AIAA JOURNAL
Vol. 32, No. 1, January 1994

Improving the Efficiency of Aerodynamic Shape Optimization

Greg W. Burgreen,* Oktay Baysal,† and Mohamed E. Eleshaky‡
Old Dominion University, Norfolk, Virginia 23529

The computational efficiency of an aerodynamic shape optimization procedure that is based on discrete sensitivity analysis is increased through the implementation of two improvements. The first improvement involves replacing a grid-point-based approach for surface representation with a Bezier-Bernstein polynomial parameterization of the surface. Explicit analytical expressions for the grid sensitivity terms are developed for both approaches. The second improvement proposes the use of Newton's method in lieu of an alternating direction implicit methodology to calculate the highly converged flow solutions that are required to compute the sensitivity coefficients. The modified design procedure is demonstrated by optimizing the shape of an internal-external nozzle configuration. Practically identical optimization results are obtained that are independent of the method used to represent the surface. A substantial factor of 8 decrease in computational time for the optimization process is achieved by implementing both of the design procedure improvements.

Introduction

A PROMISING new method for aerodynamic shape optimization has recently emerged. The unique feature of this design method is the use of aerodynamic sensitivity analysis to directly couple computational fluid dynamics (CFD) and numerical optimization techniques.[1] Numerous advantages exist for using sensitivity analysis to compute the design gradient information as opposed to the traditional and still popular finite difference approach. The primary advantage is manifested as the receipt of exact gradient information obtained at a much smaller computational cost as compared with the finite difference approach which, at times, may unwittingly provide highly inaccurate approximations of a gradient. The resulting CFD-based design procedure has been demonstrated to "automatically" determine optimal geometric surfaces that are not biased by intuition or experience in engineering.[1,2]

Discrete sensitivity analysis using CFD is currently an area of interest to several research groups.[2-5] However, integration of this new technology into a functional aerodynamic shape optimization procedure has been accomplished only recently by a few groups. The results from the applications of Baysal et al.[2] and Eleshaky and Baysal[6] have indicated that the design procedure obtains a final optimum aerodynamic shape via an evolution of successively improved shapes, although at a rather high computational cost. The purpose of this paper is to address the reasons for these high costs and, moreover, to demonstrate two major improvements for the design procedure that reduce these high costs to acceptable levels.

One important aspect of aerodynamic shape optimization is the representation of the surface to be optimized. The particular approach adopted to perform this task dictates the set of design variables to be used by the design procedure. Among the many possible methods to represent the shape are the following approaches: the definition of a new composite shape via a linear combination of shape functions,[7] the local perturbation of a baseline geometry by means of the superposition of a set of scaled "aerofunction" shapes,[8] and the analytical definition of the shape.[3,9] Although these approaches may replicate certain shapes using only a small number of design variables, they all suffer from a lack of generality for representing geometries that are not common in the aerodynamic community. That is, the shape functions or aerofunctions to be used for, say, designing a scramjet afterbody nozzle are not readily obvious or available without additional research or expertise. The latter approach is limited in that many complex geometric shapes do not easily yield to analytical descriptions; furthermore, if an analytical description is obtained, the resulting surfaces are constrained to a certain class of shapes.

In an effort to remain as general as possible, the present authors have treated the relative slopes of each surface grid point as design variables. This somewhat expensive practice was advantageous not only because of its flexibility and generality but also because of its direct correlation to grid generation. To reduce the number of design variables, the first improvement to the present design method is the use of a Bezier-Bernstein polynomial parameterization of the design surface. This procedure, which has found wide success in the grid generation field,[10] can accurately represent a complex surface shape with a relatively small number of geometric control points. Hence, a reduced set of design variables may be adopted for use in the design procedure.

A typical aerodynamic optimization may require hundreds of evaluations of the objective function, each of which requires an updated aerodynamic flowfield solution. In a design procedure based on sensitivity analysis, this demanding requirement for repetitive analyses may be effectively addressed by performing less costly flow predictions (approximate flow analyses) in lieu of complete CFD analyses.[1] Since these flow predictions are only approximations to the flowfield solution, an accurate full CFD analysis must be performed when new gradient information is required by the optimizer. However, these CFD analyses account for a large portion of the total CPU time required for an optimization problem. Hence, the second improvement involves the replacement of the current CFD algorithm, which is based on a conventional alternating direction implicit methodology, with a more efficient flow solver based on Newton's method.

Aerodynamic Optimization

For aerodynamic design applications, a typical constrained optimization problem may be stated mathematically as

Minimize

$$F[D, Q(D)] \tag{1}$$

subject to inequality constraints,

$$G_j[D, Q(D)] \leq 0 \qquad j = 1, NCONQ \tag{2}$$

Presented as Paper 92-4697 at the AIAA/USAF/NASA/OAI 4th Symposium on Multidisciplinary Analysis and Optimization, Cleveland, OH, Sept. 21-23, 1992; received Nov. 12, 1992; revision received June 11, 1993; accepted for publication June 14, 1993.

*Graduate Research Assistant, School of Mechanical and Aerospace Engineering. Student Member AIAA.

†Professor, School of Mechanical and Aerospace Engineering. Senior Member AIAA.

‡Postdoctoral Fellow, Department of Mechanical Engineering and Mechanics; currently at Alexandria University, Egypt. Member AIAA.

$$H_k(D) \leq 0 \qquad k = 1, NCOND \tag{3}$$

and to side constraints,

$$D_i^L \leq D_i \leq D_i^U \qquad i = 1, NDV \tag{4}$$

where F is the objective function and D is the vector of NDV design variables. For aerodynamic shape optimization, the design variables are geometric-type. That is, they describe the geometric shape of the aerodynamic configuration and, hence, influence the aerodynamic flowfield through the boundary conditions. Mathematically, this implicit dependence is denoted as $Q = Q(D)$, where Q is the vector of conserved variables representing the flowfield solution.

The $NCONQ$ aerodynamic inequality constraints G_j restrict quantities that are functionally dependent on the flowfield variables (e.g., magnitudes of force coefficients or actual flow variable values), whereas the $NCOND$ geometric inequality constraints H_k place limits on quantities that depend only on the geometric-type design variables (e.g., geometric thicknesses, angles, or curvatures). The side constraints limit the allowable values of the design variables within lower and upper bounds, D^L and D^U.

The optimization method employed in this paper is the numerical search technique known as the method of feasible directions, as implemented by Vanderplaats and Moses.[11] In the mth design iteration of the optimization process, new design points are obtained by iteratively updating the vector of design variables by

$$D^{m+1} = D^m + \alpha S^m \tag{5}$$

where α is a scalar multiplier. The NDV-dimensional search direction vector S is determined from the first-order gradient information ∇F, ∇G, and ∇H.[11] An incremental search is performed among the new design points that lie along direction S until the lowest objective function in that direction is found. This search is termed a one-dimensional search. If during the one-dimensional search the design encounters a constraint or the objective increases, the mth design iteration terminates. Then, assuming that the optimization convergence criteria have not been satisfied, a new search direction S^{m+1} is computed, and the optimization proceeds in the newly computed direction.

The gradient of the objective function ∇F and the constraints ∇G and ∇H are referred to as the sensitivity coefficients. These gradients may be evaluated by finite differences. However, this simple approach is not only computationally expensive but also may produce, at times, highly inaccurate gradient approximations due to uncertainties in choosing the proper finite difference step size. Alternatively, the sensitivity coefficients may be determined analytically by

$$\nabla F \equiv \frac{\partial F(D, Q)}{\partial D} = \frac{\partial F}{\partial Q} + \left(\frac{\partial F}{\partial Q}\right)^T \frac{\partial Q}{\partial D} \tag{6}$$

$$\nabla G_j \equiv \frac{\partial G_j(D, Q)}{\partial D} = \frac{\partial G_j}{\partial D} + \left(\frac{\partial G_j}{\partial Q}\right)^T \frac{\partial Q}{\partial D} \qquad j = 1, NCONQ \tag{7}$$

$$\nabla H_k \equiv \frac{\mathrm{d}H_k(D)}{\mathrm{d}D} \qquad k = 1, NCOND \tag{8}$$

The analytical evaluation of the sensitivity coefficients within the aerodynamic design frame has been the subject of much research recently. In particular, discrete sensitivity analysis procedures have been developed to compute the sensitivity coefficients. A detailed discussion of such procedures is presented by Eleshaky.[12] The object of aerodynamic sensitivity analysis is the efficient and accurate calculation of the sensitivity coefficients, which is imperative for practical aerodynamic design at reasonable costs.

Table 1 Aerodynamic shape optimization procedure

1) Formulate the optimization problem and select an initial vector of design variables.
2) Obtain the optimization gradient information.
   a) Perform a CFD analysis for the current design.
   b) Perform the aerodynamic sensitivity analysis, i.e., compute the sensitivity coefficients ∇F, ∇G, and ∇H.
3) Obtain the search direction vector S and initialize α.
4) Perform a one-dimensional search along S.
   a) Increment α and update D to obtain a new design point.
   b) Regenerate the computational grid for the new design.
   c) Perform an approximate flow analysis for the new design.
   d) Evaluate $F(D, Q)$, $G(D, Q)$, and $H(D)$.
   e) If the design is improved, go to step 4a; otherwise terminate search.
5) Check the optimization convergence criteria.
   If the termination criterion is met, stop; otherwise go to step 2 (begin a new design iteration).

The general optimization procedure used in the present paper is outlined in Table 1. The reader will frequently be referred back to it for clarity.

Implicit Solution of the Flowfield Equations

The conservation form of the two-dimensional Euler equations may be written as

$$\frac{\partial Q}{\partial t} = -R(Q, M) \tag{9}$$

where the residual vector is

$$R(Q, M) = \frac{\partial f}{\partial \xi} + \frac{\partial g}{\partial \eta} \tag{10}$$

and where f and g are the inviscid flux vectors in generalized coordinates. The residual R represents the net flux balance of mass, momentum, and energy across the domain. The projected cell surface areas M are determined from the computational grid via the coordinate transformation metrics. For steady flow, the residual R is equal to zero [since the unsteady term of Eq. (9) vanishes].

Applying Euler implicit time differencing to the unsteady term and linearizing the inviscid flux vectors in time, one obtains the discrete implicit formulation of Eq. (9) as

$$\left[\frac{I}{\Delta t} + \frac{\partial R}{\partial Q}\right]^n \Delta Q^n = -R(Q^n, M) \tag{11}$$

where

$$\frac{\partial R^n}{\partial Q} = \delta_\xi \left(\frac{\partial f}{\partial Q}\right)^n + \delta_\eta \left(\frac{\partial g}{\partial Q}\right)^n \tag{12}$$

The superscript n denotes the time level, ΔQ^n is the update vector $Q^{n+1} - Q^n$, and δ represents a numerical difference operator. In this study, Eq. (11) is discretized in space using a cell-centered control volume formulation. The inviscid flux vectors and flux Jacobians, $\partial f/\partial Q$ and $\partial g/\partial Q$, are evaluated using the flux-vector-splitting technique of van Leer. The cell interface Q values are determined using a second-order-accurate upwind biased MUSCL-type scheme with van Albada flux limiting.[13] The unfactored Jacobian $\partial R/\partial Q$ is a large sparse unsymmetric matrix having nine 4×4 block diagonals.

Newton's Method

By allowing the time step to approach infinity, the linear system of Eq. (11) becomes

$$\frac{\partial R^n}{\partial Q} \Delta Q^n = -R(Q^n, M) \tag{13}$$

Direct solution of this linear system obtains a "numerically exact" solution for ΔQ^n. This scheme is referred to as Newton's method and is known for its high rates of convergence.[14] Because of the nonlinear nature of the flowfield equations and the functional dependence of R on Q^n, more than one Newton iteration is necessary to drive the residual to zero. If the initial (or a subsequent) flowfield solution is close to the final solution, quadratic convergence to a steady state is obtained. However, if the initial solution is far from the final solution, the iterative process either diverges or requires many iterations to obtain quadratic convergence (and, consequently, may become somewhat expensive).

The exact inverse of the Jacobian matrix $\partial R/\partial Q$ is typically obtained by direct inversion techniques. For small two-dimensional problems, highly vectorized solvers based on lower/upper (LU) decomposition can perform this matrix inversion quite efficiently.[15] The primary disadvantage of direct inversion solvers is that the computational times and memory requirements for LU decomposition rapidly become impractical as the size of the problem increases.

Alternating Direction Implicit Schemes

The most popular approaches for solving Eq. (11) are based on approximate factorization formulations, in which the left-hand-side operator is split into two one-dimensional operators.[16] The resulting equation becomes

$$\left[\frac{I}{\Delta t} + \delta_\xi \left(\frac{\partial f}{\partial Q}\right)\right]^n \left[\frac{I}{\Delta t} + \delta_\eta \left(\frac{\partial g}{\partial Q}\right)\right]^n \Delta Q^n = -R(Q^n, M) \quad (14)$$

This spatially split two factor approximation of the left-hand side introduces $\mathcal{O}(\Delta t^2)$ factorization error, which adversely affects convergence properties of the scheme. Nevertheless, use of an alternating direction implicit (ADI) solution method results in a stable and convergent numerical scheme for small-to-moderate time step sizes. The ADI scheme involves solving a sequence of easily invertible equations. Each equation requires the inversion of a tri- or pentablock diagonal coefficient matrix, either of which can be accomplished very efficiently and requires little computational storage. The primary disadvantage of ADI schemes is their relatively slow rates of convergence to a steady state.

Aerodynamic Sensitivity Equation

Modern CFD methods provide the numerical foundations on which the present aerodynamic shape optimization procedure is based. In this section, it is shown how aerodynamic sensitivity analysis is derived from these CFD foundations.

For aerodynamic shape optimization, all of the design variables D are geometric type. Consequently, the computational grid and its coordinate transformation metrics about the aerodynamic body will assume the functional forms $X = X(D)$ and $M = M[X(D)]$, where X represents the two-dimensional vector of grid points making up the computational mesh in the physical plane, that is, $X \equiv \{\bar{x}, \bar{y}\}$. Furthermore, because of the *implicit* dependence of the flowfield Q on the design variables D, the residual R of Eq. (9) takes the following functional form:

$$R = R\{Q(D), M[X(D)]\} \quad (15)$$

This implicit relationship, which has its genesis in CFD, forms the basis of aerodynamic sensitivity analysis.

The sensitivity coefficients are an integral part of the optimization procedure (cf. step 2b of Table 1). In the discrete sensitivity analysis approach, one of two formulations can be used to compute these coefficients.[17]

Direct Differentiation Formulation

If the flow derivatives $\partial Q/\partial D$ are obtained, the sensitivity coefficients may be easily computed from Eqs. (6–8). One approach for computing the flow derivatives is developed as follows: given a steady-state flow solution, the residual R is equal to zero and may be analytically differentiated with respect to the design variables to give the sensitivity equation

$$\frac{\partial R(Q, M)}{\partial D} = \left(\frac{\partial R}{\partial Q}\right)_M \frac{\partial Q}{\partial D} + \left(\frac{\partial R}{\partial M}\right)_Q \frac{\partial M}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D} = 0 \quad (16)$$

or

$$\frac{\partial R}{\partial Q}\frac{\partial Q}{\partial D_i} = -\frac{\partial R}{\partial M}\frac{\partial M}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D_i} \qquad i = 1, NDV \quad (17)$$

where $\mathrm{d}X/\mathrm{d}D_i$ are the grid sensitivity terms. The flow derivatives $\partial Q/\partial D_i$ may be directly obtained from Eq. (17), which must be solved for NDV right-hand sides (RHS) corresponding to NDV design variables.

Adjoint Variable Formulation

Alternately, the sensitivity coefficients may be obtained using the adjoint variable formulation, which begins by substituting Eq. (17) into Eqs. (6) and (7). The resulting adjoint vectors may be conveniently defined:

$$\lambda_F^T = \left(\frac{\partial F}{\partial Q}\right)^T \left(\frac{\partial R}{\partial Q}\right)^{-1} \quad (18)$$

$$\lambda_{G_j}^T = \left(\frac{\partial G_j}{\partial Q}\right)^T \left(\frac{\partial R}{\partial Q}\right)^{-1} \qquad j = 1, NCONQ \quad (19)$$

The set of adjoint equations is thus obtained and is given by

$$\left(\frac{\partial R}{\partial Q}\right)^T \lambda_F = \frac{\partial F}{\partial Q} \quad (20)$$

$$\left(\frac{\partial R}{\partial Q}\right)^T \lambda_{G_j} = \frac{\partial G_j}{\partial Q} \qquad j = 1, NCONQ \quad (21)$$

where $\partial F/\partial Q$ and $\partial G_j/\partial Q$ are column vectors defining the partial derivatives of the objective function and the aerodynamic inequality constraints with respect to the flowfield variables. The number of linear systems to be solved is $NCONQ + 1$. The sensitivity coefficients may then be obtained by

$$\nabla F \equiv \frac{\partial F}{\partial D} - \lambda_F^T \frac{\partial R}{\partial M}\frac{\partial M}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D} \quad (22)$$

$$\nabla G_j \equiv \frac{\partial G_j}{\partial D} - \lambda_{G_j}^T \frac{\partial R}{\partial M}\frac{\partial M}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}D} \qquad j = 1, NCONQ \quad (23)$$

$$\nabla H_k \equiv \frac{\mathrm{d}H_k}{\mathrm{d}D} \qquad k = 1, NCOND \quad (24)$$

Comparing the two formulations, one finds that obtaining the sensitivity coefficients requires the solution of either NDV or $NCONQ + 1$ linear systems. Since the solution of *one* linear system for either formulation requires approximately the same amount of computational work, the formulation of the optimization problem (cf. step 1 of Table 1) dictates which method should be used to produce the sensitivity coefficients most efficiently.[17]

In this paper, the sensitivity equation is solved by computing the LU decomposition of $\partial R/\partial Q$ [or $(\partial R/\partial Q)^T$, depending on the formulation]. This procedure enjoys the distinct advantage of reusing the LU decomposition to efficiently compute the unknowns for multiple right-hand sides. That is, after the initial work of computing the inverse of $\partial R/\partial Q$, the solution for each RHS involves only inexpensive forward and backward substitutions. Therefore, the overall cost of this direct inversion procedure is relatively insensitive to the number of right-hand sides.

However, in practice, the present procedure is limited to small two-dimensional problems. The primary difficulty in extending sensitivity analysis procedures to handle large two- and three-dimensional problems is the efficient solution of the sensitivity equation. For these large cases, the memory requirements for the direct inversion (LU decomposition) techniques are prohibitive. Some recent approaches that have promise to solve these larger problems are reported in Refs. 4, 18, and 19.

Flow Solutions in the Optimization Procedure

This section outlines the underlying reasons for selecting the particular flow solution methodologies used in the optimization procedure of this paper.

Before the Sensitivity Analysis (cf. Step 2a of Table 1)

It was shown that proper derivation of the sensitivity equation is based on a steady-state CFD solution, that is, a solution with $R = 0$. Consequently, it follows that the CFD solution used to compute the sensitivity coefficients should always be highly converged. Any explicit or implicit CFD method may be used to obtain such a solution; however, it is imperative that the sensitivity equation based on this solution be a *consistent differentiation of the corresponding CFD residual with an implicit treatment of the boundary conditions.*[1,12]

In general, the *initial* CFD flowfield for the optimization process is most efficiently obtained by using current ADI methodologies. In the present study, the *remaining* CFD flowfields used to compute the sensitivity coefficients are obtained using either an ADI scheme or Newton's method. The rationale for adopting Newton's method is now discussed.

Since the computational requirements for the solution of the sensitivity equation are approximately equal to that of solving Eq. (13), the direct inversion Newton method may be used for the CFD analysis without additional memory overhead. Also, because the flow solution is continually updated during the one-dimensional search (cf. step 4c of Table 1), the final flow solution at the end of one design iteration may be a very good initial guess for the sequent design iteration. Hence, the present design methodology is ideally suited for using Newton's method to compute the CFD flowfield that is used to obtain the sensitivity coefficients.

Within the One-Dimensional Search (cf. Step 4c of Table 1)

Although highly accurate CFD solutions are necessary to compute accurate sensitivity coefficients, coarse CFD solutions may be sufficient during the numerous objective function evaluations of the one-dimensional searches. This opens the possibility of using an inexpensive flow prediction technique as an alternative to full CFD analysis.

A useful by-product of sensitivity analysis is the capability of computing an approximate flowfield solution without resorting to conventional CFD procedures.[1] This procedure is termed approximate flow analysis and is described as follows: First, the flow derivatives $\partial Q/\partial D$ at a given design point D^0 are obtained by solving the direct differentiation formulation of the sensitivity equation [Eq. (17)]. The flow solution for a neighboring design point D^* may then be approximated by

$$Q(D^*) = Q(D^0) + \sum_{i=1}^{NDV} \left.\frac{\partial Q}{\partial D_i}\right|_0 (D_i^* - D_i^0) \tag{25}$$

It has been demonstrated in Ref. 2 that this approach can successfully predict nonlinear fluid dynamic phenomena and that its accuracy is directly related to the nearness of the neighboring design point. Furthermore, the evaluations of nonlinear objective functions and aerodynamic constraints based on $Q(D^*)$ are more accurate than corresponding evaluations based on a direct linear approximation of these functions.[17] In this paper, approximate flow analyses are exclusively used to obtain the numerous flow solutions required during the one-dimensional searches of the optimization procedure. This practice has already been proven to result in significant CPU time savings.[1,2]

General Parameterization of the Grid

In aerodynamic shape optimization procedures, the shape of the body surface and its surrounding computational grid are dictated by the vector of geometric-type design variables D. This is mathematically stated as $X = X(D)$. The design variables should be selected such that the grid may be easily regenerated as the design variable vector changes (cf. step 4b of Table 1). It is thus desirable to obtain an explicit analytical function for $X(D)$, especially since the grid sensitivity terms dX/dD may then be determined analytically. Finite difference approximations of the grid sensitivity derivatives are possible but are prone to inaccuracies. A simple and general procedure for analytically relating the surface of a body to its surrounding computational grid is now described.

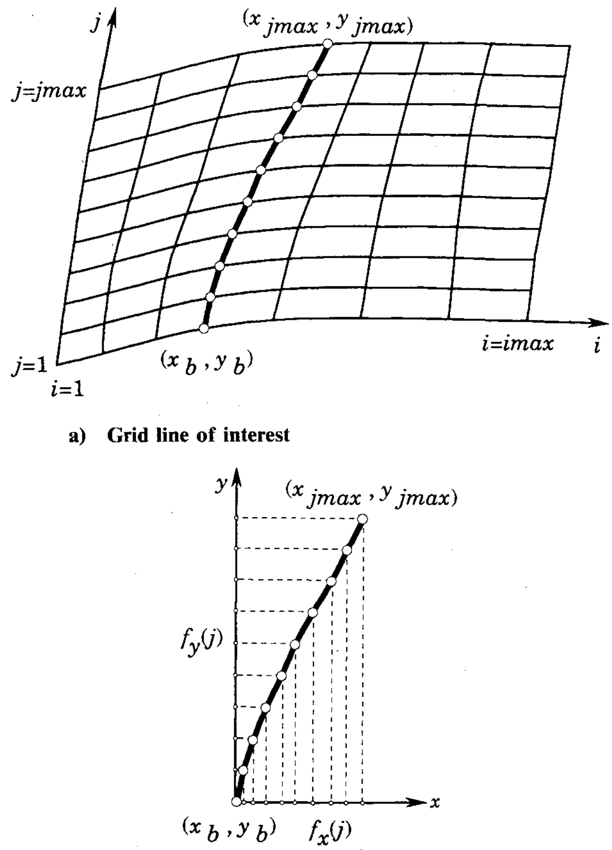
Flexible Grid Approach

The most obvious, albeit expensive, choice for geometric-type design variables is the surface grid points themselves. The task then becomes to develop an explicit relationship between the interior grid points and the design variables (in this case, the surface points).

Consider a general grid $X \equiv \{\bar{x}, \bar{y}\}$ with computational indices $\{i, j\}$ (Fig. 1a). Let the lower $j = 1$ boundary surface have $i$max nodes and be denoted by the vector $X_b \equiv \{\bar{x}_b, \bar{y}_b\}$. The grid is then parameterized in the following manner. Consider an i = const grid line that has $j$max nonuniformly distributed nodes and connects the discrete boundary nodes (x_b, y_b) and $(x_{j\max}, y_{j\max})$. The nodal distribution along this line may be projected onto the x and y axes and then characterized by two normalized distribution functions (Fig. 1b),

$$f_x(j) = \frac{x_j - x_b}{x_{j\max} - x_b} \qquad j = 1, j\max \tag{26}$$

$$f_y(j) = \frac{y_j - y_b}{y_{j\max} - y_b} \qquad j = 1, j\max \tag{27}$$



a) Grid line of interest

b) Nodal distribution functions

Fig. 1 Simple parameterization of a computational grid.

Thus, explicit relationships are obtained between the grid line's interior nodes and its boundary nodes.

Since the body surface is constantly evolving during the optimization process (cf. step 4b of Table 1), a simple adaptive grid generation scheme would have very beneficial consequences. If the surface boundary node (x_b, y_b) is displaced by $(\Delta x_b, \Delta y_b)$, a new grid line may be obtained via

$$x_{j,\text{new}} = (x_b + \Delta x_b) + f_x(j) \cdot [x_{j\max} - (x_b + \Delta x_b)] \qquad j = 1, j\max \tag{28}$$

$$y_{j,\text{new}} = (y_b + \Delta y_b) + f_y(j) \cdot [y_{j\max} - (y_b + \Delta y_b)] \qquad j = 1, j\max \tag{29}$$

Note that in Eqs. (28) and (29) each normalized distribution function is assumed to be locally invariant, and the outer boundary node $(x_{j\max}, y_{j\max})$ remains spatially fixed. Application of Eqs. (28) and (29) to each i = const grid line leads to a very simple and efficient grid regeneration procedure that accounts for a new surface shape by incrementally displacing each surface boundary node of the original grid. For small deformations of the boundary shape, the quality of the grid is not diminished.

Grid Sensitivity of the Flexible Grid

Assuming that the vector of geometric-type design variables $D \equiv \{\bar{D}_x, \bar{D}_y\}$ consists of the vector of surface boundary nodes $X_b \equiv \{\bar{x}_b, \bar{y}_b\}$, the grid sensitivity terms may be expressed by

$$\frac{\mathrm{d}X}{\mathrm{d}D} \equiv \frac{\mathrm{d}X}{\mathrm{d}X_b} \tag{30}$$

For the grid $X \equiv \{\bar{x}, \bar{y}\}$, a straightforward differentiation of Eqs. (28) and (29) (with $\Delta x_b = \Delta y_b = 0$) yields the following analytical grid sensitivity derivatives:

$$\begin{aligned} \frac{\mathrm{d}\bar{x}}{\mathrm{d}\bar{D}_x} &\equiv \frac{\mathrm{d}\bar{x}}{\mathrm{d}\bar{x}_b} = 1 - \bar{f}_x \qquad & \frac{\mathrm{d}\bar{x}}{\mathrm{d}\bar{D}_y} &\equiv \frac{\mathrm{d}\bar{x}}{\mathrm{d}\bar{y}_b} = 0 \\ \frac{\mathrm{d}\bar{y}}{\mathrm{d}\bar{D}_y} &\equiv \frac{\mathrm{d}\bar{y}}{\mathrm{d}\bar{y}_b} = 1 - \bar{f}_y \qquad & \frac{\mathrm{d}\bar{y}}{\mathrm{d}\bar{D}_x} &\equiv \frac{\mathrm{d}\bar{y}}{\mathrm{d}\bar{x}_b} = 0 \end{aligned} \tag{31}$$

This flexible grid procedure has been used with much success in previous shape optimization applications.[1,2,6] In particular, the shape was represented by a vector of design variables consisting of the local relative slope of each surface boundary node with respect to its neighboring node.

Surface Representation

The choice of surface boundary nodes as design variables leads to an extremely flexible and simple representation of a surface. However, the number of design variables (*NDV*) resulting from this choice is large. An immediate negative consequence of this is the large memory requirement for storing the *NDV* right-hand sides of the direct differentiation formulation of the sensitivity equation [Eq. (17)]. Thus, alternative methods for representing the surface need to be examined with particular attention paid to each one's compatibility within the aerodynamic design frame. Thibert[20] provides an excellent overview of the various techniques currently used in design optimization for surface representation.

Bezier-Bernstein Parameterization

The approach adopted in the present study uses a Bezier-Bernstein parameterization of the surface shape. Using this approach, a complex shape can be accurately represented with a relatively small number of geometric control points. The shape produced is smooth and does not have spurious waves between the control points. Since each grid point along the curve is defined, the simple adaptive grid generation scheme of Eqs. (28) and (29) is still applicable when using this method. The use of Bezier-Bernstein curves has also appeared in other aerodynamic design applications.[21,22]

The boundary surface contour can be represented by an Nth degree Bezier-Bernstein curve in two-dimensional space defined by

$$X_b(u) = \sum_{k=0}^{N} B_{k,N}(u)\, P_k \qquad 0 \le u \le 1 \tag{32}$$

where $X_b(u) \equiv \{\bar{x}_b(u), \bar{y}_b(u)\}$ and $P_k \equiv \{P_{x,k}, P_{y,k}\}$.

The basis functions $\bar{B}_N(u)$ are Nth degree Bernstein polynomials, which are given by

$$B_{k,N}(u) = \frac{N!}{k!\,(N-k)!}\, u^k (1-u)^{N-k} \tag{33}$$

The two curve control parameters are the normalized computational arclength u along the curve and the vector of $N + 1$ geometric coefficients P, which are called the Bezier control points.

For many reasons, the vector of Bezier control points is a natural choice for the geometric-type design variables. First, the control points have a very geometrical interpretation. That is, the Bezier curve passes through the first and last control points, and additionally the endpoint slopes may be specified exactly. Second, the formulation of Eq. (32) is mathematically simple and numerically efficient. Last, since each individual point along the Bezier curve is influenced to some degree by every control point, this formulation conveniently lends itself to an analytical computation of the grid sensitivity terms.

Grid Sensitivity Using the Bezier Representation

Since the boundary nodes X_b are dependent on the design variables D, which are taken to be the Bezier control points P_k, the grid sensitivity terms are determined by

$$\frac{\mathrm{d}X}{\mathrm{d}D} = \frac{\partial X}{\partial X_b}\frac{\mathrm{d}X_b}{\mathrm{d}D} \equiv \frac{\partial X}{\partial X_b}\frac{\mathrm{d}X_b}{\mathrm{d}P_k} \tag{34}$$

Using Eq. (32), the term $\mathrm{d}X_b/\mathrm{d}P_k$ may be more explicitly expressed as

$$\begin{aligned} \frac{\mathrm{d}x_b}{\mathrm{d}P_{x,k}} &= B_{k,N}(u) \qquad & \frac{\mathrm{d}x_b}{\mathrm{d}P_{y,k}} &= 0 \\ \frac{\mathrm{d}y_b}{\mathrm{d}P_{y,k}} &= B_{k,N}(u) \qquad & \frac{\mathrm{d}y_b}{\mathrm{d}P_{x,k}} &= 0 \end{aligned} \tag{35}$$

Thus, by combining the flexible grid technique [Eq. (31)] with the Bezier-Bernstein parameterization of the surface contour [Eqs. (34) and (35)], the grid sensitivity terms become

$$\begin{aligned} \frac{\mathrm{d}\bar{x}}{\mathrm{d}\bar{D}_x} &= (1 - \bar{f}_x) \cdot \bar{B}_N(u) \qquad & \frac{\mathrm{d}\bar{x}}{\mathrm{d}\bar{D}_y} &= 0 \\ \frac{\mathrm{d}\bar{y}}{\mathrm{d}\bar{D}_y} &= (1 - \bar{f}_y) \cdot \bar{B}_N(u) \qquad & \frac{\mathrm{d}\bar{y}}{\mathrm{d}\bar{D}_x} &= 0 \end{aligned} \tag{36}$$

These analytical expressions explicitly describe the sensitivity of the computational mesh with respect to the Bezier control point design variables.

One potential difficulty in employing Bezier curves in the present design procedure is the initial specification of the curve control parameters. In other words, the proper specification of P_k and u, which recovers the shape and nodal distribution of the initial discretized surface boundary needs to be determined. A procedure, similar to that of Ref. 22, has been developed to solve this inverse Bezier problem and was applied to the design configuration of this paper.

Results

In this paper, different optimization strategies are defined by combining the various options for shape representation and

CFD methodologies used before the sensitivity analysis. The list of proposed optimization strategies are outlined in Table 2.

The present application is an internal-external nozzle configuration whose ramp section is redesigned to maximize the axial component of the thrust vector. This configuration has been previously examined and shape optimized by Baysal et al.[2] The salient features of the problem are illustrated in Fig. 2. The initial design shape of the ramp section is a flat surface declined at a 10-deg initial expansion angle. The physical domain is discretized using 53×41 grid points (with 48 points defining the ramp surface). The optimization problem is formulated to include three aerodynamic inequality constraints ($NCONQ = 3$) and no geometric inequality constraints ($NCOND = 0$). The purpose of the aerodynamic constraints is to limit the static pressure values at the ramp and cowl tips such that no reverse flow occurs there. The surface of the ramp section is represented by a vector of design variables consisting of either 47 local relative slopes ($NDV = 47$) or the y coordinates of six of the seven Bezier control points that define the ramp shape ($NDV = 6$). The movements of all grid points as well as the Bezier control points are restricted to the y direction. The a priori specification of the degree of the Bezier curve and also the fixed axial locations of the Bezier control points does not permit the generation of every possible shape; nevertheless, ample geometric flexibility does exist to deform the ramp to realistic shapes.

Note that the design problem is formulated such that the adjoint variable formulation of the sensitivity equation is preferred since $NCONQ + 1 < NDV$.[17]

Ramp Shape Optimization

Table 3 summarizes the computational statistics associated with the shape optimization of the ramp using the different optimization strategies on a Cray Y-MP.

Two procedural points are in order here. First, the termination criterion for the CFD analyses in this design problem was the execution of 650 cycles for the ADI solver or 3 iterations for the Newton method. Both practices were necessary to consistently yield CFD residual L_2 norms smaller than $1.E-06$. Second, the α increments [cf. Eq.(5)] were chosen such that final objective functions identical to four significant digits were obtained. This rather stringent requirement is necessary to obtain an almost identical final ramp shape for each optimization strategy. (This is because the present objective function is a weak nonlinear function of the ramp's surface shape.) Final optimization results prove to be quite sensitive to the chosen α increment since α directly affects the accuracy of the approximate flow analyses.[2] In particular, for each strategy, doubling the α increment 1) causes the objective function to deviate in the fourth decimal point, 2) produces moderately different final shapes, and 3) causes a significant degradation of performance for Newton's method (due to the receipt of inferior initial guesses).

Two immediate and substantial savings are realized by reducing the number of design variables from 47 to 6 (strategy 2). A 4.3 MWord reduction of memory was achieved due to the smaller sizes of the arrays associated with the grid sensitivity terms. More significantly, the CPU cost of each flow prediction within the one-dimensional search was decreased by 60%. This is directly due to the reduction of the number of right-hand sides when solving the direct differentiation formulation of the sensitivity equation [Eq. (17)]. After including the addi-

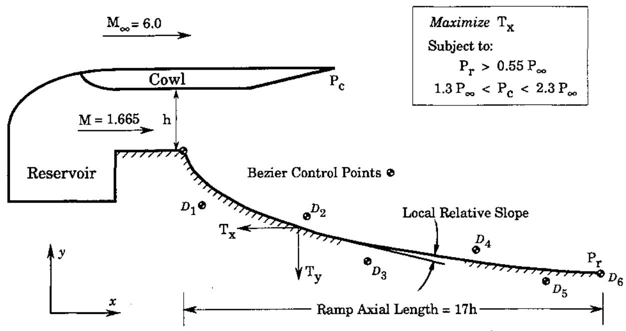


Fig. 2 Formulation of the nozzle shape optimization problem.

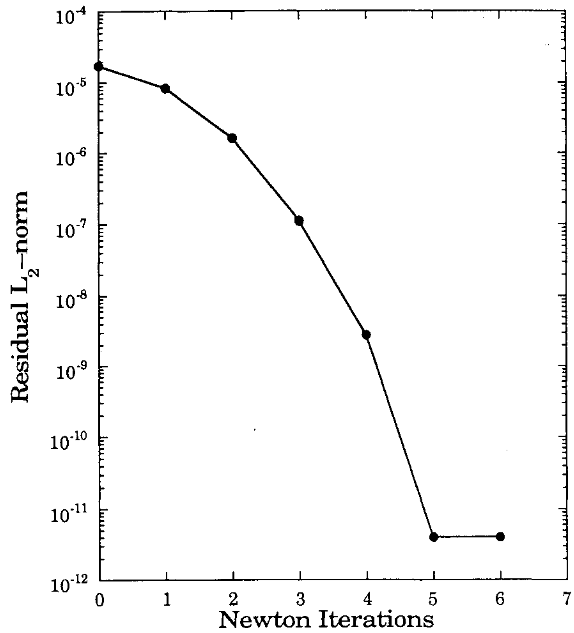


Fig. 3 Convergence history of Newton's method.

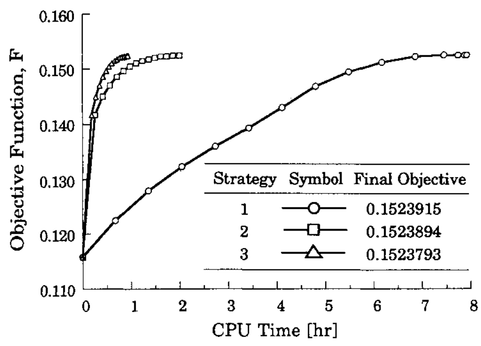


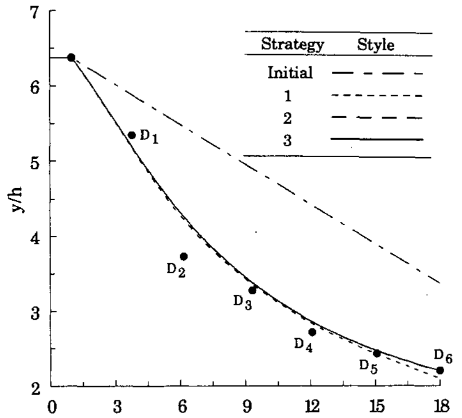
Fig. 4 Objective function time history (symbols indicate beginning of design iteration).

Table 2 Proposed strategies for the optimization procedure

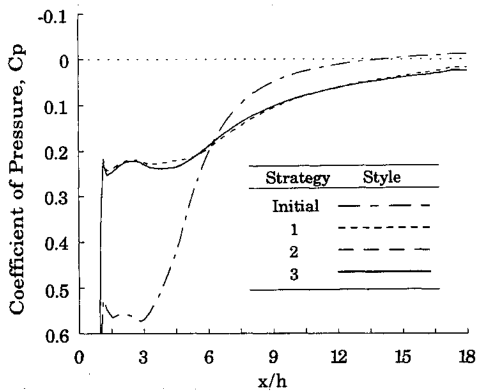
Strategy	Representation of shape	CFD methodology
1	Local slopes	ADI
2	Bezier points	ADI
3	Bezier points	Newton's Method

Table 3 Computational statistics for ramp shape optimization

Strategy	Total CPU, h	Memory, MWord	Number of sens. coeff. evaluations	CPU time per sens. coeff. evaluation, s	CPU time per CFD before sens. analysis, s	Number of flow predictions	CPU time per flow prediction, s	α increment, cf. Eq. (5)
1	8.05	11.720	14	15.5	313.0	599	40.6	0.0005
2	2.12	7.385	15	15.5	313.0	172	15.8	0.0050
3	0.97	7.385	14	15.5	40.6	166	15.8	0.0050



a) **Ramp shapes and final Bezier control point locations**



b) **C_p distribution along ramp surface**

Fig. 5 Comparison of optimized ramp design results.

tional miscellaneous computational overhead to the actual cost of a forward and backward substitution (0.071 s/RHS), one finds that the effective CPU cost for each RHS was approximately 0.6 s/RHS, which is a non-trivial cost when NDV is large.

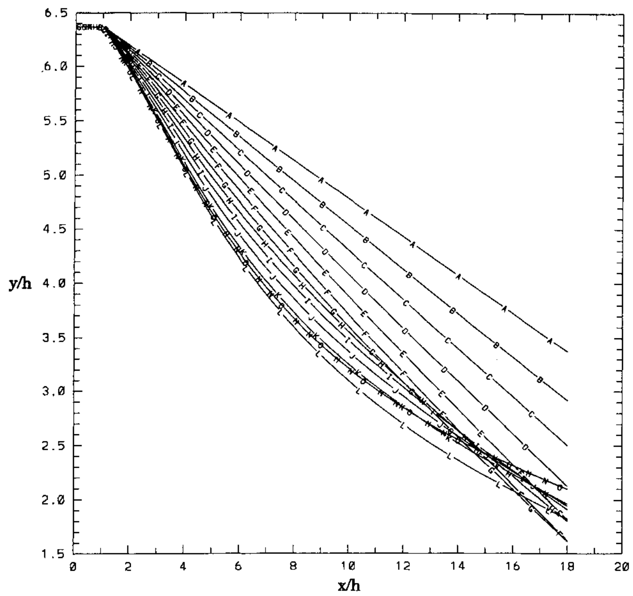
Shown in Fig. 3 is the typical convergence behavior of Newton's method during the optimization process. The observed quadratic rate of convergence indicates that good initial solutions are being provided to Newton's method; hence, it is concluded that approximate flow analysis is adequately predicting intermediate flow solutions within the one-dimensional searches. The slow and expensive rate of convergence of the ADI method is evidenced by the substantial time savings of strategy 3.

The time histories of the objective function during the optimization process are shown in Fig. 4. The symbols denote the objective function value at the beginning of each design iteration. Note the very regular and asymptotic paths of the objective functions toward their final values. This indicates that the optimizer is receiving accurate gradient information (sensitivity coefficients) for each strategy.

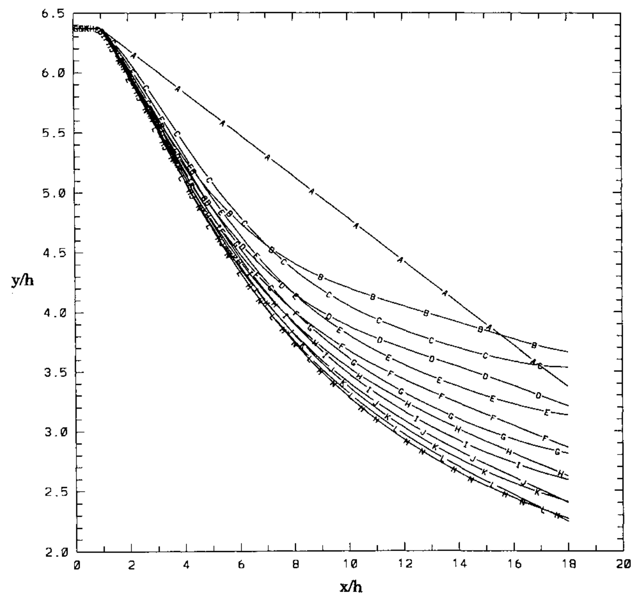
Figure 5 provides a qualitative comparison of the final ramp shapes and their corresponding C_p distributions. Note the final locations of the Bezier control points that define the ramp surface. The slight discrepancies between strategy 1 and the other two strategies may be due to an insufficient number of Bezier control points representing the initial expansion region of the ramp. The most significant observation is that practically identical optimization results are obtained that are independent of the method used to represent the surface.

An interesting feature of a shape optimization process is the evolution of the surface from its initial shape to its final optimum shape. In Fig. 6, the manner in which the ramp shape approaches its optimum appears to be markedly different between strategy 1 and strategy 2. However, a closer examination reveals that this is not true; in fact, many similarities exist in the evolution of the shape. First, note that the predominant factor influencing the component of axial thrust due to the ramp shape is the initial expansion angle at the throat exit. For the surface representation using local relative slopes (strategy 1), it is observed in Fig. 6a that the ramp first systematically approaches the optimal expansion angle before beginning to display any concavity in its shape (cf. also Fig. 4). This indicates that the physically most influential design variables (the relative slopes nearest the expansion) are the first to be driven to their optimal values during the optimization. In the final design iterations, the shape is then "fine tuned" by including influences from the rest of the design variables. This same proposition applies to the Bezier formulation of strategies 2 and 3 (Fig. 6b). Note that the physically most influential Bezier control points are the ones nearest the initial expansion. Again, the correct expansion angle is attained during the first few design iterations, and the remaining design iterations allow the ramp tip to adjust to its final shape via the physically least influential Bezier control points (those that define the aft tip section).

For strategy 2, this systematic deformation from initial to final shape is better observed in Fig. 7. In the first design iteration, the physically most influential design variables (D_1,



a) **Strategy 1 (design variables ≡ 47 local relative slopes)**



b) **Strategy 2 (design variables ≡ 6 Bezier control points)**

Fig. 6 Evolution of ramp shape (proceeding from shape A to shape N).

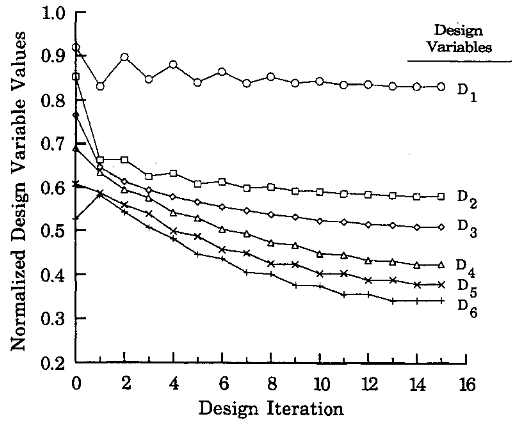


Fig. 7 History of strategy 2 design variable values (side constraints at 0 and 1).

D_2, D_3) are generously moved toward their final values, whereas the least influential design variables (D_4, D_5, D_6) change to a lesser degree. The fine-tuning process is initiated during the second design iteration and continues until the least influential design variables "damp out" to their final values. Finally, note that the side constraints (0 and 1) are not encountered during the optimization process.

Conclusions

A grid-point-based approach for surface representation was replaced with a Bezier-Bernstein polynomial parameterization of the surface. The grid-point-based representation is shown to yield a simple generalized approach for relating the computational grid to the surface boundary. This flexible grid approach led to a very simple and efficient grid regeneration procedure to account for surface shape changes. Furthermore, explicit analytical expressions for the grid sensitivity terms are obtained for this generalized approach. To reduce the large number of design variables resulting from the flexible grid approach, we adopted the Bezier-Bernstein formulation. The set of Bezier control points is shown to be a natural choice for the design variables due to its very geometrical interpretation and its ability to represent complex shapes with a relatively small number of control points. Moreover, when used in conjunction with the flexible grid approach, the Bezier formulation also yields explicit analytical expressions for the grid sensitivity terms.

Newton's method was used in lieu of an ADI method to calculate the highly converged flow solutions that are required to compute the sensitivity coefficients. The success of Newton's method in the present design procedure is due to the receipt of a good initial flow prediction at the beginning of each design iteration. These good initial predictions are obtained using approximate CFD analyses within the one-dimensional search.

The modified design procedures were demonstrated by optimizing the shape of an internal-external nozzle configuration. Appropriate computational statistics were compared to assess the effectiveness of each suggested improvement. When the surface is represented by the Bezier formulation, an almost 75% decrease in total CPU time, as well as a significant decrease in computational memory, is observed. Both savings are directly attributed to the decrease in the number of design variables. An additional 50% decrease in the total CPU time is realized by employing Newton's method in place of the ADI flow solver. Therefore, a substantial factor of 8 decrease in the computational time for the optimization process is achieved by implementing both of the suggested design improvements.

Acknowledgments

This research was supported by NASA Langley Research Center under Grant NAG-1-1188. The technical monitor was David S. Miller.

References

[1]Baysal, O., and Eleshaky, M. E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," *AIAA Journal,* Vol. 30, No. 3, 1992, pp. 718–725.

[2]Baysal, O., Eleshaky, M. E., and Burgreen, G. W., "Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equations," *Journal of Aircraft,* Vol. 30, No. 6, 1993, pp. 953–961.

[3]Taylor, A. C., III, Hou, G. W., and Korivi, V. M., "Sensitivity Analysis, Approximate Analysis, and Design Optimization for Internal and External Viscous Flows," AIAA Paper 91-3083, Sept. 1991.

[4]Elbanna, H. M., and Carlson, L. A., "Determination of Aerodynamic Sensitivity Coefficients Based on the Three Dimensional Full Potential Equation," *Proceedings of the AIAA Tenth Applied Aerodynamics Conference* (Palo Alto, CA), AIAA, Washington, DC, 1992, pp. 539–548.

[5]Frank, P. D., and Shubin, G. R., "A Comparison of Optimization-Based Approaches for a Model of Computational Aerodynamic Design Problem," *Journal of Computational Physics,* Vol. 98, No. 1, 1992, pp. 74–89.

[6]Eleshaky, M. E., and Baysal, O., "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," *Journal of Fluids Engineering,* Vol. 115, No. 1, 1993, pp. 75–84.

[7]Vanderplaats, G. N., and Hicks, R. M., "Numerical Airfoil Optimization Using a Reduced Number of Design Coordinates," NASA TM-X73151, July 1976.

[8]Aidala, P. V., Davis, W. H., Jr., and Mason, W. H., "Smart Aerodynamic Optimization," AIAA Paper 83-1863, July 1983.

[9]Sadrehaghighi, I., Smith, R. E., and Tiwari, S. N., "Grid and Design Variables Sensitivity Analyses for NACA Four-Digit Wing-Sections," AIAA Paper 93-0195, Jan. 1993.

[10]Dorrell, E. W., Jr., and Soni, B. K., "INGRID: Interactive Two-Dimensional Grid Generation," Arnold Engineering Development Center, AEDC-TR-86-49, Tullahoma, TN, Feb. 1987.

[11]Vanderplaats, G. N., and Moses, F., "Structural Optimization by Methods of Feasible Directions," *Computers and Structures,* Vol. 3, No. 7, 1973, pp. 739–755.

[12]Eleshaky, M. E., "A Computational Aerodynamic Design Optimization Method Using Sensitivity Analysis," Ph.D. Dissertation, Old Dominion Univ., Norfolk, VA, May 1992.

[13]Anderson, W. K., Thomas, J. L., and Van Leer, B., "Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," *AIAA Journal,* Vol. 24, No. 9, 1986, pp. 1453–1460.

[14]Venkatakrishnan, V., "Newton Solution of Inviscid and Viscous Problems," *AIAA Journal,* Vol. 27, No. 7, 1989, pp. 855–891.

[15]Lambiotte, J. J., Jr., "The Solution of Linear Systems of Equations on a Vector Computer," Ph.D. Dissertation, Univ. of Virginia, Charlottesville, VA, May 1975.

[16]Beam, R., and Warming, R. F., "An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation-Law-Form," *Journal of Computational Physics,* Vol. 22, No. 9, 1976, pp. 87–110.

[17]Baysal, O., and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *Journal of Fluids Engineering,* Vol. 113, No. 4, 1991, pp. 681–688.

[18]Korivi, V. M., Taylor, A. C., III, Newman, P. A., Hou, G. H., and Jones, H. E., "An Approximately Factored Incremental Strategy for Calculating Consistent Discrete CFD Sensitivity Derivatives," *Proceedings of the AIAA/USAF/NASA/OAI 4th Multidisciplinary Analysis and Optimization Conference* (Cleveland, OH), AIAA, Washington, DC, 1992, pp. 465–478.

[19]Eleshaky, M. E., and Baysal, O., "Aerodynamic Shape Optimization via Sensitivity Analysis on Decomposed Computational Domains," *Proceedings of the AIAA/USAF/NASA/OAI 4th Multidisciplinary Analysis and Optimization Conference* (Cleveland, OH), AIAA, Washington, DC, 1992, pp. 98–109; also *Computers and Fluids* (to be published).

[20]Thibert, J. J., "One Point and Multipoint Design Optimization for Airplane and Helicopter Application," *Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications,* AGARD-R-780, Paper 10, Nov. 1990.

[21]Huddleston, D. H., and Mastin, C. W., "Optimization Methods Applied to Aerodynamic Problems in Computational Fluid Dynamics," *Finite Element Analysis in Fluids, Proceedings of the Seventh International Conference on Finite Element Methods in Flow Problems,* UAH Press, Huntsville, AL, 1989, pp. 899–907.

[22]Greff, E., Forbrich, D., and Schwarten, H., "Application of Direct Inverse Analogy Method (DIVA) and Viscous Design Optimization Techniques," *Proceedings of the Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences,* edited by G. Dulikravich, Washington, DC, Oct. 1991, pp. 307–324.