AIAA JOURNAL
Vol. 34, No. 9, September 1996

Three-Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis

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

The development of an efficient and practical three-dimensional design procedure based on discrete sensitivity analysis and capable of handling large numbers of design variables is reported. The function of sensitivity analysis is to directly couple computational fluid dynamics with numerical optimization techniques, which facilitates the development of efficient direct-design methods. The Euler fluid equations are solved using a fully implicit unfactored algorithm. This new procedure is applied toward the design of three-dimensional transport wings in transonic flow. A wing geometry model that is totally based on two- and three-dimensional Bezier–Bernstein parameterizations is described. Two wing design cases are presented; one case illustrates the procedure's suitability to preliminary design, and the other demonstrates its ability to produce realistic optimal shapes, even when starting from very elementary initial geometries.

Nomenclature

A = generic coefficient matrix
arc = vector of arclength distributions
B = Bernstein polynomial
$\boldsymbol{b}$ = generic right-hand-side vector
C = preconditioning matrix
C_D, G_D = coefficient of drag
C_L, G_L = coefficient of lift
C_p = coefficient of pressure
D = design variables
$\hat{e}$ = unit basis vector
F = objective function
f = projected normalized distribution function
G = aerodynamic constraint
I = identity matrix
k = discrete computational index
L = arclength value
M, N = degree of Bernstein polynomial
P = Bezier control points
Q = conserved flow variables
R = residual
S_2, S_3 = Bezier–Bernstein surfaces
t = time
t/c = thickness-to-chord ratio
u, v = Bezier–Bernstein computational arclengths
X = computational grid
x, y, z = physical space coordinate directions
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ = vector of discrete grid points
β = trailing-edge deflection angle
δ, Δ = difference operators
θ = trailing-edge included angle
λ = adjoint vector
ϕ = generic vector of unknowns
ω = scalar relaxation parameter

Subcripts and Superscripts

n = time level
T = transpose operator
1D = concerning optimization one-dimensional search
∇F = concerning optimization gradients

Presented as Paper 94-0094 at the AIAA 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 10–13, 1994; received Feb. 26, 1994; revision received Feb. 5, 1996; accepted for publication Feb. 16, 1996.

*Graduate Research Assistant, Department of Mechanical Engineering; currently Research Assistant Professor, Department of Surgery, University of Pittsburgh Medical Center, 300 Technology Drive, Pittsburgh, PA 15219. Member AIAA.

†Professor, Department of Aerospace Engineering. Senior Member AIAA.

Introduction

AERODYNAMIC direct-design procedures have been around for a number of years and is our primary topic of interest here. These methods directly extremize some measure of merit (i.e., an objective function) for a given design problem. In this context, direct-design methods should not be confused with inverse-design methods (e.g., Ref. 1) that optimally determine the geometric shape that best matches a prescribed target distribution of some aerodynamic quantity such as pressure, Mach number, etc.

Direct-design of three-dimensional aerodynamic geometries using computational fluid dynamics (CFD) and numerical optimization techniques has only moderately evolved since first introduced in 1977. In that year, Hicks and Henne[2] extended a widely successful two-dimensional airfoil design technique to perform wing design. The major features of their innovative method include the following: 1) flow physics were predicted from the three-dimensional potential equation; 2) a finite difference approach was used to compute the sensitivity information for a gradient-based optimization code; and 3) only limited regions of the geometry were modified via perturbation shape functions. The next significant contribution to this basic methodology was by Cosentino and Holst,[3] who introduced a new spline-support technique to parametrically represent a larger portion of the design surface. Recent advances in three-dimensional direct-design methods appear in Refs. 4 and 5, in which fast, explicit CFD solvers are used to solve the Euler equations instead of the potential flow equation.

Both the performance and generality of direct-design methods are seriously impacted by the use of a finite difference approach to compute the sensitivity information. Slooff[6] correctly identifies the finite difference approach as the weakest point of a numerical optimization design strategy. In particular, the finite difference approach imposes a severe limitation on the number of design variables (NDV) to keep the computational effort within reasonable bounds (finite differenced sensitivity information requires at least NDV +1 flow analyses). In addition, the finite difference approach has the potential drawback of unwittingly introducing numerical noise into the sensitivity gradients, which leads to erroneous optimization search directions.

The problems associated with obtaining the sensitivity gradients within a direct-design method may be alleviated by either using zeroth order optimization methods that eliminate the need for the gradients (e.g., Ref. 7) or computing the sensitivity information analytically. Numerical procedures have been recently developed to efficiently and accurately calculate aerodynamic sensitivity gradients.[8–11] The computational costs of such procedures have been on the order of one complete CFD analysis. Only a few examples of three-dimensional aerodynamic sensitivity analysis have been published.[12–14]

This paper represents one of the first efforts to successfully integrate discrete aerodynamic sensitivity analysis into an efficient and functional three-dimensional direct-design procedure (see also Refs. 15–17). In particular, the two-dimensional aerodynamic shape optimization procedure of Ref. 18 is extended to treat three-dimensional geometries.

Elements of Aerodynamic Shape Optimization

The aim of aerodynamic design optimization is the minimization of an objective function $F[D, Q(D)]$ subject to constraints $G[D, Q(D)]$. Both the objective function and constraints may be nonlinear functions of the NDV design variables D and the vector of conserved flow variables Q. The major elements of the present three-dimensional direct-design code are discussed in this section, and additional details about its general optimization procedure are presented in Refs. 18–20.

Discrete Fluid Dynamic Equation

The governing fluid dynamic equations used in this study are the Euler fluid equations. After linearizing the inviscid flux vectors in time, one may write the discrete fully implicit formulation of the governing equations as

$$\left[\frac{I}{\Delta t} + \frac{\partial R}{\partial Q}\right]^n \Delta Q^n = -R(Q^n, X) \tag{1}$$

where the steady-state residual R represents the net balance of mass, momentum, and energy across the domain. In this study, Eq. (1) is discretized in space using a cell-centered control volume formulation. The inviscid flux vectors and Jacobian matrix $\partial R/\partial Q$ are evaluated using the flux-vector-splitting technique of van Leer. The cell interface Q values are determined using a spatially second-order accurate upwind-biased MUSCL interpolation with no flux limiting. Throughout this work, analytical derivatives are used for the Jacobian elements, i.e., the true Jacobian matrix is used. The numerical boundary conditions are consistently linearized and implicitly treated in Eq. (1).

Numerical Optimization Technique

The optimization algorithm employed in this paper is the method of feasible directions as applied by Vanderplaats and Moses.[21] This numerical search technique requires the first-order sensitivity gradients of both the objective function ∇F and the constraints ∇G, which are collectively referred to as the sensitivity coefficients. In this study, the sensitivity coefficients were computed analytically using an adjoint-variable formulation. For example, the analytical gradient of the objective function is expressed by

$$\nabla F \equiv \frac{\partial F(D, Q)}{\partial D_i}\hat{e}_i = \left[\left(\frac{\partial F}{\partial D_i}\right)_Q + \lambda^T \frac{\partial R}{\partial X}\frac{\partial X}{\partial D_i}\right]\hat{e}_i \qquad i \in 1, \ldots, \mathrm{NDV} \tag{2}$$

The primary obstacle to computing these analytical gradients is the accurate and efficient generation of the adjoint vector λ; this challenge has been met through the recent development of discrete aerodynamic sensitivity analysis.

Discrete Aerodynamic Sensitivity Equation

In discrete sensitivity analysis, one of two approaches can be used to form the sensitivity equation: the direct-differentiation formulation or the adjoint-variable formulation. Detailed derivations of each formulation, along with their relative merits, may be found in Refs. 10 and 19.

In this work, the adjoint-variable formulation of the sensitivity equation is exclusively used. The discrete version of the adjoint equation for the objective function takes the form

$$\left(\frac{\partial R}{\partial Q}\right)^T \lambda = \frac{\partial F}{\partial Q} \tag{3}$$

where $\partial F/\partial Q$ is the column vector defining the partial derivatives of the objective function with respect to the flowfield variables. Similar discrete adjoint equations arise for each aerodynamic constraint. In Eq. (3), $\partial R/\partial Q$ is the Jacobian matrix of the residual R and is identical to the true Jacobian matrix of the fully implicit formulation of the steady-state fluid dynamic equation. Note that the sensitivity equation is linear in its mathematical nature. Hence, no modifications or approximations can be made to either the Jacobian matrix or the right-hand-side (RHS) vectors of the sensitivity equation without compromising its true solution.

Implicit Solution Methodologies

For the present design methodology, a critical factor determining its overall computational efficiency is the use of efficient implicit solution methodologies.[18,19] This finding does not preclude the use of explicit CFD methods for flow analysis; however, it is imperative that a sensitivity equation based on such solutions incorporate a consistent differentiation of the corresponding CFD residual including boundary conditions.

The present direct-design method is ideally suited to fully implicit (Newton) methods because 1) the linear algebraic system of the fully implicit fluid dynamic equation and its numerical solution closely resembles that of the discrete sensitivity equation, and 2) during the design process, neighboring designs (and hence their flow solutions) are only incrementally different from one another. Thus, it is desired to retain a fully implicit CFD formulation within the three-dimensional design procedure. However, this in itself is a formidable numerical challenge; only a few recent examples of three-dimensional unfactored implicit CFD calculations are found in the literature.[22–26]

Hence, the major challenge for our direct-design procedure is negotiation of the large computational requirements demanded for the numerical solution of the unfactored linear systems of Eqs. (1) and (3). Several approaches from numerical linear algebra are possible to address this obstacle.

Direct Inversion Methods

Direct linear solvers based on Gaussian elimination-type decompositions suffer from large fill-in and, consequently, will result in prohibitive memory requirements and unreasonable CPU costs for practical three-dimensional problems. Out-of-core direct solvers may significantly mitigate the in-core memory requirements for these problems. However, this type of direct solver still requires large amounts of auxiliary disk storage, and if solid-state disks are not utilized, its unreasonable CPU costs may be further exacerbated by increased data transfer costs. Another viable means for reducing memory requirements is the use of domain decomposition techniques, which has been investigated recently.[13,15]

Preconditioned First-Degree Iterative Methods

Note that both the fully implicit CFD equation (1) and the sensitivity equation (3) may be considered as linear systems of the form $\boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{b}$. To solve such linear systems, a preconditioned first-degree iterative method can be written as

$$\boldsymbol{C}^n \delta\boldsymbol{\phi}^n = (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{\phi})^n \tag{4}$$

$$\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^n + \delta\boldsymbol{\phi}^n \tag{5}$$

Preconditioning leads to a more favorable condition number of $\boldsymbol{C}^{-1}\boldsymbol{A}$ compared with that of $\boldsymbol{A}$ and, hence, accelerates convergence of the iterative scheme. The choice of preconditioning matrix is crucial to the success and efficiency of this iterative scheme. Nevertheless, any preconditioning matrix that drives the RHS vector to zero may be used to obtain the correct solution to the linear system $\boldsymbol{A}\boldsymbol{\phi} = \boldsymbol{b}$. For example, Jacobi, Gauss–Seidel, or alternating direction implicit factored operators may be considered as candidate preconditioning matrices for this scheme. Korivi et al.[14,16] use this incremental iterative approach in a space-marching procedure to compute sensitivity derivatives over a three-dimensional supersonic body. The primary disadvantage of first-degree iterative methods is their relatively slow rates of convergence.

Preconditioned Conjugate Gradient Methods

Preconditioned conjugate gradient (PCG) algorithms for solving $A\phi = b$ may be regarded as second-degree iterative schemes[27]:

$$C^n \delta^2 \phi^n + C^n \delta\phi^n = (b - A\phi)^n \tag{6}$$

One inherent advantage of these schemes is that near-optimal (Newton-like) convergence rates may be sometimes attained.

Several researchers have found that preconditioned conjugate gradient-like methodologies have performed well in CFD analysis applications.[23–31] Moreover, two-dimensional direct-design optimization procedures based on PCG-like solvers have yielded significant reductions in CPU time and memory over those that utilize direct inversion solvers.[18] Consequently, the present study makes exclusive use of PCG-like linear solvers within the three-dimensional design framework. The particular solver used in this study is the generalized minimal residual (GMRES) algorithm.[32]

Like the first-degree iterative schemes, proper preconditioning is crucial to this method's performance. The preconditioner used in the present study is an incomplete lower/upper (ILU) decomposition of A as implemented by Anderson and Saad.[33] This ILU(0) preconditioner retains the same sparsity pattern as the A matrix and has been shown to yield good vector processing performance for CFD applications.[29]

Because of the lack of diagonal dominance associated with the higher-order differencing of the CFD steady-state residual, ill-conditioned linear systems result from both the time-asymptotic fully implicit CFD equation and the discrete sensitivity equation. The choice of preconditioning matrix is of vital importance to simply obtain a converged solution to these linear systems. Poor convergence is further aggravated if the effectiveness of an ILU(0) preconditioner is degraded as a result of zeroes within the bands of the coefficient matrix A. Such convergence difficulties have been reported in a CFD context by Orkwis[31] and in a design optimization context by Burgreen and Baysal.[18] In these instances, the problem was resolved by allowing fill-in to occur at the zero locations in the bands[31] or by simply reordering the equations to locate the zeroes in the outermost matrix diagonals.[18]

In the present three-dimensional design applications, the standard ILU(0)/GMRES combination failed to converge (i.e., stalled) for both the CFD and discrete sensitivity equations, and neither of the aforementioned fixes alleviated the problem. The convergence problem was finally resolved by appropriately modifying the preconditioning matrix and the RHS vectors as now described.

It is helpful to recognize that the left-hand-side (LHS) operator of Eq. (6) [and also Eq. (4)] controls the convergence process, whereas the RHS vector contains the physics of the problem and defines the accuracy of the solution. Consequently, the convergence characteristics of the preconditioned iterative schemes may be improved by choosing C to be based on a diagonally augmented version of A. In our case, we let C be defined as the ILU(0) of A_{LHS}, where

$$A_{\text{LHS}} = \frac{I}{\omega_{\text{LHS}}} + \frac{\partial R}{\partial Q} \tag{7}$$

and ω is a pseudo-time-step size. The accuracy of the solution is influenced by using the correct (or a consistent) coefficient matrix A in the RHS vector $b - A_{\text{RHS}}\phi$, where

$$A_{\text{RHS}} = \frac{I}{\omega_{\text{RHS}}} + \frac{\partial R}{\partial Q} \tag{8}$$

Options for the relaxation factors ω_{LHS} and ω_{RHS} include

$$\omega_{\text{INF}} = \infty \tag{9a}$$

$$\omega_{\text{RES}} = \omega_0 / \|R\| \tag{9b}$$

$$\omega_{\text{SER}} = \min(\omega_{\text{RES}}, \omega_{\max}) \tag{9c}$$

where ω_0 is an appropriately chosen constant, $\|R\|$ is the L_2 norm of the CFD residual, and $\omega_{\max}$ is the maximum allowable relaxation factor. In this study, $\omega_0 = 0.05$ and $\omega_{\max} = 1400$. Equation (9c) is frequently referred to as the switched evolution/relaxation strategy.

A study was performed to investigate the convergence characteristics of the PCG-like method when applied to the fully implicit CFD equation within the design process. Since the majority of CFD analyses during the optimization process require reconvergence after small design changes, the CFD analysis involved using a converged Mach 0.75 flowfield as an initial condition to compute a Mach 0.76 steady-state flowfield about a transonic transport wing ($17 \times 17 \times 43$ mesh). Two GMRES restart cycles were performed at each time step, and 20 GMRES search directions were used. The fully implicit CFD solver required 15.5 Mwords of memory and approximately 14 Cray Y-MP seconds per Newton iteration. In Fig. 1, the convergence histories for various relaxation strategies are shown. General observations from Fig. 1 include the following: 1) the use of preconditioners having no diagonal augmentation ($\omega_{\text{LHS}} = \omega_{\text{INF}}$) leads to numerical divergence; 2) preconditioners based on $\omega_{\text{LHS}} = \omega_{\text{RES}}$ become ill conditioned as $\|R\| \to 0$ and lead to stalled rates of convergence; and 3) preconditioners that retain diagonal dominance ($\omega_{\text{LHS}} = \omega_{\text{SER}}$) provide stable and convergent results. The choice of relaxation in the A_{RHS} matrix tends to affect the solution speed as well; too much relaxation ($\omega_{\text{RHS}} = \omega_{\text{SER}}$) leads to very slow linear rates of convergence.

For solution of the three-dimensional discrete sensitivity equation, the following relaxation factors were used $\omega_{\text{LHS}} = \text{const}$ and $\omega_{\text{RHS}} = \omega_{\text{INF}}$. Numerical experimentation indicated that the best convergence rates were obtained for $\omega_{\text{LHS}} = 1000$. Values much greater or lesser than this were found to result in stalled GMRES convergence. To obtain correct solutions to the linear sensitivity equation, A_{RHS} had to consist of only the true unmodified Jacobian $\partial R/\partial Q$. A solution convergence tolerance of 1E–05 was usually easily met in less than 30 GMRES restart cycles using 20 GMRES search directions.

Representation of the Design Surface

A critical element in the success of any shape optimization method is its capability to generate a great variety of physically realistic shapes. Ideally, the shape perturbation method should incorporate as much geometric flexibility as possible with as few design variables as possible.

Consider the discrete computational mesh of an elementary wing surface shown in Fig. 2. This geometrically simple wing is unswept, untwisted, and rectangular with both its chord and span equal to unity. This wing will be referred to as the unit wing. Each airfoil section of this wing is a NACA 0012 cross section that is defined in an x–z plane. Oriented at a 0-deg angle of attack, all chord lines of the unit wing lie in the $z = 0$ plane. Let the set of discrete points that describe the unit wing be denoted $\{x_0, y_0, z_0\}$.

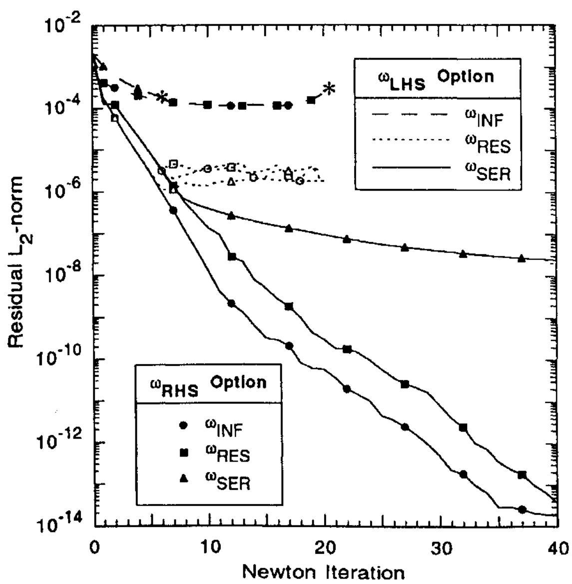

Fig. 1 CFD analysis convergence histories for various relaxation strategies.

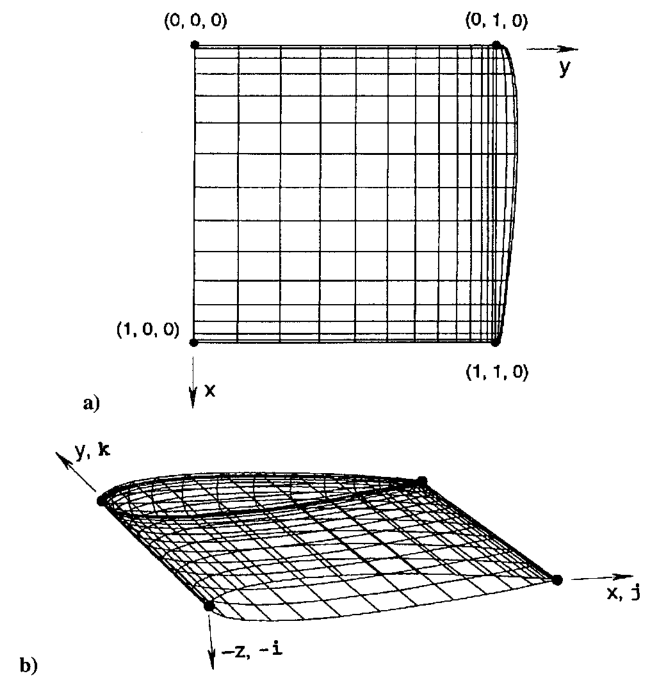

Fig. 2 Unit wing geometry: a) planform view and b) perspective view.

For design purposes, it is desired to manipulate or deform the unit wing into a new improved shape. To generate a great variety of shapes, the geometric description of a general wing should include the following features: 1) arbitrary wing section (airfoil) definitions, 2) arbitrary taper distribution, 3) arbitrary axial displacement of each airfoil section (i.e., sweep), 4) arbitrary span length, 5) arbitrary normal displacement of each airfoil section (i.e., spanwise bending), 6) arbitrary geometric twist schedule, 7) arbitrary global angle of attack, and 8) consistent and realistic treatment of wing tip region. The combined geometric deformations of features 2–4 will yield the planform shape and aspect ratio of an untwisted wing.

Wing Geometry Model

The wing geometry model was specifically developed to incorporate all of the preceding geometric features in an efficient and functional manner. Each feature is implemented as a distinct and independent geometric operation. These operations are now described.

1) This first geometric operation partially defines the airfoil sections by imposing the desired thickness and chordwise camber distributions onto the unit wing. This is accomplished by locally displacing the surface points of each airfoil section in a direction normal to its chord line. One of two approaches is used to perform this operation. The first approach limits the airfoils to the same family of shapes, whereas the second approach allows for more general airfoil definitions.

a) The airfoil thicknesses may be varied in the spanwise direction to define a wing made up of a sequence of symmetric NACA 00xx cross-sectional definitions. The wing's chordwise camber remains unchanged. Hence, only thickness scale factors as a function of span (*thkscal*) are required. The new wing is described by

$$x_A = x_0, \qquad y_A = y_0, \qquad z_A = z_0 \times thkscal(k) \tag{10}$$

(Note that the discrete computation index k runs along the y direction from the root station to the last span station before the tip region. The kth scale parameter operates on the corresponding kth airfoil section. For convenience, the discrete indices are omitted from the wing coordinates $\{x, y, z\}$.)

b) Alternatively, three-dimensional Bezier–Bernstein surfaces may be used to represent the upper and lower wing surfaces. This approach permits general distributions of both airfoil thickness and chordwise camber across the wing. More details regarding this parameterization will be given later, but suffice it to say here that the wing is described by

$$x_A = x_0, \qquad y_A = y_0, \qquad z_A = f(u, v, P) \tag{11}$$

2) Since each airfoil section of the unit wing has a chord of unity and also has its leading-edge point located on the y axis, the taper distribution may be efficiently handled by the specification of chord scale factors as a function of span (*chdscal*). This operation will simply shrink or enlarge the chord length of each spanwise airfoil section, via

$$x_B = x_A \times chdscal(k), \qquad y_B = y_A, \qquad z_B = z_A \tag{12}$$

At this point, all airfoil shapes of the wing have been fully defined.

3) The spanwise axial and normal displacements of the wing are handled by prescribing for each airfoil section the x and z locations of a specified reference point that lies on the chord line (*fchd*). In this paper, the aerodynamic center of a NACA 0012 cross section (i.e., the quarter-chord) is selected as the reference chord point. This operation requires two translation distributions as a function of span (*trnx* and *trnz*) for specifying the x and z locations of *fchd*. In addition, the taper distribution is considered to be centered about the *fchd* reference point and requires including a corresponding axial displacement:

$$\begin{gathered} x_C = x_B + trnx(k) - fchd \times chdscal(k) \\ y_C = y_B, \qquad z_C = z_B + trnz(k) \end{gathered} \tag{13}$$

4) Since the unit wing's root station lies in the $y = 0$ plane, the half-span length may be simply handled through a single scalar multiplier (*spn*):

$$x_D = x_C, \qquad y_D = y_C \times spn, \qquad z_D = z_C \tag{14}$$

At this point, a wing with complete airfoil definitions, planform shape, and spanwise bending has been defined. This was achieved through the systematic application of scaling factors and spatial translations to the unit wing.

5) The wing's geometric twist is obtained by locally rotating each airfoil section according to a twist distribution that is defined as a function of span (*twst*). Each airfoil section may be rotated about a specified reference chord point (*ftwst*); in this study, the quarter-chord location was selected:

$$\begin{aligned} x_E = {} & +(x_D - xtwst) \times \cos[twst(k)] \\ & + (z_D - ztwst) \times \sin[twst(k)] + xtwst \end{aligned} \tag{15a}$$

$$y_E = y_D \tag{15b}$$

$$\begin{aligned} z_E = {} & -(x_D - xtwst) \times \sin[twst(k)] \\ & + (z_D - ztwst) \times \cos[twst(k)] + ztwst \end{aligned} \tag{15c}$$

$$xtwst = (ftwst - fchd) \times chdscal(k) + trnx(k) \tag{15d}$$

$$ztwst = trnz(k) \tag{15e}$$

6) The angle of attack (*aoa*) is imposed by rotating the entire wing as a rigid body about the root section quarter-chord location. After the appropriate mathematical modifications have been made, this geometric transformation also may be described by Eqs. (15).

7) Finally, the new wing tip region is generated by applying analogous operations 1–6 with extrapolated geometric quantities to the unit wing tip region.

At this point, the complete wing shape has been generated. Summarizing, a new wing shape has been derived from the unit wing shape by applying a sequence of geometrical deformations based on five spanwise parameter distributions (*thkscal, chdscal, trnx, trnz,* and *twst*) and four scalar parameters (*spn, aoa, fchd,* and *ftwst*). Since the design shape of the wing depends on these parameter distributions, the manner in which these distributions are represented will dictate the type and number of design variables to be used in the shape optimization procedure.

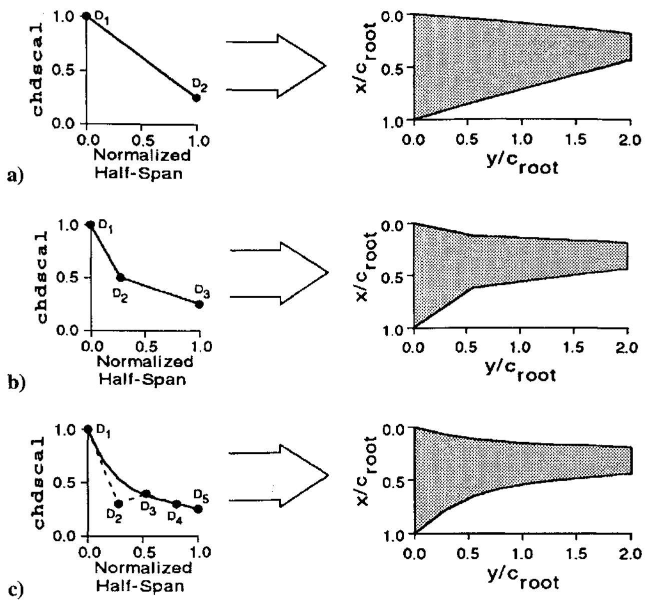

Fig. 3 Parameter distributions and their resulting planform shapes: a) linear root-to-tip variation, b) three-point piecewise continuous, and c) fourth-degree Bezier–Bernstein.

The most general treatment of the spanwise parameter distributions would be to assign a parameter value (i.e., a design variable) to each discrete spanwise station, but this approach has two obvious disadvantages. First, this approach would yield a large number of design variables, which would adversely impact the computational memory and work requirements of the design procedure. Second smoothly continuous parameter distributions are by no means guaranteed. In fact, if the design variable values of two adjacent stations are very discrepant (discontinuous), a poor aerodynamic design would likely be produced.

In many cases, a parameter distribution may be sufficiently described using a piecewise linear variation. For example, a linear taper schedule may be efficiently prescribed using only two design variables (see Fig. 3a):

$$chdscal(k) = D_1 \times [1 - y(k)] + D_2 \times y(k) \qquad (16)$$

A more general taper schedule may be produced by introducing more interior interpolation locations (see Fig. 3b). This approach is naturally suited to model geometric features that are typically prescribed in a piecewise continuous fashion, such as planform breaks, etc.

A novel approach for representing the spanwise parameter distributions has been adopted in this study. This approach employs two-dimensional Bezier–Bernstein parameterizations of the spanwise distributions (see Fig. 3c). This approach has several advantages including the following: 1) the possibility exists of modeling smoothly continuous variations; 2) a relatively small number of design variables can produce a wide range of parameter distributions; and 3) the design variables take on very geometrical interpretations.

A Bezier–Bernstein parameterization for representing two-dimensional design shapes in a direct-design procedure has been described by Burgreen et al.[18,19] An Nth-degree Bezier–Bernstein curve is defined by

$$S_2(u) = \sum_{n=0}^{N} B_{n,N}(u) \cdot P_n \qquad 0 \le u \le 1 \qquad (17)$$

The basis functions are Nth-degree Bernstein polynomials, which are given by

$$B_{n,N}(u) = \frac{N!}{n!(N-n)!} \cdot u^n \cdot (1-u)^{N-n} \qquad (18)$$

The Bezier control parameters consist of geometric coefficients P and the normalized computational arclength u. The Bezier control points P have been shown to be a natural choice for the design variables.[19]

As mentioned earlier, it is proposed to use a Bezier–Bernstein parameterization of the upper and lower surfaces of the unit wing to impose the desired wing thickness and chord camber definitions. A three-dimensional surface can be represented in the Bezier–Bernstein framework via a tensor product scheme, which is basically a bidirectional curve scheme. The three-dimensional surface has the form

$$S_3(u, v) = \sum_{n=0}^{N} \sum_{m=0}^{M} B_{n,N}(u) \cdot B_{m,M}(v) \cdot P_{nm} \qquad 0 \le u, v \le 1 \qquad (19)$$

The Bezier control points P_{nm} are arranged in a bidirectional network (see Fig. 4). In this paper, the design variables for each surface are taken as the z components of the 25 interior Bezier control points, i.e., all control points except those located on the wing's leading and trailing edges. The three-dimensional surface representation retains all of the geometrical features and computational advantages of the two-dimensional version (see Ref. 19 for more details).

Grid Adaptation Procedure

Once a new wing shape has been defined, it remains to construct the surrounding computational grid about the wing. An approach similar to that described in Ref. 19 is adopted; namely, the original surrounding grid is spatially adapted to account for the new wing shape. The spatial adaptation experienced by a typical grid line, which is described by $i$max discrete nodes, is depicted in Fig. 5.

In Ref. 19, a projected normalized distribution function defined by

$$f_x(i) = \frac{x_i - x_b^{\text{old}}}{x_{i\max} - x_b^{\text{old}}} \qquad i = 1, i\max \qquad (20)$$

is used to parameterize each surface-normal grid line in terms of the x coordinate. Each grid line is then adapted to include the new surface boundary shape via the following relationship:

$$x_i^{\text{new}} = x_b^{\text{new}} + f_x(i) \cdot \left(x_{i\max} - x_b^{\text{new}}\right) \qquad (21)$$

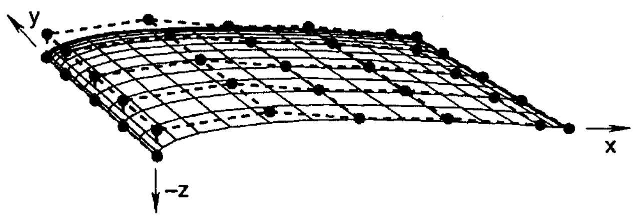

Fig. 4 Three-dimensional Bezier–Bernstein representation of the unit wing upper surface.

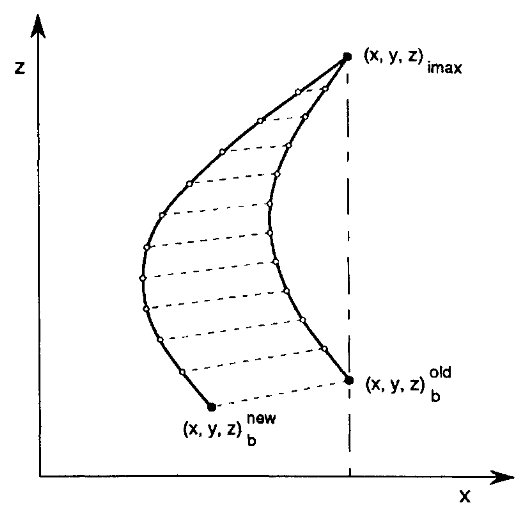

Fig. 5 Spatial adaptation of a typical surface-normal grid line.

The normalized distribution function is assumed to be locally invariant, and the outer boundary point $(x, y, z)_{i\max}$ is assumed to be spatially fixed. Relationships analogous to Eqs. (20) and (21) govern the adaptation of the normal grid line in terms of the y and z coordinates.

However, numerical problems arise if one of the coordinate values of both the outer boundary point and the surface boundary point are equal or nearly equal (cf. Fig. 5). If the denominator of Eq. (20) is identically zero, the normalized distribution function is undefined; if the denominator is very nearly zero, roundoff errors will introduce numerical noise into the adaptation procedure. This problem is circumvented by adopting a new arclength-based approach for grid adaptation. The new adapted normal grid line may be described by

$$x_i^{\text{new}} = x_i^{\text{old}} + [1 - \text{arc}(i)] \cdot \left(x_b^{\text{new}} - x_b^{\text{old}}\right) \tag{22}$$

where

$$\text{arc}(i) = \sum_{\iota=2}^{i} L_\iota \Big/ \sum_{\iota=2}^{i\max} L_\iota \tag{23a}$$

$$L_\iota = \sqrt{(x_\iota - x_{\iota-1})^2 + (y_\iota - y_{\iota-1})^2 + (z_\iota - z_{\iota-1})^2} \tag{23b}$$

Relationships similar to Eq. (22) may be written for the y and z coordinates. Note that if the surface boundary point is not relocated, then the grid line simply retains its original shape.

Finally, it is stated that all geometric deformations of the wing geometry model as well as the arclength-based grid adaptation procedure are analytically differentiable with respect to the wing model control parameters. Grid sensitivities of the wing surface points may be obtained via a chain-rule differentiation that encompasses each geometric operation.

Results

The primary intention of this section is to demonstrate the numerical capabilities of the present direct-design methodology. Toward this end, a design application of interest to the aerodynamic community is chosen, namely, transonic wing design. Two relatively simple wing design cases are considered. Unlike many of the previous wing design efforts of other researchers, the optimized wings predicted in this study have final shapes that differ considerably from their initial shapes.

The initial wing geometry is taken to be the unit wing oriented at an angle of attack of 3 deg and with a half-span length of 2 root chords. For each design case, the wing shape is optimized for inviscid transonic flow conditions. The computational domain about the wing is a coarse $17 \times 17 \times 43$ C–O grid with parabolic singular lines located at the leading and trailing edges of the wing tip. The boundary conditions at the parabolic singular lines and the coincident wake planes are implicitly treated. The coarseness of the mesh was necessitated to keep computational costs within reason during the design process. In actual detailed wing design applications, it would always be prudent to critically evaluate any improved design using higher-resolution meshes and more advanced flow physics models.

The basic wing optimization problem is formulated as follows.

Maximize:

$$C_L/C_D \tag{24}$$

Subject to aerodynamic constraints:

$$C_L \geq G_L \tag{25a}$$

$$C_D \leq G_D \tag{25b}$$

Geometric constraints at 0.00, 0.53, and 0.98 semispan stations:

$$5\,\text{deg} \leq \theta_{0.90\text{chord}} \leq 20\,\text{deg} \tag{26a}$$

$$5\,\text{deg} \leq \theta_{0.98\text{chord}} \leq 20\,\text{deg} \tag{26b}$$

$$\beta_{TE} \leq 12\,\text{deg} \tag{26c}$$

where θ is the included angle formed between the trailing-edge point and the upper and lower surface coordinates at the specified chord location. The term β is the mean angle of deflection of the trailing edge relative to the wing's angle of attack. No constraints are imposed on the wing volume or airfoil section areas. Both wing design cases in this work are performed at freestream Mach number of 0.75. (Additional wing design cases using the present design methodology for supersonic flow conditions are presented in Refs. 20 and 34.)

Different combinations of constraints and design variables are used to obtain different final wing shapes. The choice of aerodynamic constraint values G_L and G_D is critical in driving the wing design toward reasonable shapes. Note that there is no a priori guarantee that constraints will be satisfied during the optimization process or by the final design. Lift and drag coefficients are both computed from surface pressure integration, in spite of its known inaccuracy for coarse inviscid meshes.[35] In this study, inaccurate absolute drag values are manifested in the low computed values of C_L/C_D. However, it has been shown that the success of numerical optimization more crucially depends on an accurate prediction of the increments of lift and drag than it does an accurate prediction of their absolute values.[36] The number of design variables will be dictated by both the choice of included wing deformation operations and the method of representation of the spanwise distributions. For both cases, the number of design variables is much greater than the number of aerodynamic constraints; therefore the adjoint-variable formulation of the sensitivity equation should most efficiently obtain the sensitivity coefficients (for more details, see Refs. 10 and 19).

Geometrically Flexible Wing Design

The first design case is formulated to optimize a transonic wing that employs almost-full geometric flexibility of the wing geometry model. In particular, the spanwise distributions *chdscal, thkscal, trnz, trnx*, and *twst* are represented using fourth-degree Bezier–Bernstein parameterizations [cf. Eq. (17) and Fig. 3c]. For each distribution, the value of the Bezier control point located at the root section is held fixed, and the remaining four outboard control points are treated as design variables. In addition, the half-span length parameter *spn* is taken as a design variable. The wing's 3-deg angle of attack is held fixed throughout the optimization. Thus, the total number of design variables used to describe this wing is 21 (i.e., NDV = 21). Finally, the values of $G_L = 0.9$ and $G_D = \infty$ are used in the aerodynamic constraints, Eq. (25).

The wing optimization generates a quite unexpected shape, which bears no slight resemblance to a seabird's wing (Fig. 6). Although the structural integrity of such a shape is questionable, the design does possess some merit as a preliminary design concept. An upper surface shock exists across the entire wing span, and the lower surface is shock free. [The careful reader will notice that Fig. 6 (and later Fig. 9) gives the impression that the symmetry boundary condition is not being enforced; this is not the case—but is a postprocessing plotting discrepancy that results from incorrect extrapolation of cell-centered pressure values to the plane of symmetry.] The final design attains a $C_L/C_D = 6.877$ and a $C_L = 0.926$. The complete optimization required 4.58 h on a Cray Y-MP and 19.6 Mwords of memory.

To further visualize the geometric subtleties of this design, the final spanwise distributions along with their corresponding Bezier control points are shown in Fig. 7. All design variables were given a large range of side constraint bounds, and no side constraints were active or violated during the optimization. The feasibility and efficiency of using Bezier representations for the spanwise distributions is clearly demonstrated. In fact, this final design suggests that the degree of geometric flexibility of the wing needs to be reduced to produce more realistic results. Historically, this is not a type of correction commonly called for in wing design procedures.

Transport Wing Design

The degree of geometric flexibility is reduced for the second wing design case, and the complete optimization is carried out in three distinct stages. Each stage yields an optimized design for its given problem formulation. The optimization problem for stage 1 is identical to that of the previous case except the distributions *trnz, trnx*, and *twst* are here represented using a linear root-to-tip variation [cf. Eq. (16) and Fig. 3a]. The linear distribution of *trnz* is equivalent to

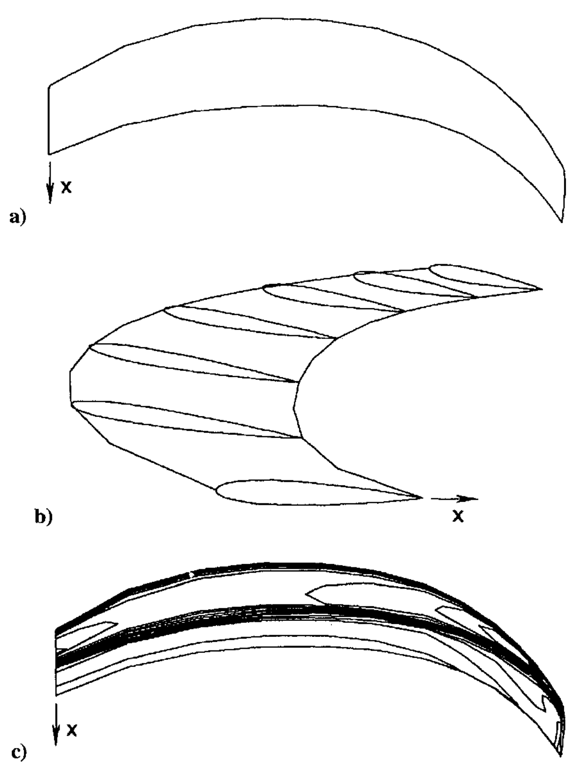

Fig. 6 Optimized geometrically flexible wing: M_∞ = 0.75 and α = 3.0 deg: a) planform view, b) perspective view, and c) upper surface pressure contours.

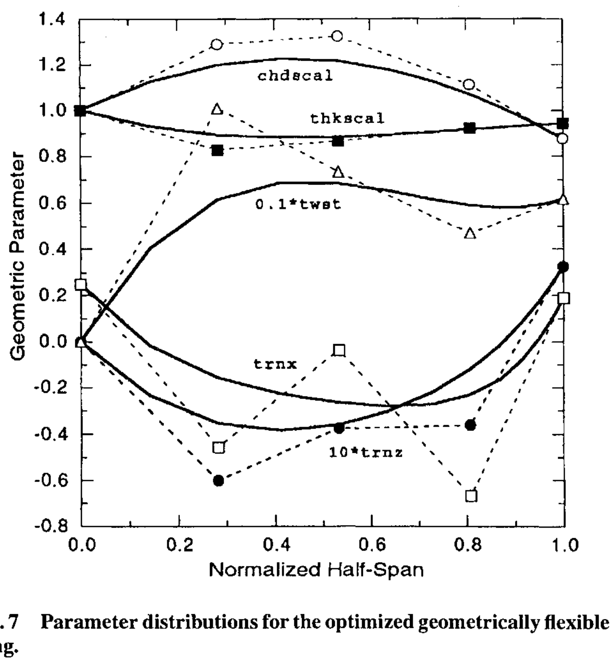

Fig. 7 Parameter distributions for the optimized geometrically flexible wing.

the specification of wing dihedral, and *trnx* now effectively dictates the sweep angle of the wing's quarter-chord line. Upper side constraints are placed on the span length and tip twist angle; namely, *spn* must be less than 2.5 root chords, which is typical of transport wings, and *twst* at the tip must be less than +0.1 deg, which prevents severe wash-in of the wing tip. Stage 2 of the optimization is simply a continuation of stage 1, but with $G_L = 0.35$ [cf. Eq. (25)]. The number of design variables for both the first and second stages is 12 (i.e., NDV = 12). Stage 3 incorporates a more general airfoil definition by replacing the *thkscal* distribution with three-dimensional Bezier–Bernstein parameterizations of both the upper and lower wing surfaces [cf. Eq. (19) and Fig. 4]. The values of $G_L = 0.9$ and $G_D = 0.04$ are used in the aerodynamic constraints,

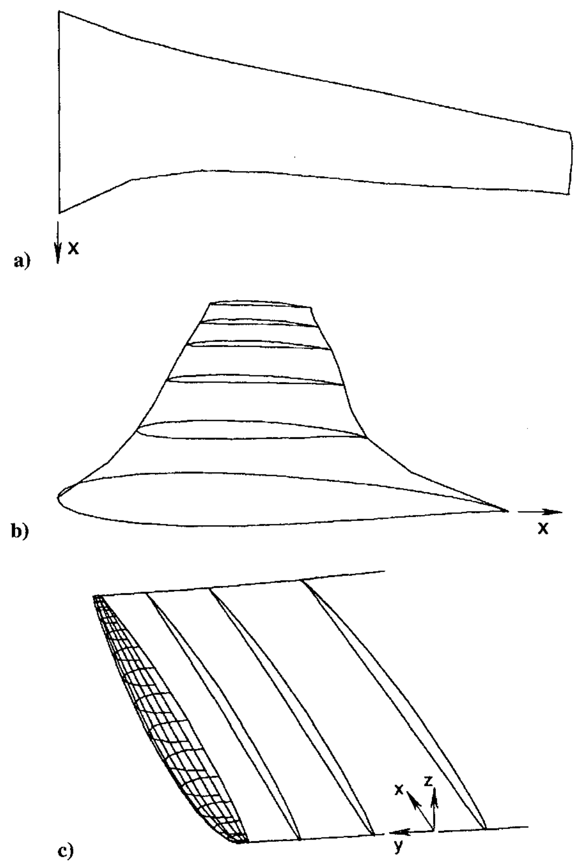

Fig. 8 Optimized design of a transport wing: M_∞ = 0.75 and α = 3.0 deg: a) planform view, b) perspective view, and c) wing tip region.

Eq. (25). The number of design variables for the third stage is 58 (i.e., NDV = 58).

The final optimized wing shape of the Mach 0.75 design is shown in Fig. 8. The geometrical features of the wing include an aspect ratio of 9.71, a taper ratio (tip chord/root chord) of 0.31, and a quarter-chord sweep angle of 9.6 deg. The optimized wing dihedral is +2.05 deg. The linear twist distribution is superimposed onto the 3-deg wing angle of attack and results in angles of incidence of +3.000 deg at the root and +3.095 deg at the tip. Figure 8b indicates that airfoil sections having slight supercritical characteristics exist along the half-span length, which is 2.5 root chords long. The wing exhibits the following airfoil section thicknesses (t/c): 11.7% at 0.0 semispan, 8.2% at 0.28 semispan, 4.1% at 0.63 semispan, and 4.2% at 0.95 semispan. Figure 8c shows that the wing tip was treated in a consistent and realistic manner. The only active geometrical-related constraints of the final design (none were violated) include the tip *twst* upper side constraint, the *spn* upper side constraint, and the minimum $\theta_{0.98\text{chord}}$ geometrical constraint at the wing tip station. Other than these influences, the wing shape was not biased in any geometrical way to attain this final optimized design.

The aerodynamic flowfield generated by this wing is no less impressive. The surface pressure contours ($\Delta C_p = 0.071$) and selected C_p distributions are shown in Fig. 9. An upper surface shock lies at approximately 65% chord along the majority of the span and then weakens and disappears at the far outboard stations. The lower surface elicits a well-behaved, shock-free flow pattern. The three-dimensional character of the flowfield is clearly observed. The optimized Mach 0.75 wing at 3-deg angle of attack attains a $C_L/C_D = 17.778$ and a $C_L = 0.794$. Further analysis indicates that this wing generates its maximum C_L/C_D when oriented at 1-deg angle of attack with a $C_L/C_D = 22.082$ and a $C_L = 0.552$.

The history of the aerodynamic coefficients during the optimization process is shown in Fig. 10, and the corresponding evolution of the wing planform shape for stages 1 and 2 (the planform shape only minutely changed during stage 3) is shown in Fig. 11. The

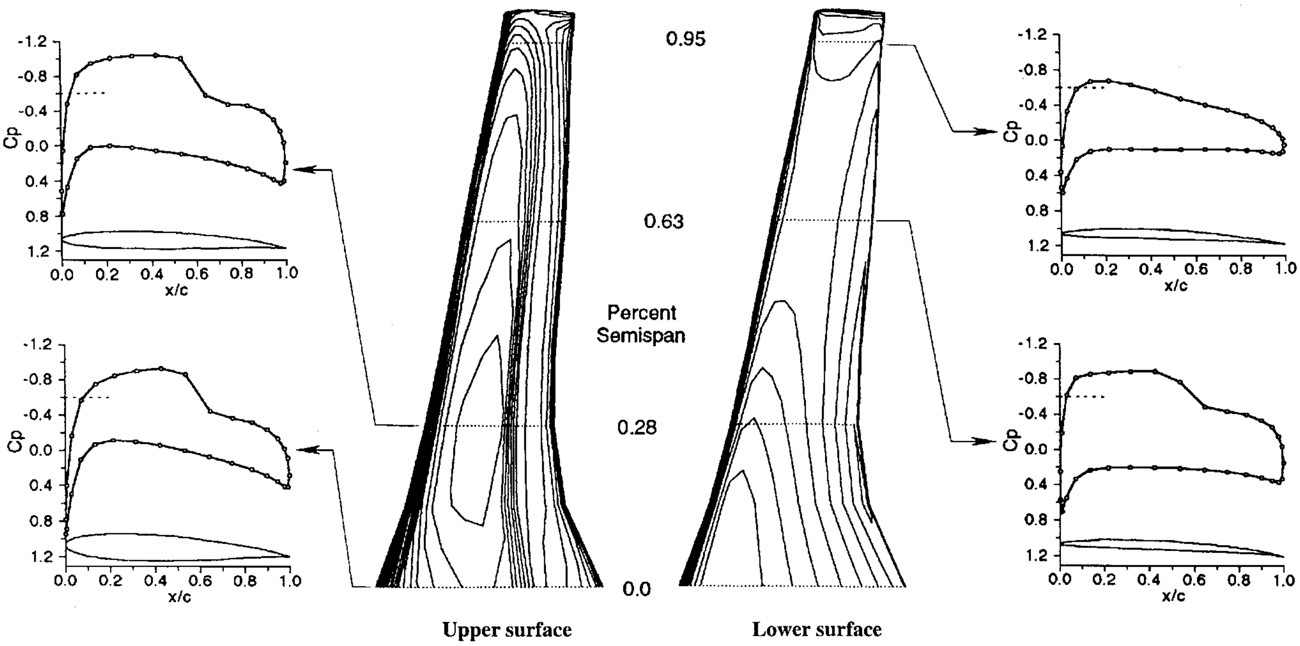

Fig. 9 Surface pressure contours and C_p distributions for transport wing design: M_∞ = 0.75 and α = 3.0 deg.

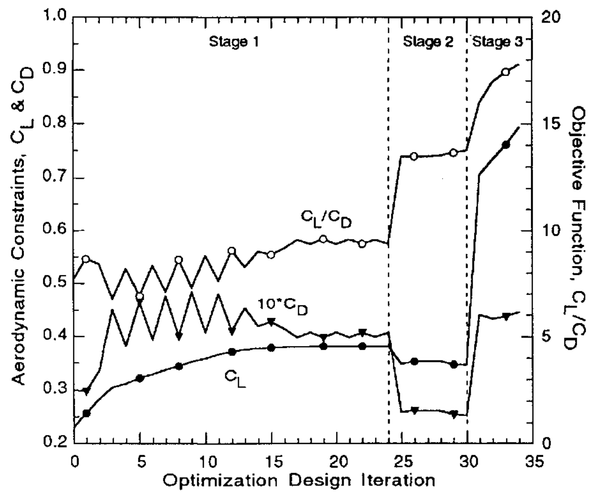

Fig. 10 History of the aerodynamic coefficients for the Mach 0.75 transport wing optimization.

choice of maximizing C_L/C_D combined with a violated C_L constraint proved to best provide an optimization search direction that led to nontrivial wing shapes. This combination kept C_D low without the explicit need for a drag constraint. Other objective function/constraint combinations generally resulted in poor designs because the gradient-based optimizer was prematurely stranded at a local maximum or terminated by conflicting constraints. By relaxing the C_L constraint in stage 2, the design method was briefly free to significantly increase C_L/C_D in an unconstrained optimization. The primary geometric changes observed during stage 2 (see Fig. 11b) were an increase in the taper ratio, an overall thinning of the wing thickness, and a sweeping back of the planform, which combine to reduce drag substantially. Stage 3 allowed for the formation of arbitrary airfoil section shapes as a result of the use of the three-dimensional Bezier parameterizations of the wing surfaces. Significant increases in C_L are observed as the near-supercritical airfoil shapes were formed. Attempts to include the Bezier surface parameterization from the beginning of the optimization resulted in poor designs having near-unit-wing planforms but with supercritical airfoil shapes. This is because the sensitivity coefficients associated with the airfoil section design variables

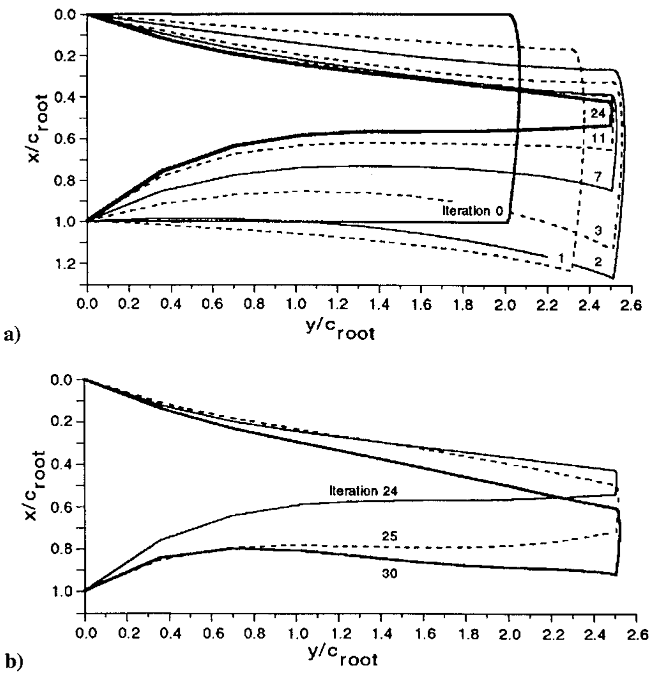

Fig. 11 Evolution of the planform shape during the Mach 0.75 transport wing optimization: a) stage 1 and b) stage 2.

overwhelmed the comparatively lesser influences associated with the other wing deformations. Finally, note that the final wing design equally violates both of the stage 3 aerodynamic constraints; this typically occurs if conflicting violated constraints compete with one another.

The computational aspects of this design case deserve detailed consideration. The complete optimization required 35 design iterations, each of which called for a sensitivity analysis. A total of 322 highly converged three-dimensional CFD analyses were performed during the optimization; this includes 35 $\mathrm{CFD}_{\nabla F}$ and 287 $\mathrm{CFD}_{\mathrm{1D}}$ analyses. [The terms $\mathrm{CFD}_{\nabla F}$ and $\mathrm{CFD}_{\mathrm{1D}}$ denote the CFD analysis performed before each sensitivity analysis and within the one-dimensional searches, respectively (see Ref. 18 for more details).] The CFD flow solutions were converged to residual L_2 norms of $\mathrm{TOL}_{\nabla F} = 1E-09$ and $\mathrm{TOL}_{1D} = 1E-08$. Each $\mathrm{CFD}_{\nabla F}$ required

118.6 s on a Cray Y-MP; each CFD_{ID} required 75.4 s; and each sensitivity analysis required 283.2 s. The complete optimization required a total of 10.26 h on a Cray Y-MP. The total CPU time may be accounted for by the following percent usage: $CFD_{ID} = 59\%$ of the total CPU time, sensitivity analyses = 27%, $CFD_{\nabla F} = 11\%$, and the remaining 3% was expended on the optimization algorithm and data transfer operations. The required memory was 18.3 Mwords for stages 1 and 2 (NDV = 12) and 29.8 Mwords for stage 3 (NDV = 58).

The most noteworthy aspect of the present design procedure is its demonstration of the essential role that discrete sensitivity analysis plays in the development of an efficient and practical three-dimensional design procedure that involves a large number of design variables. If a finite difference approach had been adopted for the calculation of the sensitivity coefficients, the total CPU time required for the sensitivity analyses alone is estimated to be 28 h on a Cray Y-MP.

This section is closed with a brief comment about the uniqueness of the optimized wing designs. We do not claim attainment of global optimality for either of the present designs. Rather, the two cases presented were the most interesting and instructive of all of the cases we investigated. Our limited experience has consistently indicated that design spaces for transonic wing design are topologically complex and full of local maximums. As alluded to by earlier statements, seemingly minor changes in a problem formulation may (and frequently did) result in radically different final designs. Thus, it is imperative that design engineers acquire some knowledge or feel of their problem's design space. This knowledge typically derives from experiential, intuitive, and heuristic means—although it is the cumulative gain from all three modes that best enables designers to effect useful design improvements.

Conclusions

The latest developments toward a practical three-dimensional direct-design procedure have been reported. Distinctive aspects of this design procedure include the following: 1) using a fully implict algorithm, the flow physics are predicted from the Euler equations; 2) discrete sensitivity analysis is used to compute the optimization gradient information; and 3) the entire surface geometry is modeled by using both two- and three-dimensional Bezier–Bernstein parameterizations. The computational efficiency of this design procedure is in large part a result of the use of discrete sensitivity analysis, which permits the efficient treatment of a large number of design variables, and also a result of the exclusive use of low-memory preconditioned conjugate gradient-like methodologies to provide inexpensive solutions to the fully implicit CFD equation and the sensitivity equation. Proper preconditioning was found to be a vital element in achieving stable and convergent three-dimensional solution methods.

A wing geometry model has been described that generates very general wing shapes by applying a sequence of geometrical deformations. Five spanwise parameter distributions and four scalar parameters are required by the model. When used within the present design procedure, the geometry model was capable of realizing non-intuitive wing design concepts as well as realistic wing designs.

Acknowledgments

This research was supported by NASA Langley Research Center under Grant NAG-1-1188. The technical monitor was David S. Miller. The authors would like to thank Bruce Wedan of NASA Langley Research Center for providing us with his code to generate the initial unit wing grid.

References

[1]Ratcliff, R. R., and Carlson, L. A., "Direct-Inverse Transonic Wing-Design Method in Curvilinear Coordinates Including Viscous Interaction," *Journal of Aircraft*, Vol. 28, No. 12, 1991, pp. 803, 804.

[2]Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," AIAA Paper 77-1247, Aug. 1977.

[3]Cosentino, G. B., and Holst, T. L., "Numerical Optimization Design of Advanced Transonic Wing Configurations," *Journal of Aircraft*, Vol. 23, March 1986, pp.192–199.

[4]Chang, I.-C., Torres, F. J., and van Dam, C. P., "Wing Design Code Using Three-Dimensional Euler Equations and Optimization," AIAA Paper 91-3190, Sept. 1991.

[5]Reuther, J., Cliff, S. E., Hicks, R. M., and van Dam, C. P., "Practical Design Optimization of Wing/Body Configurations Using the Euler Equations," AIAA Paper 92-2633, June 1992.

[6]Slooff, J. W., "Computational Procedures in Aerodynamic Design," *Computational Methods in Potential Aerodynamics*, edited by L. Morino, Springer–Verlag, New York, 1987, pp. 415–461.

[7]Gregg, R. D., and Misegades, K. P., "Transonic Wing Optimization Using Evolution Theory," AIAA Paper 87-0520, Jan. 1987.

[8]Elbanna, H. M., and Carlson, L. A., "Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regimes," *Journal of Aircraft*, Vol. 27, No. 6, 1990, pp. 507–518.

[9]Drela, M., "Viscous and Inviscid Inverse Schemes Using Newton's Method," AGARD Rept. 780, Paper 9, March 1990.

[10]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.

[11]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.

[12]Elbanna, H. M., and Carlson, L. A., "Determination of Aerodynamic Sensitivity Coefficients Based on the Three-Dimensional Full Potential Equation," AIAA Paper 92-2670, June 1992.

[13]Eleshaky, M. E., and Baysal, O., "Preconditioned Domain Decomposition Scheme for Three-Dimensional Aerodynamic Sensitivity Analysis," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference*, Open Forum Paper, AIAA, Washington, DC, 1993, pp. 1055, 1056.

[14]Korivi, V. M., Taylor, A. C., III, Hou, G. W., Newman, P. A., and Jones, H. E., "Sensitivity Derivatives for Three-Dimensional Supersonic Euler Code Using Incremental Iterative Strategy," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference*, Open Forum Paper, AIAA, Washington, DC, 1993, pp. 1053, 1054.

[15]Eleshaky, M. E., and Baysal, O., "Shape Optimization of a 3-D Nacelle Near a Flat Plate Wing Using Multiblock Sensitivity Analysis," AIAA Paper 94-0160, Jan. 1994.

[16]Korivi, V. M., Newman, P. A., and Taylor, A. C., III, "Aerodynamic Optimization Studies Using a 3-D Supersonic Euler Code with Efficient Calculation of Sensitivity Derivatives," AIAA Paper 94-4270, Sept. 1994.

[17]Reuther, J., and Jameson, A., "Aerodynamic Shape Optimization of Wing and Wing-Body Configurations Using Control Theory," AIAA Paper 95-0123, Jan. 1995.

[18]Burgreen, G. W., and Baysal, O., "Aerodynamic Shape Optimization Using Preconditioned Conjugate Gradient Methods," *AIAA Journal*, Vol. 32, No. 1, 1994, pp. 2145–2152.

[19]Burgreen, G. W., Baysal, O., and Eleshaky, M. E., "Improving the Efficiency of Aerodynamic Shape Optimization Procedures," *AIAA Journal*, Vol. 32, No. 1, 1994, pp. 69–76.

[20]Burgreen, G. W., "Three-Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis," Ph.D. Dissertation, Dept. of Mechanical Engineering, Old Dominion Univ., Norfolk, VA, May 1994.

[21]Vanderplaats, G. N., and Moses, F., "Structural Optimization by Methods of Feasible Directions," *Computers and Structures*, Vol. 3, July 1973, pp. 739–755.

[22]Vanden, K. J., and Whitfield, D. L., "Direct and Iterative Algorithms for the Three-Dimensional Euler Equations," AIAA Paper 93-3378, July 1993.

[23]Paoletti, S., and Vitaletti, M., "An Unfactored Implicit Scheme for 3D Inviscid Transonic Flows," AIAA Paper 92-2668, June 1992.

[24]Whitaker, D. L., "Three-Dimensional Unstructured Grid Euler Computations Using a Fully-Implicit, Upwind Method," AIAA Paper 93-3337, July 1993.

[25]Luo, H., Baum, J. D., Löhner, R., and Cabello, J., "An Implicit Three-Dimensional Finite Element Solver for Unstructured Meshes," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference*, Open Forum Paper, AIAA, Washington, DC, 1993, pp. 1027, 1028.

[26]Nielsen, E. J., Anderson, W. K., Walters, R. W., and Keyes, D. E., "Application of Newton-Krylov Methodology to a Three-Dimensional Unstructured Euler Code," AIAA Paper 95-1733, April 1995.

[27]Wong, Y. S., and Hafez, M. M., "Preconditioned Conjugate Gradient Methods for Transonic Flow Calculations," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 81-30, Hampton, VA, Sept. 1981.

[28]Wigton, L. B., Yu, N. J., and Young, D. P., "GMRES Acceleration of Computational Fluid Dynamics Codes," AIAA Paper 85-1494, June 1985.

[29]Venkatakrishnan, V., "Preconditioned Conjugate Gradient Methods for the Compressible Navier–Stokes Equations," *AIAA Journal*,Vol. 29, No. 7, 1991, pp. 1092–1100.

[30]Ajmani, K., Ng, W., and Liou, M., "Generalized Conjugate-Gradient Methods for the Navier–Stokes Equations," AIAA Paper 91-1556, April 1991.

[31]Orkwis, P. D., "A Comparison of Newton's and Quasi-Newton's Method Solvers for the Navier–Stokes Equations," AIAA Paper 92-2644, June 1992.

[32]Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[33]Anderson, E., and Saad, Y., "Solving Sparse Triangular Linear Systems on Parallel Computers," *International Journal of High Speed Computing*, Vol. 1, No. 1, 1989, pp. 73–95.

[34]Burgreen, G. W., and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization of Supersonic Delta Wings," AIAA Paper 94-4271, Sept. 1994.

[35]Van Dam, C. P., and Nikfetrat, K., "Accurate Prediction of Drag Using Euler Methods," *Journal of Aircraft*, Vol. 29, No. 3, 1992, pp. 516–519.

[36]Cosentino, G. B., and Holst, T. L., "Reply to R. C. Lock," *Journal of Aircraft*, Vol. 24, No. 8, 1987, pp. 575, 576.