Grid Generation for Multidisciplinary Design and Optimization of an Aerospace Vehicle—Issues and Challenges

Jamshid A. Samareh

NASA Langley Research Center, M.S. 159
Multidisciplinary Optimization Branch
Hampton, Virginia 23681 USA
j.a.samareh@larc.nasa.gov

Abstract

The purpose of this paper is to discuss grid generation issues and to challenge the grid generation community to develop tools suitable for automated multidisciplinary analysis and design optimization of aerospace vehicles. Special attention is given to the grid generation issues of computational fluid dynamics and computational structural mechanics disciplines.

Introduction

Design of an aerospace vehicle is multidisciplinary in nature, and multidisciplinary design optimization (MDO) exploits the synergism of primary, mutually interacting phenomena to improve the design. For more information on MDO methods, readers should consult the recent article by Zang and Green [1] and the special issue of *AIAA Journal of Aircraft* [2], both devoted to MDO. A reliable design of an advanced aerospace vehicle requires high-fidelity tools such as computational fluid dynamics (CFD) and computational structural mechanics (CSM). These tools require detailed grid models.

This paper assumes that readers are familiar with the field of grid generation. These proceedings and the *Handbook of Grid Generation* [3] contain excellent discussions on most grid generation techniques. During the past two decades, tremendous progress has occurred in the geometry modeling and grid generation (GMGG) field. However, the lack of automated commercial GMGG tools is still a major barrier to routine applications of high-fidelity tools such as CFD and CSM for MDO of aerospace vehicles. Current commercial grid generation tools are not suitable for an automated design and optimization environment. The following sections will provide a brief overview of the critical elements of GMGG specific to an automated MDO process and elaborate on the reasons current tools are not suitable.

Geometry Parameterization

The choice of shape parameterization technique has an enormous impact on the formulation and implementation of an MDO solution. Reference [4] reviews and evaluates several shape parameterization techniques. Today's computer-aided design (CAD) systems are capable of creating dimension-driven (parametric) models that can capture the designer's intent. The CAD systems have evolved into powerful feature-based solid modeling (FBSM) tools that can model most mechanical parts. For a more detailed account of FBSM CAD systems and role of CAD in MDO, readers are referred to Ref. [5].

Most CAD systems provide access to their geometry operations through an application-programming interface (API), which can automate the grid generation process. Tools [6] and standards that can unify the API for popular CAD systems are under development. The Object Management Group (OMG) has a proposal under review for standard CAD services that could enable the interoperability of CAD and grid generation tools through a common object request broker architecture (CORBA) interface. For more details, readers are referred to <http://www.omg.org> (search "TC Work in Progress" for CAD Services RFP).

Gradient-based optimization requires geometry sensitivity with respect to the design variables; analytical geometry sensitivity can prove to be difficult to obtain from a commercial CAD system. Finite difference methods can be used to approximate the geometry sensitivity (e.g., see Ref. [7]). For more details, readers are referred to Ref. [5].

Model Abstraction

Analysis tools such as CFD and CSM usually require simplification and abstraction of the design model. This requirement is the most cumbersome aspect of the grid generation process. Unfortunately the model is often rebuilt from scratch, relying on the judgment of skilled analysts to remove details from the design, and duplicating much of the work of creating the original geometry. Armstrong et al. have proposed a set of operations that can facilitate the derivation of abstract models from CAD geometry [8,9]. Automation of model abstraction will allow a design process to use a set of hierarchical models with different levels of fidelity, which in turn will promote the use of variable-fidelity analysis and optimization.

Automatic Grid Generation

The complexity of geometry models is increasing in today's design environment; a CAD model often uses thousands of curves and surfaces to represent an aircraft. This level of complexity highlights the importance of automation.

CAD Report magazine predicts that firms with design automation tools will thrive in the next decade [10]. Despite this and other similar predictions, commercial grid generation tools are interactive and require complex input. These tools are the most labor-intensive and time-consuming aspect of the design process. The interactive tools are appropriate for initial setup and for off-line visualization and interpretation, but their effectiveness in an automated MDO environment is very limited at best.

Several techniques could automate the grid generation process. Feature-based grid generation (FBGG) is one such technique. Unfortunately, the approach is not available in a commercial grid generation tool yet. With this technique, the grid is generated for each base feature. As each CAD feature is combined with other features by means of a Boolean operation to form the model, the individual feature grids could be combined by the same Boolean operation to form a new grid. As a result, design changes would have little or no effect on the grid generation process, and generating a new grid for a variant of an existing design would be easy. Another possibility is to create a grid for an abstract model by suppressing the features unnecessary for analysis. The next two subsections provide overview of the CFD and CSM grid generation methods.

CFD Grid Generation

CFD analyses use the detailed definition of the external geometry, which can be represented as a solid model. With multiblock structured grid methods comes the problem of block topology creation. Existing techniques have simplified the topology creation, but the process still is not fully automated for a general class of geometries [11–12]. Successful applications of automated structured grid regeneration exist for a specific class of geometries (see Refs. [13–16]). These methods generally rely on regenerating or deforming an existing field grid.

If the topology of a solid model is available, then unstructured and Cartesian grid generation techniques can be fully automated. Fortunately, efforts in unstructured and Cartesian grid generation now appear to concentrate on automation and grid quality [3,17].

CSM Grid Generation

CSM analysis tools must be able to analyze the external and internal geometries, and today's commercial CAD systems can represent these geometries by a solid model. Then, these models need to be decomposed into a set of beams, shells, and solid elements. CSM grid generation tools can be categorized by their methods of decomposition. These methods are based on either decomposition of a solid model into solid elements or dimensional reduction of a solid model into a mix of beam, shell, and solid elements. Most commercially available tools belong to the first category and are generally based on either an octree, Delaunay, or advancing front approach.

The second category of CSM grid generation tools convert a solid model based on dimensional reduction of solid models, to an equivalent mix of beam, shell, and solid elements. This category is especially important for aerospace applications, where the model usually is made of solid, shell, and beam elements. Operations also have been defined and implemented [18,19] for dimensional reduction of three-dimensional solid models.

Grid Sensitivity Analysis

Grid sensitivity is defined as the partial derivative of the grid-point coordinates with respect to a design variable. The sensitivity analysis is an essential building block of gradient-based optimization. Despite recent advances in sensitivity analysis, very few grid generation tools currently provide analytical grid-point sensitivity [13].

The sensitivity derivatives of a response t with respect to the design variable vector $\overline{v}$ can be written as

$$\frac{\partial t}{\partial \overline{v}} = \frac{\partial t}{\partial \overline{R}_f} \frac{\partial \overline{R}_f}{\partial \overline{R}_s} \frac{\partial \overline{R}_s}{\partial \overline{R}_g} \frac{\partial \overline{R}_g}{\partial \overline{v}} \tag{1}$$

The term $\overline{R}_f$ is the field (volume) grid, $\overline{R}_s$ is the surface grid, and $\overline{R}_g$ is the geometry description. 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 coordinates of the field grid-points. Readers are referred to Ref. [2], which contains an overview of recent advances in sensitivity analysis for CFD, CSM, and other fields. The second term on the right-hand side of Eq. (1) is the vector of the field grid-point sensitivity derivatives with respect to the surface grid-points. The field grid generator must provide the sensitivity derivatives of the field grid-points with respect to the surface grid-points. The third term on the right-hand side of Eq. (1) denotes the surface grid sensitivity derivatives with respect to the geometry description; the surface grid generation tools must provide this term. The fourth term on the right-hand side of Eq. (1) signifies the surface geometry sensitivity derivatives with respect to the design variable vectors; geometry construction tools such as CAD must provide this term.

Zang and Green have reviewed current methods and tools for sensitivity analysis [1]. There are four techniques for computing the sensitivity derivatives: automatic differentiation, complex variables, manual differentiation, and finite-difference approximation. If the source codes are available and are written in

FORTRAN or C, these codes can be differentiated either manually or with automatic differentiation tools.

Automatic differentiation tools such as ADIFOR [20] (Automatic Differentiation of Fortran) or ADIC [21] (Automatic Differentiation of C) can simplify and automate the differentiation process. Argonne National Laboratory maintains a web site on computational differentiation tools* such as ADIC† and ADIFOR‡. These are preprocessing tools. For example, ADIFOR accepts as input a FORTRAN code, along with specifications of the input and output variables. ADIFOR then produces an augmented FORTRAN code that contains the original analysis capability plus the capability for computing the derivatives of all the specified output quantities with respect to all the specified input quantities. Another attractive alternative is the use of the complex variable technique [22,23]. Of course, a hand-coded differentiation will probably be more efficient in terms of both computation time and computer memory [1].

For structured grids, it is possible to use finite differences to calculate the sensitivity derivatives, but there are accuracy issues that must be considered. Figure 1 shows a high-speed civil transport (HSCT) with seven planform design variables and the errors involved in using a finite-difference approximation for shape sensitivity derivative calculations. This error behavior is typical of finite-difference approximations for sensitivities. For larger step sizes the truncation error is predominant, and for smaller step sizes the round-off error is predominant.

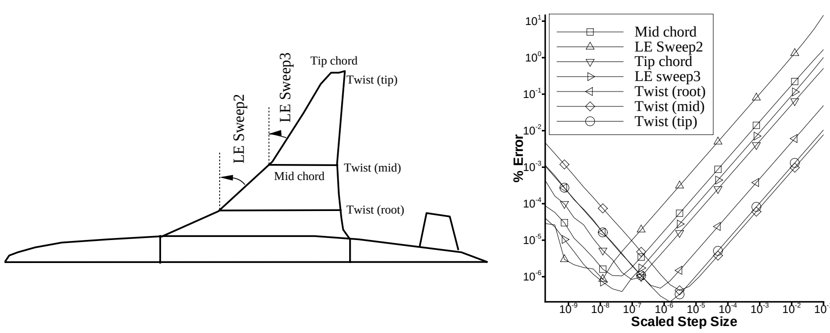

Figure 1. HSCT model and error due to forward-difference approximation.

---

* http://www-unix.mcs.anl.gov/autodiff

† http://www-fp.mcs.anl.gov/adic

‡ http://www-unix.mcs.anl.gov/autodiff/ADIFOR

An optimal step size exists where the error is minimum. This optimal step size is different for each design variable and output function, and also would vary for each optimization cycle. Estimating the error involved in finite-difference approximation of sensitivity derivatives is difficult. More details on the finite-difference approximation error can be found in Refs. [24,25].

Grid Regeneration and Deformation

During the design optimization process, the design surfaces are perturbed. These perturbations and the sensitivity derivatives must be transferred to the surface grid and propagated into field grids. Two basic techniques are available to propagate the surface grid-point movements into the field: 1) grid regeneration and 2) grid deformation. Both of these techniques will be discussed for structured and unstructured grids.

Structured Grid Generation and Deformation

Most structured grid regeneration and deformation techniques are based on transfinite interpolation (TFI). Gaitonde and Fiddes used a grid regenerating technique based on TFI with exponential blending functions [26]. The choice of blending functions has a considerable influence on the quality and robustness of the field grid. Soni proposed a set of blending functions based on arc length [27]; such a set is extremely effective and robust for grid regeneration and deformation. Soni’s algorithm has been incorporated into most commercial structured grid generation packages. Jones and Samareh presented an algorithm for general multiblock grid regeneration and deformation based on Soni's blending functions [13], and also provided analytical sensitivity derivatives by using the automatic differentiation tool ADIC [21].

Hartwich and Agrawal used a variation of the TFI method for the field grid deformation [15]. 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. [14] used a modified TFI approach with blending functions based on arc length. They used finite-difference approximation to compute the sensitivity derivatives for the field grid. Leatham and Chappell used a Laplacian technique, more commonly used for unstructured grid deformation [16].

Unstructured Grid Generation and Deformation

For unstructured grids with large geometry changes, a new grid may need to be generated at the beginning of each optimization cycle. However, for gradient calculations, many small changes must be made, and to regenerate the grid for each design variable perturbation would be too costly. In addition, the new, perturbed grid may not have the same number of grid points and/or the same connectivity; either of these situations will result in discontinuous sensitivity derivatives. Botkin has introduced a local regridding procedure that operates only on the specific edges and faces associated with the design variable changes [28]. Similarly, Kodiyalam, Kumar, and Finnigan used a grid regeneration technique based on the assumption that the solid model topology stays fixed for small perturbations [29]. Solid model topology comprises the number of grid-points, edges, and faces. Any change in the topology will cause the model regeneration to fail. To avoid such a failure, a set of constraints among design variables must be satisfied, in addition to constraints on their bounds.

Batina presented a grid deformation algorithm that models grid edges with springs [30]. The spring stiffness k_m for a given edge jk is taken to be inversely proportional to the element edge length. Then, 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:

$$\bar{\delta}^{n+1} = \frac{\sum_m k_m \bar{\delta}^n}{\sum_m k_m}, \text{ where } k_m = \frac{1}{\left|\bar{r}_j - \bar{r}_k\right|} \tag{2}$$

The term $\bar{\delta}$ is the deformation, and the summation is over all the edges of the elements. Equation 2 is similar to a Laplace operator, which has a diffusive behavior. Zhang and Belegundu proposed a similar algorithm to handle large grid movement [31]. They used the ratio of the cell Jacobian to the cell volume for the spring stiffness.

Crumpton and Giles found the spring analogy inadequate and ineffective for large grid perturbations [32] and proposed a formulation based on the heat conduction equation:

$$\nabla \circ \left[k_m \nabla\left(\bar{\delta}\right)\right] = 0, \quad k_m = \frac{1}{\max(V, \varepsilon)} \tag{3}$$

The term ε is a small positive number needed to avoid a division by zero. This technique is similar to the spring analogy [30], 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 movement. This rigid body movement avoids rapid variations in deformation, thus eliminating the possibility of small cells having very large changes in volume, which could lead to negative cell volumes. With Eq. (3), Crumpton and Giles used an underrelaxed Jacobi iteration, with the nonlinear k_m evaluated at the previous iteration [32].

Conclusions

The major grid generation challenges for an automated aerospace MDO application with high-fidelity analysis tools are:

- automation of geometry abstraction
- automation of grid generation
- calculation of CAD-based analytical sensitivity

---

[1] Zang, T. A., and Green, L. L., "Multidisciplinary Design Optimization Techniques: Implications and Opportunities for Fluid Dynamics Research," AIAA Paper 99-3798, June 1999. (Available at http://techreports.larc.nasa.gov/ltrs/)

[2] Livne, E., ed., Journal of Aircraft, Vol. 36, No. 1, January 1999.

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

[4] Samareh, J. A., "A Survey of Shape Parameterization Techniques," CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, NASA-CP-1999-209136, pp. 333–343, June 1999. (Available at http://techreports.larc.nasa.gov/ltrs/)

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

[6] Haimes, R., and Follen, J. G., "Computational Analysis PRogramming Interface," Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations, pp. 663–672, July 1998. (See http://raphael.mit.edu/capri/docs.html)

[7] Kwak, B. M., and Uphaus, F., "A CAD-Integrated Structural Design Optimization Software Using Finite-Difference Sensitivity Analysis Approach," Optimization in Industry-II, Engineering Foundation, New York, June 1999.

[8] 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 University, MS, pp. 643–662, 1998.

[9] Sheffer, A., Blacker, T., and Bercovier, M., "Steps Toward Smooth CAD-FEM Integration," Numerical Grid Generation in Computational Field Simulation, Mississippi State University, MS, pp. 705–714, 1998.

[10] "CAD in the 2000s," CAD Report, Vol. 20, No. 1, pp. 1–5, January 2000.

[11] 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, Chapter 11, 1990.

[12] Dannenhoffer, J. F., "Automatic Blocking for Complex Three-Dimensional Configurations," Conference on the Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics, NASA-CP-3291, pp. 123–142, May 1995.

[13] Jones, W. T., and Samareh, J. A., "A Grid Generation System for Multidisciplinary Design Optimization," AIAA Paper 95-1689, June 1995. (Available online at http://techreports.larc.nasa.gov/ltrs/)

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

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

[16] 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 University, MS, pp. 533–542, 1998.

[17] Aftosmis, M. J., Delanaye, M., and Haimes, R., "Automatic Generation of CFD-Ready Surface Triangulation from CAD Geometry," AIAA Paper 99-0776, January 1999.

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

[19] Price, M. A., Sabin, M. A., and Armstrong, C. G., "Fully Automatic Quad and Hex Meshing," The Proceedings of 5th International Conference on Reliability of Finite Element Methods for Engineering Applications, NAFEMS, Glasgow Nel, Amsterdam, pp. 356–367, 1995.

[20] Bischof, C., Carle, A., Khademi, P., and Mauer, A., "ADIFOR 2.0: Automatic Differentiation of Fortran 77 Programs," IEEE Computational Science and Engineering, Vol. 3, No. 3, pp. 18–32, 1996.

[21] Bischof, C., Roh, L., and Mauer, A., "ADIC—an Extensible Automatic Differentiation Tool for ANSI-C," Software—Practice and Experience, Vol. 27, No. 12, pp. 1427–1456, 1997.

[22] Squire, W., and Trapp, G., “Using Complex Variables to Estimate Derivatives of Real Functions,” SIAM Review, Vol. 10, No. 1, pp. 110–112, 1998.

[23] Anderson, K., Nielson, E., Whitfield, D., and Newman, J., “Sensitivity Analysis for the Navier–Stokes Equations on Unstructured Meshes using Complex Variables,” AIAA Paper 99-3294, 14th AIAA Computational Fluid Dynamics Conference, Norfolk, VA, 1999. (Available at http://techreports.larc.nasa.gov/ltrs/)

[24] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., “Computing Finite-Difference Approximation to Derivatives for Numerical Optimization,” SOL 80-6 (Contract DAAG29-79-C-0110), Dept. Operations Research—SOL, Stanford University, CA, May 1980.

[25] Lott, J., Haftka, R. T., and Adelman, H. M., “Selecting Step Sizes in Sensitivity Analysis by Finite Difference,” NASA TM-86382, August 1985.

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

[27] Soni, B. K., “Two- and Three-Dimensional Grid Generation Internal Flow Applications,” AIAA Paper 85-1526, January 1985.

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

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

[30] Batina, J. T., “Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes,” AIAA Paper 89-0115-CP, January 1989.

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

[32] 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, pp. 1285–1300, 1997.