

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 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 (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 application consisting of nonlinear computational fluid dynamics, detailed computational structural mechanics, and a simple performance module.

Nomenclature

A	wing area	α	angle of attack, deg
AR	wing aspect ratio	Δ	total deformation
B	Bernstein polynomial	δ	deformation
b	wing span	θ	twist angle, deg
C	chord	Λ	leading edge sweep angle, deg
C_D, C_L	coefficients of drag and lift	λ	wing taper ratio
c	camber	ξ, η, ζ	coordinates of deformation object
d	degree	ρ	twist radius
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 the solid elements		
\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		

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

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

This paper is a work of the U.S. Government and is not subject to copyright protection in the United States.

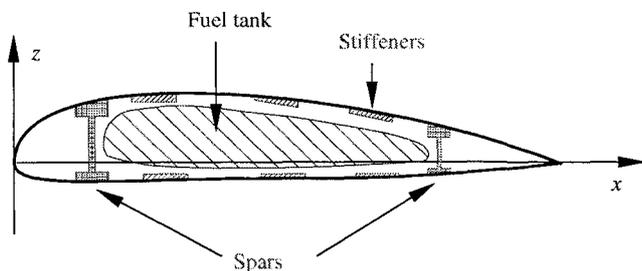


Fig. 1 Internal components of a wing.

Superscripts

T transpose of the matrix

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. 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 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 on geometry modeling and grid generation for design and optimization.¹

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. 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 aerody-

dynamic design variables, 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 and Grandhi,² Ding,³ and Samareh⁴ for surveys of shape optimization and parameterization.

In a multidisciplinary application, the parameterization must be compatible with 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: detailed CSM and CFD grids based on a CAD model take several months to develop. To fit the MSO process into product development cycle times, the MSO must rely on parameterizing the analysis grids. For a multidisciplinary problem, the process must also 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.

This paper presents a shape parameterization approach suitable for MSO as part of a multidisciplinary design optimization application. The approach consists of two basic concepts. The first concept is based on parameterizing the shape perturbations rather than the geometry itself. The second concept is based on using the soft object animation⁵ (SOA) algorithms for shape parameterization. The combined algorithm initially introduced by this author¹ was successfully implemented for aerodynamic shape optimization with analytical sensitivity for structured grid^{6,7} and unstructured grid⁸ CFD codes. This algorithm has also been used for multidisciplinary application of a high-speed civil transport (HSCT).^{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, which could define any arbitrary 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 using a large number of design variables were feasible, the automatic regeneration of analysis models

(e.g., CSM and CFD grids) is not possible with 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, this option is not commonly practiced. 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

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

For multidisciplinary aerodynamic-structural shape optimization using deformation (MASSOUD), the change, $\Delta\bar{R}$, is a combination of changes in thickness, camber, twist, shear, and planform:

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

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

Figures 2 and 3 depicts the typical MSO and MASSOUD 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, automating this MSO process would be very difficult. In contrast, the MASSOUD process (Fig. 3) relies on parameterizing the baseline grids and avoids the grid generation process, hence automating the entire MSO process.

Soft Object Animation

The field of SOA in computer graphics⁵ provides algorithms for morphing images¹³ and deforming models.^{14,15} 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.¹³ 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,

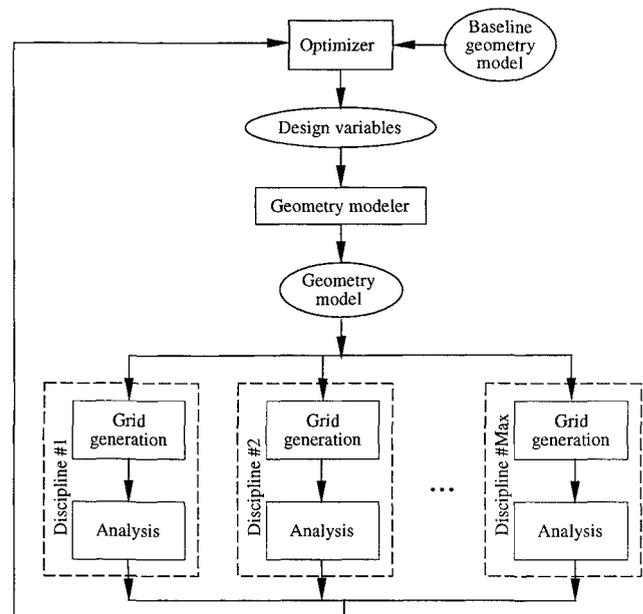


Fig. 2 A typical MSO process.

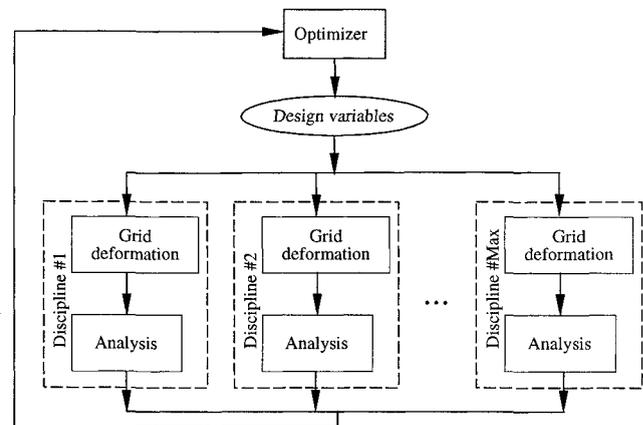


Fig. 3 The MASSOUD process.

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 (e.g., 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.¹⁴ 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 variant¹⁵ of the free-form deformation (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 manip-

ulate 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 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 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 $(\Delta \bar{R})$ and shape sensitivity derivatives $(\partial \bar{R} / \partial v)$ with the ξ, η, ζ 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 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 National Advisory Committee for Aeronautics (NACA) definition¹⁶ 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

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

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

Figures 4 and 5 show the NURBS control points in (ξ, η) and (x, y, z) coordinate systems, respectively.

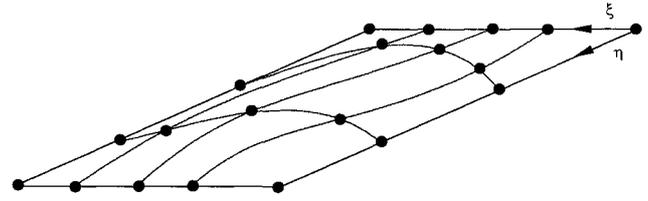


Fig. 4 Thickness and camber definitions in wing coordinate system.

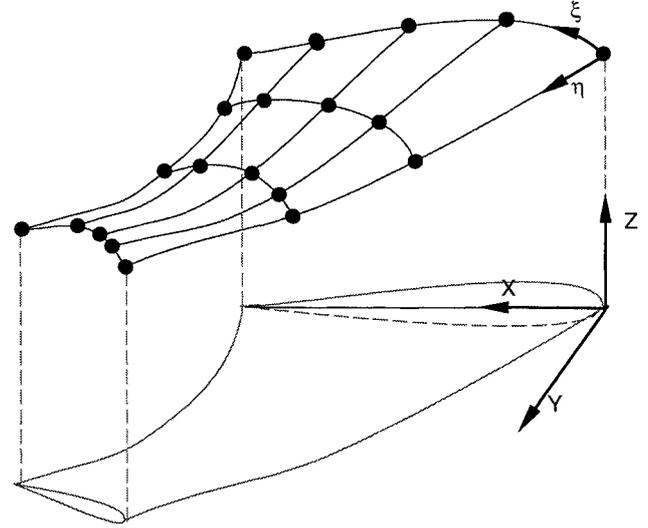


Fig. 5 Thickness and camber definitions in $x, y,$ and z coordinate system.

The control points and weights could be used as design variables.

The NURBS representation had 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 was able to handle a complex deformation and still maintain smooth surface curvature. Readers are referred to the textbook by Farin for details on the properties of NURBS representation.¹⁷ The control points were the coefficients of the basis functions, but the smoothness was controlled by the basis functions, not by the control points. The NURBS representation was local in nature, allowing the surface to be deformed locally, hence leaving the rest of the surface unchanged. Equations (3-4) served as the forward mapping between the thickness and camber design variables and the grid perturbation $(\delta \bar{R}_{th}, \delta \bar{R}_{ca})$.

The next step was to establish the backward mapping from the deformation object (i.e., 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 .

$$\xi = \%C, \quad \eta = y \quad (5)$$

To calculate $\%C$, we needed to determine the wing

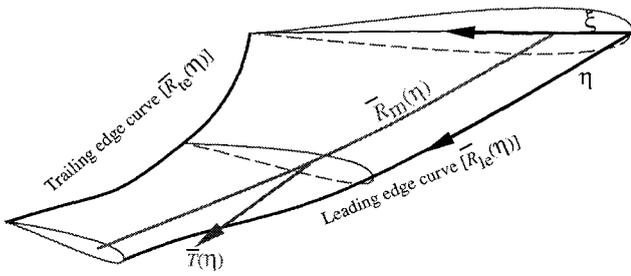


Fig. 6 Curves defining the backward mapping.

chord at each y station. The baseline CAD model provided the leading edge $\bar{R}_{le}(\eta)$, trailing edge $\bar{R}_{te}(\eta)$, wing midline $\bar{R}_m(\eta)$, and normal vector defining the airfoil plane $\bar{T}(\eta)$, as shown in Fig. 6. 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 $\bar{R}_{le}(\eta)$, $\bar{R}_{te}(\eta)$, and $\bar{R}_m(\eta)$ 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 $\bar{T}(\eta)$. We then used the following equations to find the intersection of this plane and the curves shown in Fig. 6.

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

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

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

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 $\bar{R}_m(\eta)$ to define the curve. With this approach, the deformation may be localized to a specific design area by setting allowable %C_{min}, %C_{max}, η_{min} , and η_{max} .

As the design variables (control points $\bar{P}_{i,j}$) changed, we calculated the contribution to the thickness and camber by Eqs. (3–4). 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.

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

Consequently the sensitivity, as shown in Eq. (9), was independent of the design variables ($\bar{P}_{i,d,jd}$) 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.

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. 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 was embedded in a nonlinear deformation object referred to as a twist cylinder, shown in Fig. 7. The center of the cylinder was defined by a NURBS curve, $\bar{R}_m(\eta)$. The effect of deformation was confined to a section of a wing by limiting the parameter η to vary between η_{min} and η_{max} . The η_{min} could be extended to the wing root, and the η_{max} went beyond the wing tip. The cylinder could be twisted and sheared only in a plane (twist plane) defined by a point along $\bar{R}_m(\eta)$ with a normal vector of $\bar{T}(\eta)$. The $\rho_{in}(\eta)$ and $\rho_{out}(\eta)$ were the radii of the inner and outer cylinders, respectively (see Fig. 7). 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 $\bar{S}(\eta)$ variables are defined by the NURBS representations:

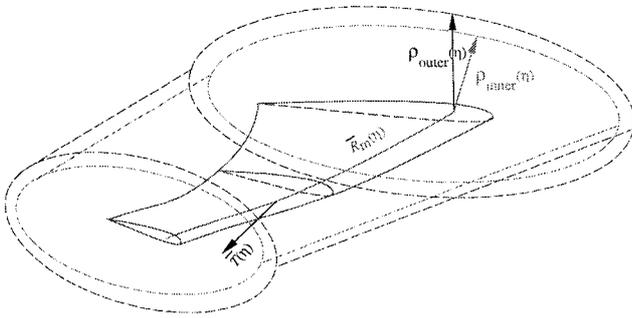


Fig. 7 Inner twist cylinder.

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

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

Similarly to thickness and camber algorithms, we used

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

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 $\rho(\eta)$, $\rho_{in}(\eta)$, $\rho_{out}(\eta)$, $\bar{T}(\eta)$, $\theta(\eta)$, and $\bar{S}(\eta)$. The point \bar{r} was rotated $\theta(\eta)$ deg about $\bar{R}_m(\eta)$ and sheared $\bar{S}(\eta)$.

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

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

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

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

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

$$\begin{aligned} \frac{\partial \bar{R}}{\partial \theta_i} &= e(\rho) \rho(\eta) \frac{\partial \theta(\eta)}{\partial \theta_i} [\cos \theta(\eta), 0, -\sin \theta(\eta)]^T \\ \frac{\partial \bar{R}}{\partial \bar{S}_i} &= e(\rho) \frac{\partial \bar{S}(\eta)}{\partial \bar{S}_i} \end{aligned} \quad (17)$$

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

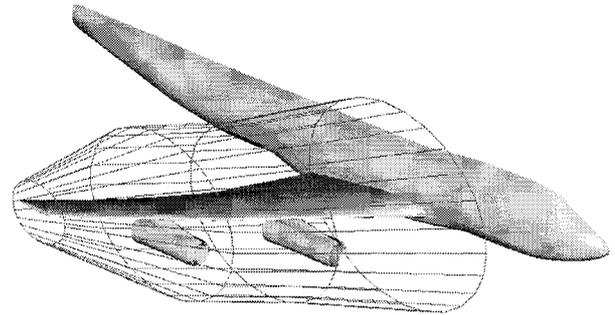


Fig. 8 Twist definition for a transport.

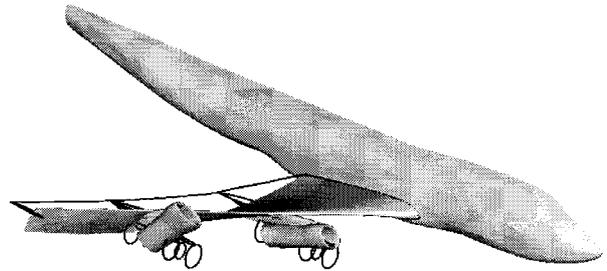


Fig. 9 Result of 45 deg twist on a transport wing tip.



Fig. 10 Planform of a generic HSCT.

$\cos \theta(\eta)$ depended on the twist design variables and were updated for every cycle of the optimization. In contrast, the term $\partial \bar{S}(\eta) / \partial \bar{S}_i$ was 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 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 10 shows the planform of a generic HSCT 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 (Λ). From these values, other planform parameters such as area (A), aspect ratio (AR), and taper ratio (λ), are defined:

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

The FFD algorithm described by Sederberg and Parry¹⁵ is ideal for deforming the polygonal models.

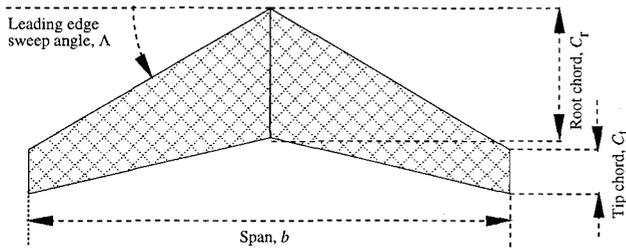


Fig. 11 Planform definition.

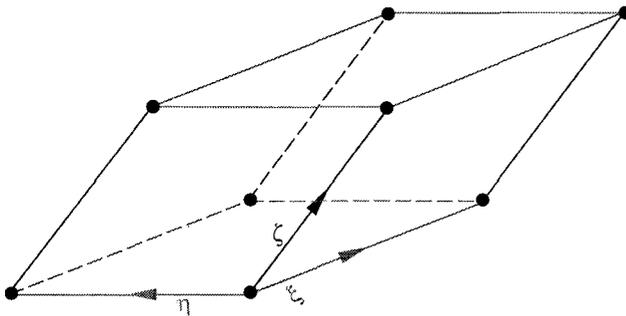


Fig. 12 Parallelepiped volume for FFD.

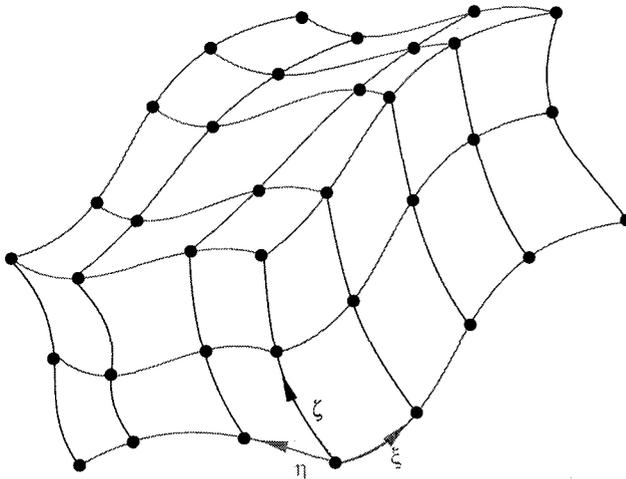


Fig. 13 NURBS volume for FFD.

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. 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 (ξ, η, ζ) . 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 ξ , η , and ζ . The relation for a parallelepiped is defined as

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

where \bar{O} is the origin of the parallelepiped, and \bar{n}_ξ , \bar{n}_η , and \bar{n}_ζ 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

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

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.¹⁵ 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 used a Bezier volume.¹⁵ Coquillart at INRIA extended the 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

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

Lamousin and Waggenspack²⁰ used multiple blocks to model complex shapes. This technique has been used for design and optimization by Yeh and Vance²¹ and also by Perry and Balling.²²

The common solid elements used in FE analysis (Fig. 14) can be used as deformation objects as well.

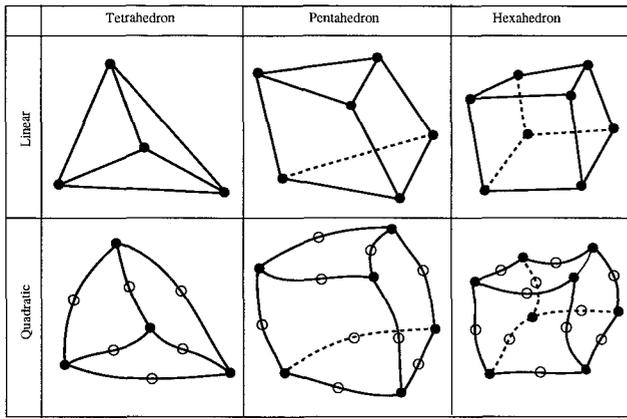


Fig. 14 FE analysis solid elements.

The mapping from the solid element coordinates is defined²³ by

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

where N_i are the FE basis functions and \bar{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 15 shows the initial and deformed models 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 was independent of the design variables (\bar{P}) and coordinates (x, y, z). Thus, we needed to calculate the sensitivity only once, at the beginning of the optimization.

Implementation

Figure 16 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 previous sections were calculated for each grid point. In the fourth step, the new grid was deformed in response to the new design variables, and the sensitivity derivatives were

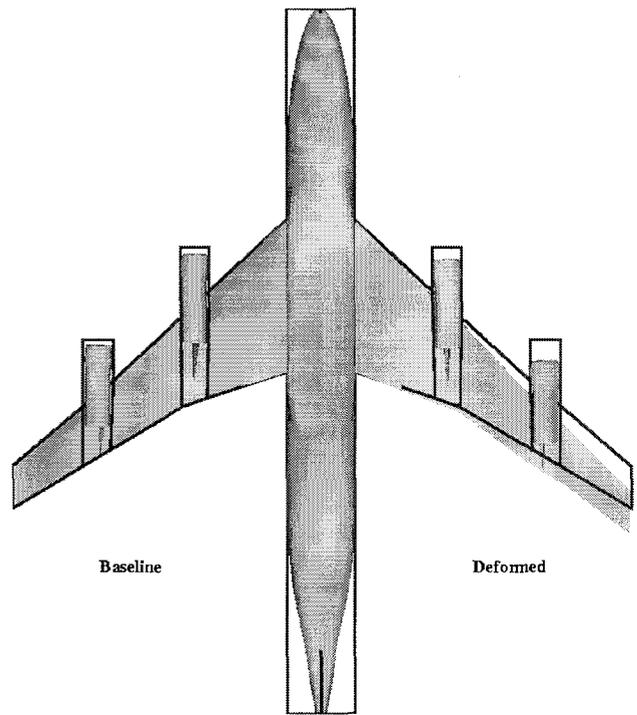


Fig. 15 Planform deformation of a transport.

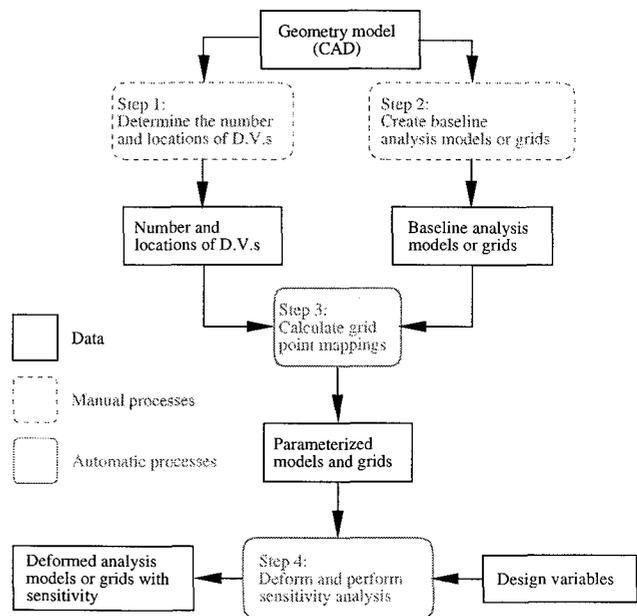


Fig. 16 MASSOUD Implementation diagram.

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.

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 (\bar{v}_i , $i = 1, i_{\max}$). It is possible to introduce a new set of design variables (\bar{w}_j , $j = 1, j_{\max}$). The sensitivity derivatives with respect to \bar{w}_j were computed based on the chain rule differentiation as

$$\frac{\partial \bar{R}}{\partial \bar{w}_j} = \frac{\partial \bar{R}}{\partial \bar{v}_i} \frac{\partial \bar{v}_i}{\partial \bar{w}_j} \quad (23)$$

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.

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

Design-Variable Sequencing

In a typical optimization problem, the number of design variables is determined a priori. However, the NURBS representation algorithms permits the use of 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 as few as three design variables (see Fig. 17). Then, the number of design variables can be increased to five by enriching the NURBS curve or surface. Enrichment is accomplished by inserting additional knots into the design NURBS curve or surface.

This method is similar to mesh sequencing and multigrid methods used in CFD to accelerate the convergence. Multigrid methods exhibit a convergence rate independent of the number of unknowns in the discretized system.²⁴ Using MASSOUD for design-variable sequencing needs to be investigated further.

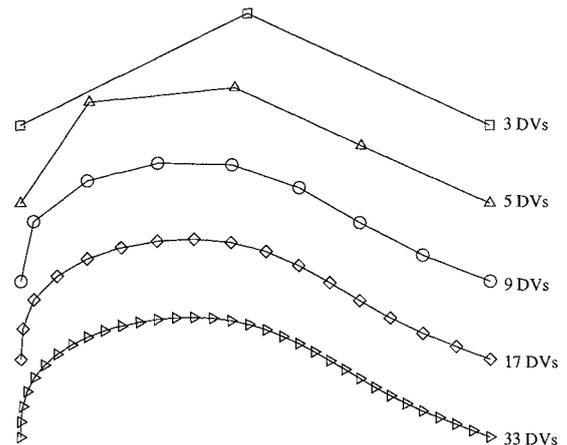


Fig. 17 A sequence of design-variable sets.

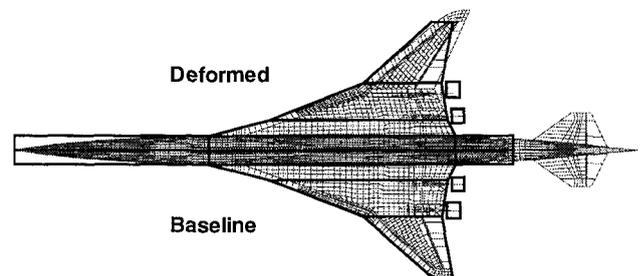


Fig. 18 Baseline and deformed CSM models of an HSCT.

Results and Conclusions

The algorithms presented in this paper have been applied for parameterizing a simple wing, a blended wing body, and several HSCT configurations. Figure 18 shows the baseline and deformed CSM grids for an HSCT. The solid lines represent the initial position of 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 for structured⁶ and unstructured⁸ CFD grids.

An aerodynamic optimization of an ONERA M6 wing was performed⁶ 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 19 shows the design cycle history for both lift and drag. In this optimization, the angle of attack is fixed, and it was found that in order to move away from the current design, the constraint on the lift coefficient had to be relaxed temporarily. This is shown clearly in the figure: 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

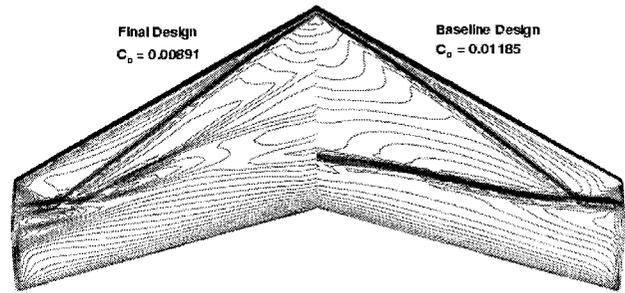
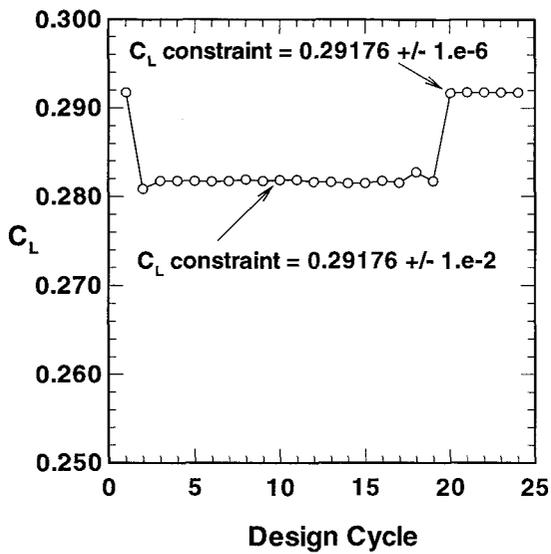


Fig. 20 Comparison of the M6 wing final design and baseline surface pressures.

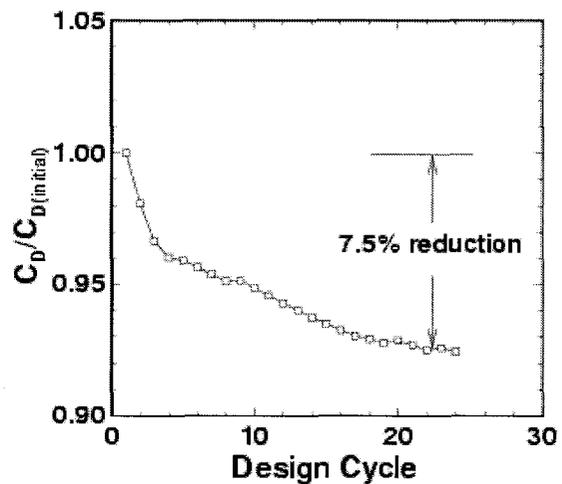
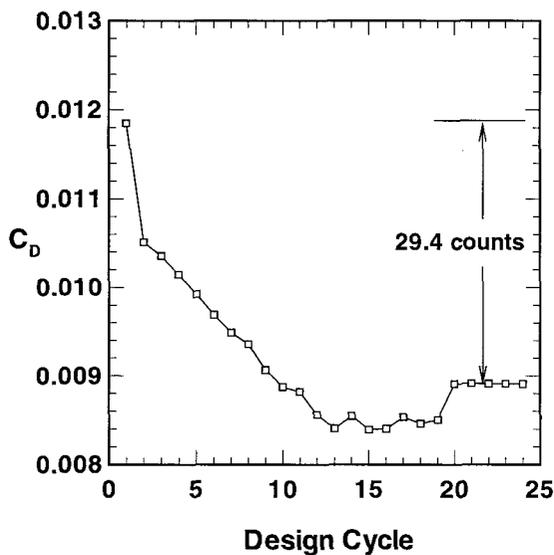


Fig. 21 Design cycle history of HSCT optimization for drag.

Fig. 19 Design cycle history of ONERA M6 wing optimization for coefficient of drag.

29 counts of drag reduction at the baseline lift. Figure 20 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 a HSCT.^{9,10} Figure 21 shows the design cycle history for aircraft drag, as measured relative to the baseline values. The figure 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 little additional reduction in drag would be obtained from additional design cycles. Figure 21 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 rep-

resented. 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 in-board spanwise location. Although not evident in the figure, the wing thickness has been slightly reduced.

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.

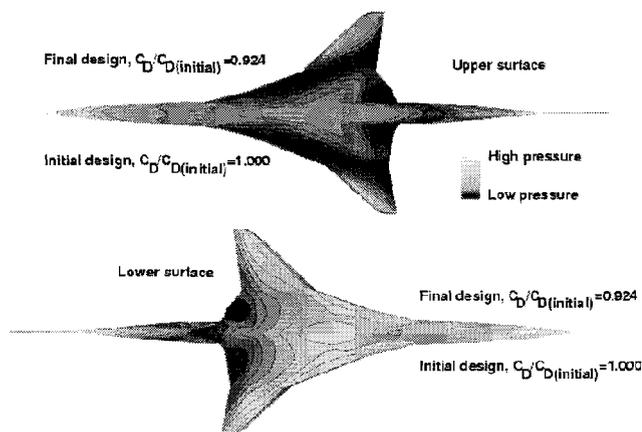


Fig. 22 Comparison of the HSCT final design and baseline surface pressures.

Acknowledgments

The author would like to thank Drs. Perry Newman and James Townsend of Multidisciplinary Optimization Branch at NASA Langley Research Center and Susan Costello of NCI Information Systems for their assistance with this paper. The author would also like to thank Dr. Robert Biedron for providing figures 19 and 20 and Joanne Walsh for providing figures 21 and 22.

References

- 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.
- 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.
- Ding, Y., "Shape Optimization of Structures: A Literature Survey," *Computers & Structures*, Vol. 24, No. 6, 1986, pp. 985-1004.
- Samareh, J. A., "A Survey of Shape Parameterization Techniques," *AIAA/CEAS/ICASE/NASA-LaRC International Forum on Aeroelasticity and Structural Dynamics Conference*, Williamsburg, Virginia, 1999, pp. 333-343.
- Watt, A., and Watt, M., *Advanced Animation and Rendering Techniques*, Addison-Wesley Publishing Company, New York, 1992.
- Biedron, R. T., Samareh, J. A., and Green, L. L., "Parallel Computation of Sensitivity Derivatives With Application to Aerodynamic Optimization of a Wing," *1998 Computational Aerosciences Workshop*, NASA CP-20857, Jan. 1999, pp. 219-224.
- 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 Computational Aerosciences Workshop*, NASA CP-20857, Jan. 1999, pp. 133-134.
- 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 Proceedings*, Sept. 1998, pp. 825-837, also AIAA-98-4809-CP.
- 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," Paper AIAA-2000-0418, Jan. 2000.

- 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," Paper AIAA-2000-0419, Jan. 2000.
- Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407-412.
- 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.
- Hall, V., "Morphing in 2-D and 3-D," *Dr. Dobb's Journal*, Vol. 18, No. 7, 1993, pp. 18-26.
- Barr, A. H., "Global and Local Deformation of Solid Primitives," *Computer Graphics*, Vol. 18, No. 3, 1984, pp. 21-30.
- Sederberg, T. W., and Parry, S. R., "Free-Form Deformation of Solid Geometric Models," *Computer Graphics*, Vol. 20, No. 4, 1986, pp. 151-160.
- Abbott, I. A., and Von Doenhoff, A. E., *Theory of Wing Sections*, Dover Publications, New York, 1959.
- Farin, G., *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, New York, 1990.
- 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.
- Coquillart, S., "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," *SIGGRAPH*, Vol. 24, No. 4, 1990, pp. 187-196.
- Lamoussin, H. J., and Waggenspack, W. N., "NURBS-Based Free-Form Deformation," *IEEE Computer Graphics and Applications*, Vol. 14, No. 6, 1994, pp. 95-108.
- 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, Sept. 1997, pp. 1-9.
- Perry, E., and Balling, R., "A New Morphing Method for Shape Optimization," Paper AIAA-98-2896, June 1998.
- Cook, R. D., Malkus, D. S., and Plesha, M. E., *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, New York, 1989.
- Hirsch, C., *Numerical Computation of Internal and External Flows*, John Wiley & Sons, New York, 1988.