Shape Optimization of Three-Dimensional Folded-Plate Structures

M. E. Botkin* and J. A. Bennett†
General Motors Research Laboratories, Warren, Michigan

The basic features of a previously described shape optimization capability included a general geometric problem description format and adaptive finite element analysis. The finite element mesh for each design step was created from the boundary information only and refined using finite element solution results. At that time, only structures in a single plane could be treated. This work has now been extended to handle structures created from an assembly of segments that may be rotated into any plane as well as to contain the nonplanar surface curvature within any segment. In order to handle these more general problems, it was necessary to introduce a more efficient mesh generation scheme. Several design examples are presented to demonstrate each of the features.

Introduction

IN a previous paper,[1] an approach to shape optimization was described that involved a boundary problem description format, an automatic triangulation scheme for creating the finite element mesh, and an adaptive mesh refinement scheme for solution accuracy. This shape optimization process involves a design description of the part based on key nodes and boundary design elements (Fig. 1). Thus, no information about the interior of the structure is available. An automatic finite element mesh generation scheme is used to produce an initial uniform triangular mesh (Fig. 2). After analysis, the mesh is automatically refined, based on identifying regions of high strain energy density gradients (Fig. 3). This approach is integrated with an optimization program to produce a shape design program that will find a minimum mass shape satisfying the displacement and stress constraints.

In the previous work, examples were limited to thin two-dimensional structures. Reference 1 suggested that this two-dimensional scheme could be extended to nonplanar structures such as sheet metal stampings or perhaps castings, which have large planar or nearly planar segments and can be modeled by a single bending-membrane finite element through the thickness.

It is convenient to think of three distinct forms of nonplanar thin structures. The first of these structures, for example, can be described by a mathematical transformation from a simple flat surface (Fig. 4a) into a cylindrical surface. Second, the surface may take the form of a general shallow shell that may not be obtained from a simple mapping relationship (Fig. 4b), but can be obtained by projection. Third, the structure may be made up of several segments that may be either planar or one of the two previously mentioned forms (Fig. 4c). In each of these forms, the ideas discussed in Ref. 1 can be used in the planar form to describe the segments, generate the mesh, and carry out the refinement. Each of these forms will be described in turn.

Mesh Generation and Refinement

Mesh Generation[2]

Many authors have stressed the need for automatically modifying the mesh as the structure changes shape, but it was observed that the commonly used mesh generation techniques, which are based upon coarse isoparametric or transformal mapping patches, imposed limitations on the ability to treat large variations in shape. While these techniques do redistribute interior nodes as the boundaries move, the aspect ratios tend to become objectionably large as the shape becomes significantly different from the initial shape. Mesh grading and solution accuracy are also difficult to control.

As an alternative to more traditional mesh generation methods, the use of fully automatic mesh generation based only upon boundary points coupled with adaptive refinement has been proposed. Given a set of uniformly spaced boundary points, this technique is capable of generating a nearly uniform initial mesh of triangular elements. Thus, as the design changes, uniform triangular meshes can be recreated at any time.

After the design model has been created, the boundaries are automatically discretized into uniform segments called the characteristic lengths (CL), which is an input value. Automatic triangulation is used to create a nearly uniform mesh from the set of boundary points and points placed uniformly throughout the region's interior having approximately the same density as the boundary points. This process of creating the uniform mesh is repeated at each step in the design for which a new set of boundary points has been generated.

Adaptive Mesh Refinement[2]

Unlike the design of fixed-configuration structures, it is not possible to assure the accuracy of the mesh as the shape changes, since the accuracy of various portions of the mesh will change. The ideas of adaptive mesh refinement can be incorporated to help resolve this difficulty.

The mesh refinement process is based upon the variation in strain energy density (SED) as a measure of the error in an element. In order to show this, consider the following relationship, which expresses the error in a solution quantity in an element in terms of the variation in that quantity:

$$|V - V_i| < C_j H_e^k |D^k V|_e$$

where V is some exact solution quantity (stress or SED), V_i the computed value, C_j a proportionality factor, H_e^k a

Received Aug. 16, 1984; revision received Feb. 19, 1985. Copyright © American Institute of Aeronautics and Astronautics, Inc., 1985. All rights reserved.

*Senior Staff Research Engineer, Engineering Mechanics Department.

†Assistant Head, Engineering Mechanics Department.

measure of the element size, and D^k a differential operator of order k (k is the order of the element). For constant strain elements ($k=1$), the error is proportional to the element size times the first variation in the solution quantity.

An example of the steps involved in the mesh refinement process can be seen in Fig. 3. A contour plot of the SED variation for the structure in Fig. 2 is shown in Fig. 3a. The elements contained within the dotted areas have undesirably high SED variations, which were selected using the following expression:

$$CV_r = \overline{\Delta E} + \beta_r(\Delta E_{\max} - \overline{\Delta E})$$

where CV_r is the SED difference cutoff value for contour r, ΔE the average SED variation for all elements, $\Delta E_{\max}$ the maximum SED variation in any element, and β_r a parameter to be selected to control the size of the contour regions. Each original element in which the SED variation exceeds CV_r will be refined by adding nodes. Once nodes have been added to all of the required elements, the final mesh as shown in Fig. 3b will be produced by retriangulating the resulting nodes.

Transformed Surfaces

The most straightforward approach to out-of-plane surface distortion is to assume some sort of shape function. Generally, if the more classical isoparametric or transformal mapping mesh generation approaches are employed, out-of-plane mapping is implicit in the formulation. These approaches, however, assume the segment is either quadrilateral or can be easily broken up into quadrilaterals. If one assumes the segments to be perfectly general arbitrary regions, it is more efficient from the modeler's point of view to use free-form mesh generation.[2] In this case, the transformation is limited to a simple cylindrical or spherical surface; examples of both were presented for single segments in Ref. 2. Only the cylindrical functions have been implemented for the multisegment case as described here. Figure 5 shows the basic parameters for definition of a cylindrical surface. Nodes N_1 and N_2 are created to locate the axis of revolution. The parameters of the shell are h and r_0, from which the radius is

$$R = (h^2 + r_0^2)/2h$$

If X_i is the local coordinate with respect to $\overline{N_1N_2}$, then the transformed local coordinates are

$$X_i' = R\sin(X_i/R)$$

$$Z_i' = R\cos(X_i/R)$$

Note that this will create a gap at each end of the total structure, which must be accounted for in the assembly process to be discussed later. It is also possible to rotate the shell about its axis of revolution.

At present, this transformation will create only fixed geometries, i.e., neither h nor r_0 can be design variables. It is possible, however, to create variable-sized cutouts that will move along the surface.

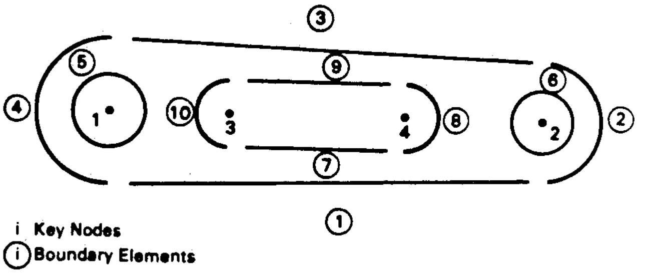


Fig. 1 Boundary elements for slotted torque arm.

Shallow Nonplanar Segments

If it is assumed that a segment such as that shown in Fig. 6 deviates only slightly from a reference plane, a finite element mesh generated on the flat surface will not be significantly distorted when merely projected onto the distorted surface. Thus, the mesh may be generated from the boundary description of the segment, including any dependence on design variables. Then, if a representation of the surface is available, which may also depend on the design variables, the out-of-plane coordinates of the mesh points may be projected to the surface by a simple interpolation

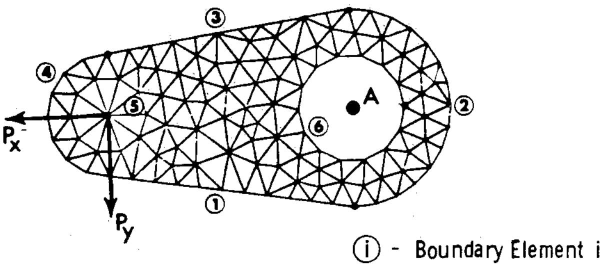
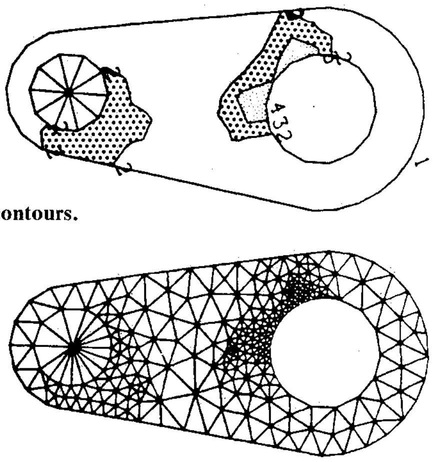


Fig. 2 Initial uniform mesh.



a) SED difference contours.

b) Refined mesh.

Fig. 3 Mesh refinement.

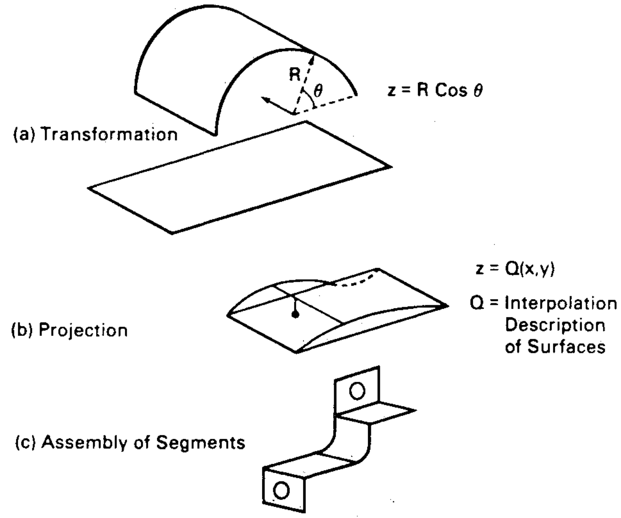


Fig. 4 Three forms of nonplanar structures.

scheme. This process is shown in the flow chart of Fig. 7. In order to describe the nonplanar behavior, a local coordinate system L_j on the surface is determined. Perpendicular to this direction, a set of design element curves H_{jk} is provided (Fig. 8). From this information, an interpolation scheme from the two-dimensional mesh must be developed.

Most general interpolation schemes require an equally spaced distribution of the points to be interpolated. Since the points to be interpolated are generated by the triangulation routine and may be refined, this condition cannot be met. However, the scheme proposed by Akima[3] does not impose such a requirement. This technique is a bivariate (two-variable) interpolation algorithm for smooth surface fitting. The fifth-degree polynomial interpolation included in this algorithm was retained even though the order was higher than necessary for our requirements, since our goal was not to study this effect but to include the ability to parameterize surface geometries using a surface interpolation scheme. It was felt that the additional computational cost associated with the more accurate interpolation was not significant enough to warrant modification of the algorithm. A set of x-y points must be provided to the interpolator with a corresponding set of known functional values Z_i. An additional set of x-y points must be provided for which the desired functional values will be computed by the interpolator.

The accuracy of this scheme can be improved if the node points are more uniformly distributed. The information available at this time in the surface description is comprised of a set of points along the exterior boundary segments at the characteristic mesh spacing CL and a set of points along the surface description H_{jk}, which may also be at a spacing of CL. However, the H_{jk} are, in general, spaced at a greater distance than CL in order to minimize the input data. This produces severely distorted triangles in the Akima interpolation. In order to resolve this problem, the interpolation mesh is enriched by passing a series of lines parallel to L_j at a spacing of CL and using a cubic spline interpolation with the boundary and surface points to create a set of points spaced at CL along this line. The out-of-plane position of the boundary points is determined by a linear interpolation between the intersection of the design surface and the edge of the segment. This scheme is shown in Fig. 9.

In order for the optimization to control the surface design, the design variables must be associated with the surface shape. Each surface design element H_{jk} can have up to six design variables, as shown in Fig. 10. The length of the element will be computed by the program, since the ends of the element terminate at the intersection with the boundary design element. Design variables a_1 and a_4 control the ends of the surface design element and, if these variables are nonzero, then the boundary of the part will also be distorted out of the plane. It is recognized that this element has some limitations and could be extended to a more general one. This element was used merely to demonstrate the overall capability.

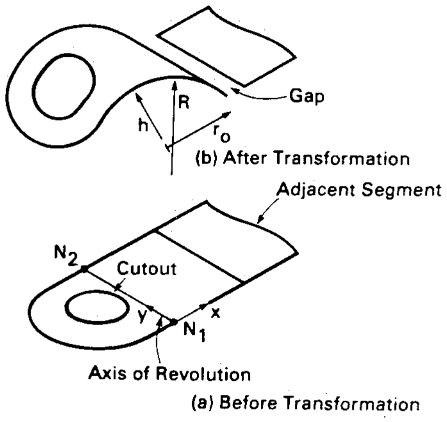


Fig. 5 Transformed surface.

Assembly of Planar Segments

The third step involves assembling the segments into a three-dimensional structure. Each of the segments discussed earlier (flat, shallow, or transformed) has been referred to a planar set of coordinates. Thus, a sequence of these segments can be assembled by allowing a rotation angle between each set of planar coordinates (Fig. 11). It is a fairly straightforward geometric process to rotate the points; however, some care is necessary to handle the connectivity and mesh refinement correctly. In addition, it may be necessary to translate segments in order to close any gaps that may exist due to the transformed surfaces discussed previously. Since each segment is generated separately from its own set of independently generated mesh points, there is a redundant set of mesh points along each line of rotation. However, since each joining boundary is generated from the same set of key nodes, they will have an identical nodal set. Therefore, one set may be discarded and the appropriate connectivities changed.

The refinements are also accomplished on the segment level as described in Ref. 2. Thus, the correct refinement on

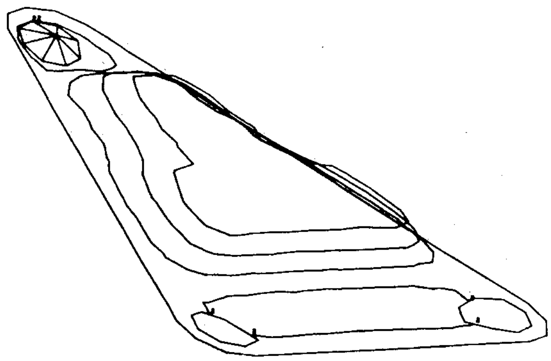


Fig. 6 Shallow nonplanar shell (contours of surface height).

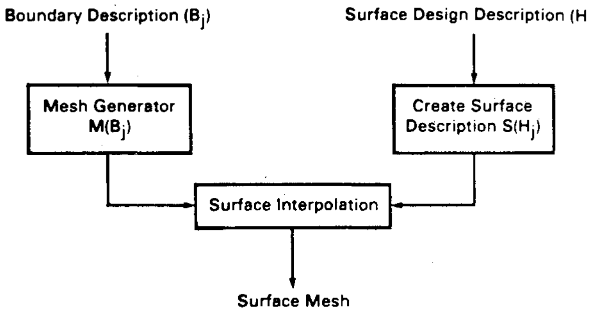

Boundary Description (B_j)
Surface Design Description (H
Mesh Generator M(B_j)
Create Surface Description S(H_j)
Surface Interpolation
Surface Mesh

Fig. 7 Flow chart for shallow surface creation.

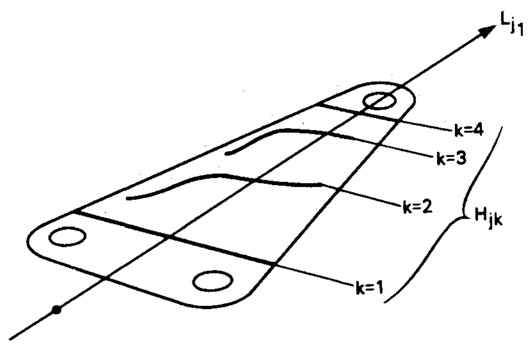


Fig. 8 Surface design elements.

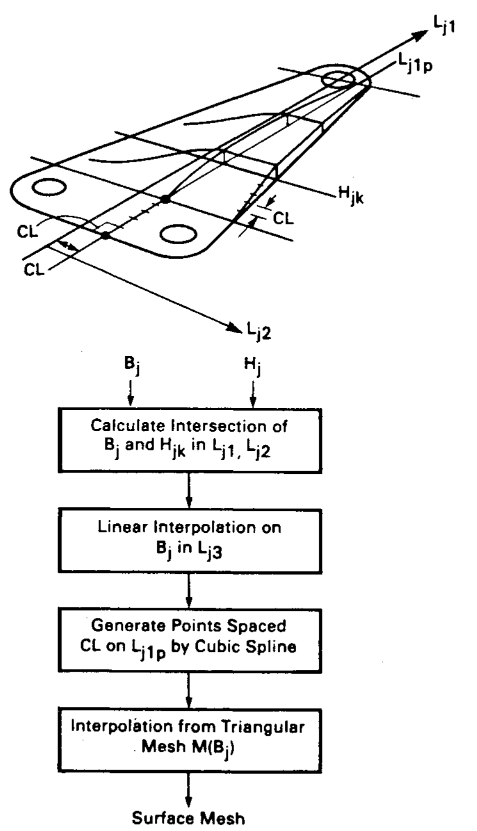

L_j1
L_j1p
H_jk
CL
CL
CL
L_j2
B_j
H_j
Calculate Intersection of B_j and H_jk in L_j1, L_j2
Linear Interpolation on B_j in L_j3
Generate Points Spaced CL on L_j1p by Cubic Spline
Interpolation from Triangular Mesh M(B_j)
Surface Mesh

Fig. 9 Addition of surface lines to improve interpolation.

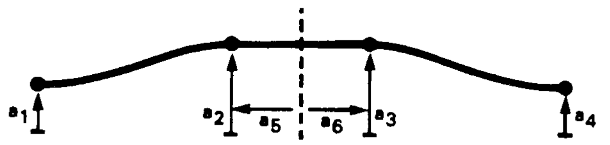


Fig. 10 Surface design element variables.

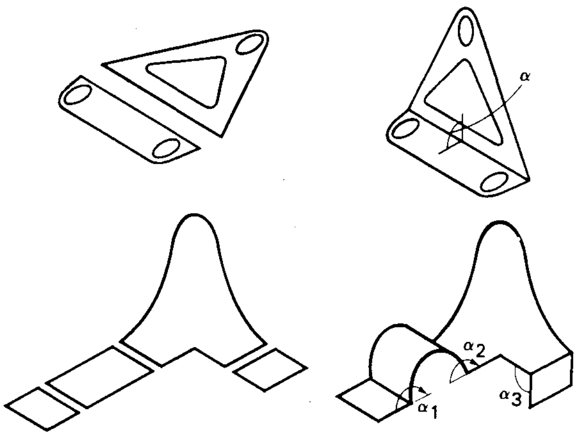


Fig. 11 Assembly and rotation of segments.

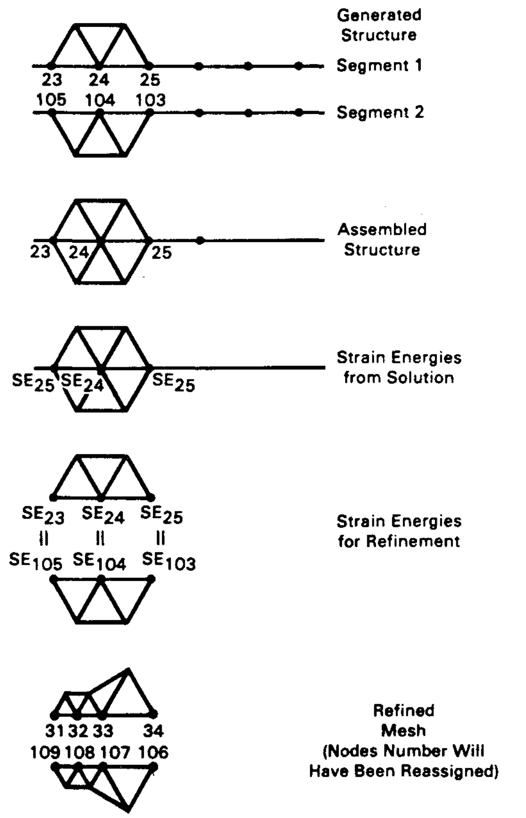


Fig. 12 Treatment of nodes on segment boundaries.

both segmented descriptions of a common boundary must be assured. As described in Ref. 2, the refinement along an edge of a triangular element is determined by the strain energy value associated with the nodes defining the edge. Thus, if the nodes on the boundary segments describing the connecting edges in the two segments have the same strain energies, the refinements will be identical and the scheme described previously can be used when the refined meshes are reconnected. This is accomplished by assigning the computed strain energies to the previously "discarded" nodes. An example of this process is shown in Fig. 12. The same schemes can be used to handle the case of nonrotated but joined edges (Fig. 13); however, this has not been implemented.

Increased Efficiency of Triangulation

Mesh generation is a major component of the shape optimization system. For the examples in Ref. 1, triangulation accounted for approximately 40% of the total CPU time or approximately equal to the time required for the finite element solution. For that reason, any efficiency gains in triangulation will have a significant effect on the total process.

A recently implemented triangulation technique[4,5] is based upon a mathematical construct called the Dirichlet tessellation. This construct is shown for two dimensions in Fig. 14. The open circles represent a set of points to be triangulated, $P_1, P_2, \ldots, P_n$. The convex polygons, shown in dotted lines, represent the set V_i, $1 \le i \le N$, in which all points are closer to P_i than any other point. This can be mathematically stated as

$$V_i = \{X\colon \|X-P_i\| < \|X-P_j\| \text{ for all } i \neq j\}$$

where $\|\cdot\|$ denotes the Euclidean distance in the plane. The boundaries of V_i (usually called a Voronoi polygon) are portions of the perpendicular bisectors of the lines joining P_i to P_j, where P_j is the node within an adjacent polygon. The assembly of polygons is referred to as the Dirichlet tessellation. The tessellation is now the basis for selecting the triangle connectivity; a node within a polygon is connected to every other node of the adjacent polygons. This produces a unique triangulation that has been proved to be as close to equilateral as possible for the given set of points.[6]

An important feature of this technique is the ease with which points can be added to an existing triangulation, as shown in Fig. 15a. It is merely necessary to identify the triangles that should be removed from the connectivity list represented by the shaded area. These triangles are those whose circumcircles (Fig. 15b) contain the new point. Finally, the new triangulation is obtained by connecting the new point to the vertices of the shaded plygon (Fig. 15c).

This technique has been found to be extremely efficient and has typically reduced the mesh generation times by more than an order of magnitude. The effect on the total system execution time has been on the order of 35%. In addition to the increased efficiency, this technique can also have a significant effect on the mesh refinement. Nodes can be added to an existing triangulation by merely re-establishing the connectivities in a localized area, as shown in Fig. 15. Due to the significant savings already gained by the new triangulation, this feature has not yet been implemented; but, it is planned for the near future.

Example Problems

Example 1—Nonplanar Plate

The structure in Fig. 16 represents a plate with design variables controlling the shallow, out-of-plane dimensions of the surface. A static load of 500 N is applied at point A in a direction normal to the *x-y* plane. In addition to stress constraints, a constraint is applied on the out-of-plane displacement at point A such that this constraint was active. A characteristic length (CL) of 0.75 cm was used to create the finite element mesh.

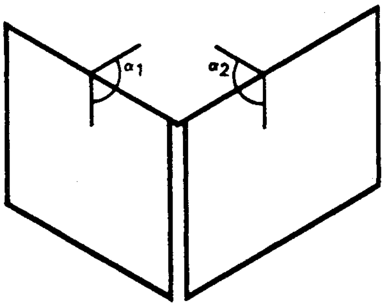


Fig. 13 Edges not joined by program.

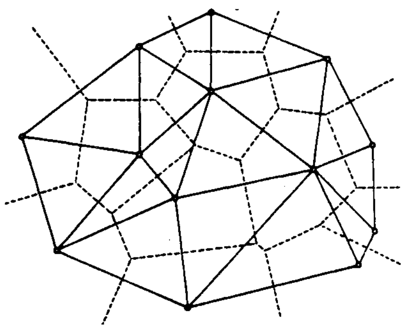


Fig. 14 Dirichlet tessellation (dashed lines) and Delaunay triangulation (solid lines).[4]

Seven design variables control the shape of the part. The size of the cutout is controlled by key node design variables. The *y*-coordinates of all three nodes and the *x*-coordinates of the bottom nodes are allowed to move. Linking is used to enforce symmetry. The out-of-plane dimensions are controlled by four design variables associated with the six line functions shown as dotted lines in Fig. 16. The values for a_1 and a_4 (Fig. 10) are zero; the values for a_5 and a_6 are shown; and the initial values for a_2 and a_3 are 1.0 cm. The remaining surface design elements (1, 2, 5, 6) are set to zero to ensure that the loading and attachment areas remain flat.

The iteration history is given in Fig. 17. After step 3, the displacement constraint is equal to the limiting value. Mass was further reduced, however, by increasing the out-of-plane stiffness so that the cutout could be increased in size. Some difficulty was encountered in steps 12-19 in following the displacement constraint, which became 9% violated in step 18. A recovery was made, however, in step 20 in which the surface design variables are in the range of 1.3 cm. In steps 20-40, the surface design variables increase to a range of 1.4-1.5 with a corresponding decrease in mass of only 2%. The initial design with the surface contours is shown in Fig. 18. The final surface design is shown in Fig. 19. Table 1 gives the design variable values for the initial and final designs.

Table 1 Design variable values, cm

Design variable number	Example 1			Example 2		
	Type[a]	Initial	Final	Type[a]	Initial	Final
1	$N^X_{(4)}$	1.00	2.0	t	0.3	0.3
2	$N^Y_{(4)}$	2.00	1.665	$R_{(11)}$	1.0	1.99
3	$N^Y_{(6)}$	5.50	11.0	$R_{(13)}$	1.0	1.641
4	t	0.30	0.3	$\mathrm{dc}_{(19)}$	0.657	0.493
5	$S_{(3)}$	1.0	1.427	$\mathrm{dc}_{(19)}$	0.10	0.099
6	$S_{(3)}$	1.0	1.428	$\mathrm{dc}_{(19)}$	5.136	3.853
7	$S_{(4)}$	1.0	1.359	$\mathrm{dc}_{(19)}$	11.412	9.497
8	$S_{(4)}$	1.0	1.5	$R_{(21)}$	1.0	1.466
9				$R_{(23)}$	1.0	1.243
10				$R_{(25)}$	1.0	2.20
11				$R_{(27)}$	1.0	1.638
12				N^X_A	23.5	20.871
13				N^X_B	20.5	19.401

[a] Numbers in parentheses refer to the element number, N^i = node (i coordinate), T = thickness, dc = double cubic, s = surface, and R = radius.

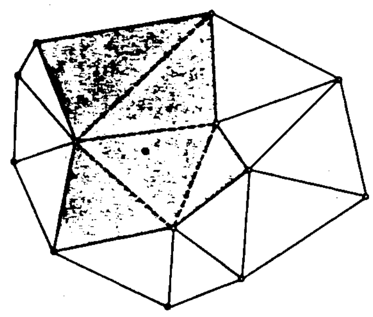


Fig. 15a Insertion polygon (shaded region).[4]

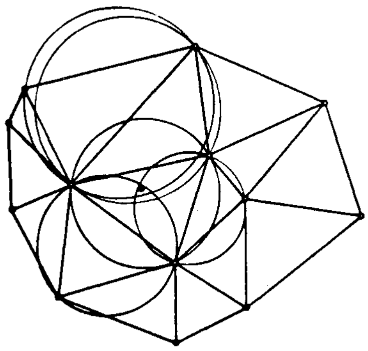


Fig. 15b Insertion of new point.[4]

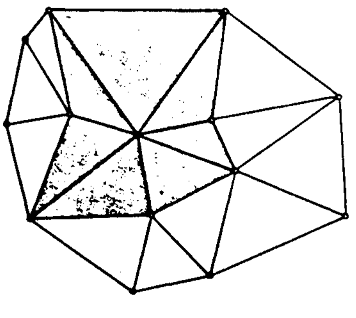
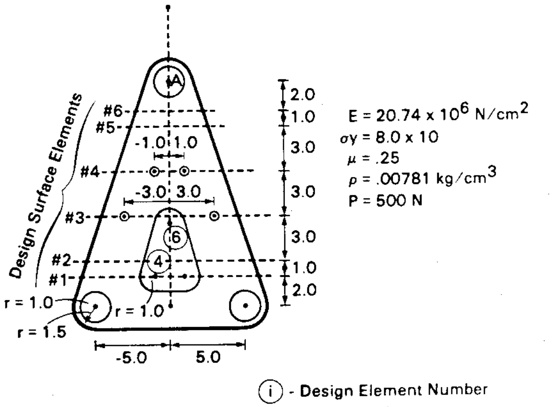


Fig. 15c Local triangulation (shaded region).[4]



(i) - Design Element Number
All Dimensions in cm

Fig. 16 Dimensions of example 1.

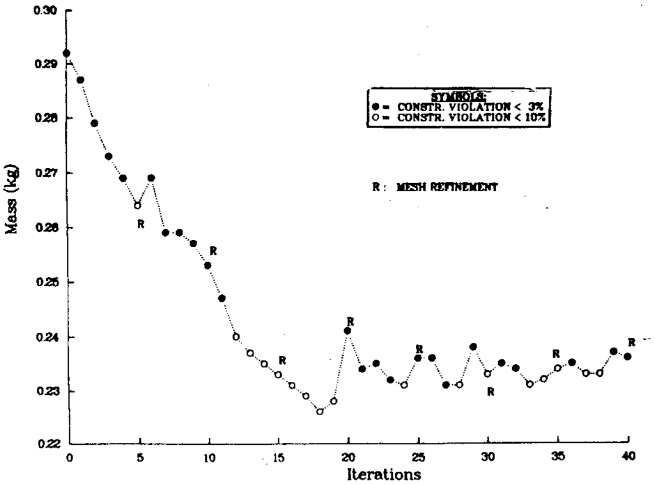


Fig. 17 Design history of example 1.

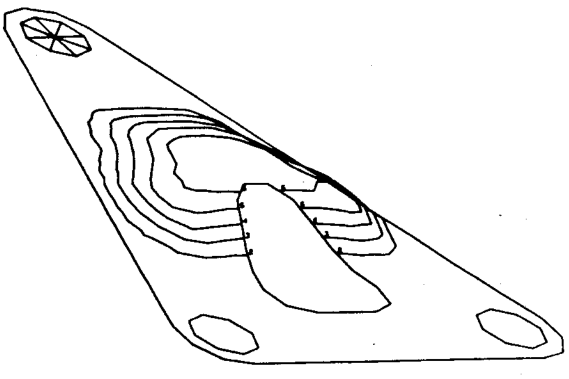


Fig. 18 Surface contour of example 1 (initial design).

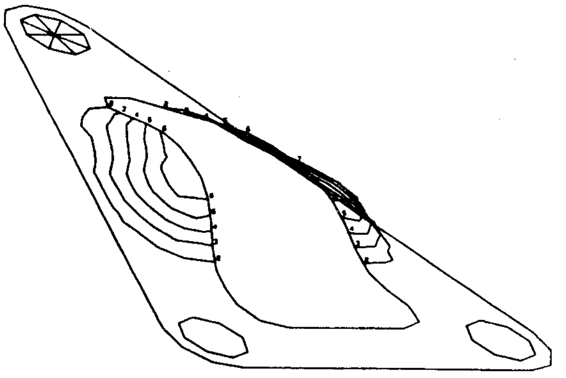


Fig. 19 Surface contour of example 1 (final design).

All Holes Have Radius of 1 cm
All Dimensions in cm
(i) - Design Element Number

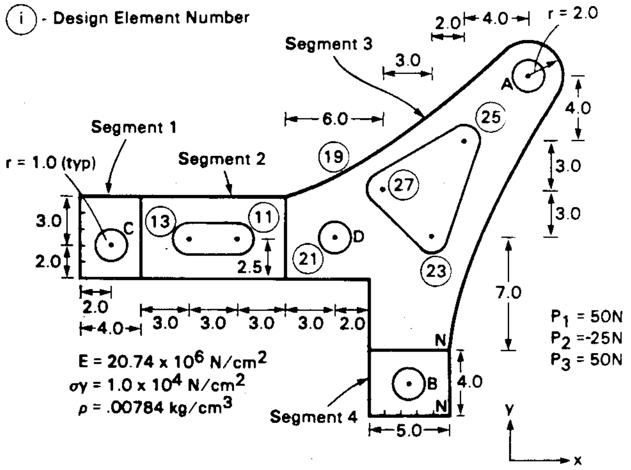


Fig. 20 Dimensions of example 2.

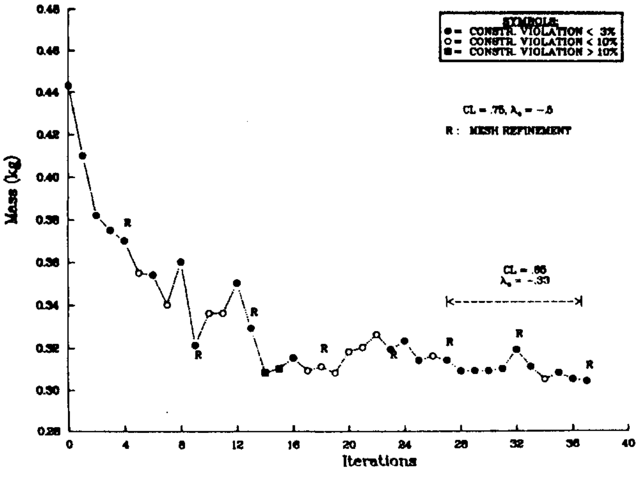


Fig. 21 Design history of example 2.

Example 2—Transmission Shift Linkage Pivot Bracket

Figure 20 shows the dimensions of a typical bracket used to change the direction of a cable running from the transmission shift linkage to the transmission. Segment 2 must be a cylindrical surface to clear the steering column. The radius of this segment is 2.86 cm. Once the coordinates of this segment have been mapped into the cylindrical coordinates, the program automatically translates segment 1 by 3.28 cm necessary to close the gap. Single-point constraints are imposed around the holes labeled C and D. Loads P_1 and P_2 are applied at point A in the y and z directions, respectively. Load P_3 is applied at point B in the y direction. A displacement constraint is applied at point A. A geometric behavior constraint of 0.5 cm was also enforced. A total of 22 design variables were used to control the shape of the bracket. A characteristic length (CL) of 0.75 cm was used, creating an initial finite element model of 2358 degrees of freedom for the unrefined mesh and 3138 for the refined mesh.

Figure 21 shows the design history. The initial shape is shown in Fig. 22. The final shape of the slot (Fig. 23) in the cylindrical segment is slightly tapered. The size of the circular portion on the left end is controlled only by geometric constraints, whereas the end on the right is controlled by

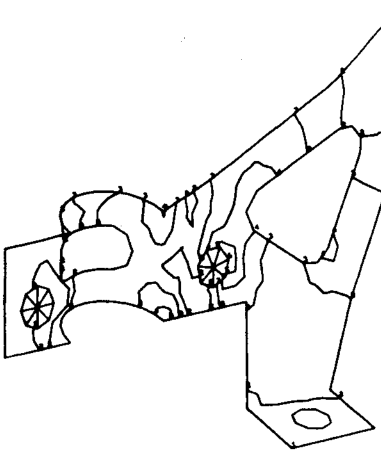


Fig. 22 Stress contour of example 2 (initial design).

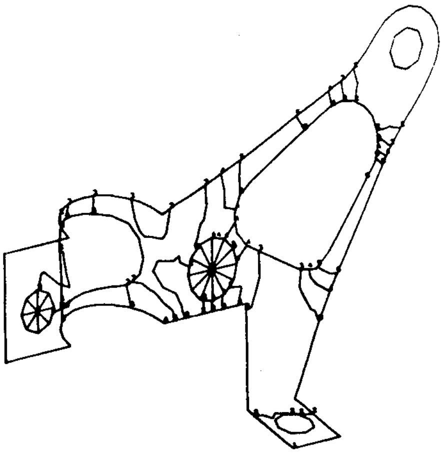


Fig. 23 Stress contour of example 2 (final design).

structural behavior. The displacement constraint became critical at step 9 and the program had some difficulties finding a good direction for several steps. Ultimately, very little mass was removed after the displacement constraint became active. Table 1 gives the design variable values for the initial and final designs.

Summary

Shape optimization using a geometric problem description and adaptive mesh refinement has been extended to structures composed of an assembly of nearly planar segments that can be translated and rotated into the third dimension. Two types of nonplanar surfaces have been described to provide added stiffness due to curvature. Deep shell curvature is provided through a coordinate transformation. General shallow shell curvature is provided through a single-coordinate projection that can also be dependent upon design variables. The increased problem size capacity required improvements in the computational efficiency. To that end, the mesh generation, which accounted for approximately 40% of the total computer time, was replaced by a new technique based upon Dirichlet tessellation, resulting in an order-of-magnitude improvement in efficiency or an overall reduction in computer time of 35%.

Two design examples have been provided to demonstrate the application of this capability. The first example demonstrates the design of an arbitrary shallow shell. The second example is a realistic part, typical of one found in an automobile. Minimum mass shapes were obtained in 30-40 iterations. Mass reductions on the order of 20-30% were obtained based upon a realistic initial shape. The design histories exhibited quite a lot of erratic behavior due to noisy and nonlinear constraints.

In general, it has been shown that it is possible to automate the design process for quite complicated structures through the integration of a parameterized geometry description, fully automatic mesh generation, finite element analysis, and optimization. The resulting capability eliminates the need for tedious data transfer of existing trial-and-error design approaches, as well as many of the repetitive steps involved.

References

[1] Bennett, J. A. and Botkin, M. E., "Structural Shape Optimization with Geometric Description and Adaptive Mesh Refinement," *AIAA Journal*, Vol. 23, March 1985, pp. 458-464.

[2] Botkin, M. E., "An Adaptive Finite Element Technique for Plate Structures," *AIAA Journal*, Vol. 23, May 1985, pp. 812-814.

[3] Akima, H., "A Method for Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points," *ACM Transactions on Mathematical Software*, Vol. 4, No. 2, June 1978, pp. 148-159.

[4] Cavendish, J. C., Field, D. A., Frey, W. H., "An Approach to Automatic Three-Dimensional Finite Element Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 21, 1985, pp. 329-347.

[5] Watson, D. F., "Computing the *N*-Dimensional Delaunay Tessellation with Applications to Voronoi Polytopes," *The Computer Journal*, Vol. 24, No. 2, 1981, pp. 167-172.

[6] Sibson, R., "Locally Equiangular Triangulations," *The Computer Journal*, Vol. 21, No. 3, 1978, pp. 243-245.