A SURVEY OF SHAPE PARAMETERIZATION TECHNIQUES

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

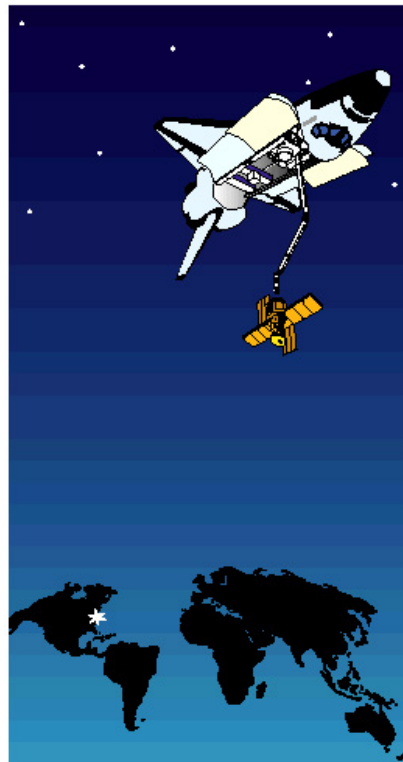# A SURVEY OF SHAPE PARAMETERIZATION TECHNIQUES

Jamshid A. Samareh *

NASA Langley Research Center, Hampton, VA 23681

This paper provides a survey of shape parameterization techniques for multidisciplinary optimization and highlights some emerging ideas. The survey focuses on the suitability of available techniques for complex configurations, with suitability criteria based on the efficiency, effectiveness, ease of implementation, and availability of analytical sensitivities for geometry and grids. The paper also contains a section on field grid regeneration, grid deformation, and sensitivity analysis techniques.

Nomenclature

B	Bernstein polynomial
$\bar{c}$	polynomial coefficients
f	response
J	cell Jacobian
k	spring stiffness
N	B-spline basis function
$\bar{P}$	coordinates of NURBS control point
$\bar{R}$	coordinates of deformed model
$\bar{r}$	coordinates of baseline model
$\bar{U}$	deformation vector
u	independent parameter coordinate
V	baseline cell volume
$\bar{v}$	design variable vector
W	NURBS weights

Subscripts

f	field (volume) grid
g	geometry
i	control point index
j	grid point index
k	grid point index
m	element index
p	degree of Bernstein polynomial and B-spline basis function
s	surface grid
ϵ	small positive number

Superscripts

i	polynomial power
n	number of design variables

*Research Scientist, Multidisciplinary Optimization Branch, AIAA Senior Member.

Introduction

IMAGINE that you have been asked to perform multi-disciplinary shape optimization (MSO) for a complete aircraft model during the preliminary design phase. During this phase, the focus is on the mathematical modeling, with sufficient accuracy, of the outside skin of an aircraft. After this phase, the geometry is frozen, and any change could be costly.

Generally, multidisciplinary design optimization (MDO) should exploit the synergism of the primary, mutually interacting phenomena to improve the design. The MDO applications commonly involve sizing, topology, and shape optimization. Sizing optimization is used to find the optimum cross-sectional area for bars and trusses and thickness for plate and shell elements. Sizing optimization is a matured technology and is available in most commercial computational structural mechanics (CSM) tools. Topology optimization is a technique for determining the optimal material distribution, which could suggest the optimum layout of the structure. Shape optimization finds the optimum shape for a given structural layout. Obviously, the selection of shape parameterization technique has enormous impact on the formulation and implementation of the optimization problem. This paper reviews and evaluates the available shape parameterization techniques.

Over the past several decades, single discipline shape optimization has been successfully applied to two-dimensional and simple three-dimensional configurations.[1,2] In recent years, there has been a growing interest in the application of MSO to complex three-dimensional configurations.[3] The MSO for a complete airplane configuration is a challenging task, especially if the MSO application is based on high-fidelity analysis tools. The analysis models, also referred to as grids or meshes, are based on some or all of the airplane components.

The aerodynamic analysis uses the detailed definition of the skin shape, also referred to as the outer mold
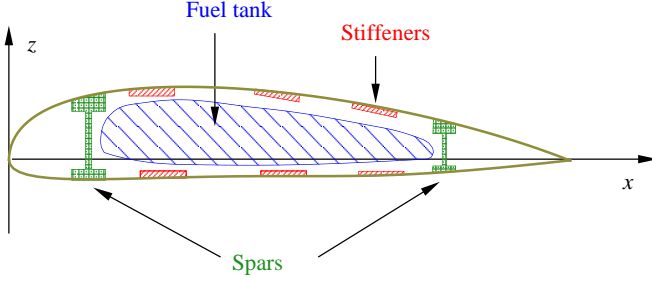
Fig. 1   Internal components of a wing.

line, whereas the CSM tools use all components. Generally, the structural model only requires a relatively coarse grid, but it must handle very complex internal and external geometries. In contrast, the computational fluid dynamics (CFD) field grid is very fine, but it only needs to model the external geometry. The MSO of an airplane must treat not only the external geometry (i.e., wing skin, fuselage, flaps, nacelles, and pylons), but also the internal structural elements such as spars, ribs, and fuel tanks (see Fig. 1). The treatment of internal structural elements is especially important for detailed finite element (FE) analysis.

For a high-fidelity MSO application to be successful, the parameterization model must yield a compact and effective set of design variables so the solution time would be feasible. For more details, readers are referred to an overview paper by this author[4] on geometry modeling and grid generation for design and optimization.

Multidisciplinary Shape Parameterization

The complexity of geometry models is increasing for today's preliminary design applications. It is not unusual for a computer-aided design (CAD) model to use over twenty thousand curves and surfaces to represent an aircraft. This level of complexity underscores the importance of automation. With any multidisciplinary application come the problems of consistent and accurate shape parameterization.

The shape 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. For a multidisciplinary problem, the application must also use a consistent parameterization across all disciplines. An MDO application requires a common geometry data set that can be manipulated and shared among various disciplines.

In addition, an accurate sensitivity derivative analysis is required for gradient-based optimization. The sensitivity derivatives are defined as the partial derivatives of the geometry model or grid-point coordinates with re-

spect to a design variable. The sensitivity derivatives of a response, f, with respect to the design variable vector, $\bar{v}$, can be written as

$$\frac{\partial f}{\partial \bar{v}} = \left[\frac{\partial f}{\partial \bar{R}_f}\right]\left[\frac{\partial \bar{R}_f}{\partial \bar{R}_s}\right]\left[\frac{\partial \bar{R}_s}{\partial \bar{R}_g}\right]\left[\frac{\partial \bar{R}_g}{\partial \bar{v}}\right] \qquad (1)$$

where $\bar{R}_f$ is the field (volume) grid, $\bar{R}_s$ is the surface grid, and $\bar{R}_g$ is the geometry. In some of the CSM literature, the sensitivity derivatives are referred to as the design velocity field.

The first term on the right-hand side of Eq.(1) represents the sensitivity derivatives of the response with respect to the field grid point coordinates. For a detailed discussion, readers are referred to Refs. 1,2,5 for CSM and to Refs. 6–8 for CFD disciplines. Newman et al.[6] have provided an overview of the recent advances in steady aerodynamic shape-design sensitivity derivative analysis and optimization based on advanced CFD. The second term on the right-hand side of Eq.(1) is vector of the field grid-point sensitivity derivatives with respect to the surface grid points. The sensitivity derivative vector must be provided by the field grid generator, but few grid generation tools have the capability to provide the analytical grid-point sensitivity derivatives.[9] The third term on the right-hand side of Eq.(1) denotes the surface grid sensitivity derivatives with respect to the shape design variables, which must be provided by the surface grid generation tools. The fourth term on the right-hand side of Eq.(1) signifies the geometry sensitivity derivatives with respect to the design variable vectors; this must be provided by the geometry construction tools.

Figure 2 shows a high-speed civil transport with seven planform design variables. Figure 3 shows errors involved in using a central-difference approximation for shape sensitivity derivative calculations for the high-speed civil transport shown in Fig. 2. This error behavior is typical of finite-difference approximations to sensitivities. For larger step sizes, the truncation error is predominant, and for smaller step sizes, the round-off error is predominant. There is an optimal step size where the error is minimum. This optimal step size is different for each design variable, and it would also vary for each optimization cycle. As a result, it is difficult to estimate the error involved in finite-difference approximation of sensitivity derivatives. If the source codes are written in FORTRAN or C, and are available, they can be differentiated with automatic tools* such ADIFOR[10] or ADIC.[11]

An important ingredient of shape optimization is the availability of a model parameterized with respect to

---

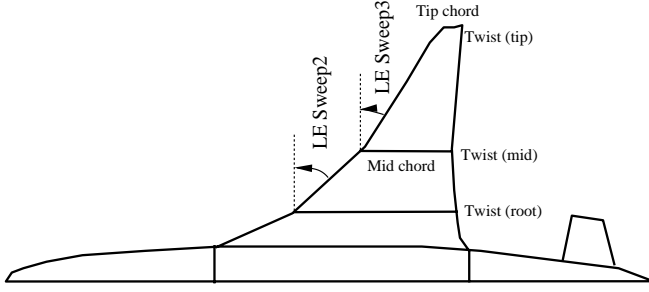*Argonne National Laboratory maintains a www site with information on automatic differentiation tools <http://www-unix.mcs.anl.gov/autodiff>

Fig. 2   Design variables for a high-speed civil transport.

a) Baseline model   b) Deformed model
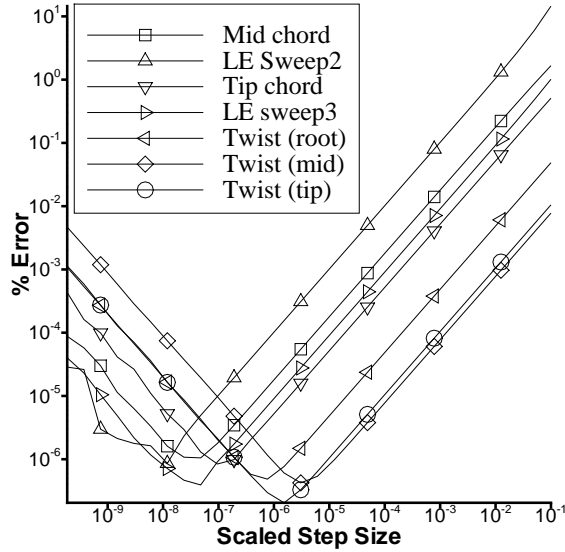
Fig. 4   Domain element.

Fig. 3   Error in using central-difference approximation for shape sensitivity derivatives calculations.

the airplane shape parameters such as planform, twist, shear, camber, and thickness. The parameterization techniques are divided into the following categories: basis vector, domain element, partial differential equation, discrete, polynomial and spline, CAD-based, analytical, free form deformation (FFD), and modified FFD. Readers are referred to reports by Haftka and Grandhi[1] and Ding[2] for surveys of shape optimization and parameterization up to 1981. The present focus is on some recent developments in the area of shape parameterization for complex models and their suitability for MSO applications. The suitability criteria are based on the efficiency, effectiveness, ease of implementation, and availability of analytical sensitivities for geometry and grid models.

Basis Vector Approach

Pickett et al.[12] proposed a technique that combines the second through fourth terms of Eq.(1) into a set of basis vectors. The shape changes can be expressed as
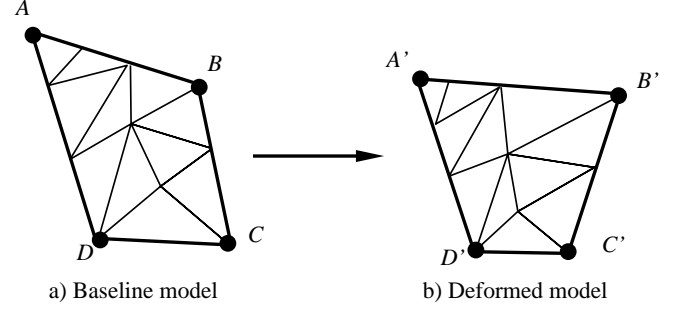
$$\bar{R} = \bar{r} + \sum_i \bar{v}_i \bar{U}_i \qquad (2)$$

where $\bar{R}$ is the design shape, $\bar{r}$ is the baseline shape, $\bar{v}_i$ is the design variable vector, and $\bar{U}_i$ is design perturbation based on several proposed shapes. Assuming that the reduced basis is constant throughout the optimization cycle, this technique is a good approach and is available in most commercial CSM codes.[13–16] However, it is difficult to generate a set of consistent basis vectors for multiple disciplines. As a result, this method can be applied only to problems involving a single discipline with relatively simple geometry changes.

Domain Element Approach

The domain element approach is based on linking a set of grid points to a macro element, domain element, that controls the shape of the model. Figure 4a shows a domain element with four nodes ($A–D$) for the baseline model. As the nodes of the domain element move ($A'–D'$), the grid points belonging to the domain will move as well (see Fig. 4b). 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.[14] The domain element technique is available for shape optimization in some commercial software.[16] This method is useful only for problems with relatively simple geometry changes.

Partial Differential Equation Approach

Bloor and Wilson[17] presented an efficient and compact method for parameterizing the surface geometry of an aircraft. The method views the surface generation as a boundary-value problem and produces surfaces as the solutions to elliptic partial differential equations (PDE). Bloor and Wilson showed that it was possible to represent an aircraft geometry in terms of a small set of design variables. Smith et al.[18] extended the PDE approach to a class of airplane configurations. Included in this definition were surface grids, volume grids, and grid sensitivity derivatives for CFD. The general airplane
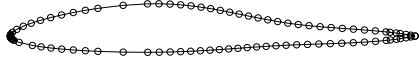
Fig. 5　Airfoil designed by a set of points.

configuration had wing, fuselage, vertical tail, horizontal tails, and canard components. Grid sensitivity was obtained by applying the automatic differentiation tool ADIFOR.[10]

Using the PDE approach to parameterize an existing complex model is time-consuming and costly. Also, because this method can only parameterize the surface geometry, it is not suitable for the MSO applications that must model the internal structural elements such as spars, ribs, and fuel tanks. As a result, this method is suitable for problems involving a single discipline with relatively simple external geometry changes.

Discrete Approach

The discrete approach is based on using the coordinates of the boundary points (see Fig. 5) as design variables (e.g., Refs. 19,20). This approach is easy to implement, and the geometry changes are limited only by the number of design variables. However, it is difficult to maintain a smooth geometry, and the optimization solution may be impractical to manufacture, as pointed out by Braibant and Fleury.[21] To control smoothness, one could use multipoint constraints and dynamic adjustment of lower and upper bounds on the design variables. For a model with a large number of grid points, the number of design variables often becomes very large, which leads to high cost and a difficult optimization problem to solve.

The natural design approach is a variation of the discrete approach that uses a set of fictitious loads as design variables (e.g., Ref. 22). These fictitious loads are applied to the boundary points, and the resulting displacements, or natural shape functions, are added to the baseline grid to obtain a new shape. Consequently, the relationship between changes in design variables and grid-point locations is established through a finite element analysis. Zhang and Belegundu[23] provided a systematic approach for generating the sensitivity derivatives and several criteria to determine their effectiveness. The typical drawback of the natural design variable method is the indirect relationship between design variables and grid-point locations.

For an MDO application, grid requirements are different for each discipline. So, each discipline has a different grid and a different parameterized model. Consequently, using the discrete parameterization approach for an MDO application will result in an inconsistent parameterization.

The most attractive feature of the discrete approach

is the ability to use an existing grid for optimization. The model complexity has little or no bearing on the parameterization process. It is possible to have a strong local control on shape changes by restricting the changes to a small area. When the shape design variables are the grid-point coordinates, the grid sensitivity derivative analysis is trivial to calculate; the third and fourth terms in Eq.(1) can be combined to form an identity matrix.

Polynomial and Spline Approaches

Use of polynomial and spline representations for shape parameterization can obviously reduce the total number of design variables. For example, Fig. 6 shows an airfoil definition with only nine control points. Braibant and Fleury[21] 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. Automatically taken into account are the additional optimization constraints most often needed to avoid unrealistic design when the shape variables are the grid-point coordinates. The analytical sensitivity derivatives with respect to the design variable vector can be computed efficiently and accurately.

For example, a curve can be described as the polynomial

$$\bar{R}_g(u) = \sum_{i=0}^{n-1} \bar{c}_i u^i \qquad (3)$$

where n is the number of design variables, and u is the parameter coordinate along the curve. The $\bar{c}_i$ is a set of coefficient vectors corresponding to three-dimensional coordinates, and the components of these vectors can be used as design variables. The sensitivity derivatives of geometry, $\bar{R}_g$, with respect to $\bar{c}_i$ is u^i. The polynomial representation in Eq.(3) is in the power basis form, and the $\bar{c}_i$ coefficient vectors convey very little geometric insight about the shape. Also, the power basis form is prone to round-off error if there is a large variation in the magnitude of the coefficients. Nevertheless, the polynomial form is a powerful and compact representation for shape optimization of simple curves (e.g., Refs. 24,25).

The Bezier representation is another mathematical form for representing curves and surfaces. For example, a Bezier curve can be described by

$$\bar{R}_g(u) = \sum_{i=1}^{n} \bar{P}_i B_{i,p}(u) \qquad (4)$$

where n is the number of control points (design variables), and the $B_{i,p}(u)$ are degree p Bernstein polynomials. The $\bar{P}_i$ are the control points (forming a control polygon), and they are typically used as design variables.
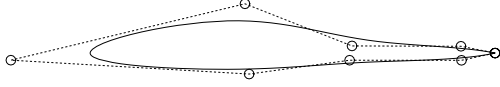
Fig. 6 Airfoil designed by a set of control points.

Readers are referred to Farin[26] for further discussions on the properties of Bezier form. The Bezier form is a far better representation than the power basis, even though mathematically equivalent. The control points are more closely related to the curve position. In fact, the control points approximate the curve. Also, the computation of Bernstein polynomials is a recursive algorithm, *de Casteljau* algorithm,[26] which minimizes the round-off error. The convex hull of the Bezier control polygon contains the curve. This property is very useful, especially in defining the geometric constraints. The first and the last control points are located exactly at the beginning and the end of the curve, respectively. The sensitivity derivative of geometry, $\bar{R}_g$, with respect to $\bar{P}_i$ is $B_{i,p}(u)$, the Bernstein polynomial functions. These functions are independent of the Bezier control points (i.e., design variables); therefore, the sensitivity derivatives stay fixed during the optimization cycles.

The Bezier form is an effective and accurate representation for shape optimization of simple curves (e.g., Ref. 27). However, complex curves require a high-degree Bezier form. As the degree of a Bezier curve increases, so does the round-off error. Also, it is very inefficient to compute a high-degree Bezier curve. To use Bezier representation for a complex curve, one can use several low-degree Bezier segments to cover the entire curve. The resulting composite curve is referred to as a spline or, more accurately, a B-spline. A multisegmented B-spline curve can be described by

$$\bar{R}_g(u) = \sum_{i=1}^{n} \bar{P}_i N_{i,p}(u) \tag{5}$$

where $\bar{P}_i$ are the B-spline control points, p is the degree, and $N_{i,p}(u)$ is the i-th B-spline basis function of degree p. In addition to the desirable properties of the Bezier representation, the low-degree B-spline form can represent complex curves efficiently and accurately. The sensitivity derivatives of geometry, $\bar{R}_g$, with respect to $\bar{P}_i$ is $N_{i,p}(u)$, the B-spline basis function. Similar to a Bezier form, the sensitivity derivatives of a B-spline curve stay fixed during the optimization cycles.

There are some limited applications in the literature that are based on polynomial and spline representations. Cosentino and Holt[28] optimized a transonic wing configuration by using a cubic spline representation for two-dimensional airfoils that define a wing geometry. Then, they used the position of the spline control points—in

particular those points that affect the wing region wetted by supersonic flow—as design variables to be optimized. In a design case study on the Lockheed C-141B aircraft, they reduced the number of design variables from 120 to 12 by using the cubic spline technique. In recent years, Schramm and Pilkey[29] used a B-spline representation to perform structural shape optimization for the torsion problem with direct integration and B-splines. Similarly, Anderson and Venkatakrishnan[30] used B-splines for aerodynamics design optimization with an unstructured grid CFD code.

The only drawback of the regular B-spline representation is its inability to represent implicit conic sections accurately. However, a special form of B-spline, nonuniform rational B-spline (NURBS), can represent most parametric and implicit curves and surfaces without loss of accuracy.[26] NURBS can represent quadric primitives (e.g., cylinders, cones) as well as free-form geometry.[26] There are some implicit surfaces (e.g., helix and helicoidal)[31] that cannot be directly converted to NURBS, but these surfaces are not common in most aerospace applications. A NURBS curve is defined as

$$\bar{R}(u) = \frac{\sum_{i=1}^{n} N_{i,p}(u) W_i \bar{P}_i}{\sum_{i=1}^{n} N_{i,p}(u) W_i} \tag{6}$$

where the $\bar{P}_i$ are the control points, W_i are the weights, and the $N_{i,p}$ are degree p B-spline basis functions. Similar to the Bezier form, the sensitivity derivatives of a NURBS with respect to the control points are fixed during the optimization cycles. However, if the weights are selected as design variables, the sensitivity derivatives will be a function of the weight design variables. Schramm et al.[32] have successfully used the two-dimensional NURBS representations for shape optimization.

Despite recent progress, it is still difficult to parameterize and construct complex, three-dimensional models based solely on polynomial and spline representations. Complex shapes require a large number of control points, and optimization is prone to creating irregular[21] or wavy[33] geometry. Nevertheless, these techniques are well suited for two-dimensional or simple three-dimensional models.

CAD-Based Approach

Use of CAD systems for geometry modeling could potentially save development time for an MDO application. For a more detailed account of the role of CAD in MDO, readers are referred to Ref. 4. Most solid modeling CAD systems use either a boundary representation (B-Rep) or a constructive solid geometry method to represent a

physical, solid object.[34] Based on a complete mathematical definition of a solid, it is possible to create a complete geometry that is suitable for detailed CFD and CSM codes.

Feature-based solid modeling (FBSM) CAD systems[35] are capable of creating dimension-driven objects. These systems use Boolean operations such as intersection and union of simple features. Examples of simple features include holes, slots (or cuts), bosses (or protrusions), fillets, chamfers, sweep, and shell. Today's CAD systems allow designers to work in a three-dimensional space while using topologically complete geometry (solid models) that can be modified by altering the dimensions of the features from which it was created. The most important capability of FBSM is the ability to capture the design intent. The FBSM tools have made design modification much easier and faster. The developers of FBSM CAD systems have put the "design" back in CAD. Because FBSM CAD tools enable today's design engineers to create a new, complete, and parametric model for a configuration, these tools are being incorporated into the design environment.

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.[36] discussed issues involved in using a CAD system for an MDO application. They identified the analytical sensitivity derivative calculations as a one of the important integration issues. Blair and Reich[37] presented a vision to integrate an FBSM CAD system with full associativity into a virtual design environment. Within such an environment, however, calculations of the analytical sensitivity derivatives of geometry with respect to the design variables could prove to be difficult.

It is possible to relate some design variables to the NURBS control points.[38] Then the analytical sensitivity derivatives can be calculated outside the CAD system. For some limited cases, the analytical shape sensitivity derivatives can be calculated based on a CAD model;[38] 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. He et al.[39] presented a procedure for integrating CAD and CAE systems to support geometry- and detailed-analysis-based optimization. The sensitivity derivatives were calculated by a finite-difference approximation.

So, it is not a trivial matter to incorporate FBSM CAD systems into a design optimization, and it is even more difficult to use them for an MDO application. Also, it is still a challenging task to parameterize an existing model that is not parametric.

Analytical Approach

Hicks and Henne[40] introduced a compact formulation for parameterization of airfoil sections. The formulation was 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 the first computation gives the baseline geometry. The shape functions are smooth functions based on a set of previous airfoil designs. Elliott and Peraire[25] and Hager et al.[41] used a formulation similar to that of Hicks and Henne, but a different set of shape functions. This method is very effective for wing parameterization, but it is difficult to generalize it for a complex geometry.

Free Form Deformation Approach

The field of soft object animation (SOA) in computer graphics[42] provides algorithms for morphing images[43] and deforming models.[44, 45] These algorithms are powerful tools for modifying shapes: they use a high-level shape deformation, as opposed to manipulation of lower level geometric entities. 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 the grid-point coordinates of an analysis model to a number of design variables. Consequently, the SOA algorithms can serve as the basis for an efficient shape parameterization technique.

Barr[44] 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 grid points of a polygon model or the control points of a parametric curve and surface. Sederberg and Parry[45] presented another approach for deformation, based on the 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. Coquillart[46] extended a Bezier parallelepiped to a nonparallelepiped cubic Bezier volume.

Lamousin and Waggenspack[47] modified FFD to in-

clude NURBS definition and multiple blocks to model complex shapes. The modified technique has been used for design and optimization by Yeh and Vance[48] and Perry and Balling.[49] Yeh and Vance[48] developed an application based on NURBS where the user can change the shape of a virtual object and examine the effect the shape change has on the displacement of the structural deformation and stress distribution throughout the object. Perry et al.[50] successfully used FFD algorithm for the optimization of an automobile air conditioning duct system.

Hsu et al.[51] presented a method to directly manipulate the object, which creates a more intuitive and transparent environment for FFD. Borrel and Rappoport[52] presented a simple, constrained deformation that allows the user to define a set of constraint points, giving a desired displacement and radius of influence for each. Each constraint point determines a local B-spline basis function centered at the constraint point, falling to zero for points beyond the radius. This technique directly influences the final shape of the deformed object.

Multidisciplinary Aero/Struc Shape Optimization Using Deformation (MASSOUD) Approach

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 into the product development cycle times, the MSO must rely on the parameterization of the analysis grids, for which the FFD algorithm is ideal. 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 author developed a set of modifications to the original SOA algorithms to alleviate this and other drawbacks; the modified algorithm set is referred to as MASSOUD.[53]

MASSOUD is a novel parameterization approach for complex shapes suitable for a multidisciplinary design optimization application. The approach consists of three basic concepts: 1) parameterizing the shape perturbations rather than the geometry itself, 2) utilizing SOA algorithms used in computer graphics, and 3) relating the deformation to aerodynamics shape design variables such as thickness, camber, twist, shear, and planform.

The MASSOUD formulation is independent of grid topology, and that makes it suitable for a variety of analysis codes such as CFD and CSM. The analytical sensitivity derivatives are available 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 analysis tools (e.g., nonlinear CFD and detailed FE modeling). The report by this author[53] contains the implementation

details of parameterizing for planform, twist, dihedral, thickness, and camber. The results presented were for a multidisciplinary optimization application consisting of nonlinear CFD, detailed CSM, performance, and a simple propulsion module.

Typically, the optimization starts with an existing wing design, and the goal is to improve the wing performance by using numerical optimization. The geometry changes (perturbations) between the initial and optimized wings are very small,[28,40] but the difference in wing performance can be substantial. By parameterizing the shape perturbations instead of the shape itself, MASSOUD reduces the number of shape design variables. Throughout the optimization cycles, the surface grid can be updated as

$$\bar{R}(\bar{v}) = \bar{r} + \bar{U}(\bar{v}) \qquad (7)$$

where $\bar{r}$ is the baseline grid, $\bar{R}$ is the deformed (perturbed) grid, $\bar{U}$ is the change (perturbation), and $\bar{v}$ is the design variable vector. It takes far fewer design variables to parameterize the shape perturbation $\bar{U}$ than to parameterize $\bar{r}$ itself.

The MASSOUD algorithm has been used for parameterizing a simple wing, a blended wing body, and several high-speed civil transport configurations. The algorithm has been successfully implemented for aerodynamic shape optimization with analytical sensitivity derivatives with structured grid[54] and unstructured grid CFD[55] codes. In addition to ease of use and implementation, MASSOUD has the following benefits: 1) parameterization is consistent, 2) the analytical sensitivity derivatives are available, 3) complex existing grids can be parameterized, 4) there is a strong local control, 5) smoothness can be controlled, and 6) few design variables are required.

Summary of Multidisciplinary Shape Parameterization

Figure 7 presents a summary and rating of the nine approaches surveyed in this paper. There are three ratings: 1) good (thumb-up), 2) fair (neutral), and 3) poor (thumb-down). The summary uses ten criteria that are important for multidisciplinary applications of complex, three-dimensional configurations.

- Consistent: Is the parameterization consistent across multiple disciplines?

- Airplane shape design variables: Are the design variables directly related to the airplane shape design variables such as camber, thickness, twist, shear, and planform?

- Compact: Does the parameterization provide a compact set of design variables?

Fig. 7   Comparisons of parameterization approaches.

- Smooth: Does the shape perturbation maintain a smooth geometry?

- Local control: Is there any local control on shape changes?

- Analytical sensitivity: Is it feasible to calculate the sensitivity analytically?

- Grid deformation: Does the parameterization allow the grid to be deformed?

- Setup time: Can a shape optimization application be set up quickly?

- Existing grid: Does the parameterization allow the existing grid to be reused?

- CAD: Is there a direct connection to the CAD system?

Field Grid Movement and Sensitivity Derivatives

The parameterization techniques are used to move the grid points and geometry of the design surfaces. The next step is to propagate the changes and sensitivity into the field. The field sensitivity derivatives can either be calculated analytically or approximated with finite differences. As discussed before, there is some error involved in the finite-difference approximation that could slow the optimization.

For a CFD calculation, the field (volume) grid may contain several million grid points. There are two basic techniques to propagate the surface grid-point movements into the field: 1) grid regeneration and 2) grid deformation.

Structured Field Grid Movement

Most structured grid regeneration and deformation techniques are based on transfinite interpolation (TFI). Gaitonde and Fiddes[56] used a regenerating grid technique based on using TFI with exponential blending functions. The choice of blending functions has a considerable influence on the quality and robustness of the field grid. Soni[57] proposed a set of blending functions based on arc length that is extremely effective and robust for grid regeneration and deformation. His algorithm has been incorporated in most commercial structured grid generation packages.

Jones and Samareh[9] presented an algorithm for grid regeneration and deformation based on Soni's blending functions, and they also provided analytical sensitivity derivatives by using an automatic differentiation tool, "ADIC".[11] The method is suitable for a general, multiblock, three-dimensional volume grid deformation. The idea of volume grid deformation was also used by Hartwich and Agrawal.[58] They introduced two new techniques: 1) the use of the "slave/master" concept to semiautomate the process and 2) the use of a Gaussian distribution function to preserve the integrity of grids in the presence of multiple body surfaces. Reuther et al.[8] used a modified TFI approach with blending functions based on arc length, and they used finite-difference approximation to compute the sensitivity derivatives for the field grid.

Leatham and Chappell[59] used the Laplacian technique, commonly used for unstructured grid deformation, for moving structured grids. They have been successful in deforming structured grids with this technique.

Unstructured Field Grid Movement

For unstructured grids with large geometrical changes, Botkin[60] proposed to regenerate a completely new grid at the beginning of each optimization cycle. However, for gradient calculations many small changes must be made, and it would be too costly to regenerate the grid for each design variable perturbation. Botkin has introduced a local regridding procedure that operates only on the specific edges and faces associated with the design variables being perturbed. Similarly, Kodiyalam et al.[61] used a grid regeneration technique based on the assumption that the solid model topology stays fixed for small perturbations. The solid model topology contains

the number of grid points, edges, and faces. Any change in the topology will cause the model regeneration to fail. To avoid such as failure, a set of constraints must be satisfied among design variables, in addition to constraints on their bounds.

For a dynamic aeroelastic case with unstructured grids, Batina[62] presented a grid deformation algorithm that models grid edges with springs. The spring stiffness for a given edge j–k is taken to be inversely proportional to the element edge length as

$$k_m = \frac{1}{|\bar{r}_j - \bar{r}_k|}.$$ (8)

The grid movement is computed through predictor and corrector steps. The predictor step is based on an existing solution from the previous cycle, and the corrector step uses several Jacobi iterations of the static equilibrium equations by using

$$\bar{U}^{n+1} = \frac{\sum k_m \bar{U}^n}{\sum k_m}$$ (9)

where the sum is over all edges of the elements. This is similar to a Laplace operator, which has a diffusive behavior. In contrast to its use for dynamic aeroelasticity, the previous optimization cycle may not provide a good initial guess to be used by the corrector step.

Zhang and Belegundu[23] proposed a similar algorithm to handle large grid movement. The equation for grid update is similar to Batina's[62] approach,

$$\bar{R}^{\text{new}} = \frac{\sum k_m \bar{R}^{\text{old}}}{\sum k_m}, \quad \text{where} \quad k_m = \frac{8|J|}{V},$$ (10)

J is the cell Jacobian defined within cell parametric coordinates, and V is the cell volume.

Crumpton and Giles[63] found the spring analogy to be inadequate and ineffective for large grid perturbations. They proposed a technique based on using the heat transfer equation

$$\nabla.\{k_m \nabla(\bar{U})\} = 0 \quad \text{where} \quad k_m = \frac{1}{\max(V, \epsilon)},$$ (11)

V is the cell volume, and ϵ is a small positive number needed to avoid a division by zero. This technique is similar to the spring analogy,[62] except that it uses the cell volume for k_m. The coefficient k_m is relatively large for small cells. Therefore these small cells, which are usually near the surface of the body, tend to undergo rigid body motion. This rigid body movement avoids rapid variations in $\bar{U}$, thus eliminating the possibility of small cells

having very large changes in volume, which could lead to negative cell volumes. Crumpton and Giles[63] used an underrelaxed Jacobi iteration, with the nonlinear k_m evaluated at the previous iteration.

Summary

The results of this study are summarized in Fig. 7. Traditional shape parameterization techniques are not suitable for application to multidisciplinary shape optimization for complex, three-dimensional configurations. At first look the CAD approach appears to be ideal, but there are some unresolved issues, such as analytical sensitivity, that require more research. In the interim, the MASSOUD approach will be useful. Ideally, the CAD and MASSOUD approaches can be combined to form a powerful parameterization tool for multidisciplinary shape optimization application. This combined approach will 1) be automated, 2) provide consistent geometry across all disciplines, 3) provide analytical sensitivity derivatives, 4) fit into the product development cycle times, and 5) have a direct connection to the CAD systems used for design.

Acknowledgments

References

[1]Haftka, R. T. and Grandhi, R. V., "Structural Shape Optimization—A Survey," *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, 1986, pp. 91–106.

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

[3]"Special Issue: Multidisciplinary Design Optimization," *Journal of Aircraft*, edited by E. Livne, Vol. 36, Jan. 1999.

[4]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.

[5]Yang, R. J. and Botkin, M. E., "Accuracy of the Domain Method for the Material Derivative Approach to Shape Design Sensitivities," *AIAA Journal*, Vol. 25, No. 12, 1987, pp. 1606–1610.

[6]Newman, J. C., Taylor, A. C., Barnwell, R. W., Newman, P. A., and Hou, G. J., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 87–96.

[7]Jameson, A., "Essential Elements of Computational Algorithms for Aerodynamic Analysis and Design," NASA CR-97-206268, Dec. 1997.

[8]Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 51–60.

[9]Jones, W. T. and Samareh, J. A., "A Grid Generation System for Multidisciplinary Design Optimization," Paper 95–1689, AIAA, Jun. 1995.

[10]Bischof, C., Carle, A., Khademi, P., and Mauer, A., "AD-IFOR 2.0: Automatic Differentiation of Fortran 77 Program," *IEEE Computational Science & Engineering*, Vol. 3, No. 3, 1996, pp. 18–32.

[11]Bischop, C., Roh, L., and Mauer-Oats, A., "ADIC: An Extensible Automatic Differentiation Tool for ANSI-C," *Software: Practice and Experience*, Vol. 27, Dec. 1997, pp. 1427–1456.

[12]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. 498–494.

[13]Kodiyalam, S., Vanderplaats, G. N., and Miura, H., "Structural Shape Optimization With MSC/NASTRAN," *Computers & Structures*, Vol. 40, No. 4, 1991, pp. 821–829.

[14]Leiva, J. P. and Watson, B. C., "Automatic Generation of Basis Vectors for Shape Optimization in the GENESIS Program," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference Proceedings*, AIAA, Sep. 1998, pp. 1115–1122, also AIAA-98–4852-CP.

[15]Kilroy, K., editor, *NASTRAN Software User's Manual*, MSC, CA, 1998.

[16]Anonymous, editor, *GENESIS Structural Optimization Software*, VMA, CO, 1997.

[17]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.

[18]Smith, R. E., Bloor, M. I. G., Wilson, M. J., and Thomas, A. T., "Rapid Airplane Parametric Input Design (RAPID)," *AIAA 12th Computational Fluid Dynamics Conference*, AIAA, Jun. 1995, pp. 452–462, also AIAA–95–1687-CP.

[19]Campbell, R. L., "An Approach to Constrained Aerodynamic Design With Application to Airfoils," Report NASA-TP-3260, Nov. 1992.

[20]Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum Aerodynamics Design Using the Navier-Stokes Equations," Paper 97–0101, AIAA, 1997.

[21]Braibant, V. and Fleury, C., "Shape Optimal Design Using B-Splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, Aug. 1984, pp. 247–267.

[22]Belegundu, A. D. and Rajan, S. D., "A Shape Optimization Approach Based on Natural Design Variables and Shape Functions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 66, Jan. 1988, pp. 87–106.

[23]Zhang, S. and Belegundu, A. D., "A Systematic Approach for Generating Velocity Fields in Shape Optimization," *Structural Optimization*, Vol. 5, No. 1–2, 1993, pp. 84–94.

[24]Taylor, C. A., Hou, G. J., and Korivi, V. M., "Sensitivity Analysis, Approximate Analysis, and Design Optimization for Internal and External Flows," Paper 91–3083, AIAA, Sep. 1991.

[25]Elliott, J. and Peraire, J., "Practical Three-Dimensional Aerodynamic Design and Optimization Using Unstructured Meshes," *AIAA Journal*, Vol. 35, No. 9, 1997, pp. 1479–1486.

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

[27]Baysal, O. and Ghayour, K., "Continuous Adjoint Sensitivities for General Cost Functions on Unstructured Meshes in Aerodynamic Shape Optimization," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference Proceedings*, Sep. 1998, pp. 1483–1491, also AIAA–98–4904-CP.

[28]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.

[29]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, Aug. 1993, pp. 251–268.

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

[31]Letcher, J. S. and Shook, M., "NURBS Considered Harmful for Gridding (Alternative Offered)," *4th International Meshing Roundtable*, Sandia National Laboratories, Oct. 1995, pp. 253–264.

[32]Schramm, U., Pilkey, W. D., DeVries, R. I., and Zebrowski, M. P., "Shape Design for Thin-Walled Beam Cross Sections Using Rational B-Splines," *AIAA Journal*, Vol. 33, No. 11, 1995, pp. 2205–2211.

[33]Jameson, A. and Reuther, J., "A Comparison of Design Variables for Control Theory Based Airfoil Optimization," *6th International Symposium on Computational Fluid Dynamics*, Vol. 4, Sep. 1995, pp. 101–107.

[34]LaCourse, D. E., *Handbook of Solid Modeling*, McGraw-Hill, New York, 1995.

[35]Shah, J. J. and Mantyla, M., *Parametric and Feature-Based CAD/CAM*, John Wiley & Sons, New York, 1995.

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

[37]Blair, M. and Reich, G., "A Demonstration of CAD/CAM/CAE in a Fully Associative Aerospace Design Environment," Paper 96–1630, AIAA, Apr. 1996.

[38]Hardee, E., Chang, K.-H., Choi, K. K., Yu, X., and Grindeanu, I., "A CAD-Based Design Sensitivity Analysis and Optimization for Structural Shape Optimization Design Applications," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference Proceedings*, Sep. 1996, pp. 77–87, also AIAA–96–3990–CP.

[39]He, B., Rohl, P. J., Irani, R. K., Thamboo, S. V., and Srivatsa, S. K., "CAD and CAE Integration With Application to the Forging Shape Optimization of Turbine Disks," Paper 98–2032, AIAA, Apr. 1998.

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

[41]Hager, J. O., Eyi, S., and Lee, K. D., "A Multi-Point Optimization for Transonic Airfoil Design," Paper 92–4681-CP, AIAA, Sep. 1992.

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

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

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

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

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

[47]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.

[48]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.

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

[50]Perry, E., Balling, R., and Landon, M., "A New Morphing Method for Shape Optimization," Paper 98–4907, AIAA, Sep. 1998.

[51] Hsu, W. M., Hughes, J. F., and Kaufman, H., "Direct Manipulation of Free-Form Deformation," *Computer Graphics*, Vol. 26, No. 2, 1992, pp. 177–184.

[52] Borrel, P. and Rappoport, A., "Simple Constrained Deformations for Geometric Modeling and Interactive Design," *ACM Transactions on Graphics*, Vol. 13, No. 2, 1994, pp. 137–155.

[53] Samareh, J. A., "A Novel Shape Parameterization Approach," Tech. Rep. NASA–TM–1999-209116, Mar. 1999.

[54] 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.

[55] 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.

[56] Gaitonde, A. L. and Fiddes, S. P., "A Three-Dimensional Moving Mesh Method for the Calculation of Unsteady Transonic Flows," *Aeronautical Journal*, Vol. 99, No. 984, 1995, pp. 150–160.

[57] Soni, B. K., "Two- and Three-Dimensional Grid Generation for Internal Flow Applications," Paper 85–1526, AIAA, 1985.

[58] Hartwich, P. M. and Agrawal, S., "Method for Perturbing Multiblock Patched Grids in Aeroelastic and Design Optimization Applications," Paper 97–2038, AIAA, 1997.

[59] Leatham, M. and Chappell, J. A., "On the Rapid Regeneration of Hybrid Grids Due to Design Driven Geometry Perturbation," *The Sixth International Conference on Numerical Grid Generation in Computational Field Simulation*, Mississippi State, MS, 1998, pp. 533–542.

[60] Botkin, M. E., "Three-Dimensional Shape Optimization Using Fully Automatic Mesh Generation," *AIAA Journal*, Vol. 30, No. 5, 1992, pp. 1932–1934.

[61] Kodiyalam, S., Kumar, V., and Finnigan, P., "Constructive Solid Geometry Approach to Three-Dimensional Structural Shape Optimization," *AIAA Journal*, Vol. 30, No. 5, 1992, pp. 1408–1415.

[62] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," Paper 89–0115-CP, AIAA, Jan. 1989.

[63] Crumpton, P. I. and Giles, M. B., "Implicit Time-Accurate Solutions on Unstructured Dynamic Grids," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 11, 1997, pp. 1285–1300.