JOURNAL OF AIRCRAFT
Vol. 38, No. 6, November–December 2001

Novel Multidisciplinary Shape Parameterization Approach

Jamshid A. Samareh*
NASA Langley Research Center, Hampton, Virginia 23681

A multidisciplinary shape parameterization approach is presented. The approach consists of two basic concepts: 1) parameterizing the shape perturbations rather than the geometry itself and 2) performing the shape deformation by means of the soft object animation algorithms used in computer graphics. Because the formulation presented is independent of grid topology, we can treat computational fluid dynamics and finite element grids in the same manner. The proposed approach is simple, compact, and efficient. Also, the analytical sensitivity derivatives are easily computed for use in a gradient-based optimization. This algorithm is suitable for low-fidelity, for example, linear aerodynamics and equivalent laminated plate structures, and high-fidelity, for example, nonlinear computational fluid dynamics and detailed finite element modeling, analysis tools. The implementation details of parameterizing for planform, twist, dihedral, thickness, camber, and free-form surface are given. Results are presented for a multidisciplinary application consisting of nonlinear computational fluid dynamics, detailed computational structural mechanics, and a simple performance module.

Nomenclature

A	=	wing area
AR	=	wing aspect ratio
B	=	Bernstein polynomial
b	=	wing span
C	=	chord
C_D	=	coefficient of drag
c	=	camber
d	=	degree
e	=	scale factor for twist and shearing
N	=	B-spline basis function
$\bar{n}$	=	normal vector
$\bar{O}$	=	origin of parallelepiped
$\bar{P}$	=	coordinates of NURBS control point
$\bar{Q}$	=	coordinates of solid element
$\bar{R}$	=	coordinates of deformed model
$\bar{r}$	=	coordinates of baseline model
$\bar{S}$	=	shearing vector
$\bar{T}$	=	twist plane
t	=	thickness
u	=	parameter coordinate
$\bar{v}$	=	MASSOUD design variable vector
W	=	NURBS weights
X, Y, Z	=	Cartesian coordinates of deformed model
x, y, z	=	Cartesian coordinates of baseline model
α	=	angle of attack, deg
Δ	=	total deformation
δ	=	deformation
θ	=	twist angle, deg
Λ	=	leading-edge sweep angle, deg
λ	=	wing taper ratio
ξ, η, ζ	=	coordinates of deformation object
ρ	=	twist radius

Subscripts

ca	=	camber
I, J, K	=	total numbers of control points
i, j, k	=	indices for NURBS control point
id, jd	=	design variable indices
in	=	inner
L	=	wing lower surface
le	=	leading edge
m	=	midline
out	=	outer
p	=	degree of B-spline basis function in i direction
pl	=	planform
q	=	degree of B-spline basis function in j direction
r	=	root
sh	=	shear
te	=	trailing edge
t	=	tip
th	=	thickness
tw	=	twist
U	=	wing upper surface

Superscript

T	=	transpose of the matrix

Introduction

MULTIDISCIPLINARY shape optimization (MSO) finds the optimum shape for a given structural layout. Performing MSO for a complete airplane configuration is a challenging task with high-fidelity analysis tools. The analysis models, also referred to as grids or meshes, are based on some or all of the airplane components, such as skin, ribs, spars, and the stiffeners. The MSO of an airplane must treat not only the wing skin, fuselage, flaps, nacelles, and pylons, but also the internal structural elements such as spars and ribs (see Fig. 1). The treatment of internal structural elements is especially important for detailed finite element (FE) analysis. For a high-fidelity MSO process to be successful, the process must be based on a compact and effective set of design variables that yields a feasible and enhanced configuration. For more details, readers are referred to an overview paper by this author on geometry modeling and grid generation for design and optimization.[1]

Shape parameterization is the first step for an MSO process. Over the past several decades, shape optimization has been successfully applied for two-dimensional and simple three-dimensional configurations. In a multidisciplinary application, the shape parameterization must be compatible with and adaptable to various analysis tools ranging from low-fidelity tools, such as linear aerodynamics

Received 1 November 1999; revision received 10 January 2001; accepted for publication 30 April 2001.

*Senior Research Scientist, Multidisciplinary Optimization Branch, Mail Stop 159; j.a.samareh@larc.nasa.gov. Senior Member AIAA.

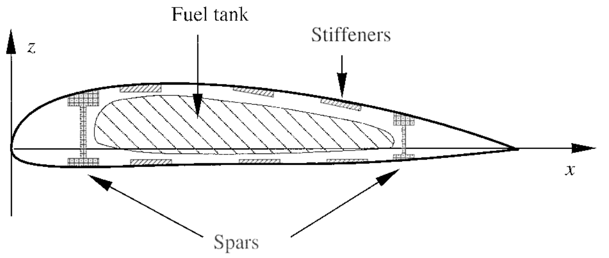

Fig. 1 Internal components of wing.

and equivalent laminated plate structures, to high-fidelity tools, such as nonlinear computational fluid dynamics (CFD) and detailed computational structures mechanics (CSM) codes. Creation of CFD and CSM grids is time consuming and costly for a full airplane model: Detailed CSM and CFD grids based on a CAD model take several months to develop. For a multidisciplinary problem, the process must use a geometry model and parameterization consistently across all disciplines. Gradient-based optimization requires accurate sensitivity derivatives of the analysis model with respect to design variables.

The shape parameterization techniques can be divided into eight categories: basis vector, domain element, discrete, analytical, free form deformation (FFD), partial differential equation (PDE), polynomial and spline, and CAD-based (see Ref. 2 for more details). A brief description is provided next.

The shape parameterization techniques are based on parameterizing either the disciplinary models (grids) or the geometry (from which the disciplinary grids can be generated).[2] The advantage of parameterizing the grid is that the grid topology stays fixed throughout optimization; hence, the grids can be regenerated (deformed) automatically. The disadvantage is that the large shape changes could produce unacceptable grids due to the fixed topology. The most attractive feature of parameterizing the grid is the ability to use an existing grid for optimization. The basis vector, domain element, discrete, analytical, FFD, and our approach belong to the grid parameterization category. Use of these approaches (with the exception of discrete approach) would result in a consistent shape parameterization for multidisciplinary applications.

The advantage of geometry parameterization is that only one representation needs to be parameterized. However, the optimization process will require automatic grid-generation tools, which may not be available for all disciplines. The PDE, polynomial and spline, and CAD-based approaches belong to geometry parameterization category.

Basis vector approach[3] is based on parameterizing grids. It uses several proposed grids to represent the design shape. The proposed grids must share the same grid topology. This approach also provides a mechanism to transfer the sensitivity data to most commercial analysis codes. We followed a similar approach to transfer the sensitivity information to commercial analysis codes.

The domain element approach is based on linking a set of grid points to a macroelement (domain element) that controls the shape of the model.[4] As the nodes of the domain element move, the grid points belonging to the domain will move as well. The movement is based on an inverse mapping between the grid points and the domain element, and the parametric coordinates of the grid points with respect to the domain element are kept fixed through the optimization cycles. The domain element technique is available for shape optimization in some commercial software. This method is very efficient, and it is relatively simple to implement. We followed a similar approach to parameterize the planform shape, but we used multiple three-dimensional domain elements.

The discrete approach is based on using the grid-point coordinates as design variables. This approach is easy to implement, and only the number of design variables limits the geometry changes. Because the shape is perturbed by moving individual grid points, a smooth geometry is difficult to maintain, and the optimization solution may be impractical to manufacture, as pointed out by Braibant and Fleury.[5] For a model with a large number of grid points, the number of design variables often becomes very large and may lead to high cost and a difficult optimization problem to solve. A multidisciplinary design optimization (MDO) application often requires parameterization of multiple dissimilar grids, for example, CFD and CSM. Because the discrete approach is not capable of parameterizing dissimilar grids simultaneously, it cannot guarantee a consistent shape parameterization across multiple disciplines.

FFD is a subset of the soft object animation (SOA)[6] algorithms used in computer graphics for morphing images and deforming models. The FFD algorithms are used to deform a baseline model to create the design model. We have used the SOA approach to model the twist, shear, and planform.

The analytical approach[7] provides a compact formulation for parameterization of airfoil sections. The formulation is based on adding shape functions (analytical functions) linearly to the baseline shape. The contribution of each parameter is determined by the value of the participating coefficients (design variables) associated with that function. All participating coefficients are initially set to zero, so that the first computation gives the baseline geometry. The shape functions are smooth functions based on a set of previous airfoil designs. This method is very effective for wing parameterization. We used a similar concept to model the effect of section design variables.

The PDE approach considers the design surface generation as a boundary-value problem, and it produces design surfaces as the solutions to elliptic PDEs. Bloor and Wilson[8] presented an efficient and compact method for parameterizing the surface geometry of an aircraft. Bloor and Wilson showed that it was possible to represent aircraft geometry in terms of a small set of design variables. Smith et al.[9] extended the PDE approach to a class of airplane configurations.

Polynomial and spline approaches represent design surfaces with the polynomial and spline representations. Use of polynomial and spline representations for shape parameterization obviously can reduce the total number of design variables. Braibant and Fleury[5] showed that Bezier and B-spline curves are well suited for shape optimization. A polynomial can describe a curve in a very compact form with a small set of design variables. Cosentino and Holst[10] optimized a transonic wing configuration by using a cubic-spline representation for two-dimensional airfoils that define wing geometry. In a design case study on the Lockheed C-141B aircraft, Cosentino and Holst reduced the number of design variables from 120 to 12 by using the cubic-spline technique. Schramm and Pilkey[11] used a B-spline representation to perform structural shape optimization for a torsion problem with direct integration and B-splines. Similarly, Anderson and Venkatakrishnan[12] used B-splines with an unstructured grid CFD code for aerodynamic design optimization. The polynomial and spline techniques are well suited for two-dimensional and simple three-dimensional models. Complex three-dimensional models are made of many curves and surfaces; as a result, these curves and surfaces are difficult to model outside of a CAD system. Also, complex models require a large number of control points, and optimization is prone to creating irregular or wavy geometry.

CAD-based approach uses the commercial feature-based solid modeling CAD systems to create the design surfaces. To parameterize an existing model is still a challenging task in today's CAD systems, and the models created are not always good enough for automatic grid-generation tools. Feature-based solid modeling (FBSM) CAD systems are capable of creating dimension-driven objects. Even though use of parametric modeling in design would make the FBSM tools ideal for optimization, existing FBSM tools are not capable of calculating sensitivity derivatives analytically. Townsend et al.[13] discussed issues involved in using a CAD system for an MDO application. The computer codes for commercial CAD systems are very large; to differentiate the entire system with automatic differentiation tools may not be feasible. Therefore, calculation of the analytical sensitivity derivatives of geometry with respect to the design variables could prove to be difficult within a commercial CAD environment. For some limited cases, the analytical shape sensitivity derivatives can be calculated based on a CAD model; however, this method will not work under all circumstances. One difficulty is that, for some perturbation of some dimensions, the topology of the CAD part may be changed. Another way to calculate the sensitivity derivatives is to use finite differences, as long as the perturbed geometry has the same topology as the unperturbed geometry. Both methods, the analytical and finite difference approximations, have their difficulties and limitations.

This paper presents a shape parameterization approach suitable for MSO as part of a MDO application. This approach was implemented in a computer code called multidisciplinary aerodynamic-structural shape optimization using deformation (MASSOUD). The combined algorithm was successfully implemented for aerodynamic shape optimization with analytical sensitivity for structured grid[14,15] and unstructured grid[16] CFD codes. This algorithm has also been used for multidisciplinary application of a high-speed civil transport (HSCT).[17,18] The approach consists of two basic concepts, which are expanded in the next two sections. The first concept is based on parameterizing the shape perturbations rather than the geometry itself. The second concept is based on using the SOA[6] algorithms for shape parameterization.

Parameterizing the Shape Perturbations

Typically, the optimization starts with an existing wing design, and the goal is to improve or redesign the wing performance by using numerical optimization. The geometry changes (perturbations) between initial and optimized wing are very small,[7,10] but the difference in wing performance can be substantial. This is demonstrated in Fig. 2, where the airfoil shapes and the corresponding coefficient of pressure distributions are shown for M6 and NACA0012 airfoils. An effective way to reduce the number of shape design variables is to parameterize the shape perturbations instead of parameterizing the shape itself. For example, 89 points are used to represent the airfoils shown in Fig. 2. However it only takes nine control points to represent the difference between the two airfoils, as shown in Fig. 3. Throughout the optimization cycles, the analysis grid can then be updated as

$$\bar{R}(\bar{v}) = \bar{r} + \Delta\bar{R}(\bar{v}) \tag{1}$$

For MASSOUD, the change $\Delta\bar{R}$ is a combination of changes in thickness, camber, twist, shear, and planform:

$$\Delta\bar{R} = \delta\bar{R}_{\rm th} + \delta\bar{R}_{\rm ca} + \delta\bar{R}_{\rm tw} + \delta\bar{R}_{\rm sh} + \delta\bar{R}_{\rm pl} \tag{2}$$

Far fewer design variables are required to parameterize the shape perturbations $\Delta\bar{R}$ than the baseline shape $\bar{r}$ itself.

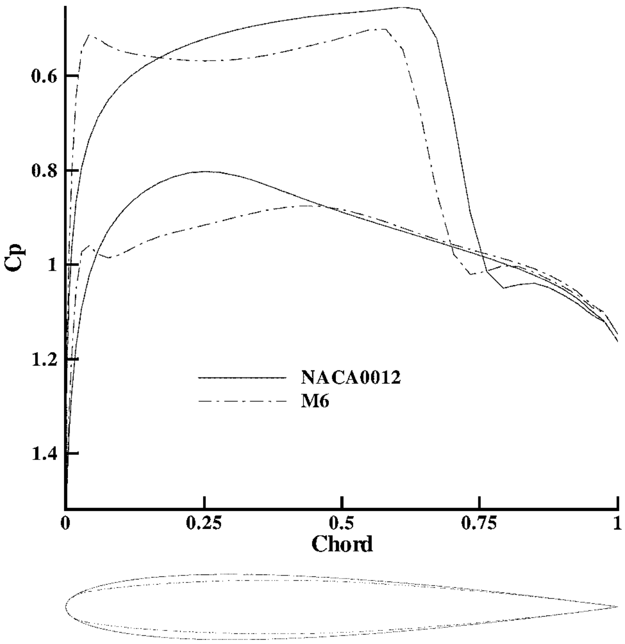

Fig. 2 Shape and coefficient of pressure for NACA and M6 airfoils.

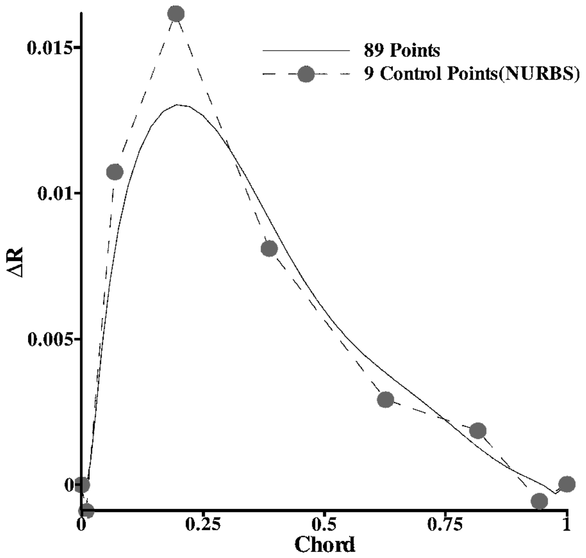

Fig. 3 Design perturbation for airfoils.

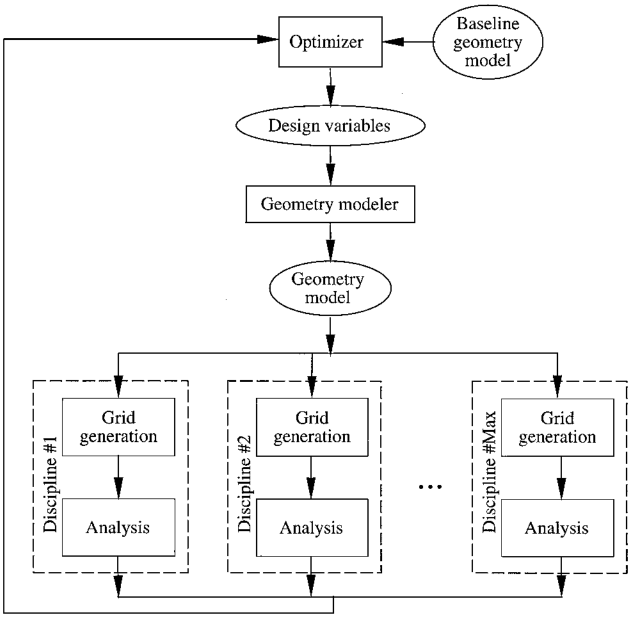

Fig. 4 Typical MSO process.

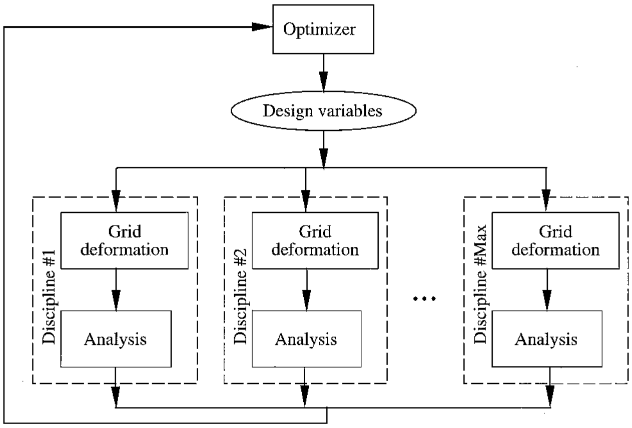

Fig. 5 MASSOUD process.

Figures 4 and 5 show the typical MSO and MASSOUD processes. In a typical MSO process (Fig. 4), a geometry modeler perturbs the baseline geometry model. Then, grids must be generated for each discipline. Because automatic grid-generation tools are not available for all disciplines, automating this MSO process would be very difficult. In contrast, the MASSOUD process (Fig. 5) relies on parameterizing and deforming the baseline grids and avoids the manual grid-generation process.

SOA

The field of SOA in computer graphics[6] provides algorithms for morphing images[19] and deforming models.[20,21] These algorithms are powerful tools for modifying the shapes: They use deformation of a high-level shape, as opposed to manipulation of lower level geometric entities. Hall presents an algorithm and provides computer codes for morphing images.[19] The deformation algorithms are suitable for deforming models represented by either a set of polygons or a set of parametric curves and surfaces. The SOA algorithms treat the model as rubber that can be twisted, bent, tapered, compressed, or expanded, but will still retain its topology. This technique is ideal for parameterizing airplane models that have external skin as well as internal components, for example, see Fig. 1. The SOA algorithms link vertices of an analysis model (grid) to a small number of design variables. Consequently, the SOA algorithms can serve as the basis for an efficient shape parameterization technique.

Barr presented a deformation approach in the context of physically based modeling.[20] This approach uses physical simulation to obtain realistic shape and motions and is based on operations such as translation, rotation, and scaling. With Barr's algorithm, the deformation is achieved by moving the vertices of a polygon model or the control points of a parametric curve and surface. Sederberg and Parry presented a variant[21] of the FFD algorithm, which operates on the whole space, regardless of the representation of the deformed objects embedded in the space. The algorithm allows a user to manipulate the control points of trivariate Bezier volumes. The disadvantage of FFD is that the design variables may have no physical significance for the design engineers. This drawback makes it difficult to select an effective and compact set of design variables. This paper presents a set of modifications to the original SOA algorithms to alleviate this drawback.

For the modified SOA algorithms presented in the next several sections, implementation will include the following common set of steps.

1) Select an appropriate deformation technique and object. This step defines the forward mapping from the deformation object coordinate system ξ, η, ζ to the baseline grid coordinate system x, y, z.

2) Establish a backward mapping from the baseline grid coordinate system x, y, z to the deformation object coordinate system ξ, η, ζ. The ξ, η, ζ mapping parameters are fixed and are independent of the shape perturbations. This preprocessing step is required only once.

3) Perturb the control parameters (design variables) of the deformation object.

4) Evaluate the grid perturbation $\Delta\bar{\boldsymbol{R}}$ and shape sensitivity derivatives $(\partial\bar{\boldsymbol{R}}/\partial\bar{v})$.

The following sections provide recipes for using SOA algorithms for parameterizing airplane models for thickness, camber, twist, shear, and planform changes.

Thickness and Camber

We used a nonuniform rational B-spline (NURBS) representation as the deformation object for thickness and camber parameterization. The NURBS representation combined the desirable properties of NACA definition[22] and spline techniques, and it did not deteriorate nor destroy the smoothness of the initial geometry.

The changes in thickness and camber were represented by

$$\delta\bar{\boldsymbol{R}}_{\mathrm{th}}(\xi,\eta)=\frac{\sum_{i=0}^{I}N_{i,p}(\xi)\sum_{j=0}^{J}N_{j,q}(\eta)W_{i,j}\bar{\boldsymbol{P}}_{\mathrm{th}_{i,j}}}{\sum_{i=0}^{I}N_{i,p}(\xi)\sum_{j=0}^{J}N_{j,q}(\eta)W_{i,j}} \tag{3}$$

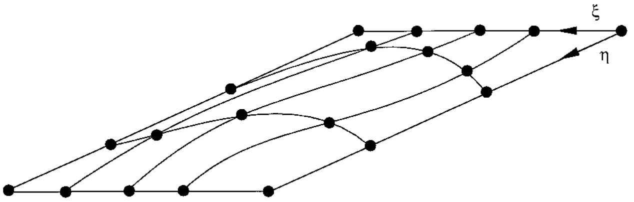

Fig. 6 Thickness and camber definitions in wing coordinate system.

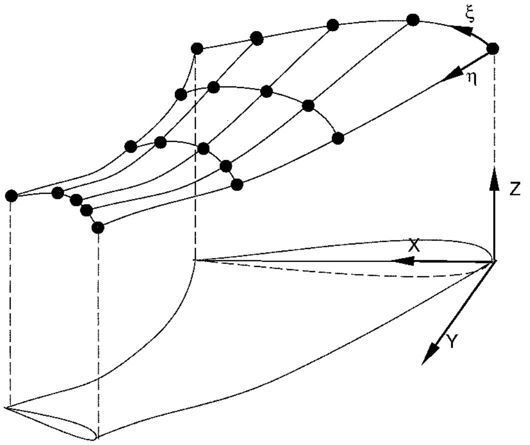

Fig. 7 Thickness and camber definitions in *x*, *y*, and *z* coordinate system.

$$\delta\bar{\boldsymbol{R}}_{\mathrm{ca}}(\xi,\eta)=\frac{\sum_{i=0}^{I}N_{i,p}(\xi)\sum_{j=0}^{J}N_{j,q}(\eta)W_{i,j}\bar{\boldsymbol{P}}_{\mathrm{ca}_{i,j}}}{\sum_{i=0}^{I}N_{i,p}(\xi)\sum_{j=0}^{J}N_{j,q}(\eta)W_{i,j}} \tag{4}$$

Figures 6 and 7 show the NURBS control points in ξ, η and x, y, z coordinate systems, respectively. The control points and weights could be used as design variables.

Readers are referred to the textbook by Farin for details on the properties of NURBS representation.[23] The NURBS representation has several important properties for design and optimization. A NURBS curve of order p, having no multiple interior knots, is $p-2$ differentiable. As a result, the NURBS representation is able to handle a complex deformation and still maintain smooth surface curvature. The control points are the coefficients of the basis functions, but the smoothness is controlled by the basis functions, not by the control points. The NURBS representation is local in nature, allowing the surface to be deformed locally, hence leaving the rest of the surface unchanged. Equations (3) and (4) served as the forward mapping between the thickness and camber design variables and the grid perturbation, $\delta\bar{\boldsymbol{R}}_{\mathrm{th}}$ and $\delta\bar{\boldsymbol{R}}_{\mathrm{ca}}$.

The next step was to establish the backward mapping from the deformation object, that is, the NURBS surface, coordinates, ξ, η, to the baseline model coordinates, x, y, z. The percentage of chord $\%C$ was used for ξ, and the spanwise location y was used for η:

$$\xi=\%C, \qquad \eta=y \tag{5}$$

To calculate $\%C$, we needed to determine the wing chord at each y station. The baseline CAD model provided the leading edge $\bar{R}_{\mathrm{le}}(\eta)$, trailing edge $\bar{\boldsymbol{R}}_{\mathrm{te}}(\eta)$, wing midline $\bar{\boldsymbol{R}}_{m}(\eta)$, and normal vector defining the airfoil plane, $\bar{\boldsymbol{T}}(\eta)$, as shown in Fig. 8. The curve defining the wing midline did not have to be at the center of the wing, but needed to be somewhere between the upper and the lower wing surfaces. The $\bar{\boldsymbol{R}}_{\mathrm{le}}(\eta)$, $\bar{\boldsymbol{R}}_{\mathrm{te}}(\eta)$, and $\bar{\boldsymbol{R}}_{m}(\eta)$ were used to separate points on the upper surface from points on the lower surface.

Because we knew η for each grid point, we were able to define a plane that passed through the grid point by means of a normal vector defined by $\bar{\boldsymbol{T}}(\eta)$. We then used the following equations to find the intersection of this plane and the curves shown in Fig. 8:

$$\bar{\boldsymbol{T}}(\eta)\cdot[\bar{\boldsymbol{r}}-\bar{\boldsymbol{R}}_{\mathrm{le}}(\eta)]^{T}=0 \tag{6}$$

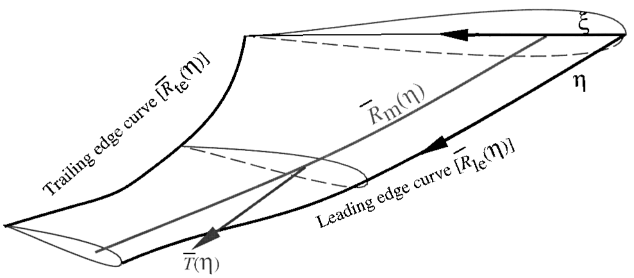
ξ
η
Trailing edge curve $[\bar{R}_{te}(\eta)]$
$\bar{R}_m(\eta)$
Leading edge curve $[\bar{R}_{le}(\eta)]$
$\bar{T}(\eta)$

Fig. 8 Curves defining the backward mapping.

$$\bar{\boldsymbol{T}}(\eta)\cdot[\bar{\boldsymbol{r}}-\bar{\boldsymbol{R}}_{\text{te}}(\eta)]^T = 0 \tag{7}$$

$$\bar{\boldsymbol{T}}(\eta)\cdot[\bar{\boldsymbol{r}}-\bar{\boldsymbol{R}}_{m}(\eta)]^T = 0 \tag{8}$$

Equations (6–8) must be solved for each grid point in the model. For a high-order NURBS curve, Eqs. (6–8) are nonlinear and can be solved by the Newton–Raphson method. The solution to Eqs. (6–8) for each η was a set of three points located at the leading edge, the trailing edge, and the center. The $\%C$ was calculated based on the leading- and trailing-edge points. Next, we needed to separate the grid points defining the wing model into upper and lower surfaces. We connected the three points obtained from Eqs. (6–8) to form a curve that separated the upper surface from the lower surface. This curve need not represent the camber line accurately, and a wing with drooping leading edge or with highly cambered airfoil sections may require more than one $\bar{\boldsymbol{R}}_m(\eta)$ to define the curve. With this approach, the deformation may be localized to a specific design area by setting allowable $\%C_{\min}$, $\%C_{\max}$, $\eta_{\min}$, and $\eta_{\max}$.

As the design variables (control points $\bar{\boldsymbol{P}}_{\text{th}_{id,jd}}$ and $\bar{\boldsymbol{P}}_{\text{ca}_{id,jd}}$) changed, we calculated the contribution to the thickness and camber by Eqs. (3) and (4). The advantage of this process was that the sensitivity of grid-point location with respect to design variables was only a function of the B-spline basis functions,

$$\frac{\partial \bar{\boldsymbol{R}}}{\partial \bar{\boldsymbol{P}}_{\text{th}_{id,jd}}} = \frac{\partial \bar{\boldsymbol{R}}}{\partial \bar{\boldsymbol{P}}_{\text{ca}_{id,jd}}} = \frac{N_{id,p}(\xi)N_{jd,q}(\eta)W_{id,jd}}{\sum_{i=0}^{I} N_{i,p}(\xi)\sum_{j=0}^{J} N_{j,q}(\eta)W_{i,j}} \tag{9}$$

Consequently, the sensitivity, as shown in Eq. (9), was independent of the design variables $\bar{\boldsymbol{P}}_{id,jd}$ and the coordinates x, y, z. Thus, we needed to calculate the sensitivity with respect to thickness and camber only at the beginning of the optimization.

Figure 9 shows four deformed airfoil shapes; the airfoil shape is the root airfoil of a blended-wing body. The dashed line is the baseline shape, and the solid lines are the deformed shapes. The baseline airfoil shape was defined with 89 control points. The shape deformations were achieved with only two NURBS control points, that is, design variables, for the camber as defined by Eq. 4.

Twist and Shear

The twist angle is defined as the difference between the airfoil section incident angle at the root and each airfoil section incident angle. Similarly, the shear (dihedral) is defined as the difference between the airfoil leading-edge z coordinate for the root and the z coordinate at each airfoil section. If the twist angle at the tip is less than the twist at the root, the wing is said to have a washout, which could delay the stall at the wing tip. Also, as the wing washout increases, the wing load shifts from outboard to inboard. As a result, the spanwise distribution of the twist angle plays an important role in the wing performance.

The SOA algorithms are used to modify the wing twist and shear distribution. Barr presented a series of SOA algorithms for twisting, bending, and tapering an object.[20] Watt and Watt referred to these algorithms as nonlinear global deformation.[6] Sederberg and Greenwood[24] extended Barr's ideas[20] to handle complex shapes. Modified versions of these algorithms are presented in this paper.

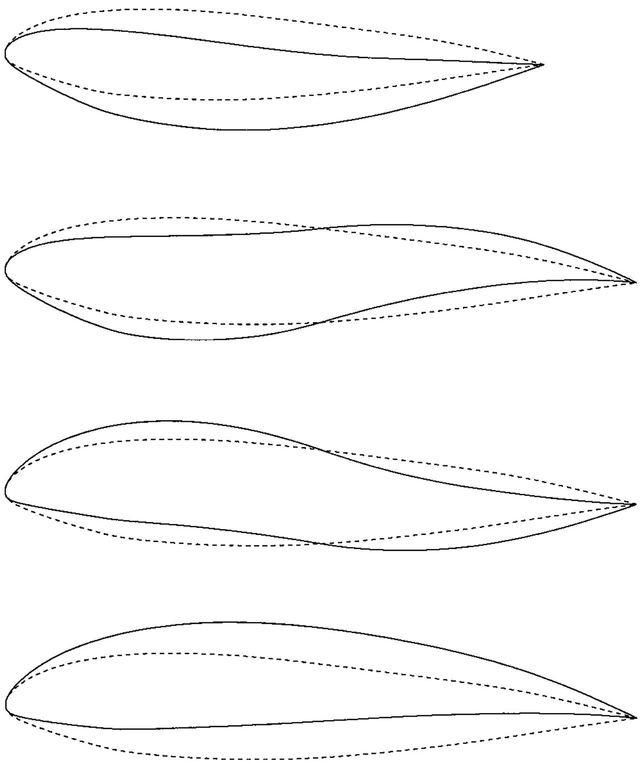

Fig. 9 Deformed airfoil shapes for blended-wing body.

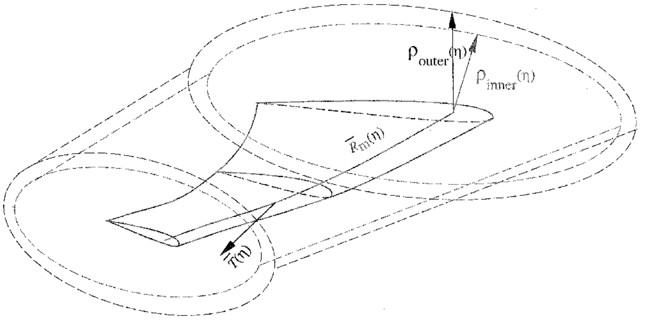

Fig. 10 Inner twist cylinder.

To modify the twist and shear distributions, the wing was embedded in a nonlinear deformation object referred to as a twist cylinder, shown in Fig. 10. The center of the cylinder was defined by a NURBS curve $\bar{\boldsymbol{R}}_m(\eta)$. The effect of deformation was confined to a section of a wing by limiting the parameter η to vary between $\eta_{\min}$ and $\eta_{\max}$. The $\eta_{\min}$ could be extended to the wing root, and the $\eta_{\max}$ went beyond the wing tip. The cylinder could be twisted and sheared only in a plane (twist plane) defined by a point along $\bar{\boldsymbol{R}}_m(\eta)$ with a normal vector of $\bar{\boldsymbol{T}}(\eta)$. The $\rho_{\text{in}}(\eta)$ and $\rho_{\text{out}}(\eta)$ were the radii of the inner and outer cylinders, respectively (see Fig. 10). The deformation had no effect on grid points located outside the outer cylinder, and the effect of deformation was scaled linearly from the outer cylinder to the inner cylinder. This linear blending allowed us to blend the deformed region with the undeformed region in a continuous manner.

The $\theta(\eta)$ and $\bar{\boldsymbol{S}}(\eta)$ variables are defined by the NURBS representations:

$$\theta(\eta) = \frac{\sum_{i=0}^{I} N_{i,p}(\eta)W_i\theta_i}{\sum_{i=0}^{I} N_{i,p}(\eta)W_i} \tag{10}$$

$$\bar{\boldsymbol{S}}(\eta) = \frac{\sum_{i=0}^{I} N_{i,p}(\eta)W_i\bar{\boldsymbol{S}}_i}{\sum_{i=0}^{I} N_{i,p}(\eta)W_i} \tag{11}$$

Similarly to thickness and camber algorithms, we used

$$\eta = y, \qquad \bar{\boldsymbol{T}}(\eta) = (0, y, 0)^T \tag{12}$$

The second step for twist and shear deformation was to establish the forward mapping from the deformation object (twist cylinder) coordinate system η to the model coordinate system x, y, z. We used Eq. (8) to determine η. Once η was determined, we calculated the local $\rho(\eta)$, $\rho_{\text{in}}(\eta)$, $\rho_{\text{out}}(\eta)$, $\bar{\boldsymbol{T}}(\eta)$, $\theta(\eta)$, and $\bar{\boldsymbol{S}}(\eta)$. The point $\bar{\boldsymbol{r}}$ was rotated $\theta(\eta)$ deg about $\boldsymbol{R}_m(\eta)$ and sheared $\bar{\boldsymbol{S}}(\eta)$:

$$\delta\bar{\boldsymbol{R}}_{\text{tw}}(\eta) = e(\rho)\rho(\eta)[\sin\theta(\eta), 0, \cos\theta(\eta)]^T \tag{13}$$

$$\delta\bar{\boldsymbol{R}}_{\text{sh}}(\eta) = e(\rho)\bar{\boldsymbol{S}}(\eta) \tag{14}$$

where $e(\eta)$ was a scale factor that diminished the effect of deformation as we approached the outer cylinder,

$$e(\eta) = \begin{cases} 0, & \text{if} \quad \rho(\eta) \geq \rho_{\text{out}}(\eta) \\ \dfrac{\rho(\eta) - \rho_{\text{out}}}{\rho_{\text{in}} - \rho_{\text{out}}}, & \text{if} \quad \rho_{\text{in}} \leq \rho(\eta) < \rho_{\text{out}}(\eta) \\ 1, & \text{if} \quad \rho(\eta) < \rho_{\text{in}} \end{cases} \tag{15}$$

The sensitivity of a grid point with respect to the twist and shear design variables was

$$\frac{\partial\bar{\boldsymbol{R}}}{\partial\theta_i} = e(\rho)\rho(\eta)\frac{\partial\theta(\eta)}{\partial\theta_i}[\cos\theta(\eta), 0, -\sin\theta(\eta)]^T \tag{16}$$

$$\frac{\partial\bar{\boldsymbol{R}}}{\partial\bar{\boldsymbol{S}}_i} = e(\rho)\frac{\partial\bar{\boldsymbol{S}}(\eta)}{\partial\bar{\boldsymbol{S}}_i} \tag{17}$$

The term $\partial\theta(\eta)/\partial\theta_i$ was independent of the twist design variables θ_i [see Eq. (10)]. However, $\sin\theta(\eta)$ and $\cos\theta(\eta)$ depended on the twist design variables and were updated for every cycle of the optimization. In contrast, the term $\partial\bar{\boldsymbol{S}}(\eta)/\partial\bar{\boldsymbol{S}}_i$ was independent of shear design variables $\bar{\boldsymbol{S}}_i$ [see Eq. (11)].

Figure 11 shows the inner twist cylinder and the CFD surface grid for a commercial transport. Figure 12 shows the result of twisting the wing 45 deg at the tip. This amount of twist is large and unrealistic, but demonstrates the effectiveness of the SOA.

Planform Parameterization

The wing planform is typically modeled with a set of two-dimensional trapezoids in the x–y plane. Figure 13 shows the planform of a generic HSCT that uses two trapezoids. As shown in Fig. 14, each trapezoid is defined by the root chord C_r, tip chord C_t, span b, and sweep angle Λ. From these values, other planform parameters, such as area A, aspect ratio AR, and taper ratio λ, are defined

$$A = (b/2)(C_r + C_t), \qquad AR = b^2/A, \qquad \lambda = C_t/C_r \tag{18}$$

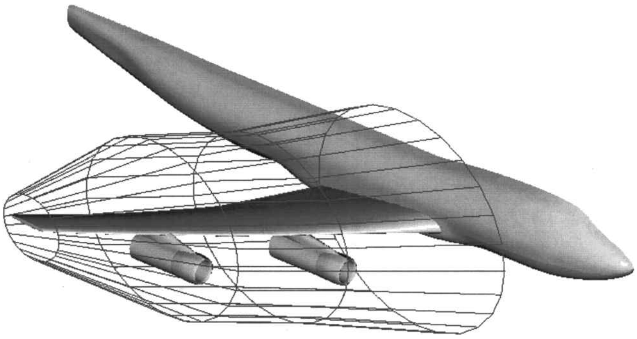

Fig. 11 Twist definition for a transport.

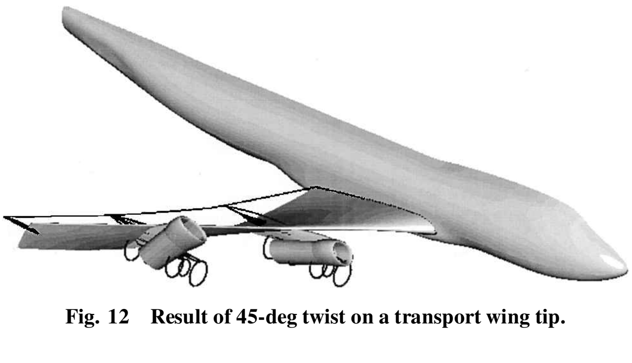

Fig. 12 Result of 45-deg twist on a transport wing tip.

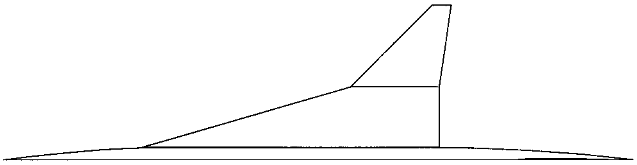

Fig. 13 Planform of a generic HSCT.

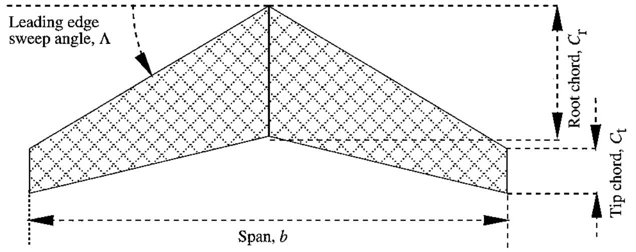

Fig. 14 Planform definition.

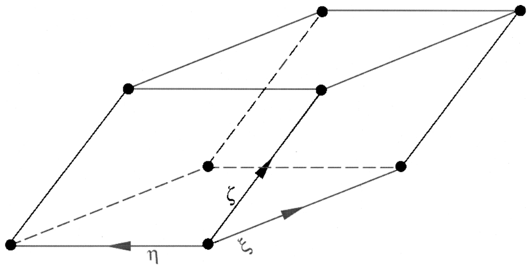

Fig. 15 Parallelepiped volume for FFD.

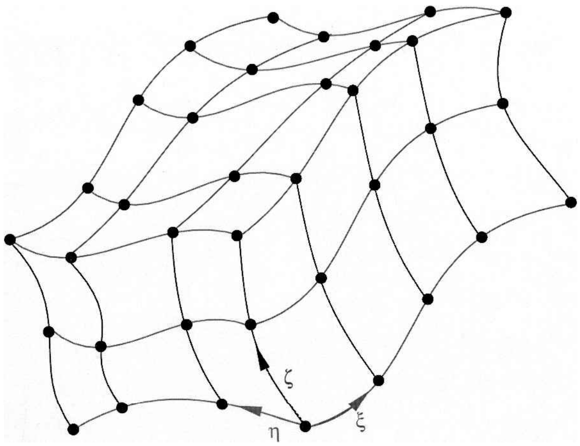

Fig. 16 NURBS volume for FFD.

The FFD algorithm described by Sederberg and Parry[21] is ideal for deforming the polygonal models. Like other SOA algorithms, this algorithm maintains the polygon connectivity, and the deformation is applied only to the vertices of the model. The FFD process is similar to embedding the grid inside a block of clear, flexible plastic (deformation object) so that, as the plastic is deformed, the grid is deformed as well. Deformation of complex shapes may require several deformation objects. The shapes of these deformation objects are not arbitrary. In fact, the shapes must be three-dimensional parametric volumes and could range from a parallelepiped as shown in Fig. 15 to a general NURBS volume as shown in Fig. 16. The block is deformed by perturbing the vertices that control the shape of the deformation block, for example, corners of the parallelepiped. For parametric volume blocks, parameters controlling the deformation are related through the mapping coordinates ξ, η, ζ. These coordinates are used in both forward and backward mapping.

Figure 15 shows a general parallelepiped defined by a set of control points forming three primary edges or directions along ξ, η, and ζ. The relation for a parallelepiped is defined as

$$\bar{\boldsymbol{r}}(\xi,\eta,\zeta) = \bar{\boldsymbol{O}} + \bar{\boldsymbol{n}}_\xi \xi + \bar{\boldsymbol{n}}_\eta \eta + \bar{\boldsymbol{n}}_\zeta \zeta \tag{19}$$

where $\bar{O}$ is the origin of the parallelepiped and $\bar{\boldsymbol{n}}_\xi$, $\bar{\boldsymbol{n}}_\eta$, and $\bar{\boldsymbol{n}}_\zeta$ are the unit vectors along the parallelepiped primary edges in the ξ, η, and ζ directions, respectively. Equation (19) defines a mapping between the deformation object (parallelepiped) and the grid points. The grid points $\bar{r}$ are mapped to the coordinates of the parallelepiped, ξ, η, and ζ, as

$$\xi = \frac{\bar{\boldsymbol{n}}_\eta \times \bar{\boldsymbol{n}}_\zeta \cdot (\bar{\boldsymbol{r}} - \bar{\boldsymbol{O}})}{\bar{\boldsymbol{n}}_\eta \times \bar{\boldsymbol{n}}_\zeta \cdot (\bar{\boldsymbol{n}}_\xi)}, \qquad \eta = \frac{\bar{\boldsymbol{n}}_\xi \times \bar{\boldsymbol{n}}_\zeta \cdot (\bar{\boldsymbol{r}} - \bar{\boldsymbol{O}})}{\bar{\boldsymbol{n}}_\xi \times \bar{\boldsymbol{n}}_\zeta \cdot (\bar{\boldsymbol{n}}_\eta)}$$

$$\zeta = \frac{\bar{\boldsymbol{n}}_\xi \times \bar{\boldsymbol{n}}_\eta \cdot (\bar{\boldsymbol{r}} - \bar{\boldsymbol{O}})}{\bar{\boldsymbol{n}}_\xi \times \bar{\boldsymbol{n}}_\eta \cdot (\bar{\boldsymbol{n}}_\zeta)} \tag{20}$$

A grid point is inside the parallelepiped if $0 \leq \xi, \eta, \zeta \leq 1$.

The FFD technique based on the parallelepiped is very efficient and easy to implement. This technique is suitable for local and global deformation. The only disadvantage is that the use of the parallelepiped limits the topology of deformation. To alleviate this disadvantage, Sederberg and Parry proposed to use nonparallelepiped objects.[21] They also noted that the inverse mapping would be nonlinear and would require significant computations.

Another popular method to define FFD is to use trivariate parametric volumes. Sederberg and Parry[21] used a Bezier volume. Coquillart[25] extended the Bezier parallelepiped to nonparallelepiped cubic Bezier volume. This idea has been further generalized to NURBS volume by Lamousin and Waggenspack.[26] The NURBS blocks are defined as

$$\bar{\boldsymbol{r}}(\xi,\eta,\zeta) = \frac{\sum_{i=0}^{I} N_{i,p1}(\xi) \sum_{j=0}^{J} N_{j,p2}(\eta) \sum_{k=0}^{K} N_{k,p3}(\zeta) W_{i,j,k} \bar{\boldsymbol{P}}_{i,j,k}}{\sum_{i=0}^{I} N_{i,p1}(\xi) \sum_{j=0}^{J} N_{j,p2}(\eta) \sum_{k=0}^{K} N_{k,p3}(\zeta) W_{i,j,k}} \tag{21}$$

Lamousin and Waggenspack[26] used multiple blocks to model complex shapes. This technique has been used for design and optimization by Yeh and Vance[27] and also by Perry and Balling.[28]

The common solid elements used in FE analysis (Fig. 17) can be used as deformation objects as well. The mapping from the solid element coordinates is defined[29] by

$$\bar{\boldsymbol{r}}(\xi,\eta,\zeta) = \sum_i \bar{\boldsymbol{Q}}_i N_i(\xi,\eta,\zeta) \tag{22}$$

where N_i are the FE basis functions and $\bar{\boldsymbol{Q}}_i$ are the nodal coordinates of deformation objects, which are related to the design variables. The equations for inverse mapping are nonlinear for all solid elements with the exception of tetrahedron solid elements. The solid elements provide a flexible environment in which to deform any shape. Complex shapes may require the use of several solid elements to cover the entire domain.

To model the planform shape, we used hexahedron solid elements with four opposing edges parallel to the z coordinate. Then, the planform design variables were linked to the corners of the hexahedral elements. Figure 18 shows the initial and deformed models for

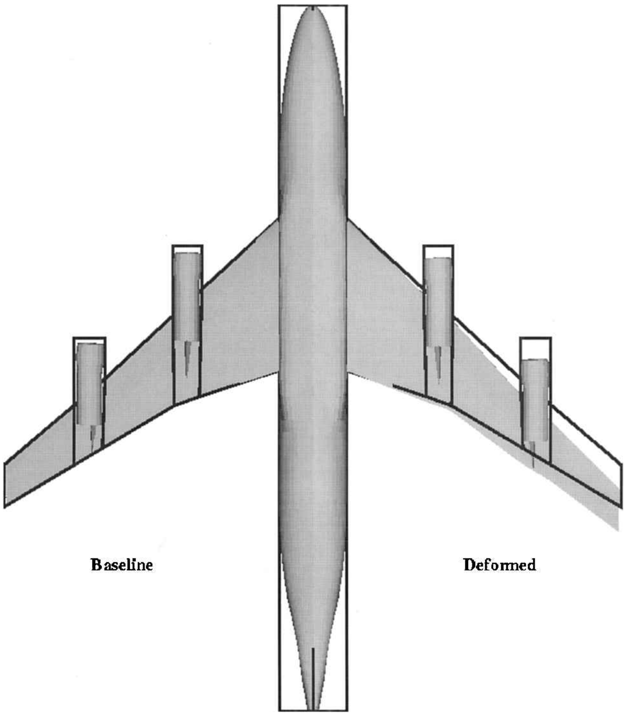

Fig. 18 Planform deformation of a transport.

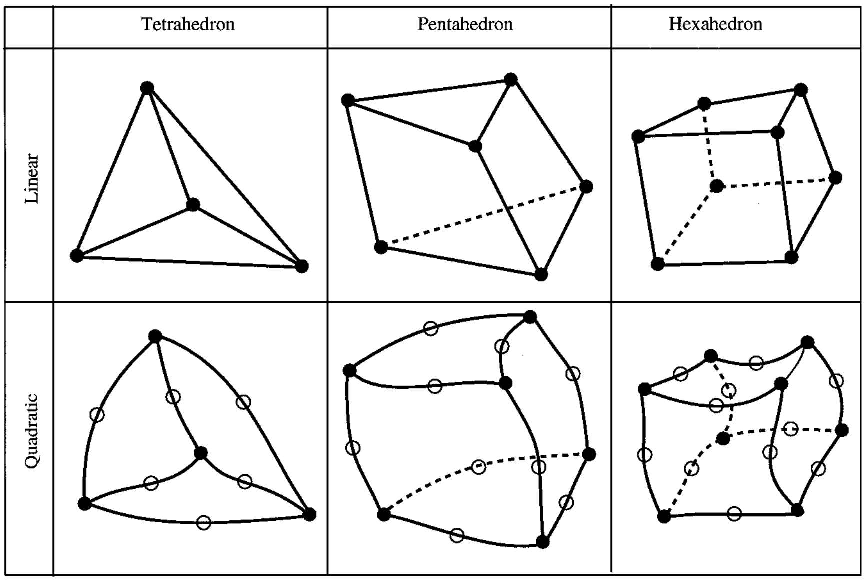

Fig. 17 FE analysis solid elements.

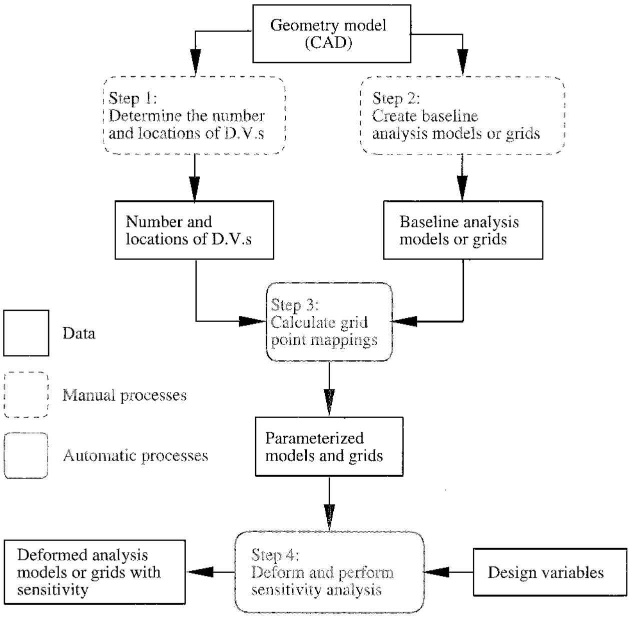

Fig. 19 MASSOUD implementation diagram.

a transport configuration. The solid lines represent the controlling hexahedron solid elements. The baseline model is on the left-hand side, and the deformed shape is on the right-hand side.

As with the camber and thickness algorithms, the sensitivity of grid-point coordinates was independent of the design variables $\bar{Q}_i$ and coordinates x, y, z. Thus, we needed to calculate the sensitivity only once, at the beginning of the optimization.

Implementation

Figure 19 shows the implementation diagram for the combined algorithm. The implementation started with a CAD model that defined the geometry. The first two steps were implemented in parallel. The first step was to determine the number and the locations of the design variables with the aid of the CAD model. In the second step, the grids were manually generated for all involved disciplines. In the third step, the forward and backward mappings described in the preceding sections were calculated for each grid point for all involved disciplines. In the fourth step, the new grids were deformed in response to the new design variables, and the sensitivity derivatives were computed as well. The third and fourth steps were completely automated. The first three steps were considered preprocessing steps and needed to be done only once.

Parameterizing CSM Models

Parameterizing CFD and CSM models appears to be similar in nature, but the CSM model parameterization has two additional requirements. First, the CSM model parameterization must include not only the OML but also the internal structural elements, such as spars and ribs. Second, the deformed CSM model must be a valid design. For example, the spars must stay straight during the optimization. The algorithms presented in this paper can easily handle the first requirement. However, if the planform design variables are not selected with care, the second requirement could easily be violated. To avoid creating an invalid CSM model, the model must be parameterized with few hexahedron solid elements, and those used must be aligned with major structural components such as spars and ribs.

Results

The algorithms presented in this paper have been applied for parameterizing a simple ONERA M6 wing,[14,16] a blended wing body, and several HSCT configurations.

An aerodynamic optimization of an ONERA M6 wing was performed[14] using a sequential linear programming technique. The objective of the optimization was to minimize the drag while maintaining the same lift as the baseline design. Figure 20 shows the design cycle history for drag. In this optimization, the angle of attack is fixed, and it was found that to move away from the current design, the constraint on the lift coefficient had to be relaxed temporarily. This is shown clearly in Fig. 20: for the first 19 design cycles, C_L was allowed to deviate by up to 0.01 from the desired value. After design cycle 19 the tolerance on the lift constraint was tightened to 10^{-6}. The net result was approximately 29 counts of drag reduction at the baseline lift. Figure 21 shows comparisons of the solutions computed on the baseline and final designs. The results indicate a significant reduction in the shock strength at most spanwise stations.

This approach has also been applied to multidisciplinary optimization of an HSCT.[17,18] Figure 22 shows the design cycle history for aircraft drag, as measured relative to the baseline values. Figure 22 shows the drag has been reduced by 7.5% relative to the baseline. Although the optimizer has not fully converged for this case, the convergence history from 20 design cycles suggests that

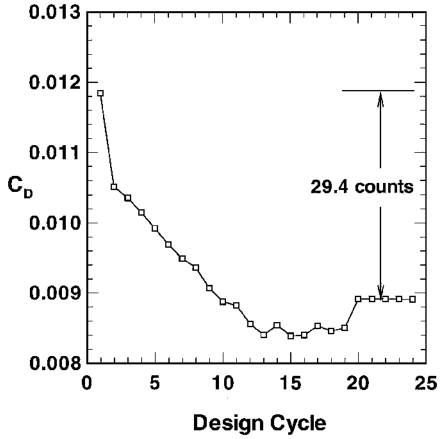

Fig. 20 Design cycle history of ONERA M6 wing optimization for coefficient of drag.

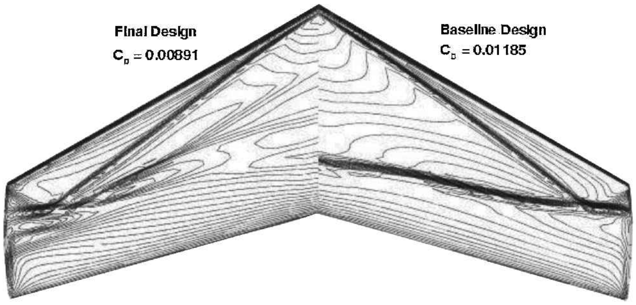

Fig. 21 Comparison of the M6 wing final design and baseline surface pressures.

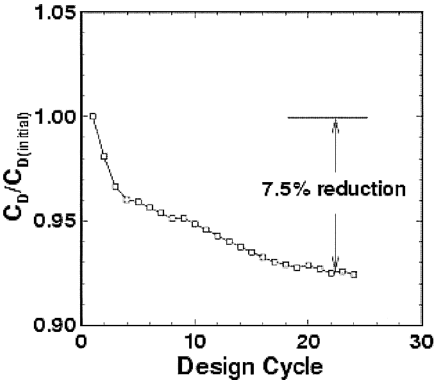

Fig. 22 Design cycle history of HSCT optimization for drag.

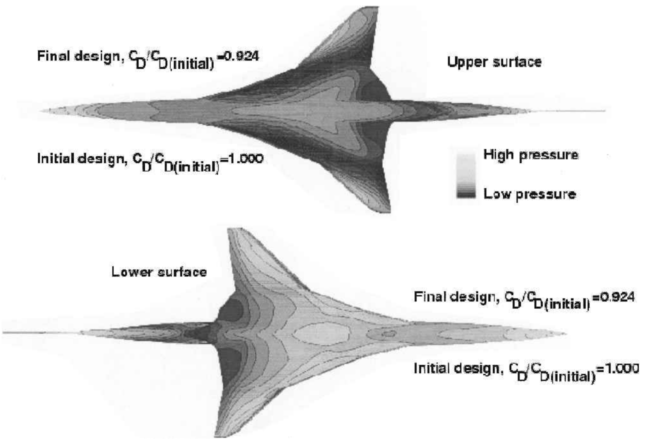

Fig. 23 Comparison of the HSCT final design and baseline surface pressures.

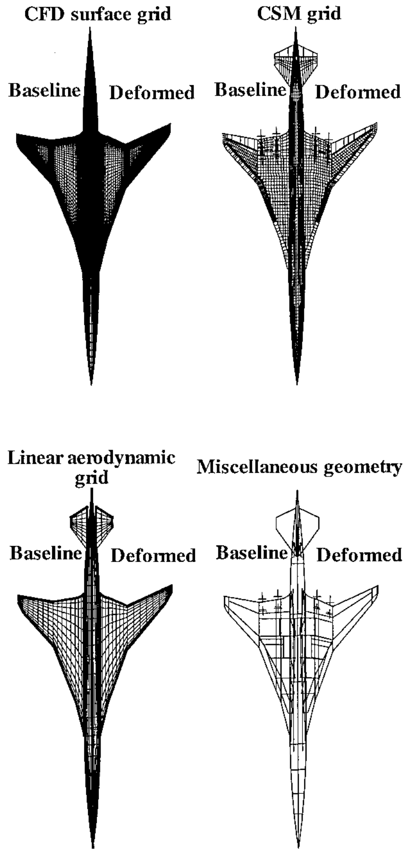

Fig. 24 Baseline and deformed models of an HSCT.

little additional reduction in drag would be obtained from additional design cycles. Figure 23 shows a comparison of the baseline and final surface pressures on both the upper and lower surfaces of a HSCT. The planform changes that occurred between the initial and final design cycles are also represented. The primary effect on the planform has been to increase the span and aspect ratio slightly and to move the outer wing leading edge break to a more inboard spanwise location. Although not evident in Fig. 23, the wing thickness has been slightly reduced. Figure 24 shows the baseline and deformed miscellaneous geometry, and CSM, CFD, and linear aerodynamic grids corresponding to the optimized shape.

Conclusions

The parameterization algorithm presented in this paper is easy to implement for an MDO application for a complex configuration. The resulting parameterization is consistent across all disciplines. Because the formulation is based on the SOA algorithms, the analytical sensitivity is also readily computed. The algorithms are based on parameterizing the shape perturbations, thus enabling the parameterization of complex existing analysis models (grids). Another benefit of parameterizing the shape perturbation is that the process requires few design variables. Use of NURBS representation provides strong local control, and smoothness can easily be controlled.

Acknowledgments

The author would like to thank Robert Biedron for providing Figs. 20–23.

References

[1] Samareh, J. A., "Status and Future of Geometry Modeling and Grid Generation for Design and Optimization," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 97–104.

[2] Samareh, J., "Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization," *AIAA Journal*, Vol. 39, No. 5, May 2001, pp. 877–884.

[3] Pickett, R. M., Rubinstein, M. F., and Nelson, R. B., "Automated Structural Synthesis Using a Reduced Number of Design Coordinates," *AIAA Journal*, Vol. 11, No. 4, 1973, pp. 494–498.

[4] Leiva, J. P., and Watson, B. C., "Automatic Generation of Basis Vectors for Shape Optimization in the GENESIS Program," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference*, AIAA, Reston, VA, 1998, pp. 1115–1122.

[5] Braibant, V., and Fleury, C., "Shape Optimal Design Using B-Splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, No. 3, Aug. 1984, pp. 247–267.

[6] Watt, A., and Watt, M., *Advanced Animation and Rendering Techniques*, Addison–Wesley, New York, 1992, Chap. 17.

[7] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.

[8] Bloor, M. I. G., and Wilson, M. J., "Efficient Parameterization of Generic Aircraft Geometry," *Journal of Aircraft*, Vol. 32, No. 6, 1995, pp. 1269–1275.

[9] Smith, R. E., Bloor, M. I. G., Wilson, M. J., and Thomas, A. T., "Rapid Airplane Parametric Input Design (RAPID)," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference*, AIAA, Washington, DC, 1995, pp. 452–462.

[10] Cosentino, G. B., and Holst, T. L., "Numerical Optimization Design of Advanced Transonic Wing Configurations," *Journal of Aircraft*, Vol. 23, No. 3, 1986, pp. 193–199.

[11] Schramm, U., and Pilkey, W. D., "Structural Shape Optimization for the Torsion Problem Using Direct Integration and B-Splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 107, No. 1–2, Aug. 1993, pp. 251–268.

[12] Anderson, W. K., and Venkatakrishnan, V., "Aerodynamics Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," AIAA Paper 97-0643, Jan. 1997.

[13] Townsend, J. C., Samareh, J. A., Weston, R. P., and Zorumski, W. E., "Integration of a CAD System into an MDO Framework," NASA TM-207672, May 1998.

[14] Biedron, R. T., Samareh, J. A., and Green, L. L., "Parallel Computation of Sensitivity Derivatives With Application to Aerodynamic Optimization of a Wing," *Proceedings of the 1998 Computational Aerosciences Workshop*, CP-20857, NASA, 1999, pp. 219–224.

[15] Green, L. L., Weston, R. P., Salas, A. O., Samareh, J. A., Townsend, J. C., and Walsh, J. L., "Engineering Overview of a Multidisciplinary HSCT Design Framework Using Medium-Fidelity Analysis Codes," *Proceedings of the 1998 Computational Aerosciences Workshop*, CP-20857, NASA, 1999, pp. 133, 134.

[16] Nielsen, E. J., and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier–Stokes Equations," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference*, AIAA, Reston, VA, 1998, pp. 825–837.

[17] Walsh, J. L., Townsend, J. C., Salas, A. O., Samareh, J. A., Mukhopadhyay, V., and Barthelemy, J. F., "Multidisciplinary High-Fidelity Analysis and Optimization of Aerospace Vehicles, Part 1: Formulation," AIAA Paper 2000-0418, Jan. 2000.

[18] Walsh, J. L., Weston, R. P., Samareh, J. A., Mason, B. H., Green, L. L., and Biedron, R. T., "Multidisciplinary High-Fidelity Analysis and Optimization of Aerospace Vehicles, Part 2: Preliminary Results," AIAA Paper 2000-0419, Jan. 2000.

[19] Hall, V., "Morphing in 2-D and 3-D," *Dr. Dobb's Journal*, Vol. 18, No. 7, 1993, pp. 18–26.

[20] Barr, A. H., "Global and Local Deformation of Solid Primitives," *Computer Graphics*, Vol. 18, No. 3, 1984, pp. 21–30.

[21] Sederberg, T. W., and Parry, S. R., "Free-Form Deformation of Solid Geometric Models," *Computer Graphics*, Vol. 20, No. 4, 1986, pp. 151–160.

[22] Abbott, I. A., and Von Doenhoff, A. E., *Theory of Wing Sections*, Dover, New York, 1959.

[23] Farin, G., *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, New York, 1990, Chap. 2.

[24] Sederberg, T. W., and Greenwood, E., "A Physically Based Approach to 2-D Shape Blending," *Computer Graphics*, Vol. 26, No. 2, 1992, pp. 25–34.

[25] Coquillart, S., "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," *SIGGRAPH*, Vol. 24, No. 4, 1990, pp. 187–196.

[26] Lamousin, H. J., and Waggenspack, W. N., "NURBS-Based Free-Form Deformation," *IEEE Computer Graphics and Applications*, Vol. 14, No. 6, 1994, pp. 95–108.

[27] Yeh, T.-P., and Vance, J. M., "Applying Virtual Reality Techniques to Sensitivity-Based Structural Shape Design," *Proceedings of 1997 ASME Design Engineering Technical Conferences*, No. DAC-3765, DETC97, American Society of Mechanical Engineers, Fairfield, NJ, 1997, pp. 1–9.

[28] Perry, E., and Balling, R., "A New Morphing Method for Shape Optimization," AIAA Paper 98-2896, June 1998.

[29] Cook, R. D., Malkus, D. S., and Plesha, M. E., *Concepts and Applications of Finite Element Analysis*, Wiley, New York, 1989, Chaps. 5, 6.