Nonlinear Elimination in Aerodynamic Analysis and Design Optimization

D. P. Young, W. P. Huffman, R. G. Melvin, C. L. Hilmes, and F. T. Johnson

The Boeing Company, P. O. Box 3707, M/S 7L-21, Seattle, WA 98124-2207

Abstract. Recent emphasis on reduction of design cycle time and cost in the design of commercial aircraft has sparked a renewed interest in design optimization in aerodynamics, structures, and aeroelastics. The constrained aerodynamic optimization problem is closely related to the problem of solving nonlinear systems of equations. In applying Newton's method to steady-state compressible CFD analysis problems, the nonlinear elimination method has been remarkably successful. In this paper we consider the implications of this experience for design optimization formulations in the general case of state equation equality constraints. This relationship between nonlinear equation solving and design optimization is illustrated by drawing on computational examples from the TRANAIR compressible CFD code. We first discuss various formulations of the PDE constrained optimization problem related to the Lagrange Newton method and the multiplier free version implementation in TRANAIR. We then discuss the nonlinear elimination method and its application to a simple nozzle problem. This method is then applied to derive various globalization methods in design optimization which are illustrated by a computational example in airfoil design. Finally, we discuss some remaining limitations and issues.

1 Introduction

Today, commercial aircraft are routinely designed using a combination of Computational Fluid Dynamics (CFD) and wind tunnel testing [24]. One historically successful 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 attempt to compute the geometry which produces such flow features. Using inverse design, the many cut-and-try iterations of repetitive analysis are replaced by a systematic computational 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. A 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 at the cruise condition will likely give reasonable off-design characteristics. A third difficulty is the indirect way in which inequality constraints on the geometric surface (arising for example from manufacturing or structural requirements) must be imposed.

These difficulties lead to the desire to develop an optimization methodology for aerodynamic design. Most of the methods for aerodynamic optimization are variations of classical methods for optimal control [3] in which special consideration must be given to computational costs and the mathematical difficulties inherent in compressible viscous analysis and design. A well formulated optimization method may help to quickly achieve a good compromise between aerodynamic or economic cost objectives and the constraints imposed on the geometry by other disciplines such as manufacturing and structures. It will also allow us to consider concurrently multiple critical flight conditions, so that iteration with off-design considerations can be reduced or eliminated. In the past few years, we have developed an optimization capability in the TRANAIR code [1, 16, 19, 27, 28]. TRANAIR is a two- and three-dimensional full potential flow code with a directly coupled boundary layer capable of handling complex geometries through the use of Cartesian grids and solution adaptive local grid refinement.

This paper discusses some of the algorithmic issues dealt with in building the TRANAIR aerodynamic optimization capability. In section 2 we discuss some of the theoretical difficulties encountered in aerodynamic optimization. In section 3 we will give an overview of optimization formulations for PDE constrained problems and in section 4 describe the methods implemented in the TRANAIR design code. In section 5 we will describe various globalization strategies motivated by nonlinear elimination and demonstrate their effectiveness with some examples. In section 6 we will describe how the optimization method is extended to multi-point design. In section 7 we consider some remaining technical issues involving robust convergence and identification of ill-posed problems.

2 Mathematical Difficulties

It has been known for over 40 years that the problem of transonic flow past an airfoil is an ill-posed problem in certain circumstances [22, 23]. Small changes in a nearly flat portion of an airfoil can cause a supersonic zone with a shock wave to appear or disappear. This is usually avoided in engineering practice since the resulting real flow is likely to exhibit undesirable unsteady oscillations. This difficulty is not remedied by using higher order mathematical modeling, e. g., the Euler equations [18]. The Navier-Stokes equations are also well known to exhibit multiple solutions as the Reynolds number is increased. As might be expected, design optimization tends to drive the design towards these impractical configurations unless constrained from doing so. It is also well known that in viscous flow, separation is a highly nonlinear and sensitive phenomena. In a boundary layer, small changes in either the airfoil shape or the outer inviscid flow can result in large changes in the boundary layer. In fact, it is questionable whether steady state solutions exist at all for the Navier-Stokes equations when the flow is separated. Further, turbulence modeling is essential to achieving realistic results for high Reynolds number flows and the dependence of the flow on these models is very nonlinear. In addition, an airframe must have acceptable performance everywhere in the flight envelope to be viable. Once this is realized, it is apparent that the aerodynamic design problem must consider at least a reasonable number of different flight conditions even if only through inequality constraints, e.g., it has been observed that controling the peak Mach number of a wing at cruise flight conditions during design will usually result in acceptable aerodynamic characteristics at higher speeds or lifts. In practice, multi-point optimization is required which increases computational cost and flow time [9].

3 Optimization Formulations

The TRANAIR code is a compressible CFD code designed to model arbitrary geometries robustly [27]. It uses a locally refined Cartesian grid that does not conform to the surface of the body, solution adaptive grid refinement, and a Newton-Krylov solution algorithm. General geometry capability and Newton-Krylov solution methodology have generally been thought to be helpful when constructing a PDE-constrained optimization code. However, it has generally been thought that solution adaptivity is an impediment because it is a source of "noise" in the objective when a black box optimization approach is employed. A further simplifying methodology utilized in TRANAIR is the use of transpiration boundary conditions to simulate the effect of actually moving the surface. This eliminates the necessity of computing sensitivities of the grid itself with respect to the design variables. One of the consequences of the use of Newton's method is the experience with inexactness of derivatives that carries over almost directly to the optimization case [28]. In addition, we will argue below that the use of a sensitivity formulation enables the implicit and cost effective generation of partial second order information.

TRANAIR discretizes and solves the flow problem on a series of grids, $l = 1, 2, \ldots, NG$. We have combined our design and optimization capability 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. For simplicity, in the following description of optimization methods, we will deal only with the discrete version of the problem which is posed on a given grid.

The discretized nonlinear 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 freestream angle of attack or Mach number. Included in $F(X,u)$ is an inhomogeneous Neumann boundary condition $T(X,u)$ used to approximate the effect of boundary motion [16]. One simple form of this condition is $T(X,u) = W \cdot \delta\hat{n}$ where W is the mass flux through the boundary and $\delta\hat{n}$ is the total change in unit normal due to the design parameters u.

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

The necessary conditions for optimality are often formulated by introducing the Lagrange multipliers $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)^T$ as independent variables [3]. 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., $\lambda_i = dI/dF_i$. 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 multipliers 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), we obtain the following equivalent necessary conditions for optimality:

$$\frac{dI}{du} \equiv \frac{\partial I}{\partial u} - \left(\frac{\partial I}{\partial X}\right)\left(\frac{\partial F}{\partial X}\right)^{-1}\left(\frac{\partial F}{\partial u}\right) = 0 \tag{5}$$

$$F(X,u) = 0.$$

The quantity dI/du 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 by m matrix

$$Q \equiv \frac{dX_i}{du_j} = -\sum_{k=1}^{n} \left(\frac{\partial F_k}{\partial X_i}\right)^{-1} \left(\frac{\partial F_k}{\partial u_j}\right).$$

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 $\partial F/\partial u$ for those u's that represent surface geometry, as described in [16]. Equation (5) gives the gradient of I assuming F is held fixed. It does not, however, tell us how to balance decreasing the value of I against making F small. Such a balance can be derived by examining the classical Lagrange-Newton method [11]. Applying Newton's method to equations (1), (2), and (3), we arrive at the following equations for the updates δu to u, δX to X, and $\delta\lambda$ to λ:

$$\begin{bmatrix} L_{u,u} & L_{u,X} & F_u^T \\ L_{X,u} & L_{X,X} & F_X^T \\ F_u & F_X & 0 \end{bmatrix} \begin{bmatrix} \delta u \\ \delta X \\ \hat{\lambda} \end{bmatrix} = - \begin{bmatrix} I_u^T \\ I_X^T \\ F \end{bmatrix} \tag{6}$$

where $\hat{\lambda} = \lambda + \delta\lambda$ and we have used the subscript notation for partial derivatives, 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. Block row operations on (6) yield the following equivalent system:

$$H\delta u = -G^T \tag{7}$$

$$F_u \delta u + F_X \delta X = -F(X,u) \tag{8}$$

$$\hat{\lambda} = F_X^{-T}[-I_X^T - L_{X,u}\delta u - L_{X,X}\delta X] \tag{9}$$

where

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

and

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

The matrix $H = d^2 I/du_j du_i$ 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 $F_X^{-1}F$, i.e., the Newton step for the state equations, is small. In any case, dropping these terms reduces convergence to two step superlinear [11]. If H is positive definite, solving equation (7) 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 when solving this quadratic program.

3.1 Block Lagrange Newton Method

One algorithm implementing the Lagrange Newton method is just a block solution method for equation (6). Given the current values for all the variables, first compute H and G from equations (10) and (11) using the current values of the Lagrange multipliers. The step in the design variables δu is obtained by minimizing QI as given by equation (12) subject to any inequality constraints. The step in the state variables is obtained by solving equation (8) for δX. Equation (9) can now be solved for $\hat{\lambda}$, the new value of the Lagrange multipliers. This completes the computation of the Newton direction. This block elimination method gives the same Newton direction as solving the large sparse system (6). Thus, the choice of whether to do the block elimination or solve the large system directly is purely a linear algebra question. The sparse approach has been advocated by Ghattas and Biros (this proceedings) as well as by Betts (this proceedings). In the sparse approach, the sensitivities are not required and the Hessian of the Lagrangian can be computed by second differences of L itself, for example. A preconditioning method for the full matrix based on a BFGS estimate of the reduced Hessian has been implemented by Biros and Ghattas [2]. This strategy will only succeed if the large linear system can be solved subject to any inequality constraints. In any Newton method for solving nonlinear systems of equations and in most optimization methods, some form of step length control is required to stabilize the algorithm. Stabilization strategies can range all the way from simple line searches to sophisticated trust region methods. These strategies are called globalizations and can have a dramatic impact on convergence as discussed in section 5.

3.2 Reduced Space Method

The Lagrange Newton method makes no assumptions about the residuals for the state equations except at convergence. This is tied up in the introduction and updating of the Lagrange multipliers. If the state equations are solvable for X for all reasonable values of u, one can implicitly eliminate the state variables from the formulation and merely minimize as a function of u the objective $I(X(u), u)$. In this formulation, the gradient is given by equation (5) and the Hessian is given by equation (10) where λ is given by equation

(4). It is assumed that $F(X, u) = 0$, i.e., that X is always updated whenever a new value of u is considered by the optimization algorithm. This reduced space method is a nonlinear elimination method applied to the Lagrange Newton system of equations and its performance will be problem dependent as discussed below.

3.3 Lagrange Newton Method Without Multipliers

It is possible to construct a simplified Lagrange Newton method without explicitly computing or updating the Lagrange multipliers by using certain simplifying assumptions. First, assume that the sensitivities Q are available. Second, if F_{XX}, F_{Xu}, F_{uX}, and F_{uu} can be neglected, the Hessian matrix H can be computed without reference to λ. This would be the case, for example, if the state equations were linear or nearly linear in both the state and control variables. The second and third terms in equation (11) can be neglected degrading convergence from quadratic to two step superlinear. In any case, if our globalization always causes F to be near zero, the terms can be neglected. In either case, G does not explicitly require computation of λ. Now equations (7) and (8) constitute a Newton like method that does not require the explicit computation or updating of the Lagrange multipliers. In TRANAIR, we substitute a more global nonlinear program for the quadratic program and never explicitly compute H, allowing an optimization code to generate an approximation using BFGS. It is an open question whether these simplifying assumptions increase the importance of globalization. Certainly, without multiplier estimates, it is difficult to see how to implement the full Lagrange Newton method. This is a question that deserves further study. Since the question is independent of the linear algebra required to solve the Lagrange Newton system, the question is most naturally studied in the context of a reduced method as given by equations (7), (8), and (9). We suggest an advantage of the sensitivity form over an adjoint form of the reduced gradient in the next section.

4 TRANAIR Solution Adaptive Method

Currently, in TRANAIR a variant of the Lagrange Newton method without multipliers is implemented. If we let superscripts denote the grid number, this method 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 $\hat{X}^{l-1}$ and calculate the sensitivities Q.

(c) Solve the optimization subproblem discussed below for δu^l. Constraints on either flow or design variables can be applied at this stage.
(d) Determine the steplength ϵ^l using some globalization procedure and update the design variables, $u^{l+1} = u^l + \epsilon^l \delta u^l$.
(e) Update the flow variables by solving equation (8) 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 below in section 5 may give different values of $\hat{X}^l$.
(f) Estimate the discretization error using $\hat{X}^l$.
(g) 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 our experience, convergence of this outer loop has taken at most two relofts of the geometry. Step 2 above is an inexact Newton method [7] for solving the necessary conditions for optimality on a fixed grid and allows the incorporation of solution adaptivity into the optimization process. Below we describe the various approximations used for the reduced gradient and Hessian.

Step 2(b) is not strictly needed for the solution of equations (7) and (8) but does enable dropping the $F(X, u)$ right hand side from equation (8) and dropping the last two terms in equation (11) for G. However, as pointed out above, if it is not carried out, it might be advisable to include Lagrange multiplier updates in the method and include the additional terms in our calculation of G and H. For highly nonlinear flow regimes, step 2(b) may be necessary in any case to prevent divergence as discussed in section 5. Step 2(c) above requires linearization of any flow or design constraints, $c(X, u)$. Assuming that the sensitivities dX_i/du_j are known, this is done using the reduced gradient formula given in equation (5):

$$\frac{dc}{du_j} = \frac{\partial c}{\partial u_j} + \sum_{k=1}^{n} \frac{\partial c}{\partial X_k} \frac{dX_k}{du_j} \tag{13}$$

Approximate Hessian information can be introduced in several ways. In the case of a least squares objective function (see section 5.4 below), 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 [11, 16]. The resulting least squares problem is currently solved with the package LSSOL [13]. In the case of a non-least-squares objective function, the quadratic programming approach given by equations (7),(8), and (9) is not fully implemented.

Instead, an approximate optimization problem is solved which assumes a linearized flow velocity $\boldsymbol{v}$. We assume that I depends on the flow only through $\boldsymbol{v}$ and actually minimize the function J given by

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

where $d\boldsymbol{v}/du$ is computed using equation (13). The gradient of J at u^0 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 [14]. 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 final active set and the BFGS approximation to the reduced Hessian for the subproblem can be saved to initialize the optimization on the next grid.

4.1 Algorithmic Issues

The storage for the sensitivities might seem at first glance to be prohibitive for large numbers of design variables. However, most aerodynamic optimization problems involve objective functions and constraints that only depend on the velocity at points on the surface of the configuration. We take advantage of this fact by storing $d\boldsymbol{v}/du$ 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 [25].

We now comment on the adjoint formulation [17] for optimal control and compare it to the sensitivity method. The two methods arise from the observation that the matrix product given above 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 4 terms of equation (10). In the adjoint method as usually implemented, no second order information is immediately availability and some acceleration strategy such as BFGS is usually relied on to improve convergence over the rate of steepest descent. In fact, if a full Newton method is desired, the apparent cost advantage of the adjoint method disappears since the reduced Hessian H requires both λ and Q. Further, with the sensitivity method, linearizations of all constraints are available immediately.

We have observed reasonably good convergence of our optimal control method especially for least squares objectives. This is probably due to the implicit use of the partial second order information available in a sensitivity method. In engineering practice, TRANAIR design is usually run for 5 to 10 design grids (each with an optimization subproblem). Because of 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 20 analyses. This is at least 2 orders of magnitude less than comparable black box methods.

However, there are still questions about the convergence of the method particularly in challenging cases. It would be of particular interest to determine the effect on convergence of including the curvature of the state equations (the last four terms in the formula for H above) and/or multiplier estimates in the formulation of the optimization subproblem.

A further interesting question concerns the comparison between the reduced space method and the block Lagrange Newton method. In the latter, the Lagrange multipliers must be explicitly computed at each step and there is controversy about whether overall convergence is enhanced by allowing "cutting across the design space."

4.2 Inexactness

The method outlined above with solution adaptive gridding and sensitivities is reliable in the presence of significant inexactness. This is due to the formulation as an inexact Newton method in which each optimization subproblem is consistent to machine precision. In fact, our sensitivities are computed using a block GMRES method in which the residual is only reduced by a factor of 20. This is exactly the convergence criterion used in the analysis version of the code. In running thousands of cases, we have been unable to find a single instance where this level of inexactness has caused a failure to converge either an analysis problem or a design optimization problem. It is much more difficult to quantify the effect of inexactness in methods such as the adjoint method or the nonlinear conjugate gradient method where a consistent subproblem is never defined.

5 Globalization

As mentioned above, the difference between what is usually called the SAND [15] (or the all at once [6] or the Lagrange Newton method [11]) and the NAND (or the discipline feasible or the reduced sequential quadratic programming (RSQP) or the reduced space method) is really globalization. An excellent discussion of many of the algorithmic issues involved is the work of Biros and Ghattas [2]. Typically, the NAND method merely enforces the state equation constraint at every optimization iteration. Complex optimization methods blur this distinction even further. In reality, there is a spectrum of methods that is liable to defy exact classification in any practical implementation [29]. In order to motivate our discussion of globalization methods in optimization, we first examine the role of globalization of Newton's method in compressible viscous fluid dynamics.

5.1 Nonlinear Elimination

When solving nonlinear systems of equations such as those arising in computational fluid dynamics, convergence can in some cases be dramatically accelerated by using the nonlinear elimination method. This method is classical but was analyzed in [20]. Suppose we want to solve the system of nonlinear equations

$$F(X) = 0 \tag{15}$$

and the system can be divided into two sets of equations F_1 and F_2 one of which might be much more linear than the other. Let us further suppose that the variables can be divided into 2 subsets X_1 and X_2 such that $F_2(X_1, X_2) = 0$ is solvable for X_2 given X_1. In this case, we can define the nonlinear function $h(X_1)$ by this latter solution X_2. We can now define a nonlinear system equivalent to (15) by

$$f(X_1) = F_1(X_1, h(X_1)) = 0. \tag{16}$$

Once (16) is solved for X_1, $F_2(X_1, X_2) = 0$ must be solved for X_2. If Newton's method is applied to (16), the Jacobian matrix is $\frac{df}{dX_1} = \frac{\partial F_1}{\partial X_1} - \frac{\partial F_1}{\partial X_2}(\frac{\partial F_2}{\partial X_2})^{-1}\frac{\partial F_2}{\partial X_1}$ and the linear Newton system to be solved would be $\frac{df}{dX_1}\delta X_1 = -f$. As can be seen by using block elimination on equation (17), the resulting δX_1 is exactly the δX_1 that would result if the Newton step for the entire system is computed by solving

$$\begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \frac{\partial F_1}{\partial X_2} \\ \frac{\partial F_2}{\partial X_1} & \frac{\partial F_2}{\partial X_2} \end{bmatrix} \begin{bmatrix} \delta X_1 \\ \delta X_2 \end{bmatrix} = - \begin{bmatrix} F_1(X) \\ F_2(X) \end{bmatrix} \tag{17}$$

Thus, the Newton direction for the reduced system (16) is just the X_1 component of the Newton direction for the full system (15). However, the globalization for the two systems is different. Given the current iterate X_1^n, a standard line search for (16) would be to choose the step length ϵ to minimize $\| F_1(X_1^{n+1}, X_2^{n+1}) \|$ where $X_1^{n+1} = X_1^n + \epsilon\delta X_1^n$ and $X_2^{n+1} = h(X_1^{n+1})$ is determined by solving exactly $F_2(X_1^n + \epsilon\delta X_1^n, X_2^{n+1}) = 0$.

The maximum computational benefit of the method is achieved when F_2 is much more nonlinear than F_1, the dimension of X_2 is small compared to X, and $F_2(X_1, X_2) = 0$ is relatively easy to solve for X_2 because of special features. An example of the last characteristic is solving the boundary layer equations given the velocity at the edge of the layer. These equations are parabolic in the streamwise direction and hence can be solved rather easily by space marching. They are also very nonlinear especially near separation. Another example of this type of situation is the compressible fluid flow equations near a strong shock wave. This is the situation discussed at length below for a simple one dimensional converging diverging nozzle problem.

5.2 Nonlinear Elimination Applied to Nozzle Flow

Differential Equation Consider a one-dimensional nozzle problem described by the differential equation

$$[A(x)\rho(u)u]_x = 0 \tag{18}$$

where $0 \leq x \leq L$. $A(x)$ is the cross sectional area of the nozzle (assumed to be converging/diverging), $u(x)$ is the velocity, and $\rho(u)$ is the density. This single equation embodies conservation of mass in the nozzle. We assume that the velocity is given as the gradient of a scalar potential, i.e., $u = \phi_x$ and that the density is given by the isentropic formula $\rho = \left[1 + \frac{1}{2}(\gamma - 1)(1 - u^2)\right]^{\frac{1}{\gamma-1}}$ where $\gamma = 1.4$. The boundary conditions are that $\phi(0) = 0$ and $\phi(L) = K$ where K is given. A typical nozzle area distribution is $A(x) = 0.6(x-1)^2 + 0.4$ where $L = 2.0$. In this formulation, we have the following identities and definitions for the pressure p, the local speed of sound c, and the local Mach number $\mathcal{M}$: $p = \rho^\gamma/\gamma$, $c^2 = \gamma p/\rho = \rho^{\gamma-1}$, and $\mathcal{M} = u/c$.

A small value of K will give a totally subsonic and symmetric (about the throat) velocity. In this formulation, $\rho = c = 1$ when $u = 1$, $u < 1$ corresponds to subsonic flow, and $u > 1$ corresponds to supersonic flow. A simple calculation shows that equation (18) is equivalent to

$$A_x \rho u - A \rho u_x (1 - \mathcal{M}^2) = 0$$

At the throat, $A_x = 0$ so that either $u_x = 0$ or $\mathcal{M} = 1$. In subsonic flow, it turns out that $u_x = 0$ at the throat whereas in transonic flow, $\mathcal{M} = 1$ at the throat. For K sufficiently large, the flow will be subsonic in the converging part of the nozzle, sonic at the throat, and supersonic downstream of the throat. The supersonic zone terminates in a shock wave followed by subsonic flow to the exit. The larger K is, the stronger the shock will be.

Discretization If we consider a mesh of points $x_i, i = 1 \ldots NX$ and let $h_i = x_{i+1} - x_i$, $x_{i+\frac{1}{2}} = 0.5(x_{i+1} + x_i)$, $u_{i+\frac{1}{2}} = (\phi(x_{i+1}) - \phi(x_i))/(x_{i+1} - x_i)$, and $\rho_{i+\frac{1}{2}} = \rho(u_{i+\frac{1}{2}})$, a typical centered difference formula would be

$$\frac{A(x_{i+\frac{1}{2}})\rho_{i+\frac{1}{2}}u_{i+\frac{1}{2}} - A(x_{i-\frac{1}{2}})\rho_{i-\frac{1}{2}}u_{i-\frac{1}{2}}}{0.5(h_{i-1} + h_i)} = 0. \tag{19}$$

The boundary conditions are $\phi(x_1) = 0$ and $\phi(x_{NX}) = K$.

First Order Density Biasing A standard form of stabilization (which is required in regions of supersonic flow to rule out expansion shocks) is upwind biasing of the density. We define the switching function μ by

$$\mu = \mathrm{MAX}\left[0, 1 - \frac{\mathcal{M}_c^2}{\mathcal{M}^2}\right]$$

where $\mathcal{M}_c$ is the cutoff Mach number. We take $\mathcal{M}_c^2 = 0.95$. In order to guarantee stability of the scheme the upwinding must be turned on at the foot of the shock where the flow is subsonic. To accomplish this, several iterations of "expanding μ" are applied. In each such iteration, we set $\mu_{i+\frac{1}{2}} = \mathrm{MAX}[\mu_{i-\frac{1}{2}}, \mu_{i+\frac{1}{2}}, \mu_{i+\frac{3}{2}}]$.

In the difference formula (19), stabilization is achieved by replacing $\rho_{i+\frac{1}{2}}$ by

$$\tilde{\rho}_{i+\frac{1}{2}} = \rho_{i+\frac{1}{2}} - \mu_{i+\frac{1}{2}}(\rho_{i+\frac{1}{2}} - \rho_{i-\frac{1}{2}})$$

Mass Flux Biasing It has been found that sonic lines are more accurately resolved by using upwind biasing of the mass flux to stabilize the finite difference scheme rather than density biasing. In this formulation, the switching function is not used. Instead we let ρ^* be the sonic density (in this case assumed to be 1.0). The sonic value of the velocity, u^* is also 1.0. In the difference formula (19), stabilization is achieved by replacing $\rho_{i+\frac{1}{2}}$ by $\tilde{\rho}_{i+\frac{1}{2}}$ given by

$$\tilde{\rho}_{i+\frac{1}{2}} = \rho_{i+\frac{1}{2}} - \frac{1}{u_{i+\frac{1}{2}}}\left[\overline{\rho_{i+\frac{1}{2}} u_{i+\frac{1}{2}}} - \overline{\rho_{i-\frac{1}{2}} u_{i-\frac{1}{2}}}\right] \tag{20}$$

where

$$\overline{\rho_{i+\frac{1}{2}} u_{i+\frac{1}{2}}} = \begin{cases} 0 & \text{if } \mathcal{M}_{i+\frac{1}{2}} < 1.0 \\ \rho_{i+\frac{1}{2}} u_{i+\frac{1}{2}} - \rho^* u^* & \text{if } \mathcal{M}_{i+\frac{1}{2}} \geq 1.0 \end{cases} \tag{21}$$

Computational Results All computations reported below were performed with first order mass flux biasing. Similar asymptotic results are found with first order density biasing except that the numbers of Newton steps are uniformly smaller. A complete Newton method was implemented with analytic computation of the Jacobian matrix. Figure 1 shows the geometry of the nozzle and the solution obtained on a very fine grid. The value of K used is 1.15, resulting in a shock wave of moderate strength.

Figure 2 shows the convergence of Newton's method for this problem on a series of uniform grids. Globalization was done using a non-derivative line search that minimized the $L2$ norm of the residual of (19). The line search routine has a tolerance that controls to what level of error the minimum of the norm of the residual is sought. A value of 10^{-6} was used in these calculations corresponding to a very accurate line search. As can be seen from the second panel of the figure, the step lengths were very close to zero except near convergence.

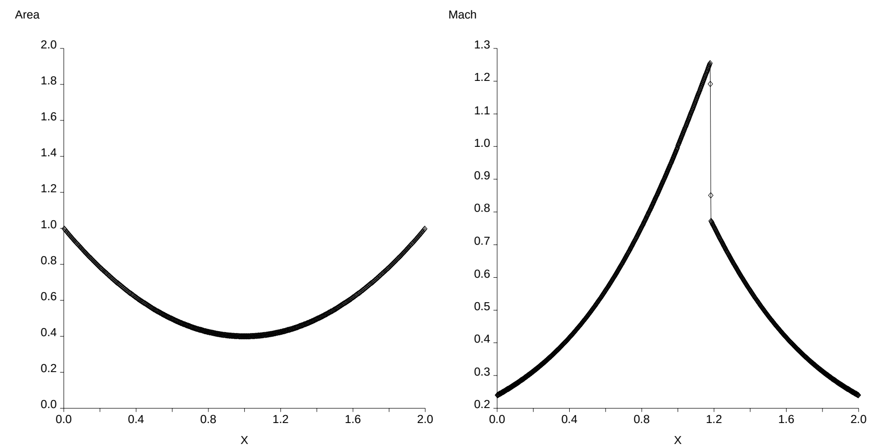

Fig. 1. Model converging diverging nozzle problem.

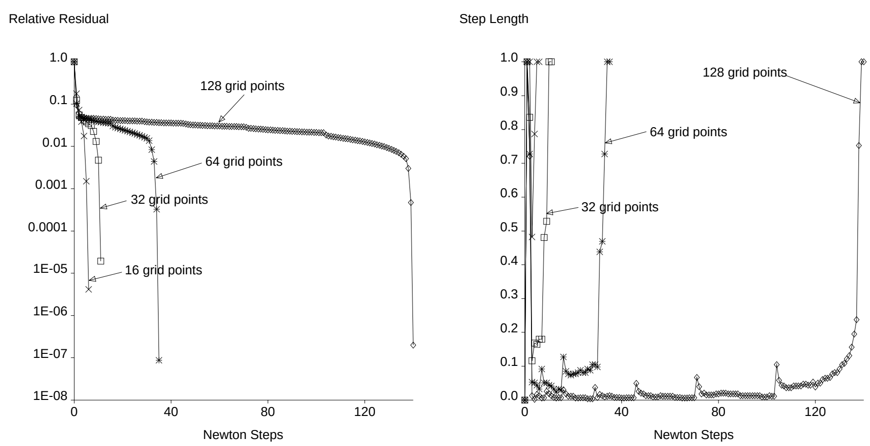

Fig. 2. Newton convergence for model problem with standard globalization.

In figure 3, we show the results of applying the nonlinear elimination method to this problem. The eliminated equations and unknowns correspond to a subdomain containing roughly 1/8th of all the grid points in the problem centered on the ultimate location of the shock wave. As can be seen from the figure, the convergence of Newton's method is dramatically improved and becomes almost independent of the density of the grid. For this problem, the nonlinearity is almost all concentrated near the shock wave. This is obvious from a consideration of the nature of equations (19), (20), and (21). With this setup, the size of the eliminated system grows linearly with the size of the original problem. The cost of the line search is increased in the nonlinear elimination method over this cost in the standard Newton method because the latter only involves computing the residual whereas the former involves solving nonlinear subproblems over the eliminated subdomain. However, total CPU time is dramatically lower with the elimination method. One effective way to control the computational cost for the nonlinear solves needed in the globalization is to use continuation in the step length ϵ. This strategy can reduce the computational cost by an order or magnitude.

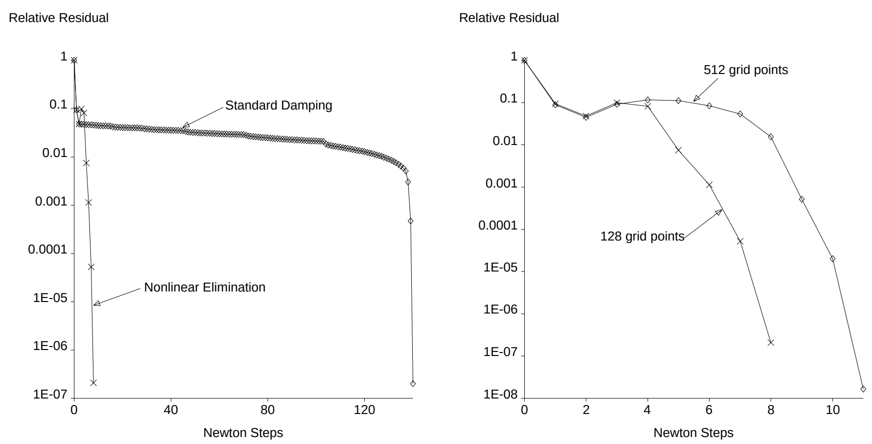

Fig. 3. Newton convergence with nonlinear elimination of the shock region.

We note that for the case of shock waves, there is usually sufficient regularity with respect to grid density to allow grid sequencing [4, 26] to work effectively. However, for second order dissipation methods, this regularity is not as reliable as would be desired. In the case of boundary layers near separation, grid regularity is poor and nonlinear elimination seems to be essential to achieving robust convergence [1, 8, 10].

As the preceding example demonstrates, it is not necessarily true that introducing additional variables will make a nonlinear system easier to solve using Newton's method. Further, the difference between Newton's method applied to the full system and Newton's method applied to the reduced system is purely a difference in globalization. An interesting extension of these ideas has been proposed by Cai and Keyes [5] in which Newton's method is nonlinearly preconditioned by a nonlinear Schwarz method. This method has also been successfully applied to the nozzle problem discussed here with similar acceleration of convergence.

5.3 Globalization Methods Implemented in TRANAIR

All Newton based methods either for analysis or optimization require some kind of globalization 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 which are classical such as the Levenberg Marquardt method or the use of an augmented Lagrangian merit function. In our 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 in some cases even though the bounds required are often problem dependent. There is some reason based on experience in analysis cases to believe that convergence would be enhanced by some type of more automatic step size control for the step in the design variables once the direction is determined by solving the optimization subproblem. The methods described below are rather obviously motivated by nonlinear elimination applied to the Lagrange Newton method [29]. To take into account the inequality constraints we introduce a merit function Φ_M [11, 12]. A popular choice is based on the augmented Lagrangian. We do not have a method to easily apply nonlinear elimination to these constraints with the exception of certain constraints on coefficients of lift and pitching moment which can be enforced exactly in the state equation solve.

Globalization Method #1

Choose ϵ^l to minimize $I(X^l + \epsilon^l Q\delta u^l, u^l + \epsilon^l \delta u^l) + \Phi_M$

Based on the observation that a poor initial guess for the state equations can result in stagnation of even a globalized Newton's method and that this situation was often encountered in step 2(b) of the solution adaptive algorithm discussed in section 4 when using Method #1, a globalization method that maintains the satisfaction of the state equations to some level of accuracy for each change in the design variables was implemented.

Globalization Method #2

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

An alternative is to 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_1 one can solve F_2 for X_2. Then one can apply nonlinear elimination to F_2. One obvious choice is to take F_2 to be the boundary layer equations and X_2 the boundary layer variables.

Globalization Method #3

> Choose ϵ^l to minimize $I(\hat{X}_2^l, X_1^l + \epsilon^l \delta X_1^l, u^l + \epsilon^l \delta u^l) + \Phi_M$, where $\hat{X}_2^l$ is determining by solving to some level of accuracy the state equation $F_2(\hat{X}_2^l, X_1^l + \epsilon^l \delta X_1^l, u^l + \epsilon^l \delta u^l) = 0$ and δX_1^l is determined by restricting $Q\delta u^l$. Set $X_2^{l+1} = \hat{X}_2^l$, $X_1^{l+1} = X_1^l + \epsilon^l \delta X_1^l$, and $u^{l+1} = u^l + \epsilon^l \delta u^l$.

The computational cost of globalization method #2 can be dramatically reduced by using continuation in ϵ to generate initial guesses for the state equation solves. In many cases, the initial guess generated by this continuation strategy is good enough that only a single Newton step is required for adequate convergence for a new value of ϵ. We can usually use the same preconditioner for all values of ϵ. Using these strategies, the cost of the globalization can be reduced to that of one or two nonlinear state equation solves on a comparable grid.

Generally, one of the salient features of compressible viscous flows is the fact that the region of attraction for Newton's method is very small. In fact, for the case of strong shocks and also for boundary layer coupling, careful attention to globalization strategies is absolutely essential to achieving the kind of reliable convergence required in the industrial engineering environment. For shock waves [26], either grid continuation, continuation in artificial viscosity, or nonlinear elimination is effective. For boundary layer coupling [1, 8], nonlinear elimination has proven the most satisfactory strategy especially near or beyond separation. This is probably due to the bifurcations known to be present in nonlinear systems representing such flow fields. Another continuation method for viscous flows that has been unexplored by the authors is continuation in Reynolds number. It is also true that generally aerodynamic optimization problems are not well posed without regularization which adds another element of difficulty. Since there are an essentially infinite number of ways to pose a given engineering optimization problem, it not always easy to determine which way is most amenable to numerical optimization or how to regularize the problem if it is ill-posed.

5.4 Computational Examples in Aerodynamic Design

In this section, we discuss the results of applying these globalization methods to four nonlinear least squares problems. One of the motivations for the introduction of these methods was the possibility that a design step could result in a sufficiently large change in the geometry that on the next grid the analysis problem nonlinear solution process would not converge. In our experience, failure to converge on a given grid is rarely improved by going on to a new grid. This did in fact happen in several cases examined here when using globalization #1. In some cases, this can be mitigated by user specified constraints. Unfortunately, specification of these constraints is problem dependent.

We note that in this case of a least squares objective, the method employed in TRANAIR is the Gauss Newton method. In the case that the residuals are nearly zero at the solution, we should expect the rapid convergence characteristic of Newton's method from a properly formulated sensitivity method. Thus, this type of test case is in some sense less challenging than the problem of minimizing a scalar objective. Suppose the cost function I is given by

$$I = \frac{1}{2}\sum_{l=1}^{q} c_l^2.$$

In this case, the reduced gradient and reduced Hessian are given by

$$\begin{aligned} \frac{dI}{du_j} &= \sum_{l=1}^{q} c_l \frac{dc_l}{du_j}, & j = 1, 2, ..., m, \\ \frac{d^2 I}{du_j du_i} &= \sum_{l=1}^{q} \frac{dc_l}{du_i}\frac{dc_l}{du_j} + \sum_{l=1}^{q} c_l \frac{d^2 c_l}{du_j du_i}, & i, j = 1, 2, ..., m. \end{aligned} \tag{22}$$

If the c_l are small, the second term in equation (22) for the Hessian can be neglected and a good approximation to the reduced Hessian is given only in terms of the sensitivities

$$R_{l,j} = \frac{dc_l}{du_j} = \frac{\partial c_l}{\partial u_j} + \sum_{k=1}^{n} \frac{\partial c_l}{\partial X_k}\frac{dX_k}{du_j}$$

We can now write equation (7) in the form $R^T R\delta u = -R^T c$. This is equivalent to the least squares problem MIN $||R\delta u + c||_2$ which can be solved by applying a constrained linear least squares solver.

The test cases discussed in this section are pressure matching cases involving a target pressure distribution obtained by analyzing an airfoil section similar to the ONERA M6 section. The initial airfoil was the NACA0012. The objective function is

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

with the weights W_l taken to be the surface panel length for the surface point in question. The pressure coefficients on the airfoil are $(c_p)_l$ and $(\hat{c}_p)_l$ are the target pressure coefficients. The first case used the flow conditions $M_\infty = 0.75$ and $\alpha = 1.0$. Figure 4 shows the initial geometry, the target geometry, and also the initial pressure distribution (on a fine grid) and the target pressure distribution.

In these cases a large number of grids was used to examine asymptotic convergence issues. For this case the grid was frozen after grid 15. The gridding is solution adaptive and the numbers of finite elements in the grids were

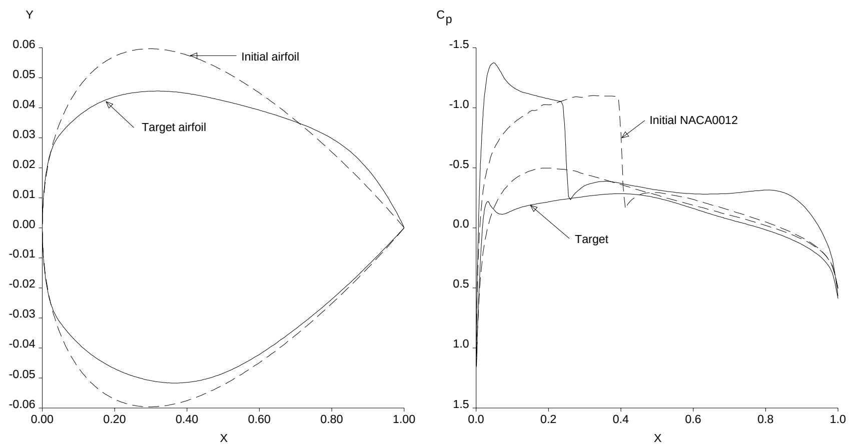

Fig. 4. Geometry and solution for Design Least Squares Case.

198, 397, 731, 1399, 2650, 5549, 6388, 7355, 8478, 9234, 10606, 12246, 13140, 14836, and 15288. This final grid is repeated a number of times. The case was run with globalization methods #1 and #2. In the inviscid case, there is little difference between the globalizations because the step size selected is almost always unity. There are some grids for which globalization #1 seems to suffer from a poor step where the agreement between the sensitivity prediction and the actual result is poor, but they are infrequent.

Figure 5 shows the convergence history for this case. Both the objective function and the estimated reduced 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 objective 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 linearized model of the state variables 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. It should be noted that finite differences are used to linearize various nonlinear components of the objective and state equations. Because of this fact, it is not expected that the reduced gradient can be reduced by a factor greater than the square root of machine precision (roughly 10^{-7} in this case) and this is born out by the computational results. The vertical line at grid 15 indicates the point at which the grid is frozen. The fixed grid results are for convenience shown starting at this point.

In these cases, solution adaptive gridding introduces a possible difficulty that might place a strain on global convergence because the target pressure

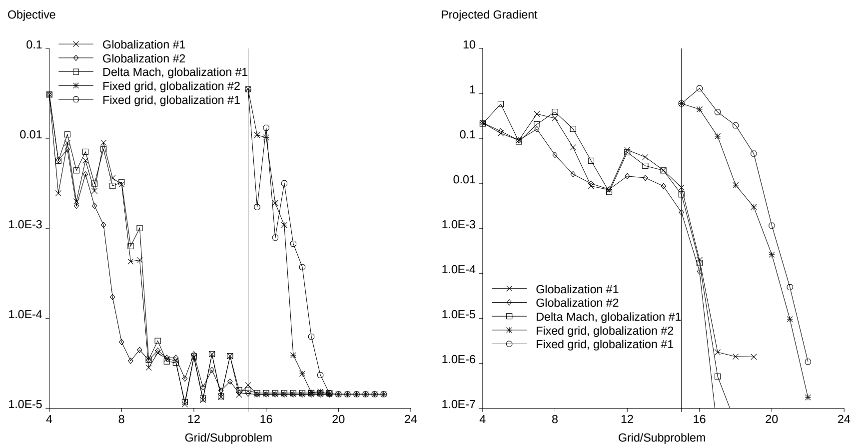

Fig. 5. Convergence of Inviscid Design Least Squares Case with Various Globalization Options.

distribution was taken from an analysis on a very fine grid and thus matching this distribution exactly on a coarse grid would result in the "wrong" geometry. To eliminate this effect as well as to dispense with the effect of potentially large changes in the grid in the solution adaptive case, another test was run. First a good solution adaptive grid was determined for the target airfoil and pressure distribution in an initial TRANAIR run. Then the design was run from the initial NACA0012 geometry with each grid being this solution adaptive grid. In this case, as can be seen in Figure 5, both globalization methods converged even though method #2 was a little faster.

A second test case illustrates the globalization methods as applied to a viscous flow case formulated using the same airfoils with a Reynolds number of 6.5 million. The C_l was specified to be the same as the inviscid case, namely, 0.3655. The pressure distributions for the initial airfoil and the target are shown in Figure 6. In this case, the effect of globalization method #2 is very significant. Using globalization method #1, convergence is achieved only by using constraints on the change in local Mach number in the optimization subproblems and line search. The value chosen was 0.2 based on experience in analysis cases. However, the final value of the objective is not quite as good as is achieved with globalization method #2. The convergence histories are given in figure 7. Using globalization method #1, the analysis portion of the code cannot converge on grid 5 with the current design variables. With globalization method #2, the optimization converges rather rapidly, the result being good even before the grid is frozen. Also in the fixed grid test, there is a significant advantage for globalization method #2. This is due to the better estimation of the true solution on the next grid taking into account the optimization step.

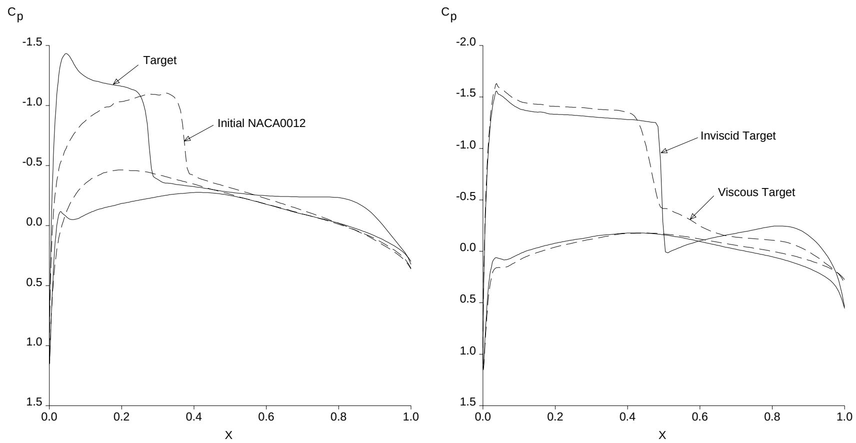

Fig. 6. Target Pressure Distributions for Viscous Case and for Challenging Case

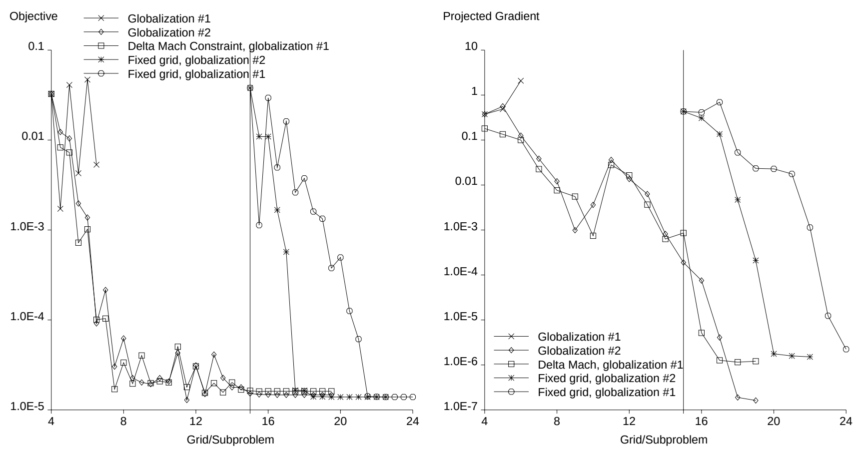

Fig. 7. Convergence of Viscous Design Least Squares Case with Various Globalization Options.

Another more challenging set of flow conditions was also run. The target pressure distribution was obtained by running the target airfoil at $M_\infty = 0.75$ and $\alpha = 2.0$. The resulting pressure distribution is shown in Figure 6 for both the inviscid and viscous cases. In this case, both globalizations perform well. The convergence histories are shown in figure 8.

This more challenging case was also run in viscous flow with a Reynolds number of 6.5 million. The C_l was once again set to match the inviscid case. This case has shock induced separation of the boundary layer. The convergence histories are shown in figure 9. In this case, there are significant convergence difficulties and even method #2 has some difficulty matching the

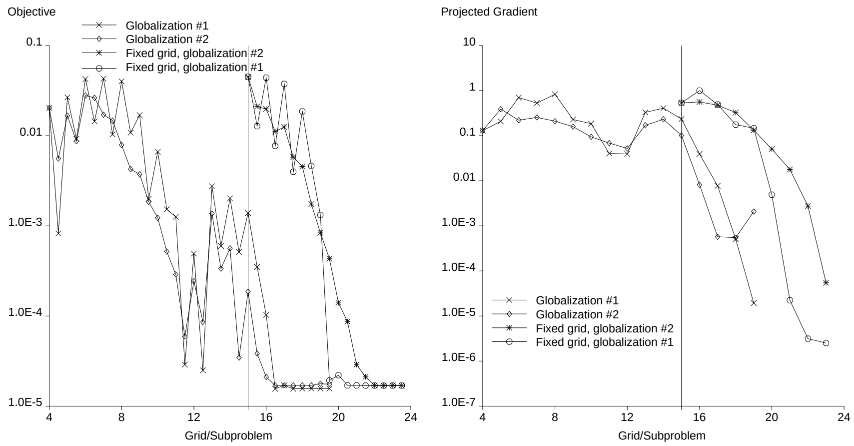

Fig. 8. Convergence of Challenging Design Least Squares Case with Various Globalization Options.

pressure distribution in the separated flow region. Globalization method #1 even with a constraint on the change in Mach number fails to sufficiently reduce the residuals of the analysis problem on grid #5. However, the results obtained with method #2 are acceptable from an engineering point of view.

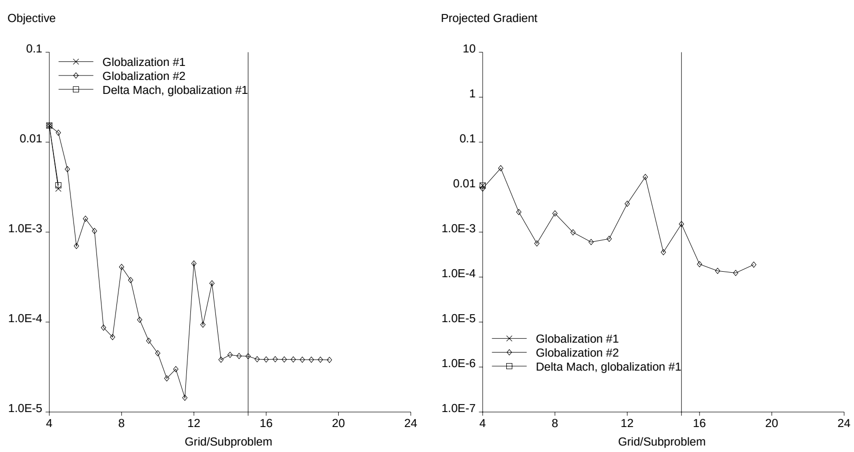

Fig. 9. Convergence of Separated Viscous Design Least Squares Case with Various Globalization Options.

All of the cases in this section were run with first order density biasing, the least nonlinear form of upwinding available in TRANAIR. With more "accurate" forms of upwinding, less reliable results can be expected. Finally, we have noted that often in cases of drag minimization such as discussed in section 8, globalization method #2 can result in designs for which the objective is somewhat lower and for which the inequality constraints are more nearly satisfied than globalization method #1. This may seem to be a minor matter, but when coupled with more robust convergence, this becomes important in multi-point design.

6 Multi-point Design

The TRANAIR design and optimization method described in section 4 has been extended to optimization at multiple flow conditions. Multi-point 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 multi-point design will be more stable than single point design; for example, in minimizing drag for an airfoil there are often many shock free single point airfoil designs none of which is viable from an engineering point of view.

The algorithm for multi-point design is very similar to that described in section 4. Step 1 is to give each flow condition solver the initial geometry (i.e., values for the design variables). Steps 2(a) and 2(b) can proceed in parallel and are completely independent for different flow conditions. Steps 2(c) and 2(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), 2(f), and 2(g) can again proceed independently for each flow condition. Details of the method and some simple computational examples can be found in [21].

With regard to globalization, in a multi-point design, lack of convergence of any analysis/sensitivity run on any grid for any flight condition brings the whole process to a halt. As is often the case in complex engineering processes, the overall probability of success is the manifold product of the probability of success on any analysis case. Thus, improvements in robustness tend to be given very high priority by the industrial engineering community. Similarly, many multi-point designs boil down to trading drag at different operation conditions against each other and against many inequality constraints. These trades are often sufficiently sensitive that a merit function based globalization process such as #1 results in less than satisfying results. Globalization method #2 can more accurately do the required trades.

7 Limitations and Remaining Difficulties

In this section, we will indicate the remaining limitations of aerodynamic design optimization by considering the problem of single point drag minimization for an airfoil in transonic flow. This problem is not well-posed in the classical sense because there may be multiple solutions. However, consideration of this problem illustrates the difficulties often encountered in practice in aerodynamic optimization.

In our test case, the drag to be minimized is the sum of the wave and profile drag. The design variables consist of 9 camber and 11 thickness modes. The thickness knots are located at 0,10,20,30,40,50,60,70,80,90, and 100% of chord. The thickness modes at 0 and 100% allow changes in the slope of thickness so that the trailing edge closure angle and the leading edge radius of curvature can change. The camber knots are located at 10,20,30,40,50,60,70,80, and 90% of chord. Values of camber and thickness are determined by a special spline fit to the values at the knots. The angle of attack is also a design variable. In figure 10, we show the initial airfoil and the final airfoil achieved with globalization methods #1 and #2. In table 1, we give the final value of C_d which is the objective function. Globalization #2 results in a somewhat lower level of the objective function. It should be noted that in this case, NPSOL was almost never able to achieve convergence for the optimization subproblems. The typical return condition was that the current solution cannot be improved and the estimated condition number of the Hessian typically became very large, roughly the reciprocal of machine precision, about 10^{14}. The estimated reduced gradient was never reduced more than a single digit indicating a remaining undiagnosed difficulty.

Table 1. Aerodynamic Forces for Various Optimization Options.

Case	C_d	Wave C_d	Profile C_d	C_l
baseline NACA0012	246.98	125.35	121.63	0.550
Camber/thickness, globalization #1	91.57	0.75	90.82	0.550
Camber/thickness, globalization #2	90.30	0.00	90.30	0.550
Curvature variables, globalization #2	90.92	0.14	90.78	0.550
Monotone curvature, globalization #2	91.09	0.08	91.01	0.550

One form of regularization involves restricting the design geometry space to "smooth" airfoil shapes. We were able to test the effect on optimization performance by using design variables that parameterized directly the curvature of the airfoil. Once again there were 20 variables describing the curvature and angle of attack was also a design variable. The variables represented coefficients for a Legrendre polynomial expansion of the curvature. The square root of arc length from the leading edge of the airfoil was used as the independent variable for these polynomials and globalization method #2 was used. As can be seen from figure 10, the resulting airfoil is much smoother even though the drag is somewhat higher. In this case, NPSOL was able to find optimal solutions for most of the subproblems and the estimated Hessian condition numbers were between 10^3 and 10^6 indicating that the optimization problem is better posed. In Figure 11, the curvatures of the optimized airfoils are shown. The airfoils optimized using splines show very rough curvature distributions while those design with curvature variables show smoother distributions. Also shown is the effect of imposing monotonicity on the curvature on a large portion of the upper and lower surfaces. As can be seen from table 1, there is a small objective function increase associated with these additional inequality constraints.

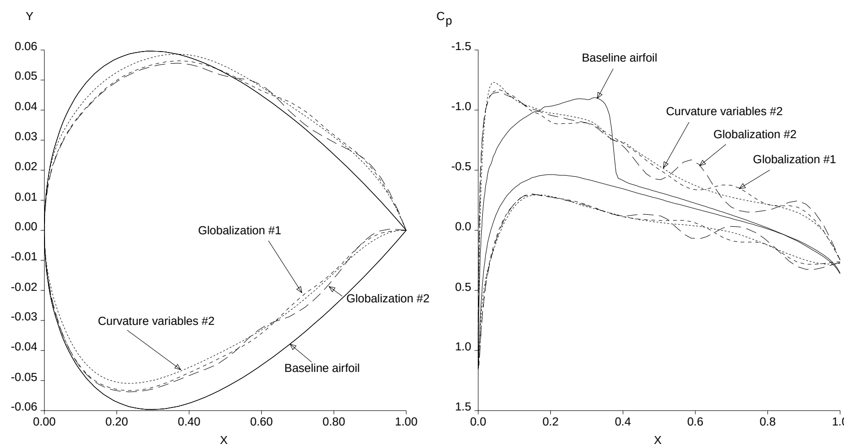

Fig. 10. Optimized Airfoils and Pressures Distributions with Various Optimization Methodologies

As mentioned above, an outstanding question is whether adding more state equation curvature information to the optimization subproblems would produce more reliable convergence. If the last 4 terms in equation (10) were added to the optimization subproblem, or if the simple quadratic program given by equation (12) were formed and solved, the result would be a true Lagrange Newton method and global convergence might be accelerated. Certainly, this issue is one of the most potentially important algorithmic issues left unaddressed by our work so far even though the computational cost and programming issues involved in examining it systematically for the case of viscous compressible flow are formidable.

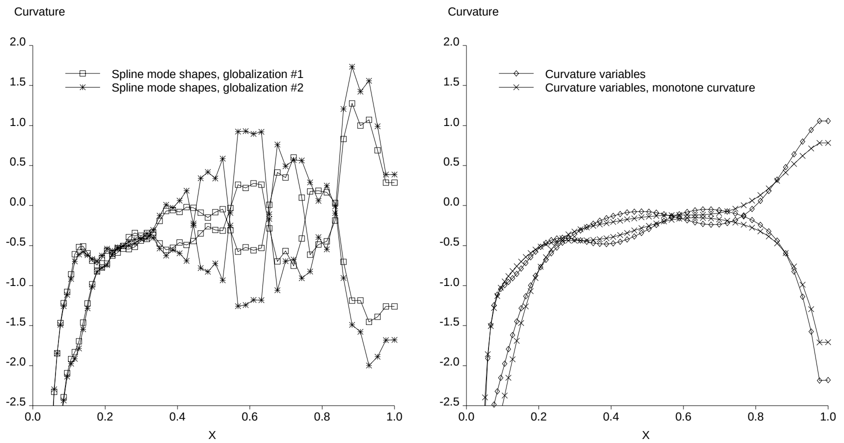

Fig. 11. Curvature Distributions of Optimized Airfoils and the Effect of Curvature Monotonicity Constraints

8 Summary

We have improved the single point design and optimization methodology in TRANAIR to include a range of globalization strategies and extended it to a multi-point design capability. Both enhancements improve the robustness and usefulness of the capability. We have offered a rationale for preferring the sensitivity method to the adjoint method in the context of a reduced gradient method by showing that some second order information is inexpensively available in the sensitivity method. We have discussed the effect in practice of staying relatively close to the manifold of solutions to the state equations which in this case are highly nonlinear.

References

1. 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. Technical Report BCSTECH-94-032, Boeing Computer Services, 1994.
2. G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur Methods for PDE-Constrained Optimization. Part I: The Krylov-Schur Solver. Technical Report, Laboratory for Mechanics, Algorithms, and Computing, Carnegie Mellon University, 2000.
3. A. E. Bryson and Y. C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control.* Hemisphere, New York, 1975.
4. X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young. Parallel Newton-Krylov-Schwarz Algorithms for the Transonic Full Potential Equation. *SIAM J. Sci. Comput.*, **19**: 246–265, 1998.

5. X.-C. Cai, and D. E. Keyes. Nonlinearly Preconditioned Inexact Newton Algorithms. submitted to *SIAM J. Sci. Comput.*
6. E. J. Cramer, J. E. Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem Formulation for Multidisciplinary Optimization. *SIAM J. Optimization*, **4**: 754–776, 1994.
7. R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton Methods. *SIAM Journal on Numerical Analysis* **19**: 400–408, 1982.
8. M. Drela. *Two-Dimensional Transonic Aerodynamic Design and Analysis Using The Euler Equations.* PhD Dissertation, MIT, 1985.
9. M. Drela. Pros and Cons of Airfoil Optimization. *Frontiers of Computational Fluid Dynamics 1998*, World Scientific, 1998.
10. M. Drela. Viscous and Inviscid Inverse Schemes Using Newton's Method. *Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications*, AGARD Report No. 780, 1990.
11. R. Fletcher. *Practical Methods of Optimization.* 2nd edn., John Wiley and Sons, New York, 1987.
12. P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* 1st edn., Academic Press, Inc., San Diego, 1981.
13. 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.
14. P. E. Gill, W. Murray, M. A. Saunders, and M. A. Wright. *User's Guide for NPSOL (Version 4.0): A FORTRAN Package Nonlinear Programming.* Stanford University Technical Report SOL86-2, Department of Operations Research, 1986.
15. R. T. Haftka and Z. Gürdal. *Elements of Structural Optimization*, Kluwer Academic Publishers, Boston, 1992.
16. 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.
17. A. Jameson. Aerodynamic Design via Control Theory. *Journal of Scientific Computing*, **3**: 233–260, 1988.
18. A. Jameson. Airfoils Admitting Non-unique Solutions of the Euler Equations. AIAA Paper 91-1625, 1991.
19. W. H. Jou, W. P. Huffman, D. P. Young, R. G. Melvin, M. B. Bieterman, C. L. Hilmes, and F. T. Johnson. Practical Considerations in Aerodynamic Design Optimization. AIAA Paper 95-1730, 1995.
20. P. J. Lanzdron, D. J. Rose, and J. T. Wilkes. An Analysis of Approximate Nonlinear Elimination. *SIAM Journal on Scientific Computing*, **17**: 538–559, 1996.
21. R. G. Melvin, D. P. Young, W. P. Huffman, F. T. Johnson, C. L. Hilmes, and M. B. Bieterman, 'Recent Progress in Aerodynamic Design Optimization', International Journal for Numerical Methods in Fluids, **30**, 205–216 (1999).
22. C. S. Morawetz. *Comm. Pure Appl. Math.*, **10**: 400, 1957.
23. C. S. Morawetz *Comm. Pure Appl. Math.*, **11**: 129, 1958.
24. P. E. Rubbert. CFD and the Changing World of Airplane Design. AIAA Wright Brothers Lecture, Sept. 1994.
25. Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, Boston, 1995.

26. D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson and S. S. Samant. Global Convergence of Inexact Newton Methods for Transonic Flow. *Int. J. Num. Meth. Fluids*, **11**: 1075–1095, 1990.
27. 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.
28. 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.
29. D. P. Young and D. E. Keyes. Newton's Method and Design Optimization. ISSTECH-96-011, May 1996.