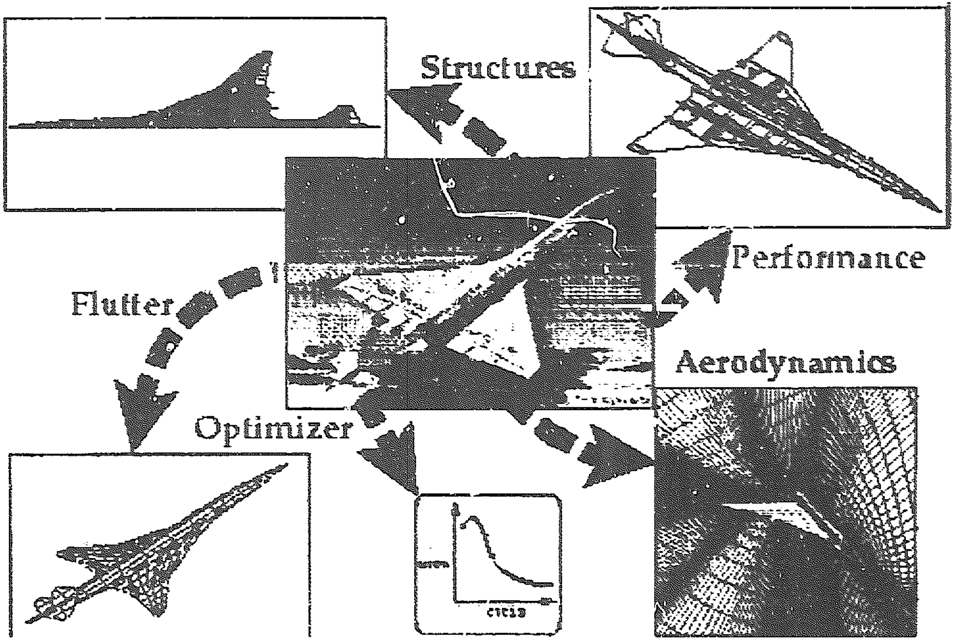
AIAA Paper

# MULTIDISCIPLINARY AERODYNAMIC-STRUCTURAL SHAPE OPTIMIZATION USING DEFORMATION (MASSOUD)

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



## 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization

September 6–8, 2000/Long Beach, CA

# MULTIDISCIPLINARY AERODYNAMIC-STRUCTURAL SHAPE OPTIMIZATION USING DEFORMATION (MASSOUD)

Jamshid A. Samareh[*]
*NASA Langley Research Center, Hampton, VA 23681*

**This paper presents a multidisciplinary shape parameterization approach. 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 in this paper is independent of grid topology, we can treat computational fluid dynamics and finite element grids in a similar 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 (e.g., linear aerodynamics and equivalent laminated plate structures) and high-fidelity (e.g., nonlinear computational fluid dynamics and detailed finite element modeling) analysis tools. This paper contains the implementation details of parameterizing for planform, twist, dihedral, thickness, camber, and free-form surface. Results are presented for a multidisciplinary design optimization application consisting of nonlinear computational fluid dynamics, detailed computational structural mechanics, and a simple performance module.**

## Nomenclature

| | |
|---|---|
| $A$ | wing area |
| $\mathcal{AR}$ | wing aspect ratio |
| $B$ | Bernstein polynomial |
| $b$ | wing span |
| $C$ | chord |
| $c$ | camber |
| $d$ | degree |
| $\epsilon$ | scale factor for twist and shearing |
| $N$ | B-spline basis function |
| $\bar{n}$ | normal vector |
| $\bar{P}$ | coordinates of NURBS control point |
| $\bar{R}$ | 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 |
| $\bar{w}$ | design variable vector |
| $X, Y, Z$ | Cartesian coordinates of deformed model |
| $x, y, z$ | Cartesian coordinates of baseline model |
| $\alpha$ | angle of attack, deg |
| $\Delta$ | total deformation |
| $\delta$ | deformation |
| $\theta$ | twist angle, deg |
| $\Lambda$ | leading edge sweep angle, deg |
| $\lambda$ | wing taper ratio |
| $\xi, \eta, \zeta$ | coordinates of deformation object |
| $\rho$ | 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 |
| $L$ | wing lower surface |
| $le$ | leading edge |
| $m$ | center |
| $p$ | degree of B-spline basis function in $i$ direction |
| pl | planform |
| $q$ | degree of B-spline basis function in $ij$ direction |
| $r$ | root |
| sh | shear |
| $te$ | trailing edge |
| $t$ | tip |
| th | thickness |
| tw | twist |
| $U$ | wing upper surface |

**Superscripts**

| | |
|---|---|
| $T$ | transpose of the matrix |

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

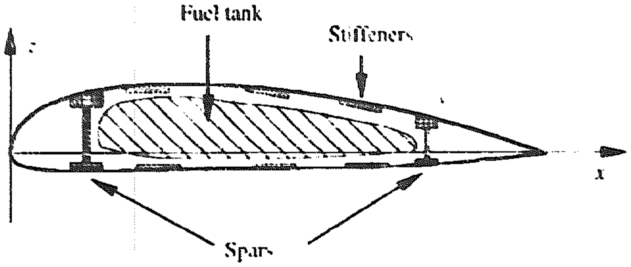


Fig. 1 Internal components of a wing.

## Introduction

MULTIDISCIPLINARY design optimization (MDO) methodology seeks to exploit the synergism of mutually interacting phenomena to create improved designs. An MDO process commonly involves sizing, topology, and shape design variables. Multidisciplinary shape optimization (MSO) finds the optimum shape for a given structural layout. It is a challenging task to perform MSO for a complete airplane configuration 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, and spars. The aerodynamic analysis uses the detailed definition of the skin, also referred to as the outer mold line (OML), whereas the computational structural mechanics (CSM) models use all components. Generally, the structural model requires a relatively coarse grid, but it must handle very complex internal and external geometries. In contrast, the computational fluid dynamics (CFD) grid is a very fine one, but it only needs to model the external geometry. 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[1] on geometry modeling and grid generation for design and optimization.

The model 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. The recent advances in computer hardware and software have made MSO applications more feasible for complex configurations. An important ingredient of aerodynamic shape optimization is the availability of a model parameterized with respect to the aerodynamic parameters, such as planform, twist, shear, camber, and thickness. The parameterization techniques can be divided into the following categories: discrete, polynomial and spline, computer-aided design (CAD), analytical, and deformation. Readers are referred to reports by Haftka,[2] Ding,[3] and Samareh[4] for surveys of shape optimization and parameterization.

In a multidisciplinary application, the parameterization must be compatible and adaptable to various analysis tools ranging from low-fidelity tools, such as linear aerodynamics and equivalent laminated plate structures, to high-fidelity tools, such as nonlinear CFD and detailed CSM codes. Creation of CFD and CSM grids is time-consuming and costly for a full airplane model: it takes several months to develop detailed CSM and CFD grids based on a CAD model. To fit the MSO process into the product development cycle times, the MSO must rely on the parameterization of the analysis grids. For a multidisciplinary problem, the process must also use a geometry model and parameterization consistently across all disciplines. For use with gradient-based optimization, the geometry model must provide accurate sensitivity derivatives of the analysis model with respect to design variables.

This paper presents an approach for shape parameterization suitable for a multidisciplinary design optimization application. The approach consists of two basic concepts. The first concept is based on parameterizing the shape perturbation rather than the geometry itself. The second concept is based on using the soft object animation[5] (SOA) algorithms for shape parameterization. The combined algorithm initially introduced by this author[1] was successfully implemented for aerodynamic shape optimization with analytical sensitivity with structured grid[6,7] and unstructured grid[8] CFD codes. This algorithm has also been applied to multidisciplinary optimization of a high-speed civil transport.[9,10]

## Parameterizing the Shape Perturbations

At first sight parameterization by splines may seem to be a viable approach for shape parameterization. The spline representation uses a set of control points to define any shape. These control points could be used as design variables for optimization. Typically over a hundred control points are required to define an airfoil section and over 20 airfoil sections to define a conventional wing. This requirement results in over two thousand control points (i.e., six thousand shape design variables) for a simple wing. The number of control points is even larger for a complete airplane model created with a commercial CAD system. The large number of control points is needed more for accuracy than for complexity.

Even if we could afford to use a large number of design variables, the automatic regeneration of analysis models (e.g., CSM and CFD grids) is not possible with the current technology. For example, it takes several months to create an accurate CSM model of an airplane. Also, traditional shape parameterization processes parameterize only the OML and are ineffective in parameterizing internal components such as spars, ribs, stiffeners, and fuel tanks (see Fig. 1).

It is possible to use any shape (e.g., a sphere) as the initial wing definition, allowing the optimizer to find the optimum wing shape; however, it is not a common practice. 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,[11,12] but the difference in wing performance can be substantial. An effective way to reduce the number of shape design variables is to parameterize the shape perturbations instead of parameterizing the shape itself. Throughout the optimization cycles, the analysis grid can be updated as

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

where $\bar{r}$ is the baseline grid, $\bar{R}$ is the deformed (perturbed) grid, $\Delta\bar{R}$ is the change (perturbation), and $\bar{v}$ is the multidisciplinary aerodynamic-structural shape optimization using deformation (MASSOUD) design variable vector. The change, $\Delta\bar{R}$, is a combination of changes in thickness, camber, twist, shear, and planform:

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

It takes far fewer design variables to parameterize the shape perturbation $\Delta\bar{R}$ than $\bar{r}$ itself.

Figures 2 and 3 contrast the typical and modified MSO processes. In a typical MSO process (Fig. 2), a geometry modeler perturbs the baseline geometry model. Because automatic grid generation tools are not available for all disciplines, it would be very difficult to automate this MSO process. In contrast, the modified MSO process (Fig. 3) relies on parameterizing the baseline grids and avoids the grid generation process, hence making it possible to automate the entire MSO process.

## Soft Object Animation

The field of SOA in computer graphics[5] provides algorithms for morphing images[13] and deforming models.[14,15] These algorithms are powerful tools for modifying shapes: they use a high-level shape deformation, as opposed to manipulation of lower level geometric entities. Hall[13] presented an algorithm and provided computer codes for morphing images. 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, while retaining its topology. This is ideal for parameterizing airplane models that have external skin as well as internal components (e.g., see Fig. 1). The SOA algorithms relate 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.

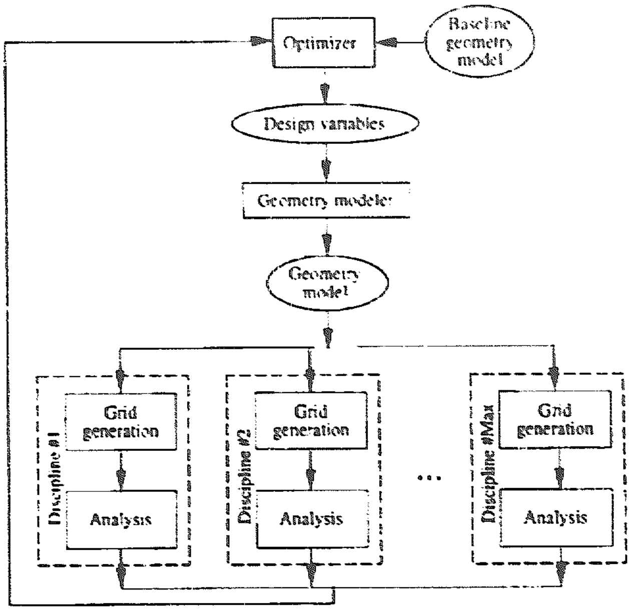


Fig. 2 A typical MSO process.

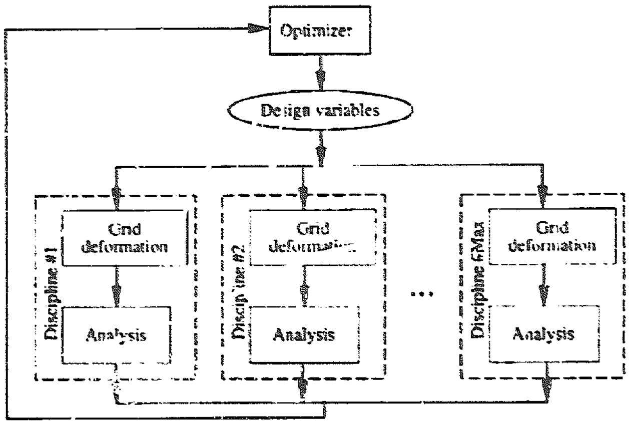


Fig. 3 The modified MSO process.

Barr[14] presented a deformation approach in the context of physically based modeling. This approach uses physical simulation to obtain realistic shape and motions and is based on operations such as translation, rotation, and scaling. With this 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[15] presented another approach for deformation based on the free-form deformation (FFD) algorithm that 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 report presents a set of modifications to the original SOA algorithms to alleviate this and other drawbacks.

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 defines the forward mapping from the deformation object coordinate system $(\xi,\eta,\zeta)$ 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 $(\xi,\eta,\zeta)$. The $\xi,\eta,\zeta$ mapping parameters are fixed and are independent of the shape perturbations. This is a preprocessing step that is required only once.

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

4. Evaluate the grid perturbation $(\Delta\bar{R})$ and shape sensitivity derivatives $(\partial\bar{R}/\partial\bar{v})$ using the $\xi,\eta,\zeta$ parameters.

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 use a nonuniform rational B-spline (NURBS) representation as the deformation object for thickness and camber parameterization. The NURBS representation combines the desirable properties of National Advisory Committee for Aeronautics (NACA) definition[16] and spline techniques, and it does not deteriorate or destroy the smoothness of the initial geometry.

The changes in thickness and camber are represented by

$$\delta\bar{R}_{\text{th}}(\xi,\eta) = \frac{\sum\limits_{i=0}^{I} N_{i,p}(\xi)\sum\limits_{j=0}^{J} N_{j,q}(\eta)W_{i,j}\bar{P}_{\text{th}_{i,j}}}{\sum\limits_{i=0}^{I} N_{i,p}(\xi)\sum\limits_{j=0}^{J} N_{j,q}(\eta)W_{i,j}} \qquad (3)$$

$$\delta\bar{R}_{\text{ca}}(\xi,\eta) = \frac{\sum\limits_{i=0}^{I} N_{i,p}(\xi)\sum\limits_{j=0}^{J} N_{j,q}(\eta)W_{i,j}\bar{P}_{\text{ca}_{i,j}}}{\sum\limits_{i=0}^{I} N_{i,p}(\xi)\sum\limits_{j=0}^{J} N_{j,q}(\eta)W_{i,j}} \qquad (4)$$

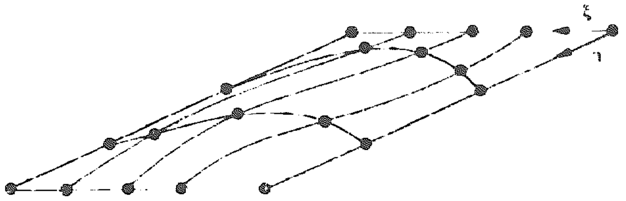


**Fig. 4 Thickness and camber definitions in wing coordinate system.**

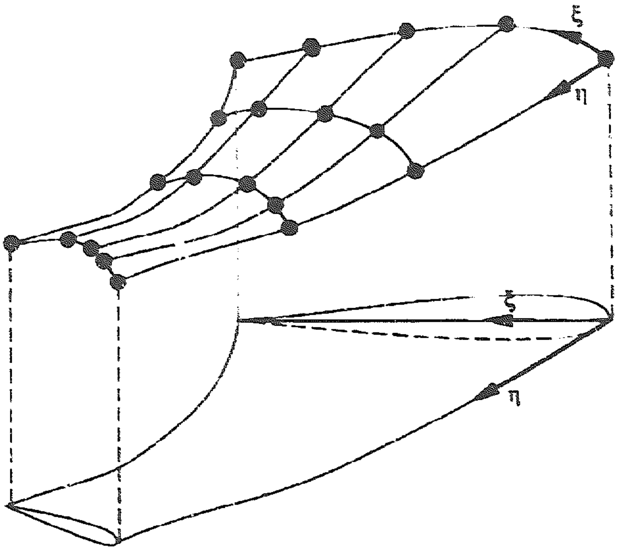


**Fig. 5 Thickness and camber definitions in $x$, $y$, and $z$ coordinate system.**

where $\bar{P}_{\text{th}_{i,j}}$ and $\bar{P}_{\text{ca}_{i,j}}$ are control points (forming a control surface) for thickness and camber, $W_{i,j}$ are the weights, and $N_{i,p}$ and $N_{j,q}$ are the $p$ and $q$ degree B-spline basis functions defined on the nonperiodic and nonuniform knot vectors. Figures 4 and 5 show the NURBS control points in $(\xi,\eta)$ and $(x,y,z)$ coordinate systems, respectively. The control points and weights could be used as design variables.

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 can handle a complex deformation and still maintain smooth surface curvature. Readers are referred to the textbook by Farin[17] for details on the properties of NURBS representation. The control points are the coefficients of the basis functions, but the smoothness is controlled by the basis functions, not 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 serve as the forward mapping between the thickness and camber design variables and the grid perturbation $(\delta\bar{R}_{\text{th}}, \delta\bar{R}_{\text{ca}})$.

The next step is to establish the backward mapping from the deformation object (i.e., NURBS surface) coordinates $(\xi,\eta)$ to the baseline model coordinates

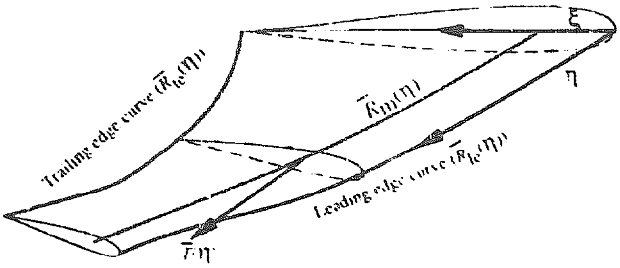

ξ
η
Trailing edge curve ($\bar{R}_{te}(\eta)$)
$\bar{R}_m(\eta)$
Leading edge curve ($\bar{R}_{le}(\eta)$)
$\bar{T}(\eta)$

**Fig. 6 Curves defining the backward mapping.**

$(x, y, z)$. The wing coordinate system—percent chord and span—is a good candidate. The percentage chord, $\%c$, is used for $\xi$, and the spanwise location, $y$, is used for $\eta$.

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

To calculate $\%c$, we need to determine the wing chord at each $y$ station. The baseline CAD model provides the leading edge ($\bar{R}_{le}(\eta)$), trailing edge ($\bar{R}_{te}(\eta)$), wing center $\bar{R}_m(\eta)$, and normal vector, defining the airfoil plane $\bar{T}(\eta)$ as shown in Fig. 6. The curve defining the wing center does not have to be at the center of the wing, but it should be somewhere between the upper and the lower wing surfaces. The $\bar{R}_{le}(\eta)$, $\bar{R}_{te}(\eta)$, and $\bar{R}_m(\eta)$ are used to separate points on the upper surface from points on the lower surface.

Because we know $\eta$ for each grid point, we can define a plane that passes through the grid point with a normal vector defined by $\bar{T}(\eta)$. We must find the intersection of this plane and the curves shown in Fig. 6,

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

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

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

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

As the design variables (control points $P_{i,j}$) change, we can calculate the contribution from the thickness and camber by Eqs. 3 and 4. The advantage of this process is that the sensitivity of grid point location with respect to design variables is only a function of the B-spline basis functions,

$$\frac{\partial \bar{R}}{\partial \bar{P}_{th_{id,jd}}} = \frac{\partial \bar{R}}{\partial \bar{P}_{ca_{id,jd}}} = \frac{N_{id,p}(\xi) N_{jd,q}(\eta) W_{id,jd}}{\sum\limits_{i=0}^{I} N_{i,p}(\xi) \sum\limits_{j=0}^{J} N_{j,q}(\eta) W_{i,j}} \tag{9}$$

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

## 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 are used to modify the wing twist and shear distribution. Alan Barr presented a series of SOA algorithms for twisting, bending, and tapering an object.[14] Watt and Watt referred to these algorithms as nonlinear global deformation.[5] Sederberg and Greenwood extended Barr's ideas to handle complex shapes.[18] Modified versions of these algorithms are presented in this paper.

To modify the twist and shear distributions, the wing is embedded in a nonlinear deformation object referred to as a twist cylinder, that is shown in Fig. 7. The twist cylinder is also used for modifying shear distribution. The center of the cylinder is defined by a NURBS curve, $\bar{R}_m(\eta)$. The effect of deformation can be confined to a section of a wing by limiting the parameter $\eta$ to vary between $\eta_{\min}$ and $\eta_{\max}$. The $\eta_{\min}$ can extend to the wing root, and the $\eta_{\max}$ can go beyond the wing tip. The cylinder can be twisted and sheared only in a plane (twist plane) defined by a point along $\bar{R}_m(\eta)$ with a normal vector of $\bar{T}(\eta)$. The $\rho_i(\eta)$ and $\rho_o(\eta)$ are the radii of inner and outer cylinders, respectively (see Fig. 7). The deformation has no effect for grid points located outside of the outer cylinder, and the effect of deformation is scaled linearly from the outer cylinder to the inner cylinder. This allows

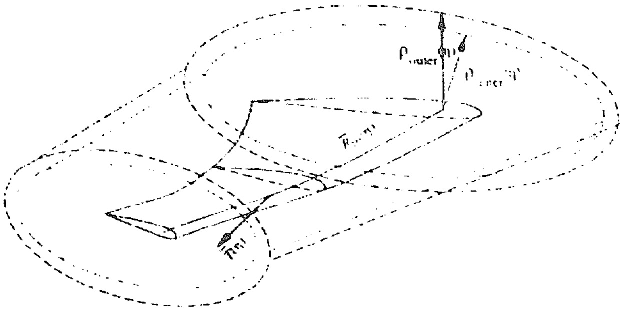


**Fig. 7 Twist definition.**

us to blend the deformed region with the undeformed region in a continuous manner.

The angle $\theta(\eta)$ defines the desired twist angle distribution, and $\bar{S}(\eta)$ defines the shearing vector. The $\theta(\eta)$ and $\bar{S}(\eta)$ variables are defined by NURBS representation:

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

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

where $\theta_i$ and $\bar{S}_i$ are the twist and shear design variables, respectively. Similar to thickness ana camber algorithms, we use

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

The second step for twist and shear deformation is to establish the forward mapping from the deformation object (twist cylinder) coordinate system $(\eta)$ to the model coordinate system $(x, y, z)$. We use Eq. 8 to determine $\eta$. Once $\eta$ is determined, we can calculate the local $\rho(\eta)$, $\rho_i(\eta)$, $\rho_o(\eta)$, $\bar{T}(\eta)$, $\theta(\eta)$, and $\bar{S}(\eta)$. The point $\bar{r}$ is rotated $\theta(\eta)$ degrees about $\bar{R}_m(\eta)$ and sheared $\bar{S}$.

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

$$\delta \bar{R}_{sh}(\eta) = \epsilon(\eta)\bar{S}(\eta) \quad (14)$$

where $\epsilon(\eta)$ is a scale factor which diminishes the effect of deformation as we approach the outer cylinder.

$$\epsilon(\eta) = \begin{cases} 0 & \text{if } \rho(\eta) \geq \rho_o(\eta) \\ \frac{\rho(\eta)-\rho_o}{\rho_i(\eta-\rho_o} & \text{if } \rho_i \leq \rho(\eta) < \rho_o(\eta) \\ 1 & \text{if } \rho(\eta) < \rho_i \end{cases} \quad (15)$$

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

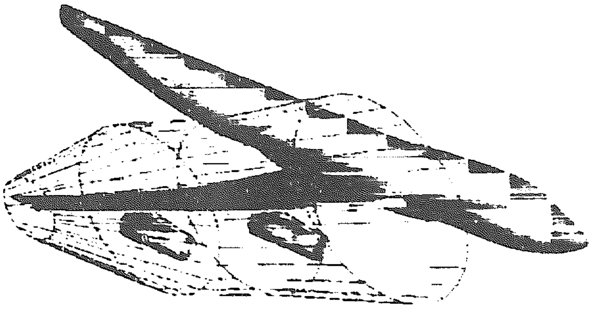


**Fig. 8 Twist definition for a transport.**

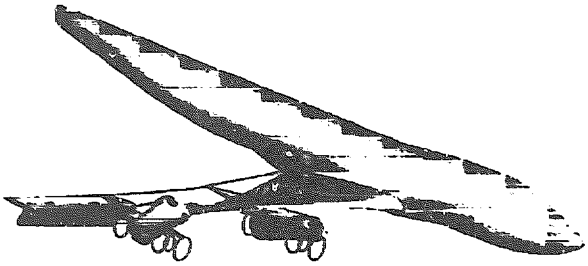


**Fig. 9 Result of 45° twist on a transport.**

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

$$\frac{\partial R}{\partial \bar{S}_i} = \epsilon(\eta)\frac{\partial \bar{S}(\eta)}{\partial \bar{S}_i} \quad (17)$$

The term $\partial\theta(\eta)/\partial\theta_i$ is independent of the twist design variables $\theta_i$ (see Eq. 10). However, $\sin\theta(\eta)$ and $\cos\theta(\eta)$ depend on the twist design variables and must be updated every cycle of the optimization. In contrast, the term $\partial\bar{S}(\eta)/\partial\bar{S}_i$ is independent of shear design variables $\bar{S}_i$ (see Eq. 11).

Figure 8 shows the inner twist cylinder for a commercial transport. Figure 9 shows the result of twisting the wing 45° at the tip. This is a large and unrealistic amount of twist, but it shows 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 10 shows the planform of a generic high-speed civil transport that uses two trapezoids. As shown in Fig. 11, each trapezoid is defined by the root chord ($C_r$), tip chord ($C_t$), span ($b$), and sweep angle ($\Lambda$). From these, other planform parameters such as area ($A$), aspect ratio ($\mathcal{R}$), and taper ratio ($\lambda$), are defined:

$$A = \frac{b}{2}(C_r + C_t), \qquad \mathcal{R} = \frac{b^2}{A}, \qquad \lambda = \frac{C_t}{C_r} \quad (18)$$

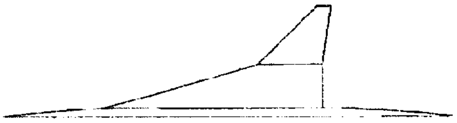


Fig. 10 Planform of a generic high-speed civil transport.

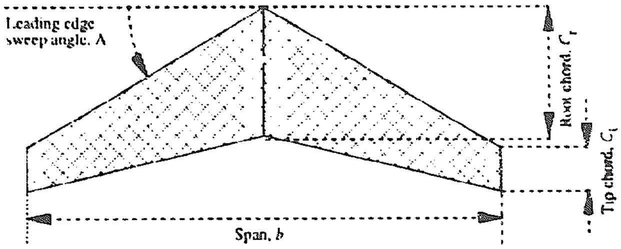


Fig. 11 Planform definition.

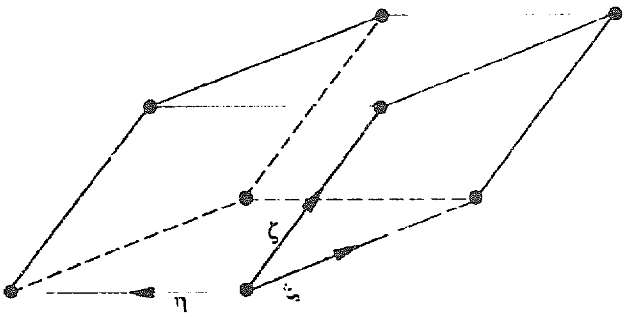


Fig. 12 Parallelepiped volume for free-form deformation.

The FFD algorithm described by Sederberg and Parry[15] is ideal for deforming the polygonal models. Like other SOA algorithms, this algorithm maintains the polygon connectiv ty, and the deformation is applied onl: 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 shape of these deformation objects is not arbitrary. In fact, they must be three-dimensional parametric volumes, which could range from a parallelepiped as shown in Fig. 12 to a general NURBS volume as shown in Fig. 13. The block is deformed by perturbing the vertices that control the shape of the deformation block (e.g., corners of the parallelepiped). For parametric volume blocks, parameters controlling the deformation are related through the mapping coordinates $(\xi,\eta,\zeta)$. These coordinates are used in both forward and backward mapping.

Figure 12 shows a general parallelepiped defined by a set of control points forming three primary edges or directions along $\xi$, $\eta$, and $\zeta$. The relation for a parallelepiped is defined as

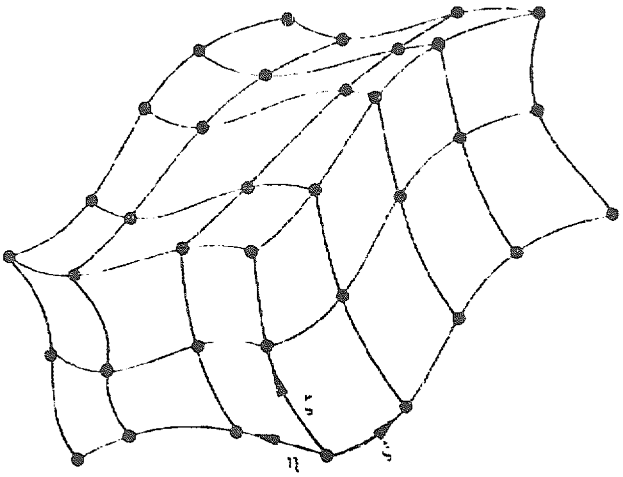


Fig. 13 NURBS volume for free-form deformation.

$$r(\xi,\eta,\zeta) = \bar{P}_0 + n_\xi\xi + \bar{n}_r\eta + \bar{n}_\zeta\zeta \qquad (19)$$

where $\bar{P}_0$ is the origin of the parallelepiped, and $\bar{n}_\xi$, $\bar{n}_\eta$, and $\bar{n}_\zeta$ are the unit vector: along the parallelepiped primary edges in $\xi$, $\eta$, and $\zeta$ directions, respectively Equation (19) defines a mapping between the deformation object (paralielepiped) and the grid ; oint. The grid points, $\bar{r}$, are mapped to the coordinates of the parallelepiped, $\xi$, $\eta$, and $\zeta$, as

$$\begin{aligned}
\xi &= \frac{\bar{n}_\eta \times \bar{n}_\zeta \cdot (\bar{r} - \bar{P}_0)}{\bar{n}_\eta \times \bar{n}_\zeta \cdot (\bar{n}_\xi)} \\
\eta &= \frac{\bar{n}_\xi \times \bar{n}_\zeta \cdot (\bar{r} - \bar{P}_0)}{\bar{n}_\xi \times \bar{n}_\zeta \cdot (\bar{n}_\eta)} \\
\zeta &= \frac{\bar{n}_\xi \times \bar{n}_\eta \cdot (r - \bar{P}_0)}{\bar{n}_\xi \times \bar{n}_\eta \cdot (\bar{n}_\zeta)}
\end{aligned} \qquad (20)$$

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. It is suitable for local and global deformation. The only drawback is that the use of the parallelepiped limits the topology of deformation. To alleviate this drawback, Sederberg and Parry proposed to use nonparallelepiped objects.[15] They also noted that the inverse mapping would be nonlinear and require significant computations.

Another popular method to define FFD is to use trivariate parametric volumes. Sederberg and Parry used a Bezier volume.[15] Coquillart at INRIA extended Bezier parallelepiped to nonparallelepiped cubic Bezier volume.[19] This idea has been further generalized to NURBS volume by Lamousin and Waggenspack.[20] The NURBS blocks are defined as

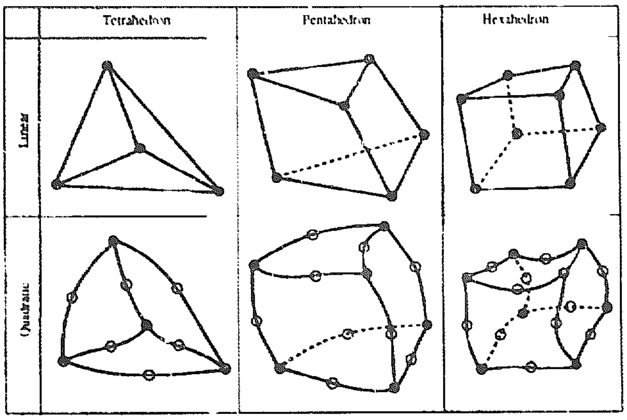


Fig. 14 FE analysis solid elements.

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

where $N$ is the B-spline basis function, and the $p1$-$p3$ are the degrees of $N$. The $\bar{P}_{i,j,k}$ are the NURBS control points that are related to the design variables. Lamousin and Waggenspack[20] used multiple blocks to model complex shapes. This technique has been used for design and optimization by Yeh and Vance[21] and also by Perry and Balling.[22]

The common solid elements used in FE analysis (Fig. 14) can be used as deformation objects. The mapping from the solid element coordinates is defined[23] by

$$\bar{r}(\xi,\eta,\zeta)=\sum_{i}\bar{P}_i N_i(\xi,\eta,\zeta) \tag{22}$$

where $N_i$ are the FE basis functions, and $\bar{P}_i$ are the nodal coordinates of deformation objects, which are related to the design variables. The equations for the inverse mapping are nonlinear for all solid elements with the exception of tetrahedron solid elements. The solid elements provide a flexible environment 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 have used hexahedron solid elements with four opposing edges parallel to the $z$-coordinate. Then, the planform design variables are linked to the corners of the hexahedral elements. Figure 15 shows the initial and deformed model for 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 is independent of the design variables ($\bar{P}_{id,jd}$) and coordinates ($x, y, z$). Thus, we need to calculate it only once, at the beginning of the optimization.

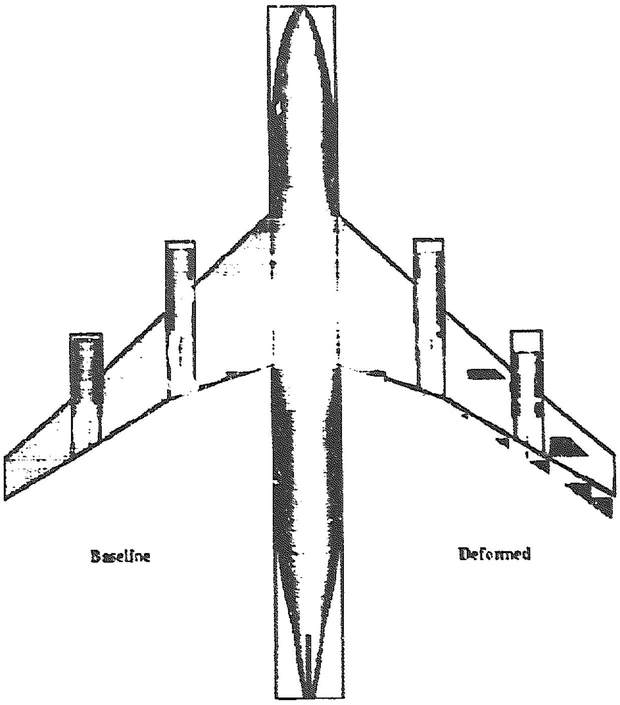


Fig. 15 Planform deformation of a transport.

## Implementation

Figure 16 shows the implementation diagram for the combined algorithm. The implementation starts with a CAD model that defines the geometry. The first two steps can be implemented in parallel. The first step is to determine the number and the locations of the design variables with the aid of the CAD model. In the second step, the grids are manually generated for all involved disciplines. In the third step, the mappings described in the previous sections are calculated for each grid point. In the fourth step, the new grid is deformed in response to the new design variables, and the sensitivity derivatives are computed as well. The third and fourth steps are completely automated. The first three steps are considered preprocessing steps and need to be done only once.

## Parameterizing Computational Structural Mechanics 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

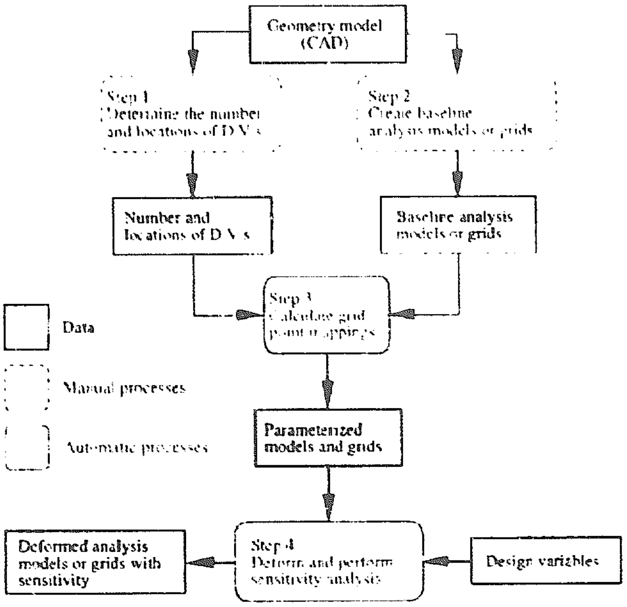

Geometry model (CAD)
Step 1 Determine the number and locations of D.V.s
Step 2 Create baseline analysis models or grids
Number and locations of D.V.s
Baseline analysis models or grids
Step 3 Calculate grid point mappings
Data
Manual processes
Automatic processes
Parameterized models and grids
Deformed analysis models or grids with sensitivity
Step 4 Deform and perform sensitivity analysis
Design variables

**Fig. 16 Implementation diagram.**

first requirement. However, if the planform design variables are not selected with care, the second requirement could easily be violated. To avoid creating invalid CSM models, planform must be parameterized with few hexahedron solid elements, and they must be aligned with major structural components such as spars and ribs.

## Sensitivity Analysis

Sensitivity derivatives are defined as the derivatives of the coordinate locations with respect to the design variables. The previous sections present a formulation for shape parameterization based on a specific set of design variables ($v_i$, $i = 1, i_{\max}$). It is possible to introduce a new set of design variables ($w_j$, $j = 1, j_{\max}$). The sensitivity derivatives with respect to $w_j$ is computed based on the chain rule differentiation as

$$\frac{\partial \bar{R}}{\partial w_j} = \frac{\partial \bar{R}}{\partial v_i}\frac{\partial v_i}{\partial w_j} \tag{23}$$

The previous sections provide techniques to compute the first term on the right-hand side. The second term is defined in a matrix form where the matrix has $i_{\max}$ rows and $j_{\max}$ columns.

$$\begin{bmatrix} \frac{\partial v_1}{\partial w_1} & \frac{\partial v_1}{\partial w_2} & \cdots & \frac{\partial v_1}{\partial w_{j_{\max}}} \\ \frac{\partial v_2}{\partial w_1} & \frac{\partial v_2}{\partial w_2} & \cdots & \frac{\partial v_2}{\partial w_{j_{\max}}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial v_{i_{\max}}}{\partial w_1} & \frac{\partial v_{i_{\max}}}{\partial w_2} & \cdots & \frac{\partial v_{i_{\max}}}{\partial w_{j_{\max}}} \end{bmatrix} \tag{24}$$

## Design-Variable Sequencing

In a typical optimization problem, the number of design variables is determined a priori. However, it is possible to use an adaptive algorithm to determine the number of design variables. The design variables are control points of a NURBS curve or surface. Optimization of a wing section could start with only three design variables (see Fig. 17). Then, the number of design variables can be increased to five by enriching the NURBS curve, which is accomplished by inserting additional knots.

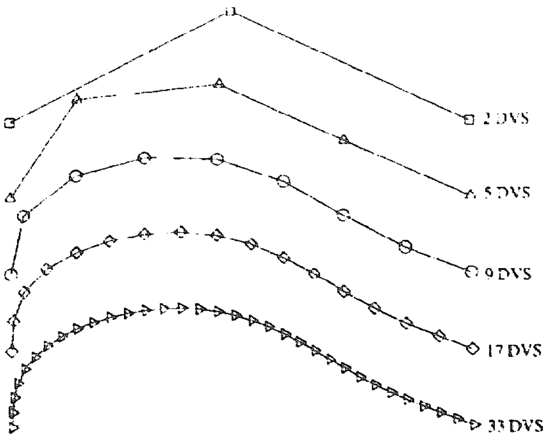


**Fig. 17 A sequence of design-variable sets.**

This method is similar to mesh sequencing and multigrid methods used in CFD to accelerate the convergence. Multigrid method exhibits a convergence rate that is independent of the number of unknowns in the discretized system. The final paper will contain results of the proposed sequencing algorithm.

## Results and Conclusions

The algorithms presented in this paper have been applied for parameterizing a simple wing, a blended wing body, and several high-speed civil transport configurations. Figure 18 shows the baseline and deformed grids for a high-speed civil transport. The solid lines represent the hexahedron solid elements controlling the planform variation. The parameterization results from this research have been successfully implemented for aerodynamic shape optimization with analytical sensitivity with structured[6] and unstructured CFD grids.[8] This approach has also been applied to multidisciplinary optimization of a high-speed civil transport.[9,10]

The parameterization algorithm presented in this paper is easy to implement for an MDO application with complex configuration. The resulting parameterization is consistent across all disciplines. Because the formulation is based on the SOA algorithms, the analytical sensitivity is also available. The algorithms are based on parameterizing the shape perturbations, thus enabling the parameterization of complex existing

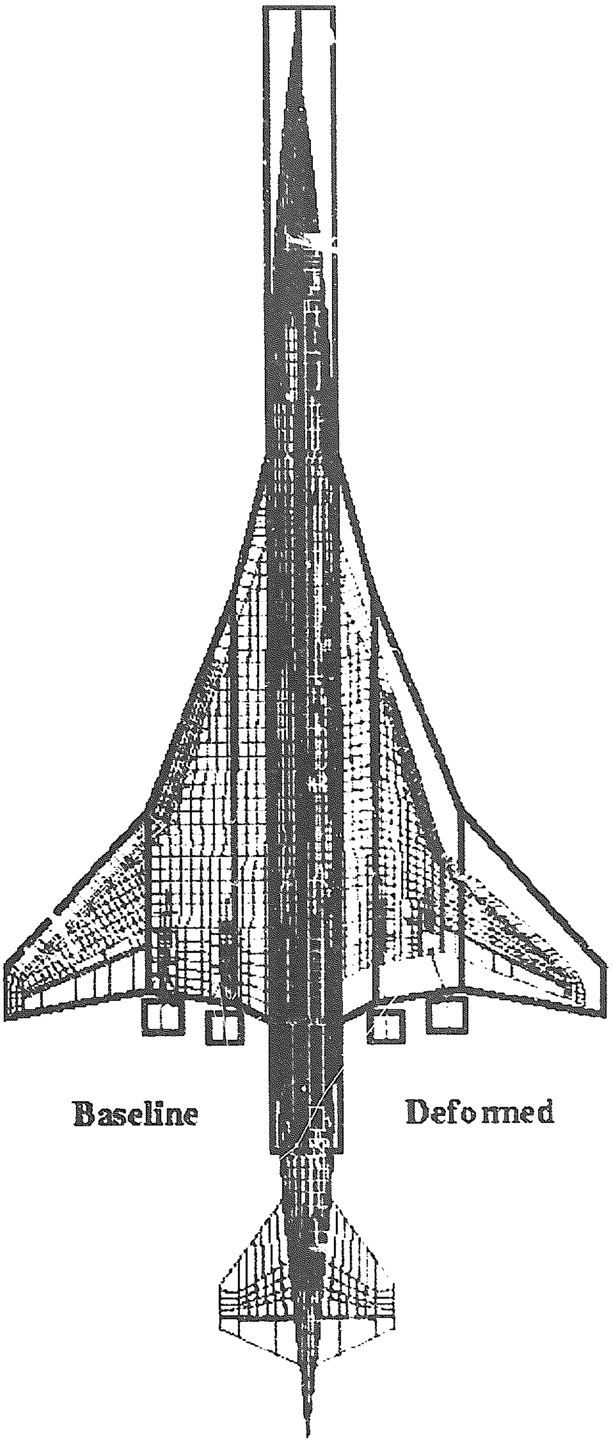


**Fig. 18 Parameterization of the CSM Model of HSCT**

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 the smoothness can easily be controlled.

The final paper will contain the proposed sequencing algorithm, the free-form surface object, and corresponding results.

## Acknowledgments

The author would like to thank Drs. James Townsend, Perry Newman and Thomas Zang of Multidisciplinary Optimization Branch of NASA Langley Research Center and Susan Costello of NCI Information Systems for reviewing this paper.

## 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] Haftka, R. T. and Grandhi, R. V., "Structural Shape Optimization – A Survey," *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, No. 1, 1986, pp. 91–106.

[3] Ding, Y., "Shape Optimization of Structures: A Literature Survey," *Computers & Structures*, Vol. 24, No. 6, 1986, pp. 985–1004

[4] Samareh, J. A., "A Survey of Shape Parameterization Techniques," *AIAA/CEAS/ICASE/NASA-LaRC International Forum on Aeroelasticity and Structural Dynamics Conference*, Jun. 1999.

[5] Watt, A. and Watt, M., *Advanced Animation and Rendering Techniques*, Addison-Wesley Publishing Company, New York, 1992.

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

[7] 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," *1998 Computer Aerosciences Workshop*, Jan. 1999, pp. 133–134.

[8] Nielsen, E. J. and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference Proceedings*, Sep. 1998, pp. 825–837, also AIAA–98–4809-CP.

[9] 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 Vehicle, Part 1: Formulation," Paper 2000–0418, AIAA, Jan. 2000.

[10] 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 Vehicle, Part 2: Preliminary Results," Paper 2000–0419, AIAA, Jan. 2000.

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

[12] 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.

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

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

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

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

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

[18] 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.

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

[20] 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.

[21] 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 in DETC97, Sep. 1997, pp. 1-9.

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

[23] Cook, R. D., Malkus, D. S., and Plesha, M. E., *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, New York, 1989.