Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

Integrated approach for shape optimization

M. El-Sayed and K. Zumwalt
Department of Mechanical Engineering
Kettering University, USA

Abstract

This paper presents an approach to shape optimization that integrates the tasks of geometric modeling, design parameterization, design characterization, response analysis, sensitivity analysis, and optimization. The developed approach integrates the response analysis, sensitivity calculation and optimization around the modeling code. All data preparation and post-processing tasks can be performed from within modeling software. To interface with commercially available software packages, several techniques for using existing functions have been developed. Other desired functions have been added using command language functions, menu customization features and external codes. An overview of the shape optimization process is given with some design examples, in order to demonstrate the ease and efficiency of the developed approach.

1 Introduction

Traditional steps in computer-aided structural analysis involve the creation of a geometric model, constructing a finite element model, performing a finite element analysis and using graphical post-processing software to evaluate the results. Because they provide a systematic method of searching for the "best design" without trying every possible solution, mathematical optimization techniques have become an important part of this process. The increased interest in using optimization techniques has also spurred the development of efficient methods for sensitivity analysis. Sensitivity analysis may be used to drive the optimization algorithm or as a separate design tool to provide the designer with quantitative data on the relationship between structural response and design parameters.

Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

Most optimization problems involving the shape and configuration of a structure are complex and computationally expensive. Most structural optimization problems are solved by: first constructing a geometric model, second creating a finite element mesh, and defining the boundary conditions using a graphical modeling package. The data is then translated from the modeling package database into one or more input files for the analysis software. The analysis and optimization steps are then carried out by linking separate a finite element, optimization, and sensitivity analysis codes together. This approach has worked well, but it is relatively inefficient since interface programs must be written to link the separate applications and provide for communication between codes. An additional problem exists in this type of approach. Since the model geometry is usually available to the analysis and optimization software only in the form of the finite element mesh, desired shape changes are controlled by defining arbitrary basis shape functions which are assigned to groups of nodes in order to control the nodal coordinate locations. This process has two undesirable aspects. The first is the additional requirement of defining a number of suitable basis shapes, which will not only control the shape of the structure but also avoid distorting the finite element mesh as the shape is changed. Once the basis functions have been selected, the appropriate combination of control parameters must be determined and then linked to the desired groups of nodes. The second problem is that changes made to the meshed finite element model are not reflected in the geometric model, which was used to create it. If this model is to be used for other tasks, i.e. the generation of numerically controlled (NC) tool programs, the optimized shape data must be converted. To solve structural optimization problems efficiently, both in terms of human and computer resources, traditionally independent areas need to be integrated. These areas are computer graphics and computational geometry; response (deformation, stress, frequency) analysis, sensitivity analysis, and numerical optimization.

In this paper it is shown how the processes of model geometry, design parameterization, analysis, and optimization control can be defined and implemented using an existing geometric modeler, namely PATRAN from MSC Software. The response analysis, sensitivity analysis, and optimization capabilities are provided by a developed multi-purpose code FESSOP (Finite Element Structural Sensitivity analysis and Optimization Program). This code has been designed to work directly with the graphics package's modeling database. This approach has several advantages over linking separate codes for shape optimization. The developed code contains all desired analysis and optimization functions and has been designed around an existing database, there is no need for interface programs, translators, separate databases, or separate input data and control files. All model and control data is obtained directly form the modeling database or a small auxiliary ASCII data file. This file contains execution control data such as type of analysis, algorithm selection, convergence criteria and iteration limits. Another advantage of this approach is that it takes advantage of the networked workstation environment. The computationally intensive tasks of response and sensitivity analysis can be

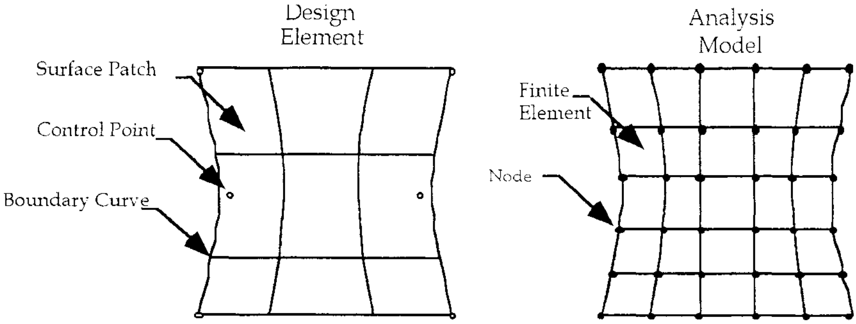
Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

delegated to a workstation while a less powerful processor manages the interactive user interface. Since the modeling database can be written in ASCII format [1], the compute server need not have the same operating system.

2 Design parameterization

The concept of the design element has been used as a means of unifying the geometric, design and analysis data, Braibant and Fleury[2]. With this approach, the structure's geometric model is divided into a relatively small number of regions, which may or may not have changeable boundaries, as shown in Figure 1. The regions having movable boundaries are designated design elements. The finite elements, which make up the analysis model are defined within the parameterization of these design elements and are determined automatically as the design elements are shaped. This approach provides a convenient and efficient way to describe the design model. The drawback of using this type of representation of the design model for a non-integrated approach is that preprocessing software has not yet directly implemented this idea and thus model creation is somewhat clumsy and tedious. Also, preprocessing packages do not support the preparation of all data required to control optimization and sensitivity analysis algorithms.

Figure 1. Typical design element and associated analysis model.

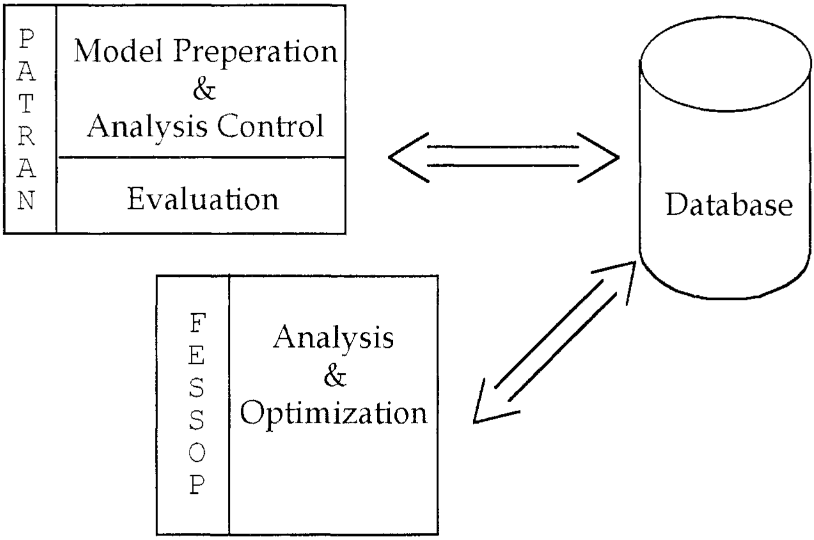


There are several approaches that may be taken to address this problem. The first is to develop modeling software from scratch with moderate capabilities. Chen [3]. Although this allows a maximum of integration the programming effort of the modeling software is prohibitive for the solution of general large problems. Another alternative is to create a separate software package which has a user interface, interface programs and translators to provide the necessary links and additional input data required to integrate the individual graphical, analysis and evaluation software packages as in Stone, Santos, and Haug [4]. The approach taken in this work is to use the application development tools that accompany a commercial software package in order to add and integrate the desired data preparation, analysis, and optimization functions. This approach eliminates the costs associated with the inter-process communication,

Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

duplication of stored data, repeated calculations, and manual preparation of data.

Figure 2. Integrated design environment.



The overall process is illustrated in Figure 2. The designer first creates the design model data. This phase includes the creation of the geometric model; the selection of the shape and response parameters, which will be used if sensitivity analysis or optimization is desired; and the generation of the initial finite element model. Analysis and optimization options are then selected and the analysis is performed. After the analysis is completed, the solution data is viewed using PATRAN's post-processing functions. If desired, changes may be made and the process repeated.

3 Design model preparation

To define the shape design model the following task must be performed:

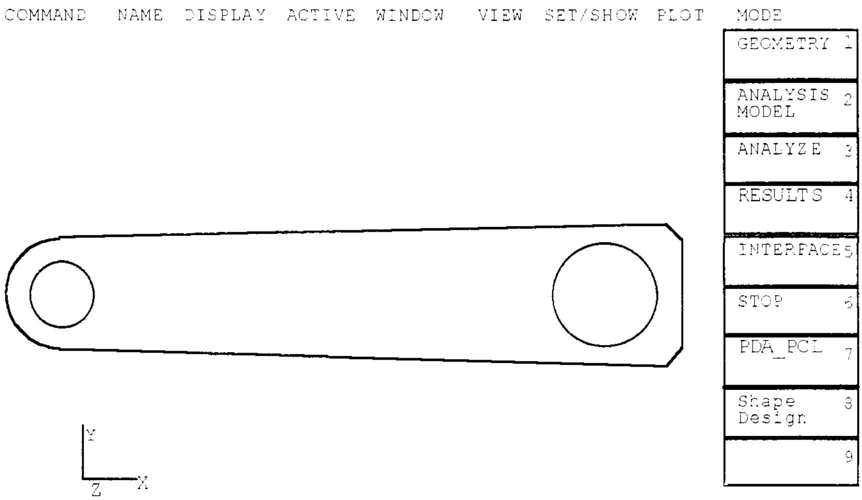
1) Create model geometry.
2) Define design elements.
3) Select shape control parameters
4) Set the move directions for control parameters.
5) Set move limits for control parameters.

Step 1, represents the normal process of creating a model for finite element analysis and is supported directly by PATRAN. The remaining tasks must be achieved using PATRAN command language (PCL) [5], and external programming. These extended functions can be made available to the designer through the menu *Shape Design,* which has been added to the standard PATRAN menu system. This enhanced system is shown in Figure 3 along with

Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

the geometric model of a torque arm, which is used to demonstrate the model creation, analysis, optimization, and evaluation steps. The surface patches that represent the interior domain, of the torque arm and entity labels, have been hidden for clarity. Submenus of the *Shape Design* menu allow access to the extended functions required to implement model creation steps 2 through 5.

Figure 3. PATRAN modeling environment.



PATRAN's model database is based on representing all curves, surfaces and volumes via cubic Hermite polynomials. Thus, lines, arcs, and spline curves are converted into individual cubic Hermite curve segments before being stored. Surface constructs are likewise converted into Coons patches using the cubic Hermite shape functions. For finite element analysis, this is a simple and flexible way to store geometric data since there is no direct relation needed between the descritized finite element model and the geometric shapes that produced it. For shape design however, this creates a number of problems. In this form, it is difficult to maintain more then C^1 continuity between connected line segments. Hermite curves are also not suited for interactive shape design since curve and surface shape depends on both control vertices and parametric slopes. The non-intuitive relationship between variations in slope and the resulting entity shape make them difficult to manipulate directly. Also, if an edge is generated using B-spline, parametric cubic spline, or Bézier options, the connectivity of the Hermite curves that are generated is not stored, nor are the control vertices directly related to the generated curves unless they are coincident with the knot points. A similar problem exists for B-spline, Bézier, and cubic spline surfaces since these are all converted to individual Coons patches.

Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

The additional information required to define these curves can be included in the database by using PATRAN's NAME feature [6]. The NAME command creates an entry into the model database containing the name of an entity set and a list of associated entities. To maintain the definition of the curves and surfaces, a name set can be defined containing the various entities required to define the particular curve or surface, i.e. control grids, knot grids, line segments, surface patches. When an edge or surface definition is needed, the database can be queried for the appropriate named set. This approach has an additional advantage, since it also allows the linking of different curves into a single boundary with discontinuities. Edge curves are designated by the key name EDGE followed by an integer id number and type code. The β-spline formulation, Bartels [7], is not offered by PATRAN, it has been added as an extended feature. Using this convention for edge representation, an entry for the upper edge of the torque arm boundary, shown in Figure 4, which is represented by a B-spline can be created manually by placing the entities in the active set.

This process can be automated through the user defined menu commands and PCL programming. The *Shape Design* menu contains two sub-menues shown in Figure 5. The first, *Design Model*, contains the functions associated with building the design model. This menu is shown in Figure 6. Options 1 and 2 of this menu provide for the definition of edges and surfaces.

Figure 4. B-spline representation of torque arm edge

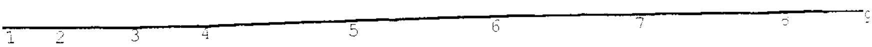
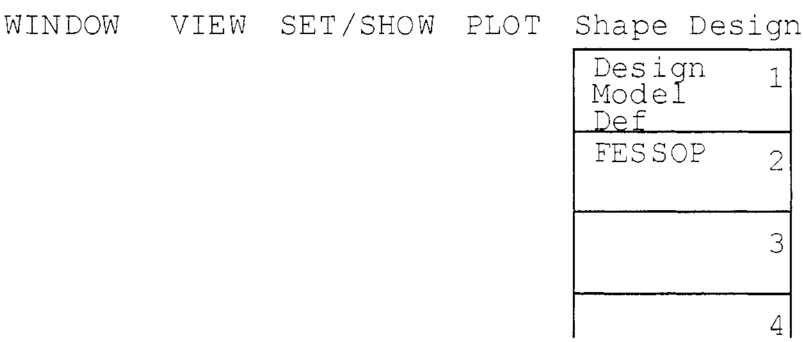


Figure 5. Shape Design Menu



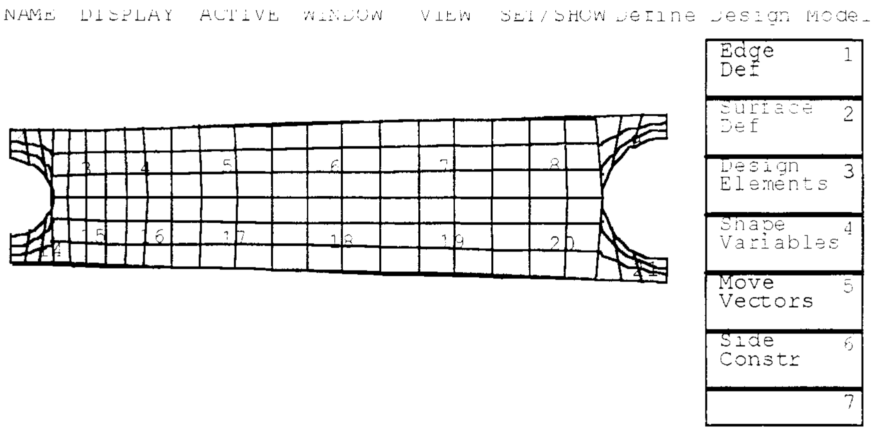
The designation of the design element in Figure 6 is accomplished by using option 3 of the *Design Model Definition* menu. The information for each design element is stored in the form of named sets, which contain the entity information required to link edge and surface patch data to the node and

Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

element tables that are generated when the finite element analysis model is created. The information in these sets is used in the calculation of sensitivity data and to update the analysis during shape optimization.

Once the moving boundaries of the model have been defined, the next step in the design model preparation is to select and define the shape design variables associated with each changeable boundary. The design variables are again defined in the database by a named set consisting of each control grid that is allowed to move. These named sets carry tags. These may be indicated by individual entries or based on boundary curve selection. Since the design variables are defined at the geometry level instead of the descritized system level, there is a direct correspondence between the design parameters and the model geometry. The advantage of this is twofold. This approach eliminates the selection of basis functions to control nodal coordinates since the shape of the design is controlled by the parametric equations of the geometric model. Thus the basis functions used are the curve or surface representations chosen to describe the model. Because the designer is free to choose from a variety of geometric description this approach allows great flexibility. Also, since the design variables control the shape of the curves and surfaces the designer need not worry about which nodes are affected. This information is maintained in the database and the finite element model is updated automatically as the design changes.

Figure 6. Design model definition menu



Associated with each shape variable is a vector, which controls the direction of movement. These shape vectors are determined by: the coordinates of the control grid, and the associated grid point. These grid points are defined in the database by named sets. As with the shape variables, the move vector grids may be assigned individually or associated with the corresponding boundary. Option 4 of the menu provides these functions.

For sensitivity analysis only, the preceding step is the last. However if optimization is to be performed, then side constraints must be defined on the

Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

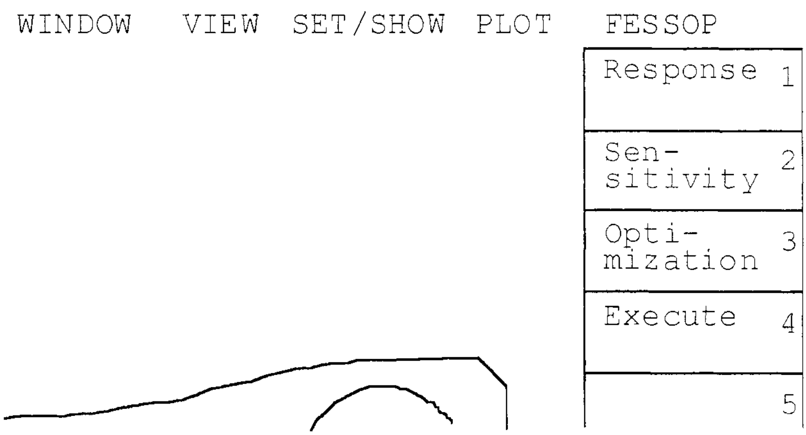
design variables. These constraints are imposed by a pair of girds on either side of the design variable, past which the control grid will not be allowed to move. This information is stored in the database and is generated using option 5.

Once the design model has been defined, the creation of the initial finite element analysis model is performed in the usual manner.

4 Analysis and optimization

To provide the analysis and optimization capability, the package FESSOP was developed. In addition to the calculation of weight, displacement and stresses, FESSOP has the capability to perform first- and second-order sensitivity analysis of weight, displacement, stress, and Mises stress using both the direct and adjoint methods. Specific formulation and implementation details can be found in Zumwalt [8]. Displacement sensitivities are calculated at the finite element nodal points. The stress sensitivities are calculated at the Gauss integration points, the element centroid, or at the nodal points. FESSOP also directly incorporates two optimization methods, Robust Feasible Directions, Vanderplaats [9], and Sequential Quadratic Programming using Quadratic Constraints, Zumwalt [8]. The analysis and optimization functions are controlled through the provided menu options.

Figure 7. Analysis and optimization menu



The code obtains all geometric and analysis data form the PATRAN database as described in the previous section. Execution of this phase takes place asynchronously so that the designer is free to perform other tasks. Depending on the options selected FESSOP produces evaluation data native in PATRAN formats. Thus there is no need for the nuisance of translating from other output formats. If optimization was selected, the code also produces a new database containing the updated model.

Transactions on the Built Environment vol 52, © 2001 WIT Press, www.witpress.com, ISSN 1743-3509

Access to the analysis and optimization capability is provide by the second menu option of Figure 3. The *FESSOP* menu option contains extended functions for the selection of analysis type, i.e. response only, response and sensitivity, or optimization. This menu is shown in Figure 7.

The first menu option *Response* allows the selection of response calculations to be made. Available options include weight, displacement, element centriod stress, average nodal stress and Mises stress. The option *Sensitivity* provides for the selection of first-order only or first- and second-order sensitivity calculations of various response measures. The *Optimization* menu allows for the selection of optimization algorithm, constraints, and the input of control information. Once the desired analysis options have been set, analysis starts by choosing *Execute* form the menu.

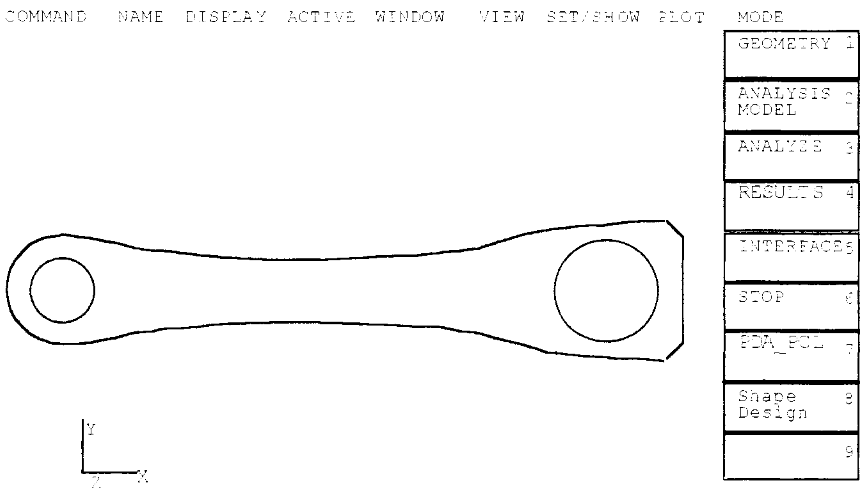
5 Output and post processing

The PATRAN software package has a variety of post-processing function for the plotting data, deformed geometries, and contours. These functions can be used to perform the following evaluation functions

1) View optimized shape
2) Deformed geometry
3) Stress contours
4) Displacement sensitivity contours
5) Stress sensitivity contours

An example of these functions is shown if Figures 9 of an optimized shape of a torque arm test case.

Figure 8. Display of optimized shape.



6 Conclusions

This paper presents an approach to provide the shape optimization capability in an integrated manner. The individual stages of the structural design process, from model creation through final evaluation have been implemented so that all tasks can be performed from within a single interactive graphical environment. The consistency of environment and total integration of the analysis and optimization tasks makes this package an efficient and easy to use tool for structural shape design and optimization.

7 References

[1] PDA Engineering, *PATRAN Plus User's Manual Volume2*, Software Products Division, PDA Engineering, 1989.

[2] Braibant, B. and Fleury, C., *Shape Optimal Design Using B-Splines*, *Computer Methods in Applied Mechanics and Engineering*, 44, pp. 247-267, 1984.

[3] Chen, H. , *A Computer Graphics Pre- and Post- Processor for Structural Optimization*, Master of Science Thesis, University of Missouri, 1987.

[4] Stone, T. A., Santos, J. and Haug, E. J., *An Interactive Pre-Processor for Structural Design Sensitivity Analysis and Optimization*, Computers & Structures, 34, 3, pp. 375-383,1990.

[5] PDA Engineering, *PATRAN Command Language Guide*, Software Products Division, PDA Engineering, 1989.

[6] PDA Engineering, *PATRAN Plus User's Manual Volume 1*, Software Products Division, PDA Engineering, 1989.

[7] Bartels, R. H., Beatty, J. C. and Barsky, B. A., *An Introduction to Splines for Use in Computer Graphics & Geometric Modeling*, Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1987.

[8] Zumwalt, K. W., *Integrated Structural Shape Optimization Using Second-Order Sensitivity Analysis and Optimization Methods.*, Ph.D. Dissertation, University of Missouri, 1992.

[9] Vanderplaats, G. N., *A Robust Feasible Directions Algorithm for Design Synthesis*, 24st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 1983.