8th AIAA/USAF/NASA/ISSMO
Symposium on Multidisciplinary
Analysis and Optimization
6-8 Sept. 2000 Long Beach, CA

A00-40046

AIAA-2000-4769

Shape Optimization using ABAQUS and VisualDOC

S. Candan*

Laboratoire de Recherches et Applications en Mécanique Avancée, Institut Francais de Mécanique Avancée, Aubière, F-63175 FRANCE

J. Garcelon†, V. Balabanov† and G. Venter†

Vanderplaats Research & Development, Inc. Colorado Springs, CO 80906

ABSTRACT

A procedure for structural shape optimization using nonlinear structural analysis commercial software ABAQUS[1] is described. Domain elements are used to define regions of shape change in the structure, while basis vectors are employed to specify allowable shape changes in the structure. Theoretical background behind both domain elements and basis vectors are discussed. Shape optimization is performed using the commercially available, general-purpose software package VisualDOC[2]. A general procedure for coupling ABAQUS and VisualDOC is discussed in detail. The proposed methodology is demonstrated using two example problems.

INTRODUCTION

Shape optimization of mechanical structures gives designers greater flexibility than traditional sizing optimization. Shape optimization is also directly applicable to components modeled using solid elements, where sizing optimization is not used. The performance gains from applying shape optimization can be significant.

There are currently only a limited number of commercial structural optimization programs that employ shape optimization. GENESIS[3] and MSC/NASTRAN[4] are two such programs. In the area of nonlinear structural analysis, designers have shied away from employing optimization to these problems because of the perceived high computational cost. In recent years, computational power has increased and the robust nature of optimizers and response surface approximations has made using optimization with nonlinear analysis feasible[5, 6, 7, 8, 9, 10].

* *Intern at Vanderplaats Research & Development, Inc.*
† *Senior Research and Development Engineer, Member AIAA*

BASIS VECTOR APPROACH

Shape optimization generally requires that alternative shapes for the structure be specified. For finite element analyses (FEA) like ABAQUS, these shapes may be represented by a set of displacement vectors, which are known as basis vectors and form the design space. The optimizer then must find a linear combination of these basis vectors that represents the optimal design.

Figure 1 shows a rectangular structure with the initial shape and a basis vector for computing the length. The length of the structure is a design variable. The side constraints on the length design variable define the possible length range.

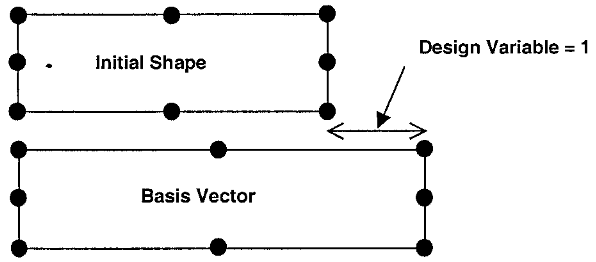


Figure 1: Initial Shape and Basis Vector

The vector difference between a basis vector and the initial shape is the perturbation vector. The equations for calculating the new node locations are given in equation (1).

$$
\begin{aligned}
X_i &= X_i^0 + \sum_j DV_j \left(XB_{ij} - X_i^0\right) \\
Y_i &= Y_i^0 + \sum_j DV_j \left(YB_{ij} - Y_i^0\right) \\
Z_i &= Z_i^0 + \sum_j DV_j \left(ZB_{ij} - Z_i^0\right)
\end{aligned}
\quad (1)
$$

where, X_i, Y_i, and Z_i, are the updated rectangular node coordinates of node i; X^0_i, Y^0_i, and Z^0_i are the initial rectangular node coordinates of node i; XB_{ij}, YB_{ij}, and ZB_{ij} are the components of the j^{th} basis vector corresponding to node i; DV_j is the value of the j^{th} design variable.

If the design variable takes on the value of 0, then the length would remain unchanged. When the design variable takes on a value between 0 and 1, the new shape is an interpolation of the original node locations and the basis vector. When the design variable value is less than 0 or greater than 1, then the new shape is an extrapolation of the basis vector.

The creation of basis vectors can be time consuming. One approach is to create special load cases. The resulting displacement fields are then added to the original node locations to create the basis vectors. This approach is used in references 3, 4, and 9. This approach, however, requires that the structure be solved to generate the basis vectors. For nonlinear analyses, this is not an attractive approach because the solution may be computationally expensive in comparison with linear solutions. Furthermore, designers have less control over the resulting basis vectors for highly nonlinear responses because of the potentially large deformations.

Another approach to creating basis vectors is based on the original geometrical features (i.e., CAD data) used to generate the mesh. In this method, the geometrical features are perturbed and a new mesh created. Each new mesh represents a basis vector. The major disadvantage of this approach is that there may not be a one-to-one correspondence between each generated mesh.

A third approach to creating basis vectors is based on applying perturbations directly to the original mesh based on pseudo-geometrical features. Examples of using this approach age given in reference 10 and a description of its implementation in GENESIS can be found in reference 11. This approach has several advantages over the other methods. First, it works directly with a given mesh and does remeshing. Second, it does not require CAD features. Third, it does not require solutions to the system of equations. Fourth, it does not have Possion's ratio effects that the first method can exhibit. One of the keys to this approach is the implementation of the pseudo-geometrical features. This is discussed in the next section.

DOMAIN ELEMENTS

Domain elements implement the pseudo-geometrical features for generating the basis vectors. Domain elements define node interpolation functions that the optimizer may scale to generate optimum shapes. Domain elements are not added to an ABAQUS model, so they do not increase the model complexity nor solution time. The basic idea is that domain elements act as "rubber bands" to stretch or contract their boarders. The "interior" nodes of domain elements then move in proportion to the movement based on the shape functions of the domain element.

Similar to isoparametric finite elements, domain elements have shape functions that define a displacement field within the element. The definition of a domain element consists of its shape type, "corner nodes", and "interior nodes". Corner and/or mid-side perturbation vectors are applied to generate the basis vectors. One of the biggest advantages of using this approach to generating shape basis vectors is that any FEA pre-processor that can generate elements and node meshes could generate domain elements.

The shape type defines the interpolating shape functions. The corner nodes identify the geometric scope of the domain element and locations where perturbations may be applied. It is important to note that the corner nodes do not have to be nodes used in the analysis. The interior nodes are analysis nodes that will be interpolated by the domain element's perturbation based on the shape functions. Table 1 provides a description of each shape type available for VisualDOC.

Table 1: Domain Types

Domain Type	No. Shape Functions	Characteristics
RBE2	1	rigid
BAR	2-3	1D elastic
TRIA3	3-6	2D elastic
QUAD4	4-8	2D elastic
BARX	2-3	axisymmetric elastic
TRIAX6	3-6	axisymmetric elastic
QUADX4	4-8	axisymmetric elastic
TETRA	4-10	3D elastic
PENTA	6-15	3D elastic
HEXA	8-20	3D elastic

References 3 and 11 provide more details about the domain element approach.

To construct the basis vectors, a user must define a domain element including the type, corner nodes and interior nodes, along with one or more perturbation vectors. Users may apply perturbation vector(s) at the corner or mid-side nodes. The procedure as described in reference 11 can be summarized in three steps: Inverse Mapping, Perturbation Calculation, and Basis Vector Generation.

Inverse Mapping

This step finds the parametric coordinates of each interior node of the domain. This may be expressed as shown in equation 2.

$$
\begin{aligned}
F_{xi} &= X_i - \sum_j X_j N_j(\xi_i, \eta_i, \zeta_i) \\
F_{yi} &= Y_i - \sum_j Y_j N_j(\xi_i, \eta_i, \zeta_i) \\
F_{zi} &= Z_i - \sum_j Z_j N_j(\xi_i, \eta_i, \zeta_i)
\end{aligned} \tag{2}
$$

where, X_i, Y_i, and Z_i are the coordinates of node i; X_j, Y_j, and Z_j are the coordinates of the j^{th} corner node; N_j are the shape functions of the domain type, and ξ_i, η_i, and ζ_i are the parametric coordinates of node i. The Newton-Raphson method is used to find ξ_i, η_i, and ζ_i such that $F_{xi} = F_{yi} = F_{zi} = 0$.

Perturbation Calculation

This step calculates the perturbation of each interior node of the domain using the domain's shape functions (equation 3).

$$
\begin{aligned}
P_{xi} &= \sum_j P_{xj} N_j(\xi_i, \eta_i, \zeta_i) \\
P_{yi} &= \sum_j P_{yj} N_j(\xi_i, \eta_i, \zeta_i) \\
P_{zi} &= \sum_j P_{zj} N_j(\xi_i, \eta_i, \zeta_i)
\end{aligned} \tag{3}
$$

where, P_{xi}, P_{yi}, and P_{zi} are the node perturbations, and P_{xj}, P_{yj}, and P_{zj} are the perturbation vector components applied to the corner and mid-side nodes. Different perturbations may be calculated for one domain by applying different sets of perturbation vectors at the corner and mid-side nodes.

Basis Vector Generation

This step adds the perturbations to each corresponding interior and corner node (equation 4).

$$
\begin{aligned}
XB_i &= X_i + P_{xi} \\
YB_i &= Y_i + P_{yi} \\
ZB_i &= Z_i + P_{zi}
\end{aligned} \tag{4}
$$

where, XB_i, YB_i, and ZB_i are the basic vector components at node i.

Domain Usage

Users define domain elements after creating the ABAQUS model. Either existing nodes or fictitious nodes that exist solely for the generation of the basis vectors may define the domain element. Once users have defined a domain element, they apply perturbation vectors to corner and/or mid-side nodes of the domain element. Users also need to define interior nodes of the domain element. Here, an ABAQUS model pre-processor is almost a necessity, particularly when using a complex ABAQUS model with multiple domain elements.

Using domain elements provides a great deal of flexibility and power to designers. A single domain element may surround the entire structure or designers can use multiple domain elements that cover only those parts of a structure to be considered for shape optimization since each domain defines a region where shape optimization applies. For those nodes that are referenced by more than one domain, the perturbations for that node are the average of the node's perturbation calculated using equation 3 for each domain element before applying equation 4.

The next section describes how ABAQUS and VisualDOC work together to perform shape optimization.

METHODOLOGY

Figure 2 shows the operational flow of VisualDOC and ABAQUS for shape optimization. The gray-shaded region shows what VisualDOC does. Each block of the flowchart is described next.

Create ABAQUS Model

Users can create the ABAQUS model using their traditional means. Typically this is done using some pre-processor, which automatically meshes the artifact and creates the necessary elements, etc. of the ABAQUS model.

Define Domain Elements and Perturbation Vectors

Creating the domain elements and perturbation vectors is best-done using the pre-processor the user created the ABAQUS model with. Currently, users can create domains and perturbation vectors using IDEAS and MSC/PATRAN. These pre-processors and modelers output the domain element and perturbation vector data for processing by VisualDOC. Alternatively, users can specify domain elements and perturbation vectors using a defined format once they have the ABAQUS model.

Generate the Basis Vectors

Next, users start VisualDOC and specify the ABAQUS model file and the domain elements and perturbation vector data file. VisualDOC's DVShape module then generates the associated basis vectors

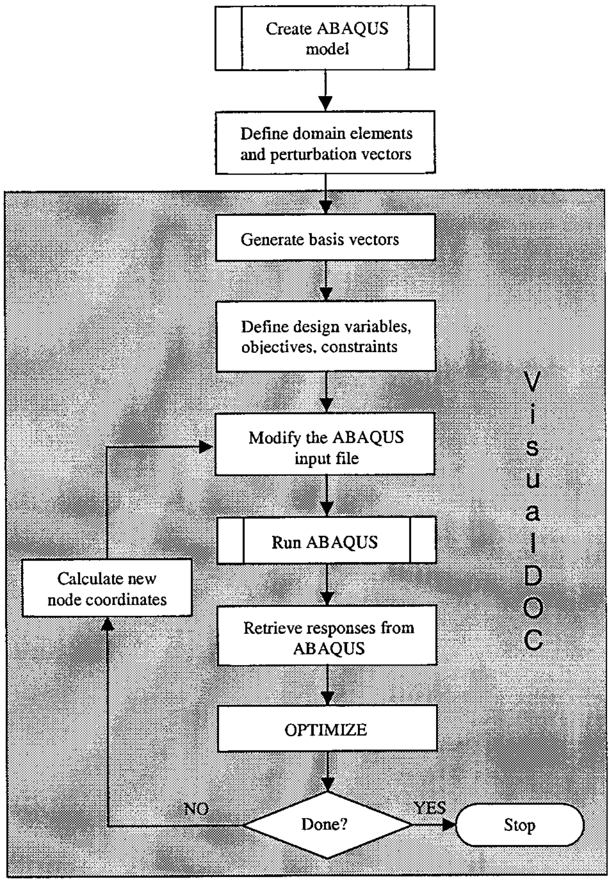


Figure 2: VisualDOC/ABAQUS Shape Optimization

using equations 2 - 4. The basis vectors are retained for use later in the optimization.

Define Design Variables, Objectives, and Constraints

The next stage is for users to create the optimization data. The VisualDOC GUI module provides a convenient and powerful tool for defining design variables, side constraints on the variables, constraints, and design objectives. Here, users must establish the relationship between basis vectors and design variables.

VisualDOC includes other relationships between design variables and design responses that makeup the objectives and constraints and the ABAQUS input file and results files. These relationships allow VisualDOC to modify the ABAQUS input file as the design progresses and to retrieve ABAQUS results from which VisualDOC calculates the design constraints and objectives.

Modify the ABAQUS Input File

This step is at the top of the optimization loop. Initially, VisualDOC does not need to modify the original ABAQUS input file; however, each cycle in the design processes generally does include some modifications to the ABAQUS input file.

VisualDOC uses several Perl scripts to interact with ABAQUS. The Perl script that modifies the ABAQUS input file is called "mogrify". "mogrify" takes a template file that describes the layout of the ABAQUS input file and a definition file that identifies specifically what needs to be modified in the input file and modifies the ABAQUS input file accordingly. The template file is an ASCII file, is included with VisualDOC, and users can extend and edit it. The definitions file provides the relationship between design variables and ABAQUS input file values. When users define design variables in VisualDOC, they also can define what values in the ABAQUS input file these correspond to.

For shape optimization, VisualDOC will actually change the node locations in the ABAQUS input file based on one or more design variable values that VisualDOC maintains. That is, shape design variables are not explicitly entered into the ABAQUS input file, but the optimized shape is implicitly characterized by modifying the node locations. A later step in the flowchart addresses this.

Run ABAQUS

At this point, VisualDOC has modified the ABAQUS input file for the current design variable values and current node locations that reflect the shape. VisualDOC starts a single ABAQUS run and waits for it to complete.

Retrieve Responses from ABAQUS

VisualDOC uses two Perl scripts that retrieve ABAQUS results. These work in a similar fashion to "mogrify" in that they require a template file that describes the general form of the file where the results are located and they requires a definitions file that specifies which responses to retrieve for the current run. These two Perl scripts are called "grok" and "aba-grok". The difference between the two scripts are that "grok" is a general-purpose script that may be used with different solvers and ASCII output files, whereas "aba-grok" works only with an ABAQUS binary, output database file. The advantage of using "aba-grok" is that the responses are retrieved as double precision, floating point values. Using double precision values during optimization reduces the influence of numerical noise and improves the stability and robustness of the optimizer.

The template files for "grok" and "aba-grok" are included with VisualDOC. These are ASCII files that users can modify. The definition files identify which specific responses are needed for optimization. When users define design responses in VisualDOC, they also can define what responses from ABAQUS are related to the design responses.

Optimize

Given the design responses, VisualDOC calculates the constraint values and objectives. Depending on the optimization algorithm VisualDOC is using, VisualDOC then creates a new set of design variable values for which it requires ABAQUS to compute responses.

Calculate New Node Coordinates

For shape optimization problems, VisualDOC must convert the perturbation vector magnitudes, which are design variables, into new node coordinates. This process is done in this stage. Once this module calculates the new coordinates, these are passed to "mogrify" along with any other design variables so VisualDOC can get another set of design responses from ABAQUS. This process is demonstrated in an example problem.

EXAMPLES

Shape Optimization of a Spinning Disk

As a first example of this technique, we will optimize a spinning disk for minimum mass with displacement, frequency, and stress constraints. The original disk model is shown in Figure 3. The disc is modeled as an axisymmetric structure with the y-axis being the axis of rotation. The disk is composed of two different materials. The outside parts are made of aluminum whereas the core is steel. The disc is subjected to a centrifugal load produced by the rotation of the disc (12 Hz). The disc is clamped along the edge x=0. The maximum displacement is limited to ±0.2 mm. The maximum Von Mises stress is limited to 200 MPa, and the lowest natural frequency must be greater than 50 Hz. The top aluminum face must remain 2 mm thick.

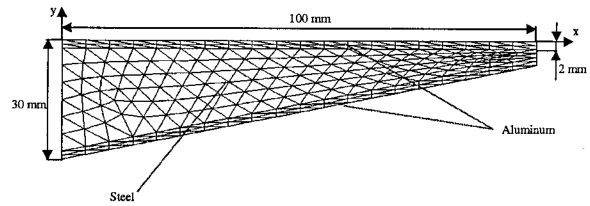


Figure 3: Original Disk Shape and Model Definition

Figure 4 shows the single domain element and three perturbation vectors used to generate the basis vectors. The domain element is an axisymmetric four-node quad element. The heavy dark lines define the outline of the domain element. The perturbation vectors are applied at two corners and one mid-side node of the domain element. In this example, we used existing structural nodes to define the domain or that portion of the structure that we could design. Since the top layer of aluminum must remain 2 mm thick and parallel with the x-axis, the domain element does not include that layer nor does it have any perturbation vectors applied.

In this example, the enclosed shape of the domain element defines the part of the structure that we are allowing to change. Since the ends of the disk must remain square with respect to the x-axis, those perturbation vectors (Vector 1and Vector 3) are vertical. The slight angle on Vector 2 allows for more flexibility on the final shape such that a nonlinear taper is possible.

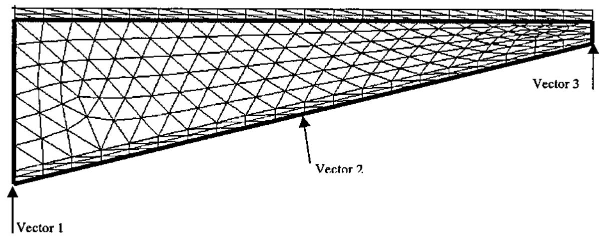


Figure 4: Axisymmetric Domain and Perturbation Vectors

There are three design variables. These are the magnitudes of the perturbation vectors, Vector 1, Vector 2, and Vector3 (Figure 4). We ran this problem using the Modified Method of Feasible Directions (MMFD) algorithm in VisualDOC. Figure 5 shows the final shape. Table 2 summarizes the results.

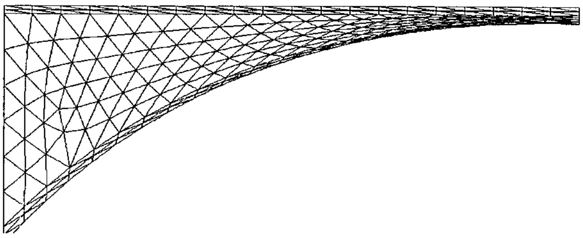


Figure 5: Optimal Shape

Table 2: Shape Optimization Results for the Disk Problem

Design Variable/Response	Initial Value	Optimum
Vector 1	0	-3.52
Vector 2	0	14.187
Vector 3	0	1.99
Displacement	-0.16 mm	-0.201 mm
Stress	200.75 MPa	117.93 MPa
Frequency	77.743 Hz	87.705 Hz
Mass	2.8211 N	2.0892 N
No. of Design Cycles		4
No. ABAQUS Calls		49

Shape optimization reduced the mass of the disk, from 2.8211 to 2.0892, a reduction of almost 26%. Initially, only the stress constraint is active. This occurred along the fixed edge at x = 0. The initial value of the displacement was -0.16, and at the optimum the displacement constraint was active. Because the disk takes on a thinner profile along the free edge accounts for the increased displacement. Even though the frequency constraint was never active the lowest natural frequency of the disk increased. This can be attributed to the lower mass of the disk. The reduction in mass also decreases the maximum stress in the disk since the centrifugal load is proportional to the mass.

Shape Optimization of a Pin

Another example of using shape optimization with ABAQUS is optimizing the shape of the interface between a pin, some epoxy adhesive, and anchor (Figure 6). The model is axisymmetric. The epoxy adhesive connects the pin to the anchor. Once the epoxy cures, the pin will carry load in tension at its free end. Mechanical resistance between the three parts prevents the pin from sliding out of the anchor. The optimization problem is to change the shape of the walls in the anchor to minimize the maximum stress in the anchor. The stresses in the pin, the anchor, and epoxy adhesive should not exceed the limits for the corresponding materials. We also applied one geometrical constraint: the inner diameter of the anchor hole should not exceed the outer diameter of the pin, because pin must be inserted into the anchor.

Figure 6 shows the domain elements and Figure 7 shows the perturbation vectors used to generate the basis vectors. The domain elements are four-node quad elements. The perturbation vectors are applied only at the two corner nodes of each domain element. Similar to the first example, we used existing structural nodes to define the domain.

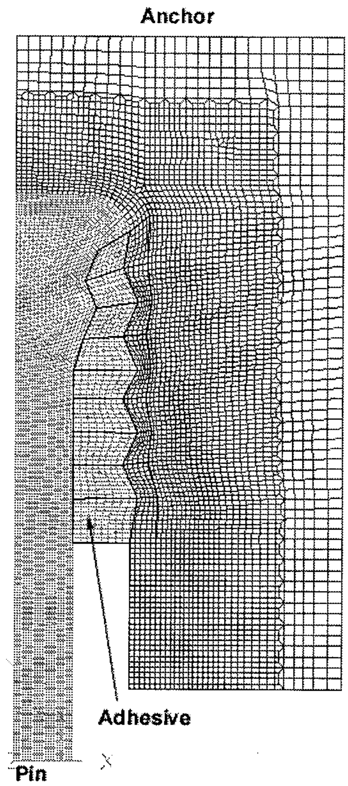


Figure 6: Domain Elements for the Pin Problem.

There are nine design variables in this problem. These are the magnitudes of the perturbation vectors (Figure 7). We ran this problem using the Response Surface Approximate optimization technique in VisualDOC. Figure 8 shows the stress contours in the solid material for initial and final shapes. Table 3 summarizes the results.

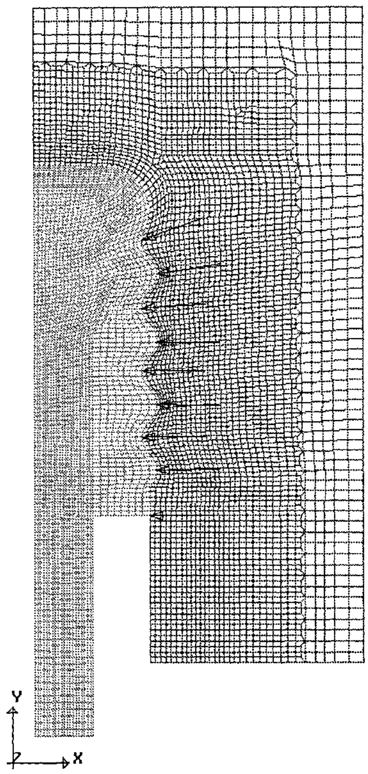
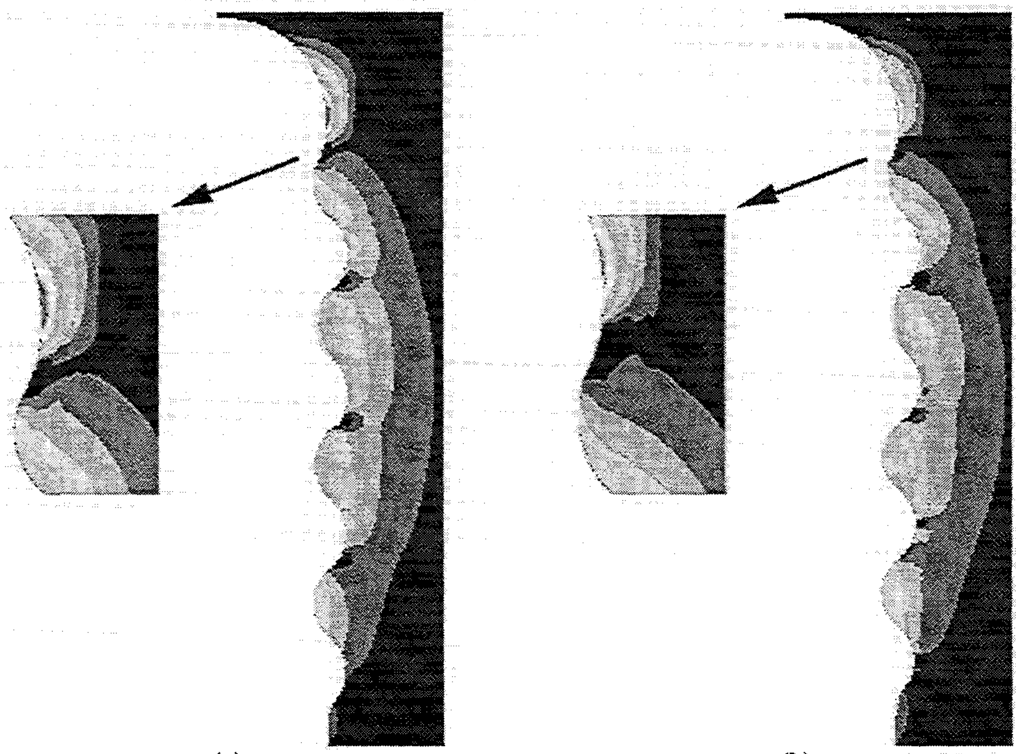


Figure 7: Perturbation Vectors for the Pin Problem.

Table 3: Shape Optimization Results for the Pin Problem

Design Variable/Response	Initial Value	Optimum
Vector 1	0	0.1576
Vector 2	0	0.0332
Vector 3	0	0.1558
Vector 4	0	-0.0275
Vector 5	0	0.2226
Vector 6	0	-0.1667
Vector 7	0	-0.3578
Vector 8	0	-0.1791
Vector 9	0	-0.3341
Normalized maximum stress	1.0	0.8885
No. Design Cycles		86
No. ABAQUS Calls		97

Although the shape of the walls did not change much, the stress distribution was improved as a result of the shape optimization (Figure 8). Comparing stresses for initial shape and for the final shape we notice that the stresses are more evenly distributed for the final shape. In addition, the magnitude of the maximum stress was reduced as a result of the optimization.



(a) (b)

Figure 8: Stress Contours for Initial (a) and Final (b) Shapes of the Pin Problem

CONCLUDING REMARKS

The methodology for structural shape optimization with ABAQUS and VisualDOC was developed and demonstrated using two example problems.

It was shown that domain elements and basis vectors could be efficiently used not only as a internal part of structural optimization codes, but as an external means to apply shape optimization to nonlinear structural analysis software in general.

VisualDOC demonstrated itself as a flexible tool to integrate nonlinear structural analysis with shape optimization.

REFERENCES

1 *ABAQUS,* Version 5.8, Theory Manual, Hibbitt, Karlsson, & Sorensen, Inc. 1080 Main Street, Pawtucket, RI 02860-4847, 1998.

2 *VisualDOC,* Version 2.0, Getting Started Manual, Vanderplaats Research & Development, Inc., 1767 S. 8th Street, Colorado Springs, CO 80906, 2000.

3 *GENESIS* Structural Optimization Software, Version 5.0 User's Manual, VMA Engineering., Colorado Springs, CO, USA, 1998.

4 Kodiyalam, S., Vanderplaats, G. N., Miura, H., "Structural Shape Optimization with MSC/NASTRAN", AIAA/ASME/ASCE/AHS/ASC 31st Structures, Structural Dynamics, and Materials Conference, Long Beach, CA, pp. 150 - 160, April, 1990.

5 Ghosh, D. K. and Vanderplaats, G. N., "Development of a Design Optimization Interface to ABAQUS," Presented at the Regional ABAQUS Users' Meeting, Los Angles, CA, September, 1997.

6 Thomas, H., "Optimization of Structures Designed Using Nonlinear FEM Analysis," AIAA/ASME/ASCE/AHS/ASC 37st Structures, Structural Dynamics, and Materials Conference, Salt Lake City, UT, pp. 604 - 608, April, 1996.

7 Wright, E. and Grandhi, R. V., "A Shape Optimization Technique for Controlling Deformation Parameters in Forging," 38st Structures, Structural Dynamics, and Materials Conference, Kissimmee, FL, pp. 1404 - 1414, April, 1997.

8 Botkin, M. E., "Shape Optimization with Stress and Buckling Constraints," 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, FL, pp. 375 - 383, September, 1994.

9 Zhang, J., "Modified Thermal Loads Approach for Automatic Generation of Basis Vectors in Shape Optimization," 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, FL, pp. 384 - 390, September, 1994.

10 Thanedar, P. B., Sankaranarayanan, S., and Chirehdast, M., "Topology and Shape Optimization for Product Design," AMD-Volume 227. Design Optimization in Industry, ASME, pp. 163 - 175, 1997.

11 Leiva, J. P. and Watson, B. C., "Automatic Generation of Basis Vectors for Shape Optimization in the GENESIS Program," 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, pp. 1115 - 1122, September, 1998.