Newton's Method and Design Optimization *

David P. Young †and David E. Keyes ‡

1 Abstract

This report discusses design optimization methods, many of which are equivalent to classical optimal control methods [1, 2] in the sense that Newton's method is applied to the necessary conditions for optimality. In this context, many design methods are minor variants of the classical Lagrange-Newton method and can be understood by using the theory for Newton's method. In many application areas, variants of Newton's method have been used that have in common an augmented line search that ensures the satisfaction of certain of the nonlinear equations at each Newton iteration. This nonlinear elimination method has recently been analyzed by Lanzkron, Rose, and Wilkes [3] and is of significant value in boundary layer coupled CFD [4, 5, 6].

We first review classical optimal control methods and then show how they can be extended to state equations derived from boundary value problems and to solution-adaptive grid methods. We then derive various proposed decomposition methods in this context and show a relationship between these methods and the inexact nonlinear elimination method. This report owes something to the spirit of [7] in the use of the relationship between methods for solving nonlinear systems of equations and optimization problems.

2 Classical Design and Optimization Methodology

Consider the problem of minimizing a scalar objective function $I(X, u)$ subject to the constraint that $F(X, u) = 0$, where $X = (X_1, X_2, \ldots, X_n)^T$ and $u = (u_1, u_2, \ldots, u_m)^T$. Typically, $F(X, u)$ is a discretization of a boundary value problem which given u can be solved for X. We assume that this solvability implies that $\partial F_i/\partial X_k$ is invertible for the values of u of interest. We 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. This is done so that the usual notation for the Jacobian of the state equations can be retained even though the standard notation in [1] uses column vectors instead, resulting in a Jacobian that is the adjoint of the usual one. We further use the convention that the Hessian matrix of second derivatives of I is obtained by taking the gradient of the adjoint of the gradient of I.

*This work was supported in part by the National Science Foundation under ECS-9527169.

†Boeing, P. O. Box 3707, M/S 7L-21, Seattle, WA 98124-0346, dpy6629@espresso.rt.cs.boeing.com.

‡Computer Science Department, Old Dominion University, Norfolk, VA 23529-0162, keyes@cs.odu.edu.

The necessary conditions for optimality are often formulated by introducing the Lagrange multipliers λ as independent variables [2]. The Lagrangian is then defined by

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

The notation A^* is used to represent the adjoint (in the real-valued discrete case, just the transpose) of A. In the presence of sufficient smoothness, necessary conditions for an optimum are that the gradient of L be zero.

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

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

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

For the discretized form of the state equations, as discussed in [1, 2], Newton's method can be applied to this system of nonlinear equations. The resulting linearized system in discrete form is

$$\begin{aligned}
\frac{\partial^2 L}{\partial u_j \partial u_i}\delta u_i + \frac{\partial^2 L}{\partial u_j \partial X_l}\delta X_l + \frac{\partial^2 L}{\partial u_j \partial \lambda_k}\delta \lambda_k &= -\frac{\partial L}{\partial u_j}^* \\
\frac{\partial^2 L}{\partial X_j \partial u_i}\delta u_i + \frac{\partial^2 L}{\partial X_j \partial X_l}\delta X_l + \frac{\partial^2 L}{\partial X_j \partial \lambda_k}\delta \lambda_k &= -\frac{\partial L}{\partial X_j}^* \\
\frac{\partial^2 L}{\partial \lambda_j \partial u_i}\delta u_i + \frac{\partial^2 L}{\partial \lambda_j \partial X_l}\delta X_l + \frac{\partial^2 L}{\partial \lambda_j \partial \lambda_k}\delta \lambda_k &= -\frac{\partial L}{\partial \lambda_j}^*
\end{aligned} \tag{4}$$

Using the subscript notation for partial derivatives, e.g., $F_{XX} = \frac{\partial^2 F}{\partial X_j \partial X_l}$, for convenience, one can compute the partial derivatives more explicitly by noting that

$$L_{\lambda,u} = F_u, \qquad L_{\lambda,X} = F_X, \qquad L_{u,\lambda} = F_u^*, \qquad L_{X,\lambda} = F_X^*, \text{ and } \qquad L_{\lambda,\lambda} = 0.$$

We can now write the linear system (4) as

$$\begin{bmatrix} I_{u,u} + \lambda^* F_{u,u} & I_{u,X} + \lambda^* F_{u,X} & F_u^* \\ I_{X,u} + \lambda^* F_{X,u} & I_{X,X} + \lambda^* F_{X,X} & F_X^* \\ F_u & F_X & 0 \end{bmatrix} \begin{pmatrix} \delta u \\ \delta X \\ \delta \lambda \end{pmatrix} = - \begin{pmatrix} I_u^* + F_u^* \lambda \\ I_X^* + F_X^* \lambda \\ F(X, u) \end{pmatrix}. \tag{5}$$

If we let $\hat{\lambda} = \lambda + \delta\lambda$, this becomes

$$\begin{bmatrix} I_{u,u} + \lambda^* F_{u,u} & I_{u,X} + \lambda^* F_{u,X} & F_u^* \\ I_{X,u} + \lambda^* F_{X,u} & I_{X,X} + \lambda^* F_{X,X} & F_X^* \\ F_u & F_X & 0 \end{bmatrix} \begin{pmatrix} \delta u \\ \delta X \\ \hat{\lambda} \end{pmatrix} = - \begin{pmatrix} I_u^* \\ I_X^* \\ F(X, u) \end{pmatrix}. \tag{6}$$

In the cases where $\partial F / \partial X$ is invertible for reasonable values of u, the Lagrange multipliers can be eliminated from the necessary conditions given by equations (1), (2), and (3). This is done by first solving equation (2) for λ. This yields

$$\lambda = -\left(\left(\frac{\partial F}{\partial X}\right)^*\right)^{-1} \left(\frac{\partial I}{\partial X}\right)^*. \tag{7}$$

If this is substituted into equation (1), we obtain the following equivalent necessary conditions for optimality:

$$\begin{aligned} \frac{dI}{du} &\equiv \frac{\partial I}{\partial u} - \left(\frac{\partial I}{\partial X}\right)\left(\frac{\partial F}{\partial X}\right)^{-1}\left(\frac{\partial F}{\partial u}\right) \\ &= \frac{\partial I}{\partial u} + \lambda^* \left(\frac{\partial F}{\partial u}\right) = 0 \\ F(X,u) &= 0. \end{aligned} \tag{8}$$

The quantity dI/du defined above is often called the reduced gradient because it is the gradient of the objective projected onto the tangent plane of the manifold of solutions of the state equations. Using discrete notation for ease of understanding, the necessary conditions can now be formulated either in terms of solving an adjoint problem (2) for λ or directly by solving the linearized state equations using multiple right-hand sides to compute each column of the n by m sensitivity matrix

$$Q \equiv \frac{\partial X_i}{\partial u_j} = \left(\frac{\partial F_k}{\partial X_i}\right)^{-1}\left(\frac{\partial F_k}{\partial u_j}\right). \tag{9}$$

The same observation applies to any other constraints $C(X,u)$ whose projected gradient may be required.

If Newton's method is applied to equation (8), we obtain the following equations for the updates $\vec{\delta u}$ to $\vec{u}$ and $\vec{\delta X}$ to $\vec{X}$:

$$H \delta u_i = -G^* \tag{10}$$

$$\frac{\partial F(X,u)}{\partial u_i}\delta u_i + \frac{\partial F(X,u)}{\partial X_j}\delta X_j = -F(X,u) \tag{11}$$

The matrix $H = d^2I/du_j du_i$ is often called the reduced Hessian and G is a modified form of the reduced gradient given above. Formulas for H and G will be derived below. If H is positive definite, solving equation (10) is equivalent to minimizing the quadratic functional

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

It is possible to impose any additional constraints involving either X and/or u at this stage. This then defines an optimization subproblem for the global optimal control problem. In the optimization literature these methods are called Lagrange-Newton or Sequential Quadratic Programming methods.

Now adding a line search to determine the step length to either equation (6) or to equations (10) and (11) results in a well defined optimization method [1] that is globally convergent and locally quadratically convergent under reasonable assumptions. The theory for inexact Newton methods [8] is applicable and allows the use of various approximations [9, 10]; which are often required in applications. The minimization method based on equation (6) is as follows:

Minimization Method #1

Given initial values of u^0, X^0, and λ^0, for $i = 1, 2, 3, \ldots, n$:

1. Compute the components for the matrix and right-hand side of equation (6) at u^{i-1}, X^{i-1}, and λ^{i-1}.
2. Solve equation (6) for δu^i, δX^i, and $\hat{\lambda}^i$.
3. Choose the step size ϵ^i so that sufficient decrease is achieved in some merit function and compute new values of the variables: $u^i = u^{i-1} + \epsilon^i \delta u^i$, $X^i = X^{i-1} + \epsilon^i \delta X^i$, and $\lambda^i = \lambda^{i-1} + \epsilon^i(\hat{\lambda}^i - \lambda^{i-1})$.
4. Go to step 1. above unless the optimality conditions are satisfied.

This method is often called the Lagrange-Newton method or in some cases the Sequential Quadratic Programming method. The minimization method based on equations (10) and (11) is as follows:

Minimization Method #2

Given initial values of u^0, and X^0, for $i = 1, 2, 3, \ldots, n$:

1. Compute the reduced Hessian and gradient, H and G of equation (10) at u^{i-1} and X^{i-1}.
2. Solve equation (10) for δu^i.
3. Solve equation (11) for δX^i.
4. Choose the step size ϵ^i and compute new values of the variables: $u^i = u^{i-1} + \epsilon^i \delta u^i$, $X^i = X^{i-1} + \epsilon^i \delta X^i$.
5. Go to 1. above unless the optimality conditions are satisfied.

This method is often called a reduced gradient method since the gradient is projected onto the manifold of solutions of the state equations. We note that these methods have been independently derived and applied by Ghattas [11]. As shown below, the two methods are mathematically equivalent in that the iterates are identical. In practice, the necessity of dropping terms that are costly or difficult to compute and other approximations required may result in different algorithms. In addition, there are several simple ways to create an optimization subproblem that is formulated as a nonlinear program rather than a quadratic program. This issue and its effect on global convergence is still being investigated. Also, the determination of the step size may vary depending on the available data.

There are many, many ways to determine the step size ϵ^i in the above algorithm. There is a vast literature on the subject [1, 12, 7]. Trust region methods are one class of methods that address this problem. Since the added cost is often negligible when the already existing method of solving the state equations is a Newton method, one could apply the nonlinear elimination method to the state equations as follows:

Line Search Method #1

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

This modified line search maintains the satisfaction of the state equations to some level of accuracy at each global Newton step. 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 nonlinear elimination to F_1 only resulting in a somewhat modified line search. For example in a CFD code the F_1 might be the boundary layer equations and the X_1 the boundary layer variables while F_2 would be the inviscid equations and X_2 the inviscid variables [4, 5, 6]. Indeed, experience from Newton analysis methods is likely to carry over to optimization methods.

Line Search Method #2

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

There are many popular choices for the merit function Mf [1, 12]. Line Search Method #2 maintains feasibility for only F_1. As discussed below, there is an "inexact" version of this nonlinear elimination method that is probably more efficient in practice.

Another possibility suggested by the nonlinear elimination method is to solve the reduced gradient equations more exactly than the state equations in the line search.

Line Search Method #3

> Choose ϵ^i to minimize $I(X^{i-1} + \epsilon^i \delta X^i, \hat{u}^i) + Mf$, where $\hat{u}^i$ is chosen by solving the first equations in the set (8), i.e., $dI/du = 0$. Set $u^i = \hat{u}^i$ and $X^i = X^{i-1} + \epsilon \delta X^i$.

This method requires a subiteration since there is usually no preexisting method to solve the required nonlinear system. The reduced gradient would be evaluated using equation (8). One could use a Newton method where the Hessian and Gradient can be simplified by the fact that they are evaluated only at points of the form $X^{i-1} + \epsilon^i \delta X^i$. This method, while it could significantly accelerate convergence, would seem to be computationally impractical. In the case that $F(X, u)$ is a linear function of X, significant simplifications would result. If the number of design variables is small, such a method might become practical.

Any of the line search methods could be used in conjunction with either of the minimization methods resulting in many possible combinations. Various approximations and levels of inexactness would result in a plethora of "methods."

We now describe how to derive equations (10) and (11) from equation (6) in order to consider the computational implications. Such a derivation is well known and is given in abbreviated form in [1]. A byproduct of this derivation will be a formula for the reduced Hessian H. One first solves the second set of equations in (5) for $\delta\lambda$ and the third set of equations for δX. These expressions are then substituted into the first set of equations. The result is equation (10), where the reduced Hessian matrix H is given by

$$\begin{aligned} H &= I_{uu} - Q^* I_{Xu} - I_{uX} Q + Q^* I_{XX} Q \\ &+ \lambda^* F_{uu} - Q^* \lambda^* F_{Xu} - \lambda^* F_{uX} Q + Q^* \lambda^* F_{XX} Q, \end{aligned} \tag{13}$$

where the subscript notation for partial derivatives is used. The modified reduced gradient G is given by

$$\begin{aligned} G^* &= I_u^* - F_u^* F_X^{-*} I_X^* - (I_{uX} + \lambda^* F_{uX}) F_X^{-1} F \\ &- F_u^* F_X^{-*} (I_{XX} + \lambda^* F_{XX}) F_X^{-1} F \\ &= (I_u - I_X F_X^{-1} F_u)^* \\ &- \left[(I_{uX} + \lambda^* F_{uX}) + F_u^* F_X^{-*} (I_{XX} + \lambda^* F_{XX}) \right] F_X^{-1} F \end{aligned} \tag{14}$$

The first two terms in the formula for G^* constitute the adjoint of the reduced gradient given by equation (8) above. The last two terms can be neglected if $F_X^{-1}F$, i.e., the Newton step for the state equations, is small. It can be shown that neglecting this term altogether results in convergence that is two-step superlinear.

Thus, solving equations (10) and (11) for δu and δX yields the same solution as solving equation (5) for $\delta u, \delta X$, and $\delta\lambda$. One of the main differences is that equation (5) is sparse, whereas the reduced Hessian matrix appearing in equation (10) is dense. In either case, the problem must be solved subject to any inequality constraints. The difficulty of solving a very large sparse linear system subject to a large number of inequality constraints has discouraged in many cases the use of Minimization Method #1.

3 Solution-Adaptive Analysis Method

To connect our theory with large-scale computing practice and to illustrate some of the additional choices necessary to create a design optimization capability, we now consider solution-adaptive grid methods for solving boundary value problems. In such a method, there is a series of grids, $l = 1, 2, \ldots, NG$. On each grid the boundary value problem is discretized and solved. The continuous differential operator will be denoted by $\mathcal{F}(X, u)$. (We will use script letters to denote continuous operators and Latin letters to represent their discretizations.) Included in $\mathcal{F}(X, u)$ are any boundary conditions required. On any given grid with n discrete unknowns, we let $F^l(X^l, u^l) = (F_1(X^l, u^l), F_2(X^l, u^l), \ldots, F_n(X^l, u^l))^T = 0$ denote the discretized state equations where, using classical optimal control terminology, the state variables, $X^l = (X_1, X_2, \ldots, X_n)^T$, are the unknowns for which the boundary value problem can be solved given values for the controls or parameters, $u^l = (u_1, u_2, \ldots, u_m)^T$. Likewise, the objective $\mathcal{I}(X, u)$ is assumed to have a continuous formulation that can be discretized along with the state and design variables.

Given an initial X^0, an analysis consists of fixing the controls at some value U and then for each grid $l = 1, 2, \ldots, NG$:

1. Discretize $\mathcal{F}(X, U)$ on grid l.
2. Solve the discrete problem $F^l(X^l, U) = 0$ for X^l to **some level of accuracy** using an initial guess derived from X^{l-1} by interpolation and/or restriction.
3. Estimate the discretization error using X^l.
4. If this estimate is not sufficiently small, use it to determine grid $l + 1$ and go to step 1 above.

4 Nonlinear Continuation Analysis Method

The above method can be considered a special case of a nonlinear continuation method in which the state equations are considered functions of a parameter Λ so that we have $\mathcal{F}(X, U, \Lambda) = 0$. In general, there will be a final value $\hat{\Lambda}$ which is believed to represent the solution sought. In fluid dynamics Λ might be the level of artificial dissipation introduced. For the Navier-Stokes equations, it might be the Reynolds number.

In such a method, there is a series of values of Λ^l, $l = 1, 2, \ldots$. For each value of Λ^l, the boundary value problem is discretized and solved. Given an initial X^0, an analysis consists of fixing the design parameters at some value U and then for $l = 1, 2, \ldots, NG$:

1. Discretize $\mathcal{F}(X, U, \Lambda^l)$.
2. Solve the discrete problem $F^l(X^l, U, \Lambda^l) = 0$ for X^l to **some level of accuracy** using an initial guess derived from X^{l-1}.
3. Estimate a reasonable value of Λ^{l+1}.
4. If $\Lambda^l \neq \hat{\Lambda}$ go to step 1 above.

This method can be combined with solution adaptivity depending on the problem being solved to produce vastly superior performance [9, 10] over a fixed grid method. In such a method, both Λ and the new grid must be chosen based on some estimates of error. In compressible fluid dynamics, solution-adaptive grid continuation in our experience is a more powerful method than viscosity continuation. However, both can be useful as ways of making the problem more regular.

5 Reduced Gradient Solution-Adaptive Design Method

A constrained optimization solution-adaptive method now can be implemented as follows:

1. Given the initial design variables u^1 and an initial guess for the state variables $\hat{X}^0$. Set the initial step size $\epsilon^0 = 1.0$.
2. For each grid $l = 1, 2, \ldots, NG$ until an optimality criterion is sufficiently small:
   (a) Discretize $\mathcal{F}(X, u)$ and $\mathcal{I}(X, u)$ on grid l. This might involve discretizing u as well as X.
   (b) Solve the discrete problem $F^l(X^l, u^l) = 0$ for X^l using initial values derived from $\hat{X}^{l-1}$ to **some level of accuracy**. A special case is to take one Newton step $F^l_X(X^{l-1}, u^l)\delta X^l = -F^l(X^l, u^l)$ and set $X^l = X^{l-1} + \delta X^l$.
   (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 state or design variables can be applied at this stage. Various approximations must often be used as discussed in [13, 10].
   (d) Determine the step length ϵ^l and update the design variables, $u^{l+1} = u^l + \epsilon^l \delta u^l$.
   (e) Estimate a new value of the the state variables $\hat{X}^l = X^l + \epsilon^l(\partial X/\partial u)\delta u^l$.
   (f) Estimate the discretization error using $\hat{X}^l$.
   (g) Use the error estimate to determine grid $l + 1$ and go to (a) above.

The step length determination in step 2(d) above could use the nonlinear elimination method in which a part of the state equations are solved more or less exactly for some subset of the state variables as discussed above. This often has a very significant advantage in that the more nonlinear state equations can be controlled so as to be kept from becoming uncomputable on the next grid. The advantages for global convergence can also be significant. This method can be combined with nonlinear continuation which "regularizes" the entire problem on early grids by, for example, using additional artificial dissipation. However, the presence of the coarse grids to obtain initial guesses for the finer grids should decrease the importance of the step size determination method used.

We note that the optimization subproblem in step 2(c) above need not be a quadratic program. We have successfully employed various nonlinear programming subproblems based on sensitivity analysis as described in [13, 10]. We note that in practice, step 2(d) must be done outside any optimization software employed to solve the subproblem because a better model of the objective and constraints (one much too costly to use in the subproblem) must be used in the line search to assure global convergence and even in some cases, the solvability of the next iterate or the existence of the required derivatives.

6 Lagrange-Newton Solution-Adaptive Design Method

If we were using the Lagrange-Newton formulation with Lagrange multipliers instead, the method would be

1. Given the initial design variables u^1 and an initial guess for the state variables $\hat{X}^1$, solve for the initial values of the Lagrange multipliers $\hat{\lambda}^1$ using equation (7).
2. For each grid $l = 1, 2, \ldots, NG$ until an optimality criterion is sufficiently small:
   (a) Discretize $\mathcal{F}(X, u)$ and $\mathcal{I}(X, u)$ on grid l.
   (b) Calculate an approximate gradient and Hessian of the Lagrangian.
   (c) Solve the system given by equation (5) for δu, δX, and $\delta\lambda$ including any inequality constraints. It may be necessary to regularize this problem with added viscosity, etc. to stabilize the process.
   (d) Determine the step length ϵ and update the variables, $u^{l+1} = u^l + \epsilon\delta u^l$, $X^{l+1} = X^l + \epsilon\delta X^l$, and $\lambda^{l+1} = \lambda^l + \epsilon\delta\lambda^l$.
   (e) Estimate the discretization error using X^{l+1}.
   (f) Use the error estimate to determine grid $l + 1$ and go to (a) above.

Once again, the line search in step 2(d) above could be used to enforce certain of the state equations.

The outer loop enables convergence to the solution of the continuous design problem. 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. Various approximations that could be used are discussed in [9, 10].

7 Variations of Lagrange-Newton Methods

We now suggest how to derive the "all at once" (AAO), "multi-discipline feasible" (MDF), and "individual discipline feasible" (IDF) methods discussed in [14] by applying the nonlinear elimination method to the Newton iteration given above.

Clearly, the AAO method is simply the use of equation (6), or equivalently equations (10) and (11), and a standard globalization method or line search for Newton's method. If during the line search we apply nonlinear elimination for the state equations, we achieve an MDF method; and if we apply nonlinear elimination for only part of the state equations, we achieve an IDF method. Many other choices are possible for the state equations and constraints to be enforced at each Newton step. Indeed, there is no *a priori* reason to expect these to correspond to discipline boundaries. The same considerations apply to inequality constraints, an important ingredient in such a method; but one not included in the MDO taxonomy. Some of these can be relaxed or modeled linearly in intermediate steps and others should be enforced as exactly as possible. We note also that in the absence of specifying the optimization method used, the "AAO", "MDF", "IDF" taxonomy is ambiguous. An "AAO" method that used an optimization method that enforced nonlinear constraints at each major iteration would in fact be an "MDF" method. Similarly, another choice of optimization code could result in an "IDF" method. Both the "MDF" and "IDF" methods in the view presented above would seem to be special cases of the method of feasible directions as discussed by Fletcher [1].

In light of the complexity of the algorithms given above, we must conclude that the MDO classification system discussed above is incomplete. Operationally, the differences between the "AAO" and "MDF" methods are relatively minor. The level of "inexactness" allowed in step 2(b) of the method of section 5 (if the step size is fixed at 1.0) or the step size determinization method in step 2(d) is all that is involved. In fact, to distinguish the methods clearly, one must draw some kind of arbitrary line in this regard. However, the nonlinearity of the state equations locally compared to the optimality conditions might make it advisable to be able to view this step as an inexact solve of the state equations. In a Newton method for the state equation, the added cost of additional Newton steps is often relatively small and the consideration of the stopping criterion boils down to how nonlinear the state equations are locally. In compressible fluid dynamics, they are often very nonlinear, which suggests a criterion that is rather tight. Some subset may be much more nonlinear than the rest, indicating that they should be solved more exactly than the rest. By relatively nonlinear, we mean that the step size in a Newton method is controlled by the nonlinear residuals for these equations; indicating that the nonlinear "scaling" for them is stronger. This can be related to the first neglected term in the Taylor series expansion on which Newton's method is based. In practice, the potential unsolvability of the more nonlinear state equations may force them to be used in the search criteria more or less exactly to make all the linearizations required at the next step well defined. This is certainly the case with boundary layer equations. In optimization this observation is often stated in rather the converse way, namely, that the constraints should be "relaxed" and not enforced until the end of the process. This probably is only true of the rather more linear constraints. In the example of boundary layer coupled CFD, the failure to enforce exactly the boundary layer equations, which are much more nonlinear, results in extremely slow convergence or even divergence and is crucial to the success of the method.

An issue of crucial importance in implementing an overall design method is the effect of the various approximations that are almost always required in the definition of the optimization subproblem. The role of inexactness in the calculation of sensitivities, constraints, etc., can also be of great importance to the overall efficiency of the method. It would further seem to be desirable to consider the nonlinear nature of the optimal control problem, e.g., whether it is well-posed or well-conditioned. Such considerations might show the use of a continuation method to be crucial to success.

A further generalization suggested by these considerations involves the use of different models of the state equations in the computation of the search direction and the determination of the step length. It is practical to use a much better model in the latter and the methods above lend themselves to these variants easily. In fact, in the case of large numbers of design variables, the asymptotic cost of a standard quasi-Newton method is by far the largest one in such a design optimization method (assuming that the linear solvers involved are reasonably good); growing at least quadratically in the number of design variables. One way to dramatically reduce the constant for this term is to simplify the function being minimized by various approximations. To reduce the asymptotic complexity, some low complexity method must be implemented to compute approximate Hessians.

This perspective also shows why one might make various choices about more or less exact satisfaction of the state equations. If some of the state equations are known to be more nonlinear than the reduced gradient conditions or than the rest of the state equations, then these should probably be enforced more or less exactly at each Newton step. As pointed out by Lanzkron, Rose, and Wilkes [3], these choices are highly problem-dependent.

A further important consideration not discussed in [14] that now presents itself has to do with additional constraints $C(X, u)$ that can involve either the state or control variables or both. These constraints can be enforced when the design variables are updated and as such can either be treated as linear or nonlinear functions of the state variables and design variables. In the former case, the nonlinear constraint will only be enforced inexactly at the new point. In the latter case, the constraint is often enforced more or less exactly by the optimization method. Considerations similar to those mentioned above regarding the state equations apply. If the constraint is highly nonlinear, it should probably be treated nonlinearly, if less nonlinear, the less expensive linear approximation should be used and could result in faster convergence of the overall algorithm. This observation points to the need for optimization methods that allow for both linear and nonlinear constraints.

We suspect that the software constraints of the state equation solver that is available will be highly determinative of the type of optimization algorithm that is feasible as well as its computational cost. If the underlying method uses Newton's method, there are clearly advantages, not the least of which is that many of the lessons about global convergence will suggest choices for an efficient optimal control algorithm. Further, issues having to do with the relative nonlinearity of the various components of the state equations are likely to have been already considered. Certainly in our experience in aerodynamics, these problem-dependent issues have consumed the vast majority of the effort and development time as well as posing the most challenging algorithmic issues. The variations discussed above that are essentially modified line search methods often in practice have a relatively small effect compared to such methods as nonlinear continuation or grid sequencing.

8 Conclusion

The "bottom line" from this discussion is that a large variety of choices can be made about enforcement of various constraints in an optimal control problem. These choices are highly problem-dependent as is the effect of these choices on global convergence. In the overall context of a solution-adaptive grid optimal control method, there are many other issues of perhaps greater weight, such as the various approximations that will be required in the calculation of the Hessian and gradient. Further, in the context of the Lagrange-Newton method, all choices of constraints to be enforced can be implemented as special cases of the nonlinear elimination method and effect only the step size selection portion of the method; which almost certainly should be implemented outside any general optimization software used. The theory and experience for nonlinear elimination methods combined with experience using Newton's method for the state equations would seem to offer guidance to those seeking the best method. Such a view highlights the importance of choosing different "models" of the objective and constraints in the optimization subproblem as opposed to the step size determination algorithm.

References

[1] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, New York, second ed., 1987.

[2] A. E. Bryson and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere, New York, 1975.

[3] P. J. Lanzkron, D. J. Rose, and J. T. Wilkes, "An Analysis of Approximate Nonlinear Elimination", SIAM Journal on Scientific Computing, 17 (1996), pp. 538–559.

[4] M. Drela, "Viscous and Inviscid Inverse Schemes Using Newton's Method", in *Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications*, AGARD Report No. 780, 1990.

[5] M. Drela, "Design and Optimization Method for Multi-element Airfoils". AIAA Paper 93-0969, 1993.

[6] 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, Aug. 1994.

[7] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall Series in Computational Mathematics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[8] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact Newton Methods", SIAM Journal on Numerical Analysis, 19 (1982), pp. 400–408.

[9] 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, July 1993.

[10] 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, Sept. 1994.

[11] O. Ghattas and J.-H. Bark, "Optimal Control of two- and three- dimensional Navier-Stokes Flows", Journal of Computational Physics, (to appear).

[12] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, Inc., San Diego, first ed., 1981.

[13] D. P. Young, W. P. Huffman, R. G. Melvin, M. B. Bieterman, F. T. Johnson, and C. L. Hilmes, "Formulations For Design Optimization", Technical Report BCSTECH-94-021-R1, Boeing Computer Services, Sept. 1994.

[14] E. J. Cramer, J. E. Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin, "Problem Formulation for Multidisciplinary Optimization", SIAM Journal on Optimization, 4 (1994), pp. 754–776.