.
5. D.P. Young, W.P. Huffman, R.G. Melvin, M.B. Bieterman, C.L. Hilmes and F.T. Johnson, 'Inexactness and global convergence in design optimization', *AIAA Paper 94-4386*, 1994.
6. A.E. Bryson and Y.C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*, Hemisphere, New York, 1975.
7. R. Fletcher, *Practical Methods of Optimization*, 2nd edn., Wiley, New York, 1987.
8. R.S. Dembo, S.C. Eisenstat and T. Steihaug, 'Inexact Newton methods', *SIAM J. Numer. Anal.*, **19**, 400–408 (1982).
9. P.E. Gill, S.J. Hammerling, W. Murray, M.A. Saunders and M.A. Wright, User's guide for LSSOL (version 1.0): a FORTRAN package for constrained linear least-squares and convex quadratic programming, *Stanford University Technical Report*, Department of Operations Research, 1986.
10. P.E. Gill, W. Murray, M.A. Saunders and M.A. Wright, User's guide for NPSOL (version 4.0): a FORTRAN package non-linear programming, *Stanford University Technical Report SOL86-2*, Department of Operations Research, 1986.
11. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1995.
12. A. Jameson, 'Aerodynamic design via control theory', *J. Sci. Comput.*, **3**, 233–260 (1988).
13. P.E. Gill, W. Murray and M.H. Wright, *Practical Optimization*, 1st edn., Academic Press, San Diego, 1981.

