AIAA JOURNAL
Vol. 30, No. 5, May 1992

Constructive Solid Geometry Approach to Three-Dimensional Structural Shape Optimization

Srinivas Kodiyalam,* Virendra Kumar,† and Peter M. Finnigan‡
General Electric Corporate Research & Development Center, Schenectady, New York 12301

This paper presents a constructive solid geometry approach to generic three-dimensional shape optimization. The problem definition and shape control are based on constructive solid geometry whereas the assets of boundary representation are exploited to specify the physics of the problem and for meshing the object. This approach is strongly coupled to an automatic mesh generator and uses to its advantage the explicit association of the finite element data with the model geometry for performing shape sensitivity analysis. Hybrid approximation methods are used to minimize the number of finite element analyses. A classical example of a cantilevered plate with a hole and a realistic aircraft turbine disk problem are solved for optimum shape using this new approach.

Introduction

THREE-dimensional shape optimization is an area where comparatively little research has been reported. More specifically, there is very little published work on generalized, fully automatic, three-dimensional shape optimization. The primary goal of this work is to develop a generic system that is capable of solving practical three-dimensional shape design problems. This system is founded on geometry, utilizing a hybrid constructive solid geometry and boundary representation approach.[1] In addition, it is strongly dependent on fully automatic mesh generation and the explicit association of the finite element data with the model geometry.

The possibility for automating the structural design process by determining the shape of complicated three-dimensional components was shown in Ref. 2. In Refs. 3 and 4, improved approximations of stresses in three-dimensional solid elements were used to optimize general three-dimensional structural shapes, resulting in significantly improved convergence characteristics. The reduced basis method,[5] which employs a dimensionality reduction technique to replace the actual design problem with a problem sufficiently small to solve with existing mathematical programming procedures, was used in Ref. 6 to optimize structural shapes.

These developments in the field of three-dimensional shape optimization have provided strong encouragement for using numerical optimization in a practical design environment. However, none of the existing approaches for solid shape optimization are built around a solid modeling system or use fully automatic mesh generation, which are two critical components of the overall automation process. Also, more general shape changes are possible when an automatic mesh generator is employed vs the use of conventional mapped mesh generation techniques.

A geometry-based approach to two-dimensional shape optimization, using automatic mesh generation, has been developed in Ref. 7. The successful application of this approach to fairly complex two-dimensional geometries has encouraged the development of this three-dimensional shape optimization methodology.

Geometry as a Basis for Finite Element Modeling

Structural shape optimization is inherently an iterative procedure. Because the geometric shape of the problem is evolving from an initial state to some converged solution, the specification of the problem must be recast periodically. To be of practical value, this type of iterative analysis needs to be performed completely under computer control, without benefit of human intervention. Therefore, automating the optimization process requires the appropriate foundation. The authors have found that using geometry as a basis,[7,8] vs directly using the finite element data, for formulating and controlling this type of problem leads to generic robust procedures for performing structural shape optimization. There are three aspects to geometry-based finite element modeling (FEM): problem formulation, geometry representation, and geometry manipulation. This section describes the philosophy and implementation details that make this approach so successful.

Figures 1 illustrate the use of geometry as a basis for problem formulation. The process begins by defining the geometry of the problem. Then the mesh control information and problem-specific attributes (material properties, loads, boundary conditions, etc.) are specified and associated with the geometry. Having the mesh control data and an automatic mesh generator, the finite element mesh is produced, with the resulting node and element data also associated with the originating geometry. This enables the attribute data to be automatically mapped from the geometry to the finite element data. Thus, geometry serves as a framework on which all other data depends, as illustrated in Fig. 2.

The second aspect of geometry-based FEM is representation. For solid models, there are two basic representational schemes: constructive solid geometry (CSG) and boundary representation (B-Rep). With CSG, an object is defined in terms of a number of primitives that can be scaled, positioned, or combined using Boolean operations. The CSG representation of an object is an ordered binary tree where the root of the tree represents the final object, the leaves, or terminal nodes, are the primitives, and the nonterminal nodes are either Boolean operators or rigid-body motions. The CSG tree can then be evaluated for purposes of producing a B-Rep. A B-Rep model is one that is described in terms of geometry (i.e., points, curves, and surfaces) and topology. Topology provides the relational information between individual geometric entities as well as providing the trimming information for sur-

Presented as Paper 91-1211 at the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, April 8-10, 1991; received April 15, 1991; revision received July 18, 1991; accepted for publication July 29, 1991.

*Mechanical Engineer, Mechanical Design Methods Program, Solid Mechanics Laboratory, Building K-1, Room 2A25, P.O. Box 8. Member AIAA.

†Mechanical Engineer, Mechanical Design Methods Program, Solid Mechanics Laboratory, Building K-1, Room 2A25, P.O. Box 8.

‡Manager, Mechanical Design Methods Program, Solid Mechanics Laboratory, Building K-1, Room 2A25, P.O. Box 8.

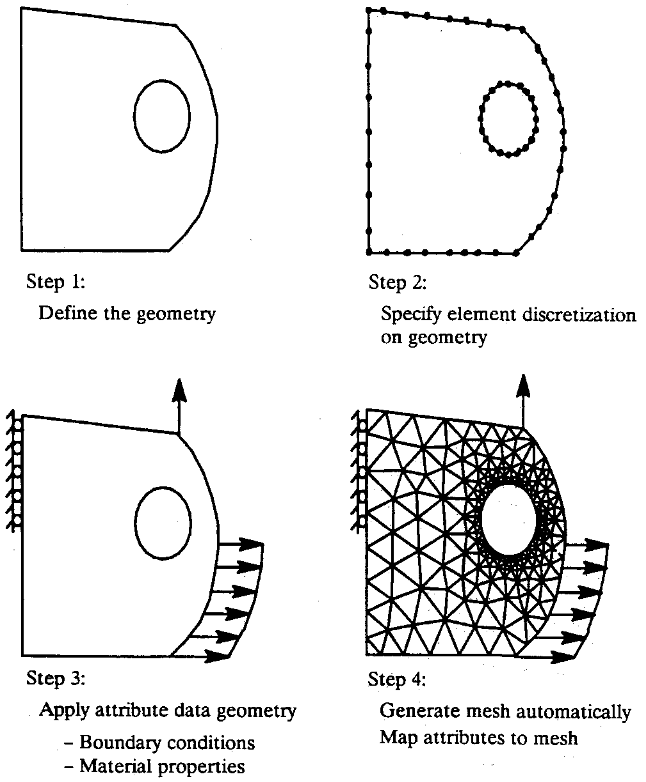

Fig. 1 Geometry-based approach to finite element modeling.

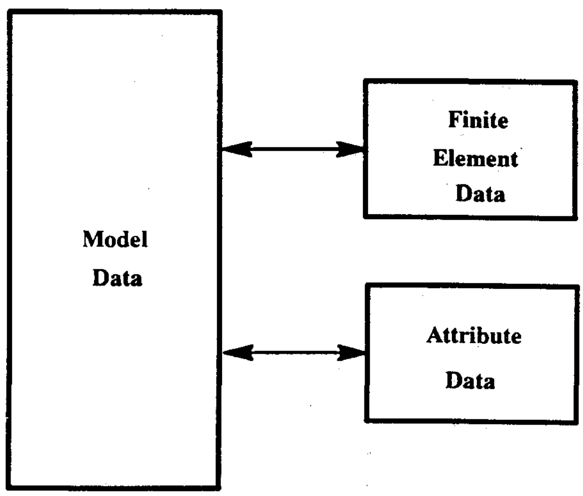

Fig. 2 Relationship among the data structures.

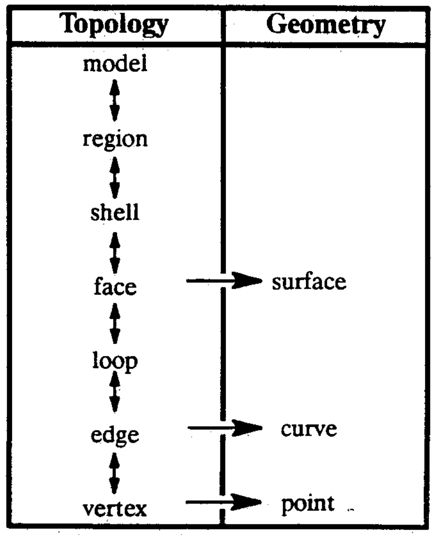

Fig. 3 Model data structures.

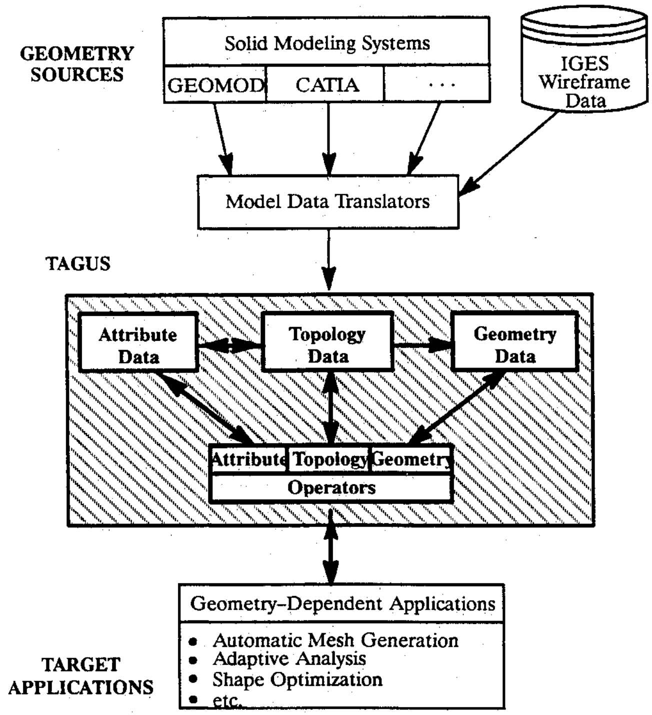

Fig. 4 Geometric modeling utility.

faces. Geometry plus topology provides a complete and unambiguous representation for a solid. Figure 3 shows a B-Rep's model data that consists of a conventional hierarchical topological model and the associated geometry. This paper is concerned with solid objects that can be described with two-manifold topology (i.e., models where an edge has at most two faces using it).

The last aspect of geometry-based FEM is the availability for a set of general data structures and operators that permit access to and manipulation of the model geometry, topology, and attribute data from within an application program. Commercial systems may have some of these capabilities, but typically in a closed environment. To address this limitation of these systems, GE Corporate Research and Development (GE-CRD) has been working on a geometric modeling utility system known as TAGUS (topology and geometry utility system). The TAGUS system, shown in Fig. 4, translates evaluated B-Rep data from a variety of commercial modeling systems, including GEOMOD, CATIA, or UNIGRAPHICS, into its neutral representation, and through its operator driven interface, bridges the gap between closed modeling systems and geometry-dependent application programs, such as shape optimization.

Fully Automatic Mesh Generation

Because three-dimensional shape optimization may require moderate to large changes in the geometry, multiple finite element meshes will be required during the optimization process. If the goal is to automate this procedure, the availability of a fully automatic mesh generator is essential. A fully automatic mesh generator is a procedure capable of producing a valid finite element mesh in a domain of arbitrary complexity, given no input past the geometric description of the object to be meshed and some mesh control parameters. Adherence to this rather specialized definition is important if the level of automation required by three-dimensional shape optimization is to be achieved. A number of approaches to automatic mesh generation have been developed, many of which are summarized by Shephard.[9] The approach that is gaining the most widespread acceptance because of its versatility is based on recursively subdividing a geometric model[10-12] to the point where element generation can be more readily performed. At GE-CRD, a fully automatic mesh generator known as OCTREE[13] is under development and is used as the basis for the three-dimensional shape optimization presented in this paper. The meshing algorithm is based on a combination of recursive spatial decomposition and Delaunay triangulation. The algorithm, depicted for two-dimensional geometry in Fig. 5, can be described as a three-step process: 1) domain decomposition or tree building, 2) element generation using Delaunay triangulation, and 3) mesh smoothing, or nodal repositioning. The resulting finite element mesh is stored in a hierarchical data structure, known as FEDS (finite element data structures), as shown in Fig. 6. The benefit of this elaborate storage of the data is that an explicit correspondence between the constituent finite element data (i.e., fe_nodes, fe_edges, and fe_faces) and the model data are established. This permits coupling of the geometry-based attribute data with the finite element data, and as will be seen later, it provides an efficient mechanism for performing shape sensitivity analysis. In addition, the third step of the OCTREE algorithm, mesh smoothing, works hierarchically off the FEDS. That is, the position of the node points are stored explicitly and parametrically with respect to the model's edges and faces. Smoothing proceeds first with respect to the model edges, then the faces, and finally within the region. The mesh smoothing operators are also used during shape sensitivity.

Constructive Solid Geometry and Shape Control

As mentioned previously, the two common methods for representing a solid model are CSG and B-Rep. Depending on the representational scheme, the methods for controlling the shape are quite different. There are advantages to each method; however, the questions that must be considered are what type of shape control is possible and what response one expects to see when modifying the geometry. Direct manipulation of the B-Rep intuitively seems like the most natural approach because of the physical control one can exert on individual points, curves, and surfaces that comprise the part. Although this approach works quite well in two dimensions,[7] there are some serious problems that emerge when dealing with three-dimensional geometries. For example, geometric entities do not exist autonomously in a model, and therefore, locally altering one geometric entity may adversely affect another. In addition, maintaining three-dimensional slope and curvature relationships among various geometric entities is virtually impossible, except for the most trivial cases. Finally, the number of possible design variables for a B-Rep model is large in comparison to a corresponding CSG model, and thus, global shape control is difficult and costly, at best. In summary, in order to use a B-Rep for three-dimensional shape control, two elements are required. First, the shape controller must have the ability to access and modify individual geometric entities under program control, without loss of model integrity. Second, it must be possible to manage geometric constraint information. The notion of general three-dimensional geometric constraint management is, however, still an unsolved problem.

Alternatively, the CSG approach offers distinct advantages, which circumvent many of the problems indicated earlier. The design variables can be any or all of the parameters used to define the CSG primitives. This typically ensures a relatively few number of unknowns, the model integrity is preserved, and global shape control is provided. One modeling system that we have found to be very useful in supporting our efforts in three-dimensional shape optimization is GEOMOD from Structural Dynamics Research Corporation (SDRC). GEOMOD is a B-Rep-based solid modeler that uses CSG input to define the object. GEOMOD possesses several features that make it attractive from both a modeling and an implementation point of view. GEOMOD, of course, provides classical means for specifying and Boolean (join, intersect, and subtract) simple primitives, such as blocks, cones, cylinders, spheres, etc. In addition, it can Boolean superprimitives (i.e., combinations of simpler primitives). There is also a facility for lofting surfaces through a set of curves to produce complex, sculptured surface models, which preserves surface continuity. Furthermore, it has the ability to define and sweep arbitrary two-dimensional cross sections for the purpose of creating another type of solid primitive. Finally, there is a two-dimensional geometry constraint manager that can be employed to help define the cross section, thereby providing another form of shape control.

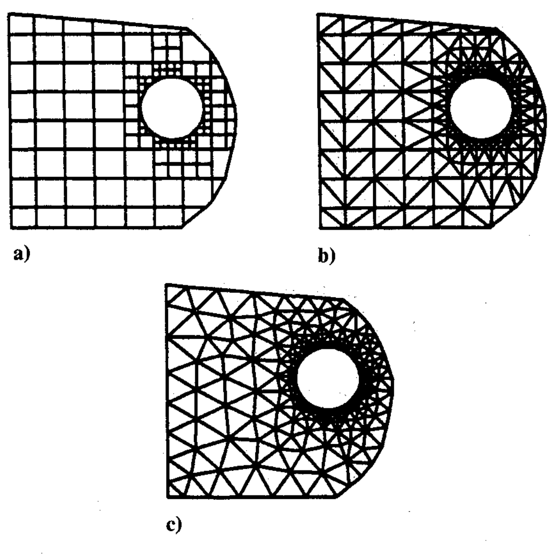

Fig. 5 Three step meshing process: a) tree building; b) element generation; c) mesh smoothing.

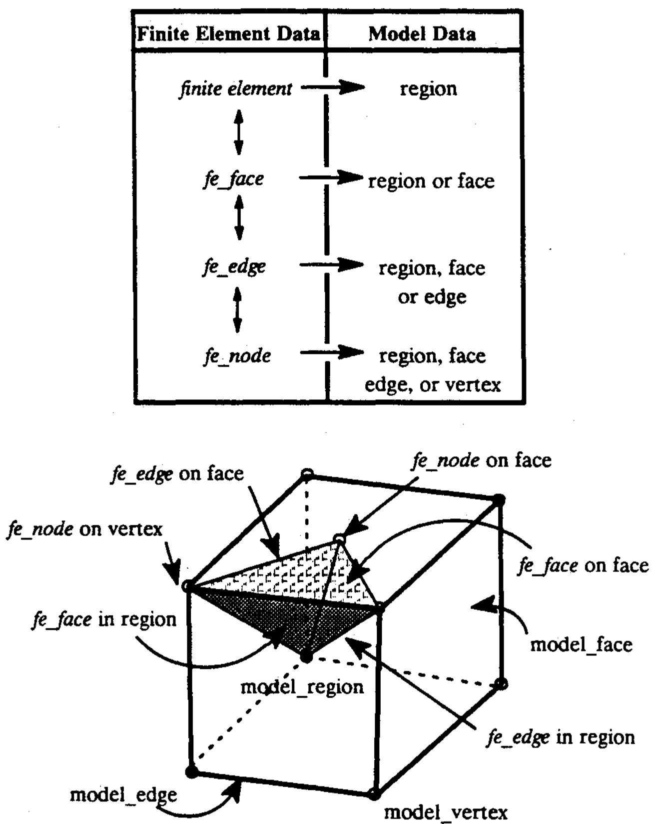

Fig. 6 Associativity between finite element data and model data.

In addition to its modeling facility, GEOMOD maintains an evaluated boundary representation in its relational database PEARL as well as providing access to the B-Rep via a library of database access routines. More recently, SDRC has added the functionality to the system, whereby it is possible to set up interprocess communications through UNIX pipes between GEOMOD and an independent application program. This facility, known as the director observer, falls short of making it an open system, but it at least provides a window into GEOMOD, thereby exposing much of its functionality.

Shape Optimization Problem

The nonlinear programming problem is stated as follows. Find the set of design variables that minimize

Objective function:

$$F(X)$$

subject to

Inequality constraints:

$$g_j(X) \leq 0$$

Side constraints:

$$x_i^l \leq x_i \leq x_i^u \tag{1}$$

where X is the vector of design variables.

In the shape optimization problem considered here, the design variables are geometry-based parameters that control the sizes of the geometric primitives that comprise an object and details of two-dimensional profiles that are used to generate swept and lofted three-dimensional objects. The objective function is the volume of the structure, and structural response constraints are imposed on displacements and stresses at various locations of the object. A geometry-based specification of the response constraints is adopted, whereby displacements can be constrained on a particular edge, stresses on a face of the model, etc. Structural response constraints can also be imposed on the whole region. Geometric requirements, such as preventing surface intersections to maintain a topologically valid model, can be treated in the design problem. Side constraints are also imposed on the design problem.

A modeling approach based on solid design primitives has been demonstrated in Ref. 14. Each primitive is identified by a text description so that the designer can work with design oriented entities rather than lower-level descriptions such as points and lines. The set of design primitives used in Ref. 14 are tailored to idealized automotive suspension systems. As mentioned earlier, two types of design variables are used in the present work: 1) primitive descriptions and 2) geometric details of lofted and swept models. The primitives considered here are generic, geometry-based primitives commonly used in the solid modeling environment.

Shape Sensitivity Analysis

Shape sensitivity analysis deals with the calculation of the sensitivity of the structural responses with respect to the shape design variables. The commonly used approaches are ones that employ the finite difference procedure and the material derivative methods.[15] A numerical scheme, using the finite difference method, is employed here for shape sensitivity analysis. For sensitivity computations, no new mesh is generated; instead, the existing finite element mesh topology is retained. The reason for this is that the shape change resulting from a small perturbation in the design variables does not distort the element shapes to an unacceptable level and, therefore, does not require remeshing. Also, when using an automatic mesh generator, even if the overall geometry changes are small, the number of elements generated can be different than in the original mesh. To avoid discontinuities, a full remeshing is avoided at the level of the sensitivity analysis.

The three-dimensional object, for a design variable perturbation, is first generated using the solid modeler. It is now necessary to reposition the finite element nodes to the new boundary of the object. This requires access to some of the low-level geometric evaluation and manipulation tools of the geometric modeler. The boundary finite element nodes are repositioned to the new boundary using the evaluation operators in TAGUS. For example, if a finite element node lies on an edge or a face, then the coordinates of this node on the new boundary are evaluated using the parametric locations of the node on the corresponding curve or surface. The finite element data structures are then modified to reflect the change in the nodal coordinates. Once all of the finite element nodes on the boundary are modified, a smoothing operation is performed to reposition the interior nodes. The same Laplacian operator, used by OCTREE operating of FEDS, positions each node point within the model region at the centroid of all of its neighboring nodes.

This procedure, for sensitivity calculations in an automatic mesh generation environment, could also be used to implement the semianalytical method provided the programmer has access to the finite element stiffness and strain-displacement matrices.

Approximate Optimization Problem

The nonlinear programming problem, represented by Eq. (1), is replaced by a sequence of approximate optimization problems involving a reduced number of constraints that are approximated as functions of the design variables. Since the approximate models for the constraint and objective functions essentially use the first-order Taylor series expansion, their evaluations require a substantially less amount of computational effort. The hybrid approximation,[16] which is more conservative than the linear and reciprocal approximations, is used for the constraint functions:

$$\tilde{g}(X) = g(X_o) + \sum_{i=1}^{n} B_i(x_i - x_{oi}) \frac{\partial g}{\partial x_i} X_o$$

where

$$B_i = 1 \quad \text{if} \quad x_{oi} \frac{{}^*\partial g}{\partial x_i} > 0$$

$$= \frac{x_{oi}}{x_i} \quad \text{if} \quad x_{oi} \frac{{}^*\partial g}{\partial_{x_i}} \leq 0 \tag{2}$$

When $B_i = 1$, the hybrid approximation is the same as direct Taylor series approximation and otherwise reduces to the reciprocal approximation.

A modification to Eqs. (2) was proposed and implemented in Ref. 6 to handle the changes in sign in shape design variables. The switch between direct and reciprocal approximations is now performed as follows:

$$B_i = 1 \quad \text{if} \quad x_{oi} \frac{{}^*\partial g}{\partial x_i} > 0 \quad \text{or} \quad x_i^l x_i^u \leq 0$$

$$= \frac{x_{oi}}{x_i} \quad \text{if} \quad x_{oi} \frac{{}^*\partial g}{\partial x_i} \leq 0 \quad \text{or} \quad x_i^l x_i^u > 0 \tag{3}$$

The approximate optimization problem is now stated as follows.

Minimize

$$F(X)$$

subject to

$$\tilde{g}_j(X) \leq 0$$

$$x_i^l \leq x_i \leq x_i^u \tag{4}$$

The approximate optimization problem is solved using the method of feasible directions,[17] programmed in the ADS[18] optimizer.

Three-Dimensional Shape Optimization System Architecture

The various components of an automated shape optimization system, outlined in this paper, have been integrated into a three-dimensional shape optimization system. This represents a major extension of the DESIGN-OPT software.[19] The complete system architecture is shown in Fig. 7. GEOMOD is used as the solid modeler for creating and modifying the geometry of the three-dimensional object. Since a procedural call cannot be made to execute GEOMOD, the newly developed interprocess communications capability in I-DEAS is used to establish a link between GEOMOD and the three-dimensional shape optimization modules. The geometry and the topology data of the three-dimensional object is extracted from the PEARL relational database using a boundary representation translator and written into an archive file format representation suitable for the geometric modeling utility TAGUS. Thus, TAGUS provides an interaction mechanism with the geometric representation derived from the modeler.

The meshing of the object is performed using the fully automatic mesh generator OCTREE. The geometry-based specification of the loads and boundary conditions are then mapped onto the finite element model. Presently, geometric requirements like preventing surface crossover and intersections can be specified on the design model. These requirements, specified by the user, are evaluated in the geometric constraint (GEOM-CONST) routines. The finite element software ANSYS is used for analysis and is executed by forking a process from the DECstation to the CONVEX minisupercomputer. The output from ANSYS(TAPE12) is processed to extract volume, displacement, and stress data. These data are then used in the design sensitivity analysis (DSA) routine to compute the gradient of the objective and constraint functions. The link between the DSA routine and GEOMOD is through UNIX pipes, using the SDRC/I-DEAS director-observer capability, in conjunction with their program files. The primitive's sizes, and details of two-dimensional profiles, are updated in the program file to reflect the perturbations in the design variables, and GEOMOD is then re-executed.

The responses computed by the analysis program and the gradient information from DSA are used to generate a hybrid approximation to the optimization problem defined in Eq. (1). This is done in the approximate analysis routine. The numerical optimizer ADS is used to solve the approximate optimization problem. During the optimization stage, no finite element analysis is performed; instead, mathematical approximations to the objective and constraint functions are used. After the convergence of the approximate optimization problem, the whole design cycle is repeated until a satisfactory convergence of the original optimization problem is obtained.

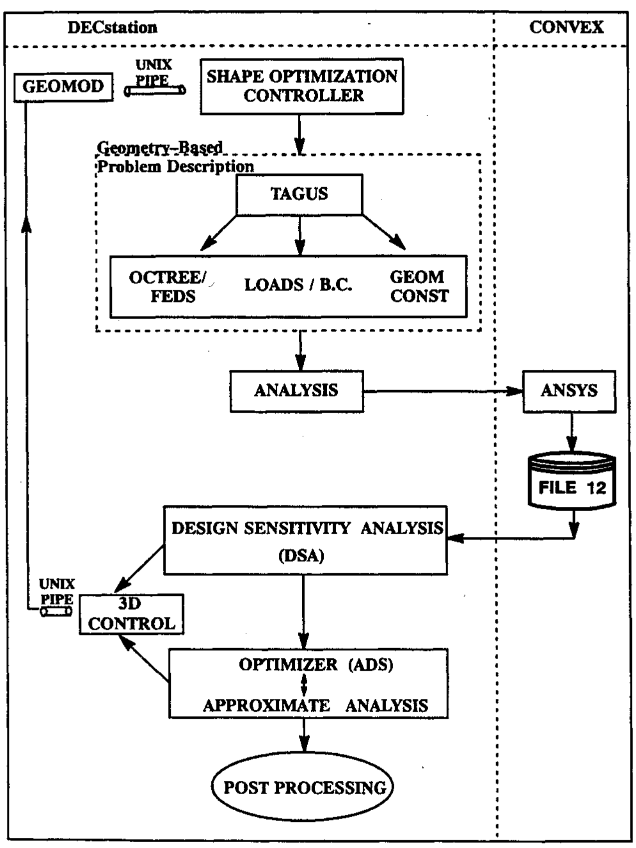

Fig. 7 Shape optimization system architecture.

Numerical Examples

Example 1: Cantilevered Plate with a Circular Hole

The three-dimensional cantilevered plate, shown in Fig. 8, is modeled using three two-dimensional profiles and lofting the profiles along the z direction. The Boolean subtraction operation, using a primitive in the form of a cylinder, is used to create a circular hole through the width of the plate. The finite element mesh of the initial design consists of 336 parabolic tetrahedron elements and a total of 684 nodes. The structure is subjected to an edge load of 20,000 N at the free end. The material properties are as follows: Young's modulus = 10×10^6 MPa, Poisson's ratio = 0.3, and allowable effective stress = 3000 MPa.

The design problem is to minimize the volume of the structure, subject to constraints on the vertical displacements at the loaded edge and effective stress at several locations of the finite element model. The top and bottom surfaces of the structure and the hole size are changed to determine the optimum shape. The design model consists of four design variables, three corresponding to details of the two-dimensional

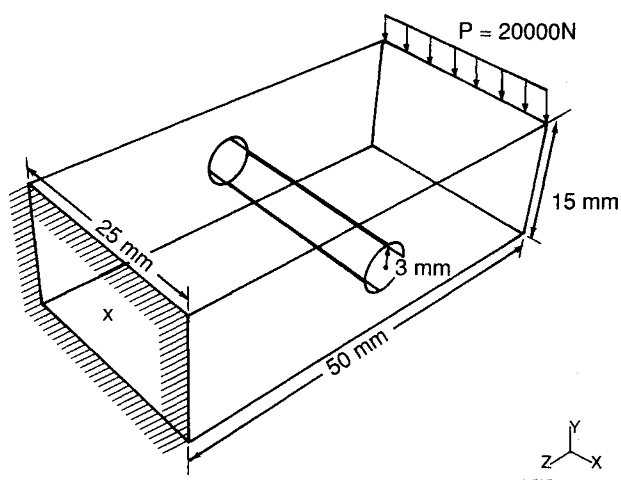

Fig. 8 Cantilevered plate with hole—initial design.

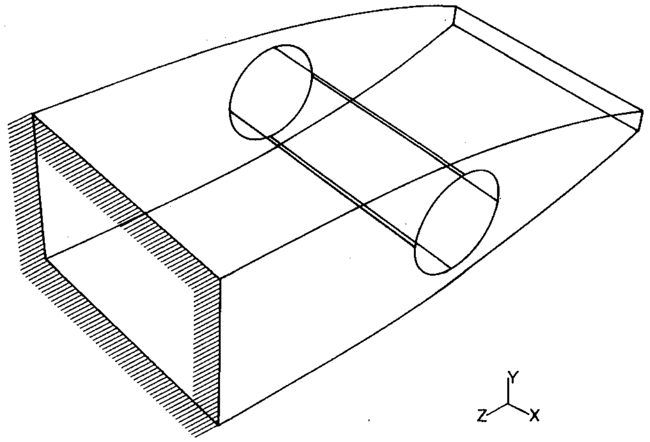

Fig. 9 Cantilevered plate with hole—final design.

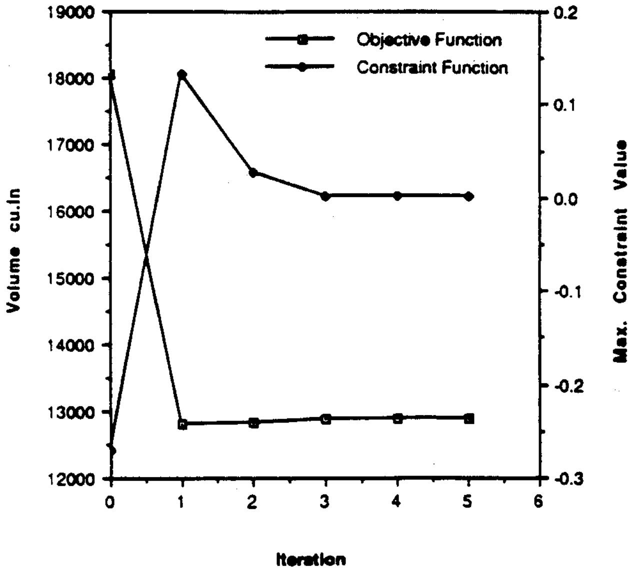

Fig. 10 Design iteration history.

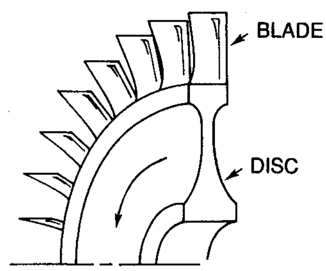

Fig. 11 Aircraft turbine disk.

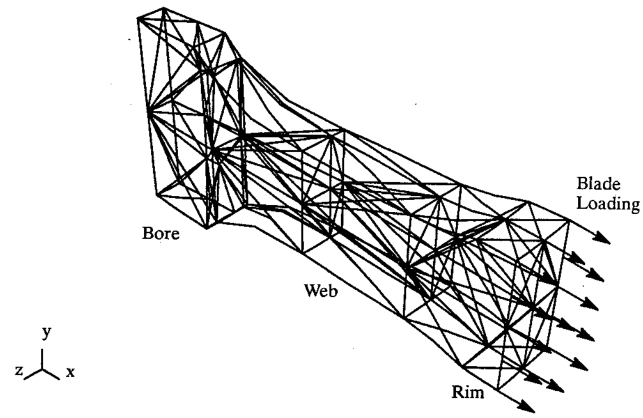

Fig. 12 Aircraft turbine disk—initial feasible design.

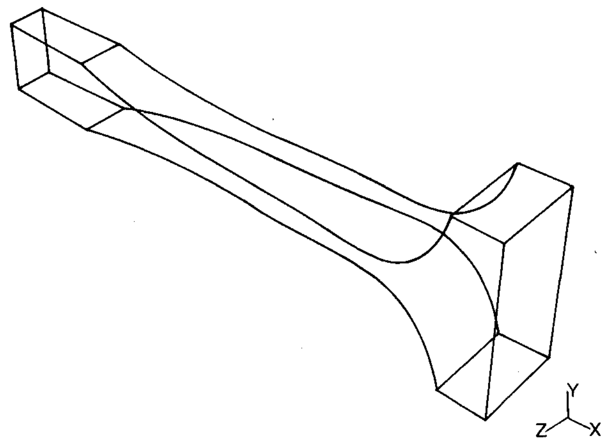

Fig. 13 Aircraft turbine disk—initial infeasible design.

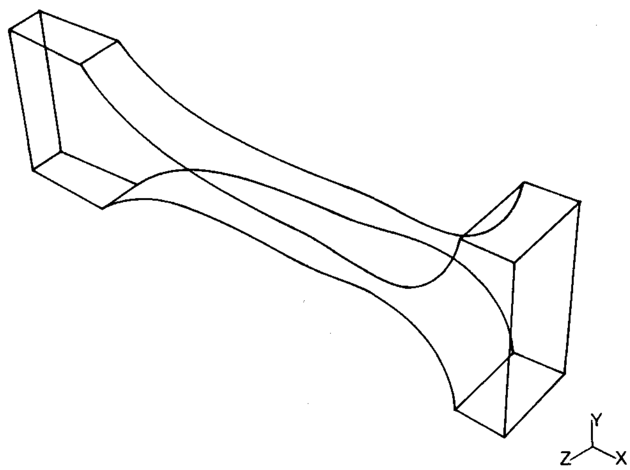

Fig. 14 Aircraft turbine disk—final design.

profiles used for generating the lofted object and the fourth variable for varying the hole size.

The initial design has a volume of 18,051 mm^3 with all of the displacement and stress constraints satisfied. The optimization process converged to a volume of 12,884.8 mm^3, in five iterations, with critical displacement constraints. The final shape is shown in Fig. 9, and the objective and constraint iteration histories are shown in Fig. 10.

Example 2: Aircraft Turbine Disk

The aircraft turbine disk, shown in Fig. 11, was originally solved as a two-dimensional shape optimization problem in Ref. 7. Invoking the concept of variational geometry, the two-dimensional cross section of the disk was described and revolved by a prescribed angle to create the three-dimensional model. The finite element mesh of the initial design, shown in Fig. 12, consisted of 144 parabolic tetrahedron elements. Appropriate symmetry boundary conditions were enforced on the front and the back faces of the disk. The disk was subjected to centrafugal loading and a pressure on the rim to simulate the forces induced by the blades. The material properties and loading are as follows:

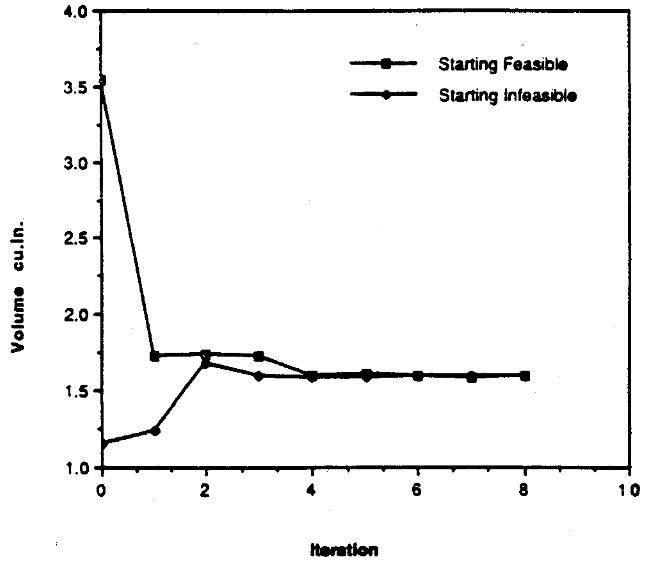

Fig. 15 Volume iteration history.

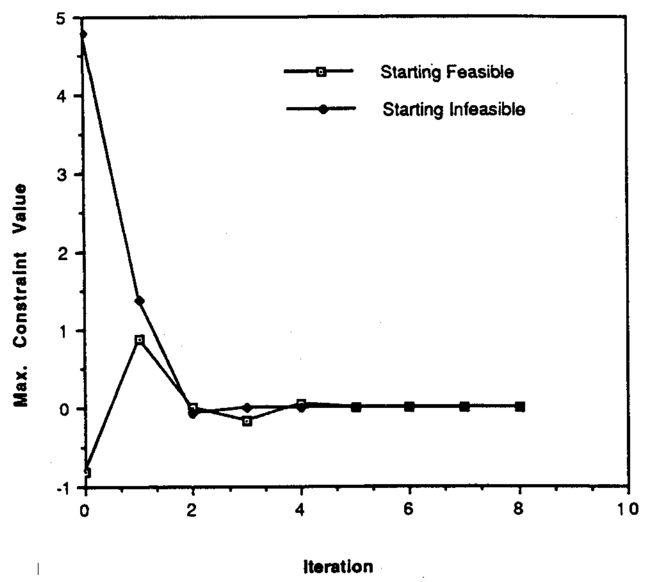

Fig. 16 Constraint iteration history.

Young's modulus	$= 27.1 \times 10^6$ psi
Poisson's ratio	= 0.3
Density	= 0.284 lb/in.^{3}
Rotational speed	= 11,500 rpm, 1204 rad/s
Blade load	= 3869.4 lb/deg
Allowable Von Mises stress	= 114,000 psi
Allowable tangential stress (rim)	= 93,000 psi

The design problem was to minimize the volume of the disk with consraints imposed on the Von Mises effective stresses over the whole region (rim, web, and bore) and tangential stress at the rim. The design model had a total of five design variables that were details of the profiles used to generate the swept model. Two very different starting designs were considered. The first design had an initial volume of 3.546 in.^{3} with all of the constraints satisfied. The second design had an initial volume of 1.154 in.^{3} with violation of effective stress constraints by over 100%. Both of these starting designs converged to a volume of 1.59 in.^{3} with all of the stresses below their allowable limits and similar shapes. In both cases, eight design iterations were required to reach the optimum, each with a different OCTREE generated finite element mesh. The initial infeasible and the final designs are shown in Figs. 13 and 14, whereas the iteration histories are provided in Figs. 15 and 16.

Conclusions

A geometry-based, fully automatic three-dimensional shape optimization capability is presently being developed. The approach is new and is based on solid modeling, specifically a hybrid CSG and B-Rep approach. It embraces the notion of high-level geometry definition and shape control via constructive solid geometry, while at the same time exploiting the benefits of boundary representation for specifying the physics of the problem and for meshing the part. More general shape changes are possible when an automatic mesh generator is employed. The system uses to its advantage the explicit association of the finite element data with the model geometry for performing shape sensitivity analysis. Also, for shape sensitivity analysis, the existing mesh topology is maintained and no new mesh is generated. Corresponding to the perturbation of the geometry, the boundary nodes of the existing mesh are modified and then an internal mesh smoothing is performed. Finally, optimization with respect to the approximate models was sufficiently accurate to overcome constraint violations quickly.

Acknowledgment

This work has been sponsored by the Defense Advanced Research Projects Agency (DARPA) under Contract MDA972-88-C-0047 for the DARPA Initiative in Concurrent Engineering Program.

References

[1]Kodiyalam, S., Finnigan, P. M., and Kuman, V., "A Hybrid CSG/B-Rep Approach to Three-Dimensional Shape Optimization," American Society of Mechanical Engineers, Paper 90-WA/CIE-1, New York, 1990.

[2]Yang, R. J., and Botkin, M. E., "A Modular Approach for Three-Dimensional Shape Optimization of Structures," *AIAA Journal*, Vol. 25, No. 3, 1987, pp. 492–497.

[3]Kodiyalam, S., and Vanderplaats, G. N., "Shape Optimization of Three-Dimensional Continuum Structures via Force Approximation Techniques," *AIAA Journal*, Vol. 27, No. 9, 1989, pp. 1256–1263.

[4]Vanderplaats, G. N., and Kodiyalam, S., "Two-Level Approximation Method for Stress Constraints in Structural Optimization," *AIAA Journal*, Vol. 28, No. 5, 1990, pp. 948–951.

[5]Pickett, R. M., Jr., 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. 489–494.

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

[7]Kumar, V., German, M. D., and Lee, S.-J., "A Geometry-Based 2-Dimensional Shape Optimization Methodology and a Software System with Applications," *Proceedings of the 3rd International Conference on CAD/CAM Robotics and Factories of the Future*, edited by B. Prasad, Vol. 2, Springer-Verlag, New York, 1989, pp. 5–10.

[8]Finnigan, P. M., Kela, A., and Davis, J. E., "Geometry as a Basis for Finite Element Automation," *Engineering with Computers*, Vol. 5, No. 5, 1989, pp. 147–160.

[9]Shephard, M. S., "Approaches to the Automatic Generation and Control of Finite Element Meshes," *Applied Mechanics Review*, Vol. 41, No. 4, 1988, pp. 169–184.

[10]Jackson, C. L., and Tanomoto, S. L., "Oct-trees and Their Use in Representing Three-Dimensional Objects," *Computer Graphics and Image Processing*, Vol. 14, No. 3, 1980, pp. 249–270.

[11]Yerry, M. A., and Shephard, M. S., "Automatic Three-Dimensional Mesh Generation for Three-Dimensional Solids," *Computers and Structures*, Vol. 20, No. 1/3, 1985, pp. 31–39.

[12]Kela, A., "Hierarchical Octree Approximation for Boundary Representation Based Geometric Models," *Computer-Aided Design*, Vol. 21, No. 6, 1989, pp. 355–362.

[13]Graichen, C., Hathaway, A. F., Finnigan, P. M., Kela, A., and

Schroeder, W. J., "A 3-D Fully Automatic Geometry-Based Meshing System," American Society of Mechanical Engineers, Paper 89-WA/CIE-4, New York, 1989.

[14]Botkin, M., "Shape Design Modeling using Fully-Automatic 3-D Mesh Generation," *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 31st Structures, Structural Dynamics and Materials Conference*, AIAA, Washington, DC, 1990, pp. 171–183.

[15]Haug, E. J., Choi, K. K., and Komkov, V., *Design Sensitivity Analysis of Structural Systems*, Academic, New York, 1986.

[16]Starnes, J. H., and Haftka, R. T., "Preliminary Design of Composite Wings for Buckling, Stress and Displacement Constraints," *Journal of Aircraft*, Vol. 16, No. 6, 1979, pp. 564–570.

[17]Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: with Applications*, McGraw-Hill, New York, 1984.

[18]Vanderplaats, G. N., *ADS User's Manual*, Version 3.0, VMA Engineering, Goleta, CA, March 1988.

[19]Kumar, V., Lee, S.-J., and German, M. D., "Finite Element Design Sensitivity Analysis and its Integration with Numerical Optimization," *Computers and Structures*, Vol. 32, No. 3/4, 1989, pp. 883–897.