JOURNAL OF AIRCRAFT
Vol. 36, No. 1, January–February 1999

Status and Future of Geometry Modeling and Grid Generation for Design and Optimization

Jamshid A. Samareh*
NASA Langley Research Center, Hampton, Virginia 23681-2199

Geometry modeling and grid generation (GMGG) have played, and will continue to play, an important role in computational aerosciences. During the last two decades, tremendous progress has occurred in GMGG; however, GMGG is still the biggest bottleneck to routine applications for complicated computational fluid dynamics and computational structures mechanics models for analysis, design, and optimization. We are still far from incorporating GMGG tools in a design and optimization environment for complicated configurations. It is still a challenging task to parameterize an existing model in today's computer-aided design systems, and the models created are not always good enough for automatic grid generation tools. Despite many advances in grid generation, the process is still the most labor-intensive and time-consuming part of the computational aerosciences for analysis, design, and optimization. In an ideal design environment, a design engineer would use a parametric model to evaluate alternative designs effortlessly and optimize an existing design for a new set of design objectives and constraints. For this ideal environment to be realized, the GMGG tools must have the following characteristics: 1) be automated, 2) provide consistent geometry across all disciplines, 3) be parametric, 4) provide sensitivity derivatives, and 5) fit in the product development cycle times. This paper reviews the status of GMGG for analysis, design, and optimization processes, and it focuses on some emerging ideas that will advance the GMGG toward the ideal design environment.

I. Introduction

IN 1975, Chapman[1] made the following prediction: "To displace wind tunnels as the principal source of flow simulations for aircraft design, computers must reach about ten thousand times the speed of the Illiac IV." By some accounts, we have already reached this goal with today's supercomputers. But still, the wind tunnels play a major role in aircraft design, which may require over 30,000 h of wind-tunnel testing.[2] The airplane design process resembles a jigsaw puzzle, requiring multidisciplinary analysis (MDA) and multidisciplinary design optimization (MDO). Geometry modeling and grid generation (GMGG) has an important role in both areas. Complexity of the geometry models is increasing; in today's preliminary design environment, it is not unusual for a computer-aided design (CAD) model to use over 20,000 curves and surfaces to represent an aircraft. This level of complexity underlines the importance of automation.

There are several critical capabilities and processes that need to be available to streamline and automate the MDA and MDO processes. First, the CAD systems must 1) be design oriented, 2) provide easy and rapid model parameterization, and 3) be able to create a complete and accurate CAD model. Second, the CAD model must be transferred to grid generator automatically and accurately. Third, the grid generators must be robust and fully automatic (push-button). Fourth, for gradient-based optimization, the process of grid sensitivity computation with respect to design variables must be easy and accurate. Fifth, MDA and MDO processes must have tools to handle multidisciplinary interactions that require a consistent CAD model for all disciplines.

In MDA and MDO environments, geometry is a common data set that must be manipulated and shared among various disciplines. In traditional design processes, MDA and MDO are performed in an ad hoc manner, with data thrown over the wall from one discipline to another with no considerations for consistency and accuracy. This cultural habit not only affects the consistency and accuracy of the processes, it also increases the design cycle time and cost. Robust and automated GMGG tools could reduce the design cycle time and cost.

There is a large volume of published research in GMGG areas; however, there are a few robust commercial GMGG tools that are ready for incorporation into the MDA and MDO processes. It takes many years to implement a published research into a robust tool. For example, the research in solid modeling (SM) became visible in the mid-1960s, and by the mid-1970s, the first generation of experimental systems had appeared[3]; these systems were based on simple, analytical solids. After decades of development, these commercial solid modelers can handle relatively complex models; however, they are not yet robust. Similarly, research in feature-based solid modeling (FBSM) technology has been conducted since the 1980s, but the FBSM has only become available in commercial CAD systems within the past five years. Also, it took more than a decade to implement the automated algorithm for the computational structures mechanism (CSM) tetrahedral grid generation[4] within the commercial computer-aided engineering (CAE) tools.

An ignored consideration in most existing GMGG tools is the sensitivity analysis that is required for the gradient-based optimization. Sensitivity is defined as the partial derivative of the geometry model or grid point with respect to a design variable. They can be calculated either analytically or by finite differences. This paper will review the essential elements of GMGG for design and optimization. These elements are CAD, SM, FBSM, standards for geometry exchange, grid generation, and geometry parameterization.

II. CAD Systems

Use of CAD systems for geometry modeling potentially could save development time in MDA and MDO environ-

Received Aug. 5, 1998; revision received Oct. 1, 1998; accepted for publication Oct. 26, 1998.

*Research Scientist, Multidisciplinary Optimization Branch, MS 159. E-mail: j.a.samareh@larc.nasa.gov. Senior Member AIAA.

ments. In the last decade, CAD systems have gone through a series of revolutionary changes; for a more detailed account of these changes, readers are referred to the CAD/CAM handbook by Machover.[5] Daratech, Inc.[6] has provided a comprehensive product, vendor, and industry information for more than 500 commercial CAD/CAM/CAE tools. The CAD systems have evolved from a two-dimensional modeling paradigm to a three-dimensional, solid, parametric, and feature-based modeling paradigm. Among today's major CAD systems, sets of functionalities are similar, but sets of flavors are different. As a result, the selection of a CAD system is more a business decision than a technical one. Three major U.S. car companies demonstrated this in that each company selected a single, but different, CAD system for the entire company.

CAD tools have arrived at this present state through three major advances over the last several decades: the incorporation of 1) nonuniform rational B-splines (NURBS), 2) SM, and 3) FBSM. In a traditional CAD system, the geometry is represented as one of many possible mathematical forms, such as Bezier, Coons patches, B-spline curves, and surfaces. However, one can use NURBS equations to represent most spline and implicit curves and surfaces without loss of accuracy.[7] NURBS can represent quadric primitives, e.g., cylinders, and cones, as well as free-form geometry. Although some surfaces, e.g., helix and helicoidal,[8] cannot be directly converted to a NURBS representation, these surfaces are not common in most aerosciences applications. The SM and FBSM systems are discussed in the next two subsections.

A. Solid Modeling

Most SM CAD systems use either a boundary representation (B-Rep) or constructive solid geometry (CSG) method to represent a physical solid object.[9] The B-Rep and CSG representations provide a complete mathematical definition of a solid object. In contrast to traditional surface modeling software, SM software have automated the process of creating solid model topology. Users do not need to trim surfaces nor keep track of relevant parts. SM CAD systems keep track of surfaces, intersection curves, and appropriate trim sections. They also keep track of the space that lies outside and inside the closed volume of the part, so that the described shapes can unambiguously be physically realized as solids. Most SM software hide the tedious topology information from users. This approach enables designers to create and modify shapes much faster than is possible with explicit surface modeling software. SM helps to avoid design errors, and it allows designers to better understand how their products will look and function before physical models are made.

Based on a complete mathematical definition of a solid, it is possible to 1) create a complete geometry that is suitable for detailed computational fluid dynamics (CFD) and CSM analyses; 2) clearly define mating conditions between parts; 3) detect interference automatically; 4) create a computer model for rapid prototyping, e.g., through stereolithography; and 5) allow reuse of solids in design. SM technology has a great potential for automating the GMGG process, but it is not yet mature. Building accurate, complicated geometry is still the Achilles heel of SM systems. Often, designers believe that their models are complete and accurate, but unseen imperfections cause problems in applications such as grid generation, data exchange, numerical control programming, and rapid prototyping. The possible imperfections are 1) free edges, 2) bad loops (inconsistent face or surface normals), 3) unacceptable vertex-edge gaps, 4) unacceptable edge-face gaps, 5) unacceptable loop closure gaps, 6) minute edges, 7) sliver faces, and 8) transition cracks. The first two imperfections are topology errors. Free edge is an edge of a face that is not shared by any other face. Bad loop occurs when the edge of a face has a wrong orientation; as a result, the face normal points in the wrong direction. Another source of error is inaccuracy in computing deviations allowed among different topological entities, such as faces, intersection curves, and vertices.[10] For example, there is no precise solution for calculating the curve of intersection between two arbitrary B-spline surfaces. Consequently, SM software must use an approximate intersection curve, which does not lie on either surface. These deviations are usually so small that they cannot be detected by rendering the solid model. Yet, in the presence of these deviations, automatic grid generation tools often fail. Also, the data translation failure is mistakenly blamed on the data exchange standards, e.g., initial graphics exchange specifications (IGES) and standard for the exchange of product model data (STEP). In reality, lack of a consistent tolerance between sending and receiving systems is the source of the problem. To avoid this problem, the CAD systems must store an intersection tolerance with each entity that defines the solid. Few CAD systems follow this approach.

This problem can be entirely avoided for a simple geometric design by using simple analytical surfaces, e.g., conics, which have exact analytical intersection curves. However, using simple analytical surfaces is not possible for an aircraft design process that relies on complex free-form surfaces.

With a complete and accurate solid model, the grid generation software may still fail. The problem is usually the sliver faces that result from patching between larger surfaces in a model. To create almost equilateral triangles, automatic grid generation tools will create an unnecessarily fine grid near these sliver faces. The resulting analysis will require a large amount of computer resources, and the analysis result will not be accurate because of excessive grid skewness.

These errors could prevent the SM technology from being used in an automated GMGG environment. Some CAD systems are finding solutions in using tolerance modeling and healing to bridge precision issues. Tolerance modeling allows the receiving system to relax its default precision requirements, but these exceptions may not be supported by all integrated CAD and CAE applications. Healing software runs the CAD model through automatic cleaning or tightening algorithms, which may make adjustments that may be unacceptable to the designer. Some SM software allow users to control tolerances, and these software can be used to correct accuracy problems. However, selecting extremely low tolerances may prohibit models from regenerating. Cleaning up these anomalies impedes the automation of grid generation and can potentially add 50% to the time it takes to go from a CAD model to a CFD or CSM grid.

For a detailed airplane design, working with a solid model requires at least an order of magnitude more computer resources than working with a surface model. For example, aircraft designers will find it difficult locating resources to create and assemble the large number of components for a wing or fuselage using current SM tools. As a result, the airplane cannot be modeled completely with current available computer hardware and SM CAD systems. Despite these problems, SM software produce much higher quality data than a user can create with a traditional surface modeling software.

Another basic problem with the solid model representation is that the design intent is not captured; the final design is not made up of features that capture the design intent. The design process is bottom-up, and the design changes are very time consuming.

B. Feature-Based Solid Modeling

Adding features to SM CAD has resolved the capturing design intent problem. Features are dimension-driven objects that are the basis of the FBSM construction techniques.[11] They 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. Although research in FBSM technology has been conducted for more than 15 years, FBSM has only become available in the commercial CAD systems in the past five years.

Today's CAD systems allow designers to work in three dimensions, using topologically complete geometry (solids) that could be modified by altering the dimensions of the features from which it was created. The FBSM has made design modification much easier and faster. The developers of FBSM CAD systems have but the D back in CAD. Today's design engineers can create a new, complete, parametric model for a configuration, and FBSM CAD can be incorporated into a design environment.

With FBSM tools, the designers must define the relationship and constraints among geometric entities for each feature in the model. This requires some additional time, thought, and planning, but it will pay off when the designer needs to change the model. As a result, the design changes are not time consuming, and it is easier to change the model to develop new variants of existing designs. For example, it is much easier and faster to model holes in a design using solid cut operations than it is to do them with traditional surface modeling tools.

The FBSM process relies on simple top-down and high-level geometric constructions. The most important capability of the FBSM is the ability to capture the design intent. Embedding this intelligence in a model allows workers who are not thoroughly familiar with a product to make changes to existing designs. Another benefit is the capability of suppressing the small features for analysis purposes.

FBSM facilitates the implementation of object-oriented design in CAD systems. For example, a screw within a product can be a unique object. In a design for a new engine, that screw may need to be replicated hundreds of times. Simply copying a single part like a screw is fairly easy with FBSM CAD systems. However, taking this example one step further, the screw has a property that describes its diameter. Suppose the overall engine design changes during a project review, and the screws need to be thicker to support more weight. Then, it should be possible to change just one copy of the screw used in the design. Through links managed by the system, all of the identical screws would reflect the new diameter. FBSM CAD systems treat components as objects, and they can do this task fairly simply. The result may be the automation of design methods used consistently within one organization, or the result may be an engineer's design changes reflected through dynamically linked objects used across multiple designs.

Object-oriented CAD makes it easier to share data between applications because it introduces a layer of abstraction between the data and the user. Instead of requiring every application to translate data between different formats, objects hold property information that describes how the data should be handled by an application. Current FBSM CAD systems offer a library of completed, or partially completed, parameterized objects, enabling one to capture a design process and knowledge, and to document it as a set of objects.

There is another object model in use today. Within the software community, Microsoft's object linking and embedding (OLE) specification provides one of the most broadly used implementations. This specification has been approved by the Design and Modeling Applications Council,† and many vendors. OLE provides a variety of services, enabling data to be shared easily across different applications. For example, the most common way to incorporate data from other applications into a single data file is to embed it as an object within an OLE document. This could facilitate a closer integration between FBSM CAD systems and CAE.

Because today's FBSM systems rely on SM techniques, created models are not always good enough for automatic grid generation tools. The CAD process may create a solid with unacceptable accuracy, e.g., cracks, with sliver faces, or with unacceptable and excessive geometric details. FBSM is not yet a mature and robust technology for complicated aerospace geometric modeling. Even though use of parametric modeling in design would make the FBSM tools ideal for optimization, existing FBSM tools do not have the capability to calculate the analytical sensitivity of a CAD model with respect to the design variables. So it is far from trivial to incorporate FBSM CAD systems into a design optimization process, and it is even more difficult to incorporate them into an MDO environment. Also, it is still a challenging task to parameterize an existing model that is not parametric. It took over 30 years for SM tools to reach today's maturity, and FBSM has been around for less than 10 years. The FBSM approach is a sound approach, but it will take another decade to mature enough for complicated aerospace MDO applications.

III. Standards for Geometry Exchange

Once the CAD model has been completed, the next step is to transfer the data to a CAE application such as a CFD or CSM code. Geometry exchange is always the biggest issue for going from CAD to CAD, or from CAD to CAE, and it could be impeding the development of automatic CAE applications. There is very little incentive for CAD companies to provide a robust tool for geometry exchange. They fear that if they provide a robust tool, they will then loosen their hold on customers.

Obviously, the best way to share data is to use the same CAD system. Generally, exchanging data among different CAD systems is an unreliable process, and so it makes sense to limit the number of CAD systems used in a process. For example, major U.S. car companies have reduced the number of data translations dramatically by selecting a single CAD system for the entire company. To exchange data between a small number of CAD systems, a direct translation is the most efficient and accurate way. These direct translation tools are expensive, but they are cost-effective for a large volume of data exchange.

If exchanging data is necessary, understanding what the data will be used for is the key ingredient for success. If a structured CFD grid must be created using imported data, then only the surface model is required. However, if an unstructured CFD grid must be generated, then transfer of solid models is the only way to satisfy this requirement without having to rebuild the geometry.

There are a number of different file formats for exchanging data among CAD and CAE systems. The most popular formats are IGES in the U.S., Standard d'Echange et de Transfer (SET) in France, Verband der Automobilindustrie (VDA) in Germany, and STEP worldwide. Table 1 shows a list of CAD representations and associated U.S. file formats to support them. For surface and solid data exchange, IGES or STEP can do the job, but to bring data from another system into an FBSM system, the only choice now is to rebuild the model manually. There is no standard file format to support the transfer of parametric data contained in an FBSM. Thus, if a parametric solid model is translated to a data exchange format and then read directly back into an FBSM, all parametric information is lost.

IGES[12] were designed in 1979. They are the most popular format in North America, and they have become reliable for production work. A survey in 1993 found that 66% of firms used IGES for data transfer.[13] The format has gone through several major revisions. It has one big flaw: the data are stored in two sections of the file, a directory section and a parameter section. Many IGES bugs have to do with mismatches between

Table 1 Standards for geometry exchange

Representation	Standards
FBSM	Not supported
Solid models	STEP and IGES
Surface models	STEP, IGES, and DXF
Tessellated models	VRML and STL

†Design and Modeling Applications Council: http://www.dmac.org.

directory and parameter sections. Also, IGES use fixed-length records, which consume a lot of space even when nothing is in them, and therefore, IGES files are very bulky. The development of IGES started in an area when punch cards were popular for putting data into computers. Even when the physical punch cards are not used, data are stored in the form of 80-character records.

STEP[14] is an international standard for the exchange of product model data (ISO 10303). The product data exchange using STEP (PDES) is an American National Standard. STEP is a better geometry standard than IGES in several areas. It is international, more compact, stores data for each entity in only one place, and uses a more modern data architecture. STEP is defined in terms of a new language, EXPRESS. The STEP Application Protocol number 203 (AP203), entitled "Configuration Controlled Design,'' encompasses the relationship between product parts, assemblies, bills of material, change authorizations, change requests, and model release information. Part 42 of AP203 provides methods for describing three-dimensional CAD geometry. Boeing and its primary contractors have been using STEP successfully to check for interference between engine parts and airframe structures.

Part 42 includes most elements found in the IGES standard, including two- and three-dimensional points, lines, arcs, B-splines, conic sections, and planar, spherical, cylindrical, ruled, NURBS, trimmed, and offset surfaces. It also contains topology information for creating solids and their boundary representation. Also, primitive solids such as blocks and spheres may define shapes. AP203 data exchange is not yet highly reliable for analysis of aerodynamics solid models. Most of these solids are created with free-form surfaces, where curves of intersection cannot be defined precisely.[10]

The most significant omission is that STEP does not have a way to describe the geometry constraints employed by FBSM CAD systems. It also lacks methods for rule-based geometry construction. STEP does not contain a history tree relating parts, and this prevents changes to individual parts. Consequently, both the feature descriptions and the parametric relationships that allow CAD models to be changed quickly will be lost in any STEP translation. There are some efforts to bring STEP into line with today's FBSM systems capabilities. The enhanced STEP could add means for capturing and exchanging parametric, constraint-based, and feature-based product models. This addition to STEP will not be a trivial exercise, because parameterization is often associated with history-based approach to modeling, whereas STEP is currently oriented firmly toward the exchange of the explicit, or snapshot, type of product model.

There are two other standards that can be used to exchange data between CAD and CAE. These standards are stereolithography (STL) and virtual reality markup language (VRML), both simple in nature. The STL format is the de facto standard for rapid prototyping. The STL file format allows for the representation of a CAD model as a set of triangles and their normals. The specification of the STL format states that the model must represent a tessellated solid, and STL may be used for automatic grid generation instead of the full CAD geometry. A limitation of STL, which is a tessellated representation, relates to accuracy. For STL models, certain features such as rounds are converted to triangles, and radius information is not accessible. The VRML standard is very similar to STL, and most CAD systems support both. In addition to tessellated data, the VRML standard supports quadrilaterals, cones, cubes, and circular cylinders. Dimensions can be queried from these models, but accuracy becomes an issue because of the approximation of the actual model. The VRML models can be used for grid generation, but they too lack accuracy.

Another important element for optimization that has been left out of all standards for data exchange is the sensitivity of CAD models with respect to design variables. Presently, there is no plan to include sensitivity in the future standards.

IV. Grid Generation

Grid generation is the first step in CAE analysis. The handbook of grid generation[15] contains excellent discussions on most grid generation techniques. There is a tremendous amount of published research on the mathematics of grid generation and its algorithms.[15–36] But, there are a few good automatic grid generation codes. The reason is that writing a good code is significantly more time consuming than writing a good paper.

Fortunately, the CAE software companies have realized that the need for stand-alone grid generation products is diminishing in favor of more integrated tools. These tools have a direct connection to CAD systems, either through a tight integration with CAD or through the data exchange standards, e.g., IGES and STEP. CAD is traditionally seen as the carrier of information about design. However, grid generation usually requires simplification and idealization of the design model. This requirement is the most cumbersome aspect of the grid generation process. Therefore, the analysis model is often rebuilt from scratch, relying upon the judgment of skilled analysts in removing details from the design and duplicating much of the work in creating the geometry. Often, integrated tools are interactive and require the design engineer to provide complex input. As a result, the current grid generation process is not yet a push button process; it is the most labor-intensive and time-consuming aspect of the computational aerosciences. It takes too many man-hours and calendar days, and it requires a grid specialist. This limits the use of analysis codes in the preliminary design. Armstrong et al.[37] proposed a set operations that could facilitate the derivation of the idealized models from CAD geometry. Sheffer et al.[38] used a virtual topology model that could significantly simplify and speed up the generation of the model for grid generation.

To incorporate grid generation tools into a design and optimization system, the tools must 1) use CAD-generated geometry, 2) handle solid models with many surfaces [$\mathcal{O}(10{,}000)$], 3) handle surfaces with bad parameterization, 4) be fully automatic (push-button), 5) be designed for nonspecialists, 6) be robust and have a short design cycle time, 7) calculate grid sensitivity, and 8) be able to create boundary layer/stretched grids.

The remainder of this section focuses on CFD and CSM grid generation methods. Even though both have the same goal of model discretization for analysis, they have different requirements. Generally, CSM requires a relatively coarse grid, but it must handle very complex internal and external geometries. In contrast, the CFD grid is very fine, but it must only model the external geometry. Both classes of grid generation techniques will be discussed in subsequent sections.

The feature-based approach has not been used in a commercial grid generation tool yet. This approach, feature-based grid generation (FBGG),[39] could automate and simplify the grid generation process for very complicated designs based on FBSM. With this technique, the grid is generated for each base feature. As each CAD feature is combined with other features using a Boolean operation to form the model, the individual feature grids could be combined using the same Boolean operation to form a new grid. As with FBSM, FBGG could be based on Boolean operations such as intersection and union of simple grids. As a result, design changes would have little or no effect on the grid generation processes, and it would be easy to generate a new grid for a variant of an existing design. Also, as with FBSM, FBGG relies on a simple, top-down, high-level grid generation construction. It is also possible to create a grid for an idealized model by suppressing the features unnecessary for analysis purposes.

It is important to note that, with respect to design optimization, very few grid generation tools can provide the grid point sensitivity required for gradient-based optimization process.[40]

A. CFD Grid Generation

The handbook of grid generation,[15] and the survey article by Thompson,[41] have provided a complete overview of the current CFD grid generation techniques. These techniques have been developed around the formulations of spatial discretization of flow equations, such as multiblock structured, unstructured tetrahedral, unstructured mixed elements, and Cartesian grids. With the exception of Cartesian grid generation methods,[42] all produce body-fitted grids. The Cartesian method is based on decomposing the domain into cells that are oriented along the three Cartesian directions (x, y, and z). The drawback of this method is that it cannot represent complex geometries accurately. However, this approach can fully automate the CFD grid generation process. It is possible to extend the Cartesian method to model viscous flows.[43,44] The accuracy of these extensions has not been fully assessed. Readers are referred to Ref. 45 for a complete overview and history of the Cartesian grid generation.

Structured and unstructured techniques have three distinct steps: 1) topology creation, 2) surface grid generation, and 3) volume grid generation. The first book on grid generation[18] contains a complete and detailed overview of structured grid generation. With multiblock structured grid methods comes the problem of block topology creation, which has not been adequately automated. Allwright[46] proposed a semiautomated technique to create the topology with the aid of a graphical computer program. Dannenhoffer[47] extended Allwright's ideas to a wide range of configurations. Stewart[48] used a search tree with a small set of rules to drive directional probing from the boundary to search for an appropriate block decomposition. This technique can be fully automated for two-dimensional geometries; however, it is difficult to extend it to three-dimensional geometries. The existing topology creation techniques have simplified the process, but the process is still not fully automated for a general class of geometries.

There have been successful applications of structured grid regeneration for a specific class of geometries.[40,49-51] These methods generally rely on regenerating or deforming an existing volume grid.

The unstructured tetrahedral, unstructured mixed elements, and Cartesian grid generation techniques require the same surface geometry topology as the solid B-Rep model. Because SM tools provide the topology information, unstructured and Cartesian grid generation techniques can be fully automated. As mentioned in the previous section, the challenge lies in transferring the topology information from SM tools to grid generation tools. There are some integrated CAD, structured, unstructured, and hybrid grid generation tools for CFD analysis, but they lack automation. Efforts in unstructured grid generation now appear to concentrate on automation and grid quality. Readers are referred to Refs. 15, 35, 41, and 52 for further discussions.

B. CSM Grid Generation

Grid generation methods for CSM applications are based on either decomposition of a solid model into solid elements,[4,35,53] or dimensional reduction of a solid model into mixed solid/shell/beam elements.[37,54,55] Most commercially available tools belong to the former category. Often these tools are based on *P*-element technology, where the elements may have curved edges. Consequently, a given part can be modeled to higher geometric accuracy with fewer elements than is possible with *H*-element (linear-edge) codes. The *P*-element technology developed at IBM's Almaden Research Center has been incorporated into several major CAD and CAE systems. This method is not good for analysis of anisotropic materials (composites), materials with nonlinear elastic curves, or systems with gaps and large nonlinear deflections. *P*-element codes require unique grid generation routines that can approximate the geometry with polynomial functions used in finite element methods (FEM). There are commercial CSM tools that have integrated CAD, grid generation, and FEM analysis into a single tool. Generally, the use of these tools requires little or no FEM experience, and they are as easy for engineers as using spelling checkers.

The CSM grid generation tools are generally based on either an octree,[4] Delaunay,[35] or advancing front[56] approach. The process of all-hexahedral grid generation has been automated for solid models.[53] A simple, grid-based approach can generate the grid in the interior of the model, and then an isomorphism technique is used to generate the elements in the boundary regions. The plastering algorithm[57] is based on an advancing front technique that generates a hexahedral grid starting from quadrilateral elements on the model boundary.

The second category of FEM grid generation tools is based on dimensional reduction of solid models, where a solid model is converted to an equivalent mixed solid/shell/beam elements. A procedure has been developed for the automatic dimensional reduction of a two-dimensional geometric model to an equivalent one-dimensional beam model. This was achieved by using the medial axis transform,[58] an alternative, skeleton-like representation of the geometric model, having properties relevant to the model. Operations also have been defined and implemented,[54,55] for dimensional reduction of three-dimensional solid models. These operations are interactive, with appropriate physical properties, such as shell thickness, beam section, moment of areas, and torsion constants, calculated automatically.

V. Geometry Parameterization

To avoid the GMGG complexity, often an aircraft is represented by a simple model during the conceptual and preliminary designs. Because simple models are neither accurate nor complete, optimization of these models could lead to an impractical design.[59,60] To use complex shapes in an MDO environment, the parameterization and geometry modeling must use or be compatible with existing FBSM CAD systems, and it must be adaptable to CFD and CSM grid generation tools. To integrate any geometry parameterization tool into a design and optimization environment, the tool must 1) use a common geometry representation for all disciplines, 2) parameterize discipline models consistently, and 3) calculate analytical grid and geometry sensitivities.

Readers are referred to Haftka[61] and Ding[62] for the survey of shape optimization up to 1986. The rest of this section will focus on some recent developments in the area of shape parameterization techniques. These techniques can generally be categorized as discrete, CAD, and free-form deformation.

A. Discrete Approach

The discrete approach is based on using coordinates of the boundary grid points as design variables.[63,64] This is easy to implement, and the geometry changes do not have a limited form. But it is difficult to maintain a smooth geometry, and the optimization process could create a problem in that the optimum design may be impractical to manufacture.[65] Also, for a model with a large number of points, the number of design variables often becomes very large, which leads to high costs and a difficult optimization problem to solve. The benefits of the discrete approach are that 1) complex and existing grids can be easily parameterized, 2) there is a strong local control, 3) analytical sensitivity is available, and 4) there is no shape limitation. However, the drawbacks of the discrete approach are 1) there are too many design variables; 2) because the grid for each discipline is parameterized separately, the parameterization is inconsistent for MDO applications; 3) discipline interaction is difficult to model for MDA and MDO applications; and 4) smoothness is not guaranteed for an unconstrained optimization.

B. CAD Approach

The second parameterization technique is based on using an FBSM CAD system. For example, Blair and Reich[66] presented a vision to integrate a FBSM CAD system with full associativity in a virtual design environment. Within such an environment, calculations of the sensitivity of geometry with respect to the design variables could prove to be difficult. For some design variables, it is possible to relate the NURBS control points to the design variables. Then the analytical sensitivity can be calculated outside the CAD system. For some limited cases, the analytical shape sensitivity can be calculated based on a CAD model.[67] However, this method will not work under all circumstances. One difficulty is that a dimension may be chosen as a design variable for which the variation of a design surface cannot be assumed to be linearly dependent.[67] The second difficulty is that for some perturbation of some dimensions, topology of the part may be changed. Another way to calculate the sensitivity is to use finite difference, as long as the perturbed geometry has the same topology as the unperturbed one. Both methods, the analytical and finite differences, have their pitfalls and limitations.

The advantages of CAD-based parameterization are that 1) parameterization is consistent, 2) complex models can be parameterized, 3) smoothness can be controlled, and 4) models require a few design variables. On the other hand, there are limitations that could make CAD-based parameterization impossible to implement. These limitations are 1) the shape is limited by the parameterization available within FBSM CAD systems, 2) it is difficult to parameterize existing nonfeature-based models, 3) analytical sensitivity is very difficult to obtain, 4) there is little local control, and 5) it is difficult to use CAD for discipline interaction within an MDO environment.

Researchers[65,68,69] have attempted to use CAD representations, such as B-splines and NURBS, for shape optimization without using a CAD system. With the absence of a CAD system, one could calculate the analytical sensitivity directly from the spline definition. But it would be very difficult to parameterize and construct complex three-dimensional models. Consequently, these techniques are well suited for two-dimensional or simple three-dimensional models.

C. Free-Form Deformation Approach

During the preliminary design phase of an aircraft, when the focus is on the mathematical modeling of the outside skin, the free-form deformation technique could serve as an effective tool with sufficient accuracy. Creation of CFD and CSM grids is time consuming and costly. Therefore, the parameterization of existing grids is necessary for shape optimization.

Hicks and Henne[70] introduced a compact formulation for parameterization of airfoil sections. The formulation is based on adding shape 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. The concept of modifying a baseline shape is very similar to free-form deformation.

The free-form deformation is very similar to morphing techniques[71,72] used in computer animation. It can simulate planform, twist, dihedral, thickness, and camber variations.[39] In a sense, the model is treated as putty or clay in areas where it can be twisted, bent, tapered, compressed, or expanded, but retains the same topology.

For example, the planform variations are modeled with a set of quadrilaterals that control the changes. Then the planform design variables are linked to a set of vectors defined at the corners of the quadrilaterals. Any CFD or CSM grid point within a quadrilateral can be mapped from a three-dimensional space ($\bar{r}$) to two-dimensional parameter space (u, v) of the quadrilateral. The change in grid point location, $d\bar{r}$, is computed based on the parametric value, u, v. The free-form deformation concept introduced by this author[39] was successfully implemented for aerodynamic shape optimization with analytical sensitivity using a structured grid[73] and an unstructured grid.[74]

In addition to ease of use and implementation, the free-form deformation has benefits such as 1) parameterization is consistent, 2) analytical sensitivity is available, 3) complex existing analysis models (grids) can be parameterized, 4) there is a strong local control, 5) smoothness can be controlled, and 6) it requires few design variables. There are a few drawbacks such as 1) shape changes are limited and 2) discipline interaction is difficult to model.

VI. Multidisciplinary Interactions

Another important issue is the strong interaction among disciplines common in an MDO environment. All disciplines share the same geometry, and must be able to communicate and share information consistently, e.g., on deflections and loads. Multidisciplinary interactions can reflect physically important phenomena in aircraft such as those occurring as a result of aeroelasticity. Correct modeling of these complex aeroelastic phenomena requires direct coupling of CFD and CSM codes. Maman and Farhat[75] outlined an algorithm for transferring information between two dissimilar grids, e.g., CFD and CSM grids. The method uses the finite element shape function to interpolate the coordinates, pressure, and displacement vectors. The local coordinates for the interpolation is computed by projecting one grid onto another. This method is very similar to methods reported in Refs. 76 and 77. Cebral and Lohner[77] used an adaptive Gaussian integration technique to improve the conservation of forces. The interpolation-based algorithms cannot guarantee conservation. However, this does not mean that they are necessarily unreliable or inaccurate, as demonstrated by Farhat and Lesoinne.[78]

The interpolation-based methods work well for discipline models having the same level of geometry details. As noted by Tzong et al.,[79] a CFD grid generally resembles the true geometry of the aircraft; they include components such as pylons, nacelle, flaps, and slats. However, CSM grids generally represent only major structural components such as wing box, which carries major loads in the spanwise direction. Flaps and slats, which carry relatively small loads, are either represented by a few simple beam elements or are completely excluded. Tzong et al.[79] and Kapania and Bhardwaj[80] developed a method based on finite element technology in which virtual work is employed to transfer the aerodynamic pressure into a CSM grid. The displacements are then converted back to a CFD grid through the reciprocal theorem. Kapania and Bhardwaj were successful in using a simplified version of this method for several modern complex wings.

Smith et al.[81] evaluated six methods for transferring information between CFD and CSM models. These methods were infinite-plate spline, multiquadric-biharmonic, NURBS, thin-plate spline, finite plate spline, and inverse isoparametric mapping (IIM) methods. Out of these six methods, they recommended studying IIM and NURBS further. They indicated that IIM showed great promise for two-dimensional applications, and it needs to be extended to three dimensions. One key ingredient for this extension is the ability to map information known on the CFD model to the CSM model, and vice versa. A typical CSM model consists of triangle and quadrilateral elements. This author[82] has shown a process by which a set of three-dimensional points, e.g., CFD model, can be mapped to either a set of triangles, quadrilaterals, or NURBS surfaces. This procedure can be used to extend the IIM for three-dimensional applications.

The interactions among various disciplines require the manipulation of the original CAD geometry stored as a set of NURBS. Currently, commercial CAD systems do not support this interaction. Also, Smith et al.[81] indicated that NURBS has excellent promise, but there were some problems that needed to be resolved before implementation. One problem was that the input data must be a structured (regular) grid. This requirement forces the data, at best, to be approximated. In most realistic cases, this step is either impossible or time consuming. In a similar study, this author[83,84] proposed and studied a method to use NURBS representation for data transfer among various disciplines. Because this method is based on a general three-dimensional least-squares representation, it does not require the input to be a structured grid. Another advantage of this approach is the control over the tradeoff between smoothness and accuracy.

VII. Conclusions

The GMGG tools are an enabling technology for traditional design processes of today, and even more so for the revolutionary, integrated, multidisciplinary design processes of tomorrow. GMGG tools must 1) be automated, 2) provide consistent geometry across all disciplines, 3) be parametric, 4) provide sensitivity derivatives, and 5) fit in the product development cycle times. Despite the large volume of published research in GMGG areas, there are a few robust commercial tools that are ready for incorporation into MDA/MDO processes.

SM tools for aerosciences applications are not mature, and to solve their technical problems would require either a new generation of surface mathematics or some sort of tolerance-passing scheme yet to be perfected. The feature-based SM technology will help us to automate the design process and perform optimization. However, it will probably take another decade to successfully implement the FBSM techniques in a commercial CAD system capable of handling the detailed design of a complete aircraft. Because of their generality and potential for automation, the unstructured grid generation techniques will become prevalent in the future for CFD applications. There are commercial CSM grid generation codes available that are fully automatic; one area of research is the dimensional reduction of solid models into mixed solid/shell/beam elements. An automatic grid generation method has been proposed in this paper. The method, feature-based grid generation, is based on features and is compatible with FBSM.

There are still a great many open issues that need to be resolved. Following is a list of research opportunities for GMGG tools and algorithms for design and optimization.

1) Tools to automatically heal/mend solid models.

2) A tolerance-free geometry representation for solid modeling.

3) Fully automatic topology creation for structured grid.

4) Feature-based grid generation using constructive solid geometry.

5) Rule/knowledge-based systems to design CSM topology.

6) Dimensional reduction of solid models to solid/shell/beam elements.

7) Tight CAD, grid generation, and CAE integration for MDO.

8) Automatic tools to idealize geometry models (remove and create geometry).

9) CAD-based tools for analytical sensitivity.

10) Object-oriented tools for design and optimization.

11) CAD tools to model the interdisciplinary interactions.

References

[1] Chapman, D. R., and Mark, H., and Pirtle, M. W., "Computers vs. Wind Tunnels for Aerodynamic Flow Simulations," *Astronautics and Aeronautics*, Vol. 13, No. 4, 1975, pp. 22–35.

[2] Roskam, J., *Airplane Design*, Vol. 8, DARcorporation, Lawrence, KS, 1990.

[3] Requicha, A. A. G., and Voelcker, H. B., "Solid Modeling: A Historical Summary and Contemporary Assessment," *IEEE Computer Graphics and Applications*, Vol. 2, No. 2, 1982, pp. 9–24.

[4] Shepard, M. S., and Yerry, M. A., "Finite Element Mesh Generation for Use with Solid Modeling and Adaptive Analysis," *Solid Modeling by Computers: From Theory to Applications*, Plenum, New York, 1984, pp. 53–80.

[5] Machover, C., *The CAD/CAM Handbook*, McGraw–Hill, New York, 1996.

[6] *CAD/CAM, CAE: Survey, Review and Buyers' Guide*, Daratech, Inc., Cambridge, MA, 1997.

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

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

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

[10] Ferguson, D. R., Lucian, M. L., and Seitelman, L., "PDES, Inc., Geometric Accuracy Team Interim Report," Boeing Information & Support Services, ISSTECH-96-013, Seattle, WA, 1996.

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

[12] *IGES: Initial Graphics Exchange Specification (IGES 5.3, 1996)*, U.S. Product Data Association, Charleston, SC, 1996.

[13] "PDES Progress Report," *Computed-Aided Design Report*, March 1993, pp. 1–6.

[14] *STEP: Product Data Exchange Using STEP*, U.S. Product Data Association, Charleston, SC, 1994.

[15] Thompson, J. F., Soni, B. K., and Weatherill, N. P. (eds.), *Handbook of Grid Generation*, CRC Press, Boca Raton, FL, 1998.

[16] Smith, R. E. (ed.), *Numerical Grid Generation Techniques*, NASA CP-2166, Hampton, VA, 1980.

[17] Thompson, J. F. (ed.), *Numerical Grid Generation, Proceedings of a Symposium on the Numerical Generation of Curvilinear Coordinate Systems and Their Use in the Numerical Solution of Partial Differential Equations*, North-Holland, New York, 1982.

[18] Thompson, J. F., Warzi, Z. U. A., and Mastin, C. W., *Numerical Grid Generation*, North-Holland, New York, 1985.

[19] Hauser, J., and Taylor, C. (eds.), *Numerical Grid Generation in Computational Fluid Mechanics*, Pineridge, Swansea, Wales, UK, 1986.

[20] Thompson, J. F., Steger, J. L., and Yoshihara, H. (eds.), *Three Dimensional Grid Generation for Complex Configurations—Recent Progress*, AGARD, CP-AG-309, March 1988.

[21] Sengupta, S., Hauser, J., Eiseman, P. R., and Thompson, J. R. (eds.), *Numerical Grid Generation in Computational Fluid Mechanics*, Pineridge, Swansea, Wales, UK, 1988.

[22] Schmidt, W. (ed.), *Applications of Mesh Generation to Complex 3-D Configurations*, AGARD, CP-464, May 1990.

[23] George, P. L., *Automatic Mesh Generation: Applications to Finite Element Methods*, Wiley, New York, 1991.

[24] Arcilla, A. S., Hauser, J., Eiseman, P. R., and Thompson, J. F. (eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, North-Holland, New York, 1991.

[25] Smith, R. E. (ed.), *Software Systems for Surface Modeling and Grid Generation*, Proceedings NASA-Conference, NASA CP-3143, Hampton, VA, 1992.

[26] Posenau, M. (ed.), *Unstructured Grid Generation Techniques and Software*, Proceedings NASA-Conference, NASA CP-10119, Hampton, VA, 1993.

[27] Knupp, P., and Steinberg, S., *Fundamentals of Grid Generation*, CRC Press, Boca Raton, FL, 1994.

[28] Lober, R. R., and Blacker, T. D. (eds.), *The 3rd International Meshing Roundtable*, Sandia National Labs., Albuquerque, NM, 1994.

[29] Weatherill, N. P., Eiseman, P. R., Hauser, J., and Thompson, J. F. (eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fluids*, Pineridge, Swansea, Wales, UK, 1994.

[30] Tautges, T. J. (ed.), *The 4th International Meshing Roundtable*, Sandia National Labs., Albuquerque, NM, 1995.

[31] Choo, Y. K. (ed.), *Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamic (CFD) Solutions*, NASA CP-3291, 1995.

[32] Mitchell, S. (ed.), *The 5th International Meshing Roundtable*, Sandia National Labs., Albuquerque, NM, 1996.

[33] Soni, B. K., Thompson, J. F., Hauser, J., and Eiseman, P. R. (eds.), *Numerical Grid Generation in Computational Field Simulation*, Mississippi State Univ., Mississippi State, MS, 1996.

[34] Carey, G. F., *Computational Grids: Generation, Adaptation, and Solution Strategies*, Taylor and Francis, London, UK, 1997.

[35] George, P. L., and Borouchaki, H., *Delaunay Triangulation and Meshing*, Hermes, Paris, 1998.

[36] Cross, M., Soni, B. K., Thompson, J. F., Hauser, J., and Eiseman, P. R. (eds.), *Numerical Grid Generation in Computational Field Simulation*, Mississippi State Univ., Mississippi State, MS, 1998.

[37] Armstrong, C. G., Bridgett, S. J., Donaghy, R. J., McCune, R. W., McKeag, R. M., and Robinson, D. J., "Techniques for Interactive and Automatic Idealisation of CAD Models," *Numerical Grid Generation in Computational Field Simulation*, Mississippi State Univ., Mississippi State, MS, 1998, pp. 643–662.

[38] Sheffer, A., Blacker, T., and Bercovier, M., "Steps Toward Smooth CAD-FEM Integration," *Numerical Grid Generation in Com-*

putational Field Simulation, Mississippi State Univ., Mississippi State, MS, 1998, pp. 705–714.

[39] Samareh, J. A., "Geometry Modeling and Grid Generation for Design and Optimization," *ICASE/LaRC Interdisciplinary Series in Science and Engineering*, Vol. 7, Hampton, VA, April 1998, Chap. 4.

[40] Jones, W. T., and Samareh, J. A., "A Grid Generation System for Multi-Disciplinary Design Optimization," AIAA Paper 95-1689, June 1995.

[41] Thompson, J. F., "A reflection on Grid Generation in the 90s: Trends, Needs, and Influences," *Numerical Grid Generation in Computational Field Simulation*, Mississippi State Univ., Mississippi State, MS, 1996, pp. 1029–1110.

[42] Melton, J. E., Berger, M. J., and Aftosmis, M. J., "3D Applications of a Cartesian Grid Euler Method," AIAA Paper 95-0853, Jan. 1995.

[43] Karman, S. L., "SPLITFLOW: A 3D Unstructured Cartesian/Prismatic Grid CFD Code for Complex Geometries," AIAA Paper 95-0343, Jan. 1995.

[44] Wang, Z. J., "An Automated Viscous Adaptive Cartesian Grid Generation Method for Complex Geometries," *Numerical Grid Generation in Computational Field Simulation*, Mississippi State Univ., Mississippi State, MS, 1998, pp. 577–588.

[45] Melton, J. E., "Automated Three-Dimensional Cartesian Grid Generation and Euler Flow Solutions for Arbitrary Geometries," Ph.D. Dissertation, Univ. of California, Davis, CA, June 1996.

[46] Allwright, S., "Multiblock Topology Specification and Grid Generation for Complete Aircraft Configurations," *AGARD Conference Proceedings on Applications of Mesh Generation for Complex 3-D Configurations*, AGARD, CP-464, 1990, pp. 11.1–11.11, Chap. 11.

[47] Dannenhoffer, J. F., "Automatic Blocking for Complex Three-Dimensional Configurations," *Surface Modeling and Grid Generation, and Related Issues in Computational Fluid Dynamics (CFD) Solutions*, NASA-CP-3291, May 1995, pp. 123–142.

[48] Stewart, M. E. M., "A Multiblock Grid Generation Technique Applied to a Jet Engine Configuration," *Software Systems for Surface Modeling and Grid Generation*, NASA CP-3143, 1992, pp. 477–485.

[49] Reuther, J., Jameson, J., Farmer, J., Martinelli, L., and Saunders, D., "Aerodynamics Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation," AIAA Paper 96-0094, Jan. 1996.

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

[51] Leatham, M., and Chapel, J. A., "On the Rapid Regeneration of Hybrid Grids Due to Design Driven Geometry Perturbation," *Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulation*, Mississippi State Univ., Mississippi State, MS, 1998, pp. 533–542.

[52] Mavriplis, D. J., "Unstructured Grid Techniques," *Annual Review of Fluid Mechanics*, Vol. 29, 1997, pp. 473–514.

[53] Schneiders, R., "Automatic Generation of Hexahedral Finite Element Meshes," *Proceedings of the 4th International Meshing Roundtable*, Sandia National Labs., Albuquerque, NM, 1995, pp. 103–114.

[54] Rezayat, M., "Midsurface Abstraction from 3D Solid Models: General Theory and Applications," *CAD*, Vol. 28, No. 11, 1996, pp. 917–928.

[55] Price, M. A., and Armstrong, C. G., "Hexahedral Mesh Generation by Medial Surface Subdivision: Part II. Solids with Flat and Concave Edges," *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 1, 1997, pp. 111–136.

[56] Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466.

[57] Blacker, T. D., and Meyers, R. J., "Seams and Wedges in Plastering," *Engineering with Computers*, Vol. 9, No. 2, 1993, pp. 83–93.

[58] Armstrong, C. G., Robinson, D. J., McKeag, R. M., Li, T. S., and Bridgett, S. J., "Medials for Meshing and More," *Proceedings of the 4th International Meshing Roundtable*, Sandia National Labs., Albuquerque, NM, 1995, pp. 277–288.

[59] Aidala, P. V., Davis, W. H., and Mason, W. H., "Smart Aerodynamic Optimization," AIAA Paper 83-1863, July 1983.

[60] Hutchison, M. G., Huang, X., Mason, W. H., Haftka, R. T., and Grossman, B., "Variable-Complexity Aerodynamic-Structural Design of a High-Speed Civil Transport Wing," AIAA Paper 92-4695, Sept. 1992.

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

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

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

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

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

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

[67] 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," AIAA Paper 96-3990, Sept. 1996.

[68] Schramm, U., Pilkey, W. D., DeVries, R. L., 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.

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

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

[71] Hall, V., "Morphing in 2-D and 3-D," *Dr. Dobb's Journal*, No. 202, July 1993, pp. 18–26.

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

[73] Biedron, R. T., Samareh, J. A., and Green, L. L., "Parallel Computation of Sensitivity Derivatives with Application to Aerodynamic Optimization of a Wing," *Computer Aerosciences Workshop*, NASA Ames Research Center, Moffett Field, CA, 1998, pp. 73, 74.

[74] Nielsen, E. J., and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations," AIAA Paper 98-4809, Sept. 1998.

[75] Maman, N., and Farhat, C., "Matching Fluid and Stricture Meshes for Aeroelastic Computations: A Parallel Approach," *Computers and Structures*, Vol. 54, No. 4, 1995, pp. 779–785.

[76] Murti, V., and Valliappan, S., "Numerical Inverse Isoparametric Mapping in Remeshing and Nodal Quantity Contouring," *Computers and Structures*, Vol. 22, No. 6, 1986, pp. 1011–1021.

[77] Cebral, J. R., and Lohner, R., "Conservative Load Projection and Tracking for Fluid-Structure Problems," *AIAA Journal*, Vol. 35, No. 4, 1997, pp. 687–692.

[78] Farhat, C., and Lesoinne, M., "A Conservative Algorithm for Exchanging Aerodynamic and Elastodynamic Data in Aeroelastic Systems," AIAA Paper 98-0515, Jan. 1998.

[79] Tzong, G., Chen, H. H., Chang, C. C., Wu, T., and Cebeci, T., "A General Method for Calculating Aero-Structure Interaction on Aircraft Configurations," AIAA Paper 96-3982, Sept. 1996.

[80] Kapania, R. K., and Bhardwaj, M., "Aeroelastic Analysis of Modern Complex Wings," AIAA Paper 96-4011, Sept. 1996.

[81] Smith, M. J., Hodges, D. F., and Cesnik, C. E. S., "An Evaluation of Computational Algorithms to Interface Between CFD and CSD Methodologies," Wright–Patterson AFB, Rept. WL-TR-96-3055, Nov. 1996.

[82] Samareh, J. A., "Unstructured Grid in NURBS Surfaces," AIAA Paper 93-3454, Aug. 1993.

[83] Samareh, J. A., "Use of CAD in MDO," AIAA Paper 96-3991, Sept. 1996.

[84] Samareh, J. A., "Aeroelastic Deflection of NURBS Geometry," *Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulation*, Mississippi State Univ., Mississippi State, MS, 1998, pp. 727–736.