THESIS REPORT
Master's Degree

SYSTEMS RESEARCH CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Numerical and Experimental Shape Optimal Design of a Hole in a Tall Beam

by : S.M. Bhandarkar
Advisor: A.J. Durelli

M.S. 88-6
Formerly TR 88-61

ABSTRACT

Title of Thesis : Numerical and Experimental Shape Optimal Design of a Hole In a Tall Beam

Sarvotham M. Bhandarkar, Master of Science, 1988

Thesis directed by : Shapour Azarm, Assistant Professor

August J. Durelli, Professor

Department of Mechanical Engineering and Systems Research Center

Shape optimal design is a structural optimization process in which the boundary shapes of structures are optimized to meet a set of requirements. This is in contrast with early strategies which dealt mostly with the so-called sizing variables such as cross-section, length and width, rather than with the shape of boundaries.

This thesis deals with the optimization of the shape of an interior discontinuity (hole) in a two-dimensional structure, namely a tall beam. The objective is to find hole shapes with approximately uniform boundary stress to reduce the weight of the beam without increasing the maximum tensile stress originally present in the beam.

Two approaches are used:

1. A numerical approach using finite element analysis, non-linear programming and interactive graphics.
2. An experimental approach using photoelasticity.

The hole shapes obtained from the two approaches match quite well. A CAD approach is also presented to simulate the photoelastic method. The methods described in the thesis can be used for a wide variety

of two-dimensional structural problems, including reduction of weight, minimization of stress concentration and other design objectives.

NUMERICAL AND EXPERIMENTAL SHAPE OPTIMAL DESIGN OF A HOLE IN A TALL BEAM

by

Sarvotham M. Bhandarkar

Thesis submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirement for the degree of
Master of Science
1988

Advisory Committee:

Assistant Professor S. Azarm

Professor A. J. Durelli

Associate Professor M. Pecht

Dedicated to my parents

ACKNOWLEDGEMENT

I would like to thank my adviser Dr. Azarm for the dedicated guidance and support he provided throughout the duration of this thesis work. This work was made possible largely through the motivation and direction provided by Dr. Azarm and his constant availability for discussions at all times.

I thank Dr. Durelli, also my adviser, for the time and effort he invested in guiding me through the experimental method and his ready availability to discuss problems.

Thanks are also due to Dr. Pecht for serving on my committee and for the useful suggestions on the written documentation.

Credit is due to Professor W. Yu, research associate from China and Dr. B. Ranganayakamma, research associate from India, for the preliminary work done by them in the numerical and experimental methods respectively.

Finally, I would like to thank Chris Russell, system manager of the CAD lab for his assistance in making available the computing facilities required for this work.

This work was supported in part by NSF (SRC) Grant CDR-85-00108.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Structural optimization is the design of structures to improve performance while meeting specific criteria dictated by functionality. The improvement in performance could be in the form of reduction in weight, minimization of stress concentration, or a combination of these and other similar design objectives.

Historically, the earliest work can be traced back to Galileo (1638), who solved (though erroneously), the problem of finding the optimum dimensions of a variable depth beam. However, most of the work having relevance today has been developed since the early part of this century. In that period several works were completed in the areas of least weight layout of idealized frameworks, optimum design of structural components (columns and beams), and minimum weight design of simple structural systems (trusses and frames). Different analysis and design techniques were used in each of these cases, most of them applicable only to certain specific classes of structural problems.

It was only in 1960 that a classic work of Schmit (1960) set forth a rather general approach to tackle a broad class of structural systems with a single numerical technique. This technique was the coupling of finite element structural analysis and non-linear programming to solve the design problem. The following two decades saw a rapid development and consolidation of this concept. However, the majority of the effort was towards solving member sizing problems in discrete structures (e.g., trusses), where the variables were member cross-sectional areas, lengths, and thicknesses.

In contrast to the problems mentioned earlier, shape optimization[1] is the design of continuum structures (as opposed discrete or skeletal structures like trusses and frameworks) wherein one or more boundaries of the continuum is considered variable in shape. Here the shape of the boundary is changed to satisfy design objectives, rather than changing the thicknesses and cross-sectional areas. Often shape design is more efficient than size design. Consider for example a hole in an infinite plate under uniaxial loading. The sizing design would attempt to add material to the plate around the edges of the hole or introduce another discontinuity in its vicinity to reduce stress concentration. The shape design on the other hand would change the shape of the hole by removing material thereby redistributing the stresses along the hole, minimizing stress concentration and giving a uniform distribution of stress around the hole. Further, the shape design would decrease the weight of the plate.

The desire to reduce structural weight without compromising structural integrity has been the main driving force behind the development of structural optimization methods, especially in aerospace applications. With the growing scarcity of energy and material resources, structural optimization finds application in the automobile industry as well. The use of expensive alloys, composite materials and the possibility of building large-scale structures in space adds further impetus to the development of new and efficient design techniques. Advances in computing and manufacturing

[1]The term 'shape optimization' is also used in literature to describe topology and configuration optimization of skeletal structures. However, throughout this thesis, its usage is restricted to refer to shape optimization of continuum structures. A recent survey by Topping (1983) reviews shape optimization of skeletal structures.

technologies have opened several new areas of exploration in the field of structural optimization, providing highly effective designs both in terms of cost and performance. Shape optimization is one of these emerging areas and the thesis attempts to examine it in some detail.

There are several different approaches to solving the shape design problem. These can be classified into three broad categories, analytical, numerical and experimental, depending on the technique used to obtain the stress distribution. The analytical method uses calculus of variations and optimality criteria methods. The numerical method couples mathematical programming (linear or non-linear) with either finite element or boundary element analysis. Most of the work, though, is based on employing the widely accepted and popular finite element analysis of the structure. An extensive survey of the techniques has been done by Haftka and Grandhi (1986) with 139 references. The experimental method is based on photoelasticity using transparent models in a monochromatic light field. Virtually all the work in this area has been done by Durelli and co-workers (see chapter 3).

1.1 Problem Statement and Assumptions

This thesis examines two approaches to shape optimization, namely the numerical approach using finite elements and an optimizer, and the experimental approach using photoelasticity. The problem considered is the shape optimal design of an interior discontinuity (hole) of an in-plane loaded tall beam.

The rectangular tall beam is shown in fig. 1. It is under uniform compressive in-plane loading on the top and is supported at each bottom-end. The objective is to remove material from within the beam by cutting out a centrally located hole (and changing its shape and size) while satisfying certain constraints.

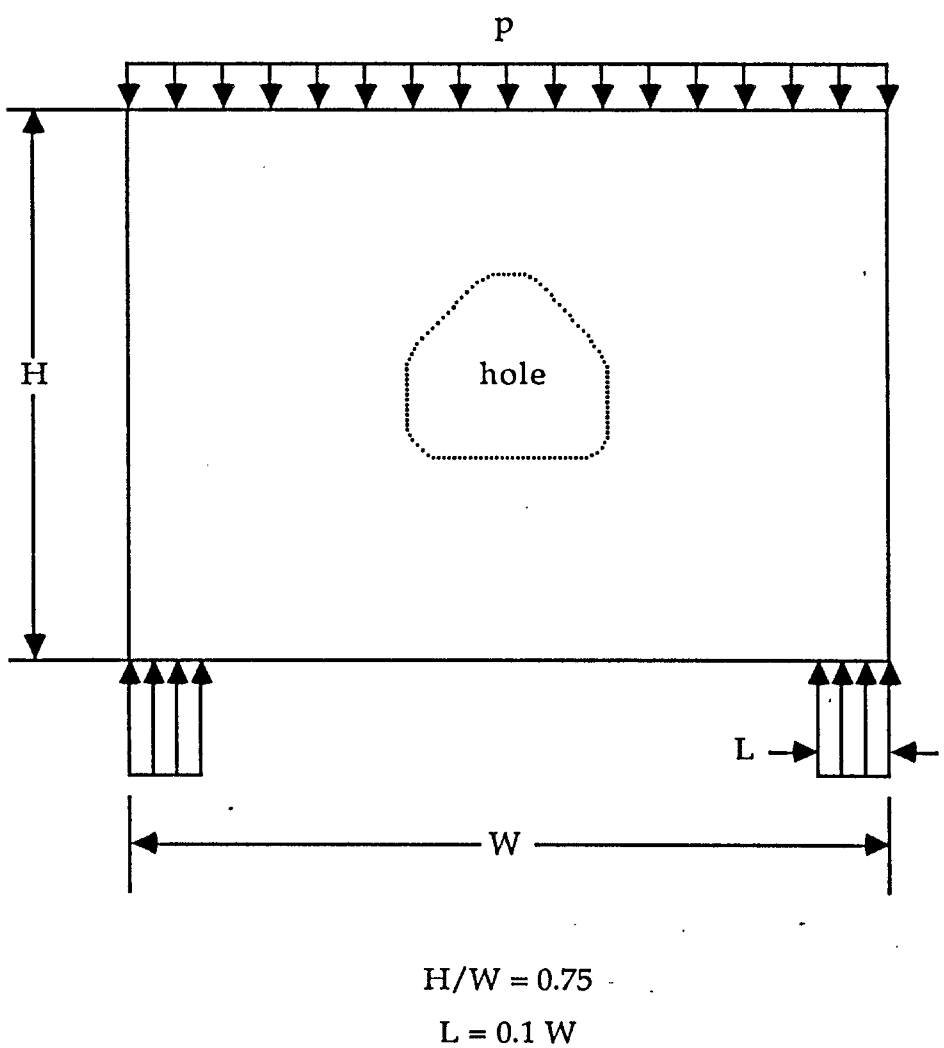


Fig. 1. Geometry of the solid beam to be optimized.

The constraints[1] of the design are :

1. The ratio of the height of the beam to its width is 0.75
2. The top, bottom and sides of the beam remain straight throughout the optimization process.
3. The load applied to the top is approximately uniform. The hole boundary is load free.
4. Each of the two uniform supports at the bottom has a length of L=0.10W.
5. The hole is symmetrical about the vertical axis of the beam.
6. The change in hole shape or size should not increase the maximum tensile stress present in the solid beam , i.e., the maximum tensile stress in the beam at any stage of the optimization process is no greater than that in the initial solid condition.
7. The tangential stress along the hole boundary remains approximately constant.
8. Buckling and out-of-plane bending are neglected.

The experimental method depends on observation of the maximum shear stress fringes which also give a direct measure of the tangential stress along load free boundaries (chapter 3). The designer therefore only needs to view the fringes to verify the maximum tensile stress constraint, assuming that the maximum occurs only on the free boundary. This is discussed in detail in chapter 3.

The use of the constraint on the maximum tensile stress assumes that the beam fails under tension. The maximum tensile stress criterion used here has been found to apply with reasonable accuracy to materials that fail

[1] The word 'constraint' as used here refers to the constraints on the design and is not to be confused with the 'constraints' formulated for the mathematical programming method of chapter 2.

under brittle fracture (Juvinall, 1967). Other criteria like Von-Mises stresses could also be used in a similar fashion.

The tangential stress along the hole boundary usually changes sign as one moves along the boundary. In such cases the tensile stress and the compressive stresses are independently kept constant over as much of the boundary as possible. The 'constants' can be different but in the case of tensile stress it should not exceed the maximum tensile stress in the solid beam.

It is assumed that not only the geometry and loading are symmetric, but that the design constraints also require a final design which is symmetric. Thus it is sufficient to analyze the half structure in the numerical method.

Apart from the constraints mentioned above, there are geometric constraints that limit the extent of the hole shape, e.g., the hole cannot cross the bottom or sides of the beam or approach them too closely.

In summary the objective of the thesis is to find optimum hole shapes that

1. Decrease the weight of the beam.
2. Have an approximately uniform stress distribution around the edges of the hole, taking into account the changing sign of the hole boundary stress as mentioned above.
3. Satisfy the maximum tensile stress constraint and the other constraints mentioned before.

1.2 Contribution of the Thesis

This thesis attempts to :

1. Solve the tall beam design problem numerically by combining finite element analysis and non-linear programming.
2. Solve the problem experimentally using photoelasticity.
3. Compare the two techniques .

4. Examine the use of designer interaction in aiding , understanding, and speeding up the overall design process.

The numerical approach combines commercially available mesh generation and non-linear programming software. Considerable use is made of the operating system command language, in interfacing various aspects of each software. Graphical capabilities of the workstations are utilized to present results in an easily understandable form.

The application of photoelasticity in the experimental method is based almost entirely on designer interaction wherein the operator changes the shape of the hole boundary on a transparent photoelastic specimen while observing the changing fringe patterns, which suggests the next change.

Though considerable literature has accumulated in the field of shape optimization, little has been done to compare the numerical and experimental approaches and to examine the advantages and disadvantages of each. In addition to comparing these two approaches, a CAD approach is presented to simulate the photoelasticity approach. This allows the designer to manually change the shape of the boundary and visualize fringe patterns on the computer as in the photoelasticity approach.

Finally, the importance of designer interaction in the numerical approach is stressed, both as an aid to enhancing solution speed and providing an understanding of the design process. Besides displaying information in a form that is interpretable by an experienced designer, the graphical displays encourage adoption by inexperienced designers and those skeptical of the design process, by allowing them to participate in and follow the design process from start to finish.

1.3 Organization of the Thesis

Chapter 2 describes the numerical approach in detail. The methodology is discussed, the results being deferred to a later chapter (4). The finite element modelling, boundary shape representation, and the non-linear programming algorithm are also discussed. Some of the problems faced during program development are also mentioned along with means by which a designer may interact with the system. Finally, the CAD simulation for the photoelasticity approach is described.

The experimental methodology using photoelasticity is described in chapter 3. Also included is a discussion of practical problems faced in the experimental set-up and operation.

The results obtained by the the numerical and experimental methods are presented and compared in chapter 4. The conclusions and recommendations for future work are discussed in chapter 5.

The appendices contain user manuals for the CAD and numerical approaches. Computer listings of important sections of the code are also included.

2 NUMERICAL METHOD

Numerical methods for shape optimization of structures combine a analysis method with an optimizer. The basic approach can be summarized as follows. The variable boundary of the structure is defined by a curve whose shape depends on the values of variables called shape variables. Sensitivity derivatives are calculated for each shape variable. These give quantitative information on how the structural response is affected by a change in a shape variable, i.e., the change in stress, displacement, etc., due to a change in the shape of the boundary. Subsequently the optimizer utilizes the sensitivity derivatives to change the values of the shape variables, in accordance with a design objective function and constraints. A new shape results. Sensitivity derivatives are again calculated for the new shape followed by another optimization. This iterative process is repeated several times until a satisfactory optimum boundary shape is obtained.

A structural analysis method is necessary to calculate the structural response for each boundary shape and for the calculation of sensitivity derivatives. The finite element method is commonly used for this purpose. It is reliable and has a large field of application. Its use however necessitates discretization of the structure, which in turn brings special problems in practical implementation. This is because the finite element mesh must be changed as the boundary shape changes. An automatic mesh regeneration procedure is required for this purpose.

Besides mesh regeneration some of the other important issues include the need for accurate evaluation of sensitivity derivatives and proper selection of design variables. These and other problems are discussed by Haftka and Grandhi (1986) in an extensive review of recent literature. Bennett and Botkin (1985), use regional mesh refinement schemes in addition to automatic mesh regeneration, for improving the accuracy of the finite element mesh. For sensitivity calculations two broad approaches are available. In one, finite difference methods are used to obtain the derivatives after finite element discretization (Haftka and Kamat, 1987). The other approach is based on differentiating the continuum equations before finite element discretization (Haug et al., 1986). There are also a number of ways of selecting the shape variables. Improper selection of variables can lead to a jagged boundary shape as shown in the next section. A robust method used by Braibant and Fleury (1985) utilizes a design element in which the boundaries are described by B-splines and Beizer curves. Finally, it should be noted that boundary elements can be used to model the boundary instead of using finite elements. Some work in this area has been done by Mota Soares et. al (1984).

2.1 Solution Process

2.1.1 Overall Approach

Finite element analysis (displacement method) is used in conjunction with sequential quadratic programming (optimizer). The variable hole boundary is represented by using a B-spline. The sensitivity derivatives are calculated using finite differences. Isoparametric eight-node quadrilateral elements are used for the finite element modelling.

Fig. 2 shows a flowchart of the overall methodology. An initial shape (i.e., initial values of the shape variables for the B-spline) is input by the designer at the beginning of the optimization. Successive iterations consist of applications of finite element analysis, optimization, and modification of the

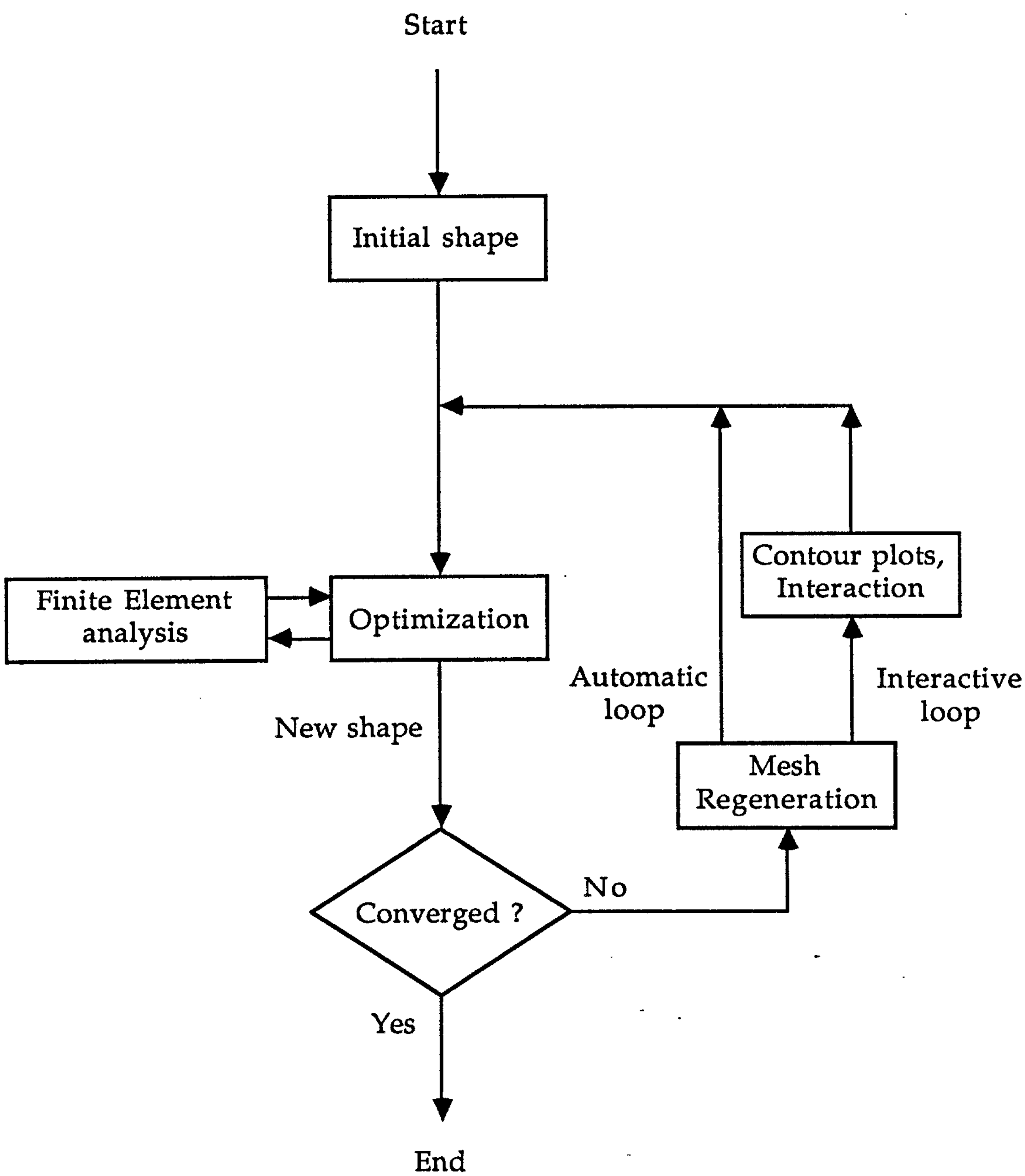


Fig. 2 Flow-chart of the numerical shape optimization process.

initial shape until a satisfactory boundary shape is obtained. Restart capabilities of the optimizer permit an optional designer interaction at the end of each iteration.

The solution process may be broken down into the following main components:

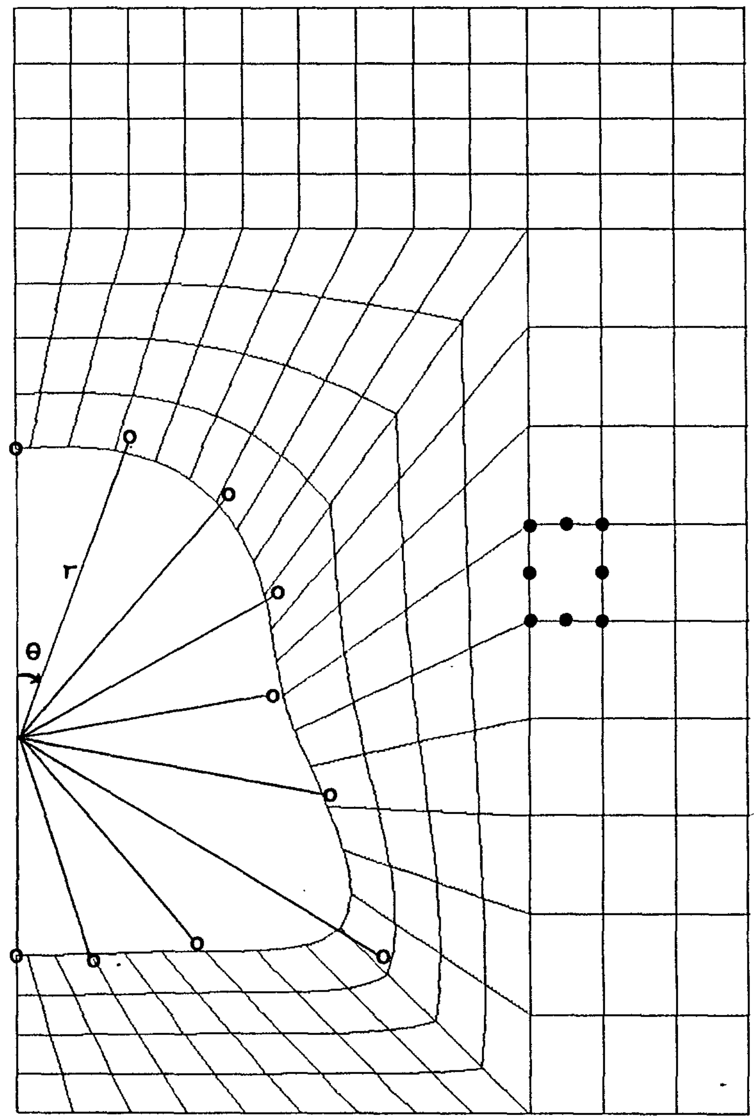
1. Boundary representation
2. Mesh regeneration
3. Finite element analysis
4. Optimization
5. Designer interaction

Each of these components are explained in detail in the following sections.

2.1.2 Boundary Representation

The hole boundary is represented by a B-spline defined by the position of ten 'control points' (Fig. 3) . Only the half structure is analyzed because of symmetry. Note that the control points are different from the boundary nodes of the finite element model. The position of the control points are determined by polar coordinates (r,θ) with the origin on the vertical axis of symmetry of the beam. For each control point the angle θ is fixed but the radius r is variable. The shape of the hole boundary can therefore be changed by varying radii r. These then are the so-called shape variables. The shape representation used here is similar to the one used by Fleury (1987).

It is important to strike a balance between desired accuracy and cost of the analysis when choosing the number of shape variables. By increasing the number of variables, more complex shapes can be modelled, but the sensitivity calculations become expensive. With this in mind a ten control



o control points

• finite element nodes

Fig. 3 Finite element mesh model for the right half beam with the hole boundary modelled with ten control points.

point B-spline was found to be sufficient to model the desired boundary shapes.

The software PATRAN (PDA Engineering, 1984) is used for finite element mesh generation and can model structures with B-splines. This is linked with a mesh regenerator for automatic mesh generation (section 2.1.3). The mathematical equation of the spline is not required since the sensitivity information is calculated from the structural response by perturbing each control point in turn and using the finite difference method (section 2.1. 5).

An important property of the cubic B-spline is that it passes through only the first and last control points (Fig. 4). The intermediate points serve as a kind of 'points of weight' or 'spring application' for the curve. This variation diminishing property of the B-spline prevents zig-zag boundary shapes and boundary shapes with sharp changes in curvature. This, however may not be possible if higher order polynomials are used. In the early stages of development, piecewise straight lines connecting the control points were used to model the variable boundary. Jagged boundary shapes were obtained at some stages (fig. 5). These not only decrease the accuracy of the finite element analysis, but also hinder the progress of the optimization. The optimization algorithm takes several iterations to recover from such jagged shapes. The occurrence of the jagged shapes has been attributed to the one-to-one correspondence between the finite element model and the design variables (Haftka and Kamat, 1987).

One more point of interest is the selection of the shape variables. At first the (x,y) coordinates of the control points were used as the shape variables (i.e., both the x and the y coordinates are variable). This creates complications in maintaining mesh integrity as the hole boundary changes shape. Rectangular variation domains have to be imposed on each control

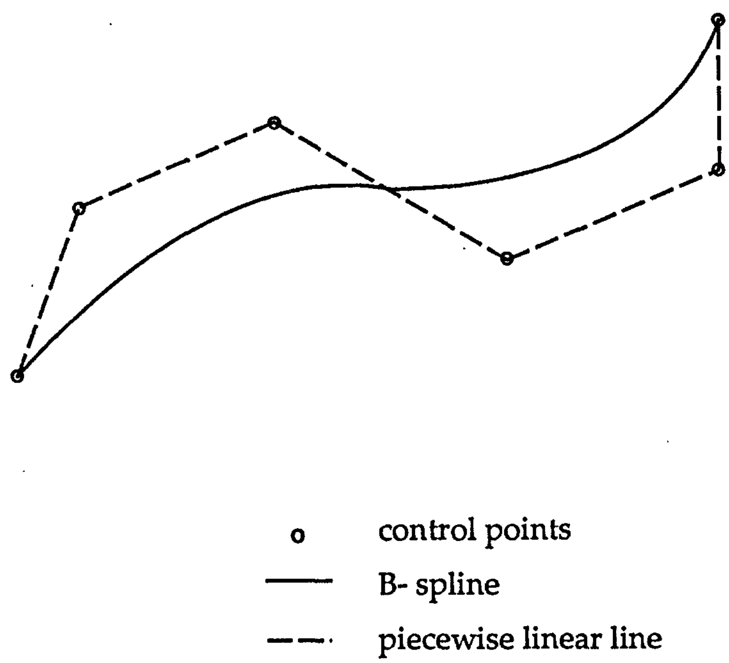


Fig. 4 Variation diminishing property of the B-spline.

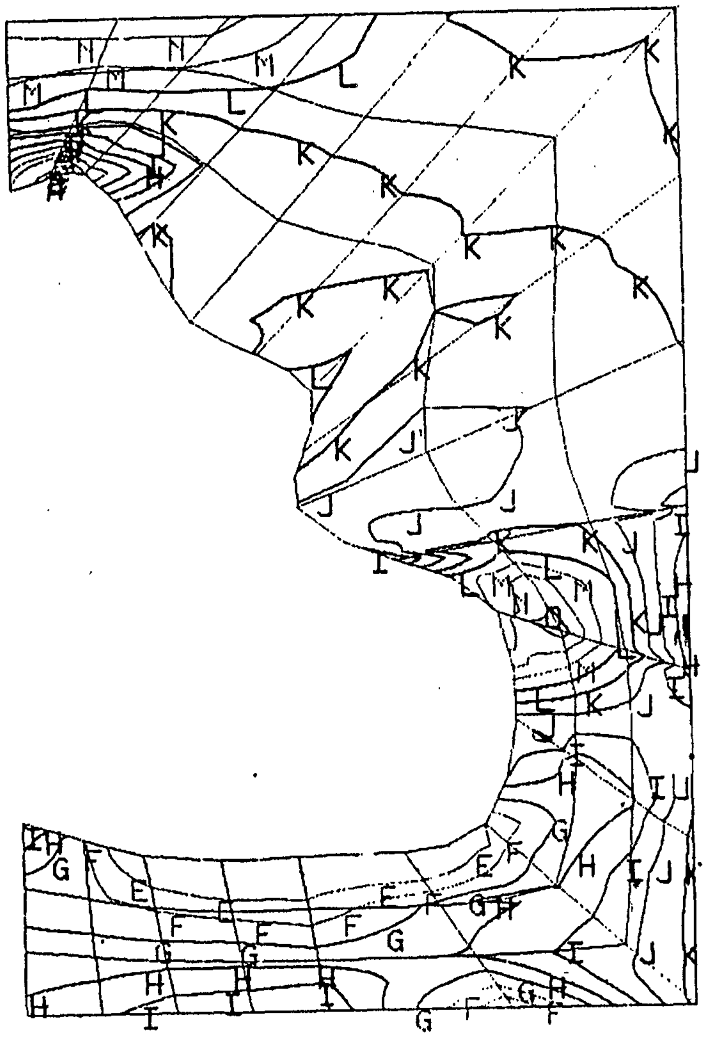


Fig. 5 Jagged hole boundary shape

point to prevent them from crossing over or approaching each other too closely. Moreover the variation domains change as the hole shape changes when the control points move in the plane. When the shape variable is r (θ being fixed) the control points are more or less equally spaced out, and as evident from fig. 3, only r has to be restricted in range. For instance an upper limit may have to be imposed on r to prevent the hole boundary from crossing the top, bottom or sides of the beam (a physically absurd condition). The limit remains same throughout the optimization process. There is also a tremendous saving in computational effort since the number of variables is reduced by half, by selecting r as the variable instead of both x and y.

It should be noted that we are trying to find the optimum shape within the range of shapes that can be modelled by the B-spline representation described above. Some hole shapes[1] like those in fig. 6, cannot be modelled by a B-spline with ten control points. Only an approximation shown by the dotted line at the notch can be accurately modelled. However it is possible to introduce additional control points in the vicinity of the notch and rerun the optimization starting from the dotted line approximation. This will take more computing time depending on the number of additional control points introduced.

2.1.3 Mesh Regeneration

One of the most important steps integral to the success of the shape optimization algorithm is the development of an automatic mesh generator. As the hole boundary changes shape, the mesh generator regenerates the finite element mesh for each of the shapes. At any stage, the mesh must be

[1] The hole boundary shape of fig. 5 with the small notch was one of the shapes actually obtained using photoelasticity (see chapters 3,4).

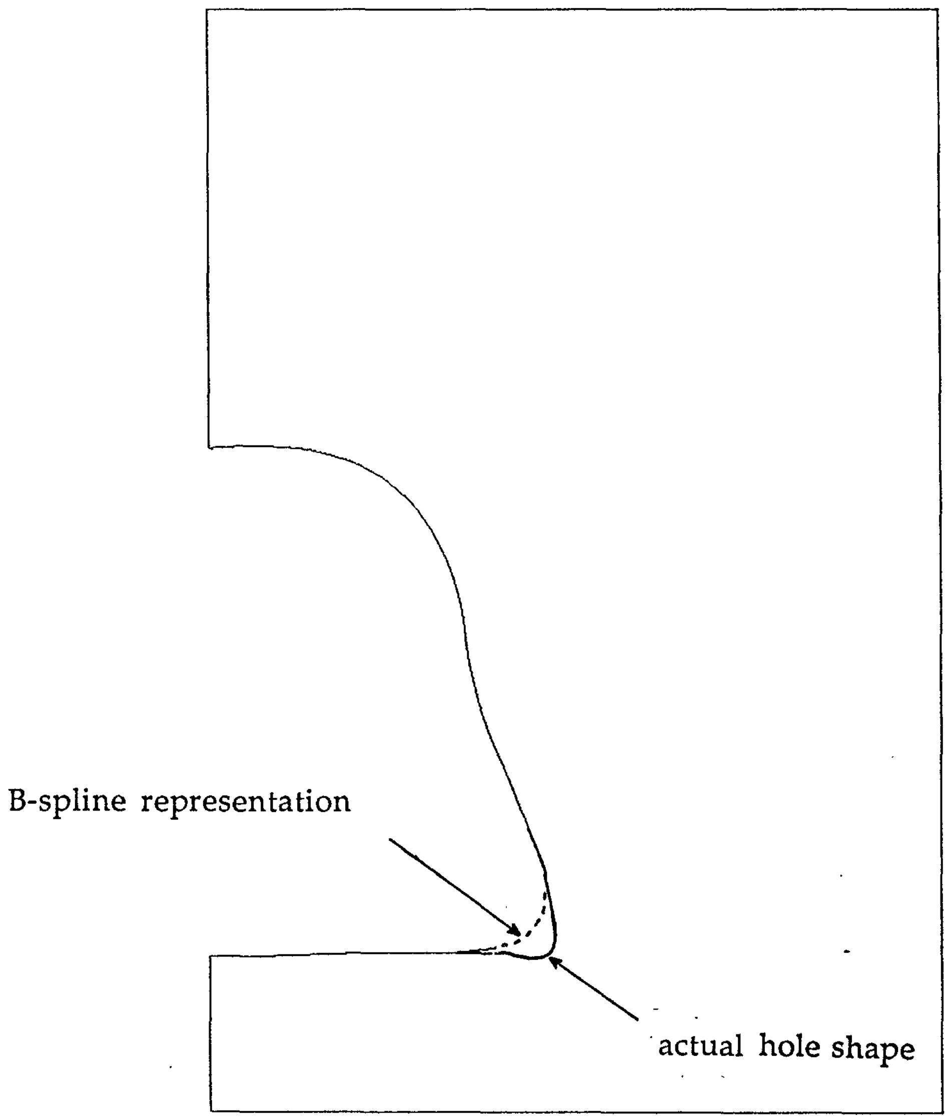


Fig. 6 Hole shape that cannot be modelled by a ten control point B-spline.

accurate enough to provide a reliable structural response. Excessive distortion of elements or their bad aspect ratios can affect the accuracy of the finite element analysis. Also a finer mesh may be needed in regions of high stress concentrations for reliability.

The mesh generator used here makes use of the way in which the basic geometry is described by PATRAN (PDA Engineering, 1984). PATRAN builds up the model by first using directives which describe the essential geometry of the model. The actual division of the beam into finite elements takes place at the final stages of model definition. This allows for easy parametric studies of the model. If the ratio (H/W) of the the tall beam is changed for instance, then only one or two initial directives (that affect the aspect ratio) would have to be changed, and the new geometry is remeshed using the same mesh type as before (since the mesh division directives which appear later on, are left unchanged). An analogous method is used for shape optimization in which the hole boundary is variable. In this case the variable boundary is the B-spline which is controlled by the control points. So to remesh the new geometry the control point coordinates have to be changed in the basic geometry definition.

The geometry directives are the grids, lines and patches. A grid is a point in space. A patch is a quadrangular area defined by four corner grids (fig. 7a). These grid points are the first entities that are defined while creating the finite element model. The patches are then created using the grid points as the corners. Subsequently the basic geometry defined by the patches is divided into nodes and elements by further PATRAN directives. Note that the grid points are not the same as the finite element nodes though the two may coincide at the corners of the patches. All the directives used for creating the

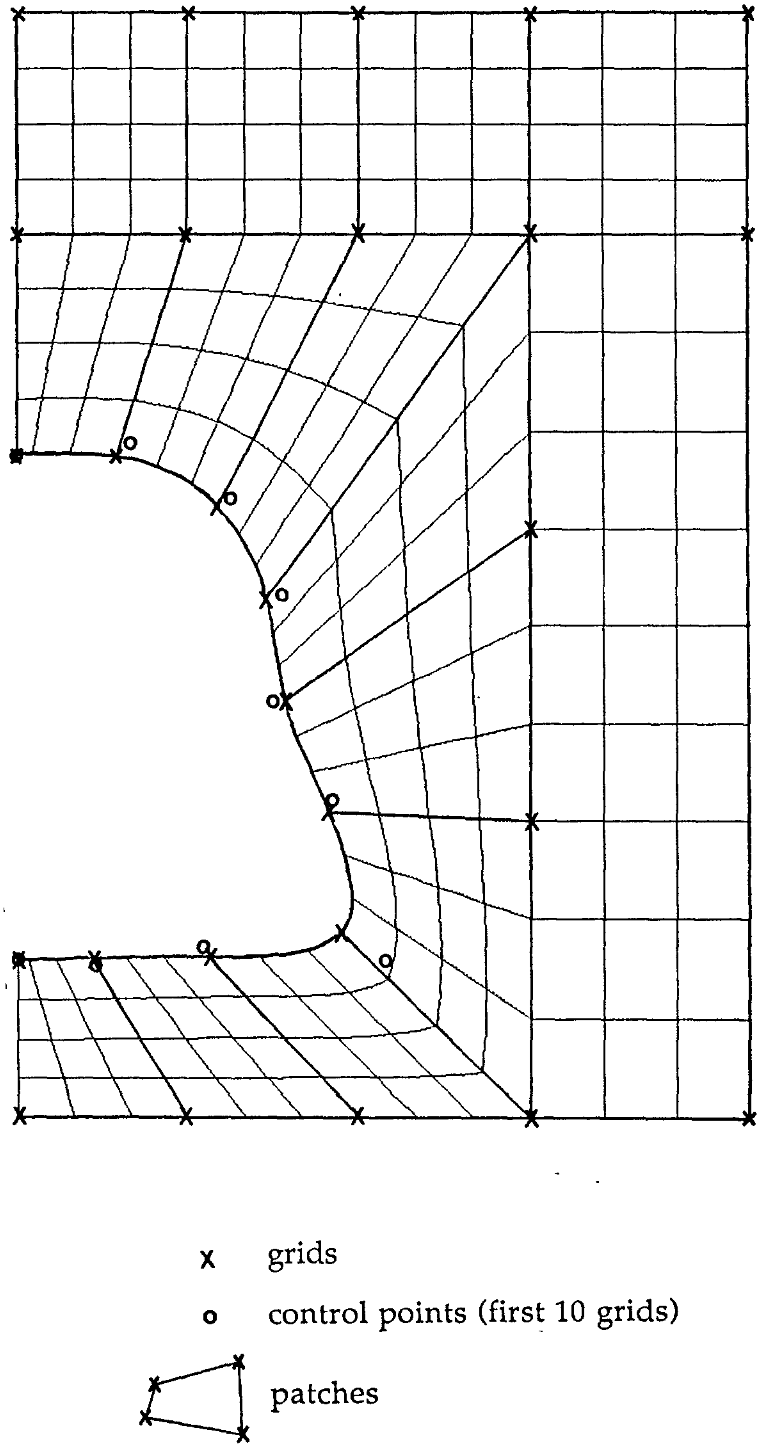
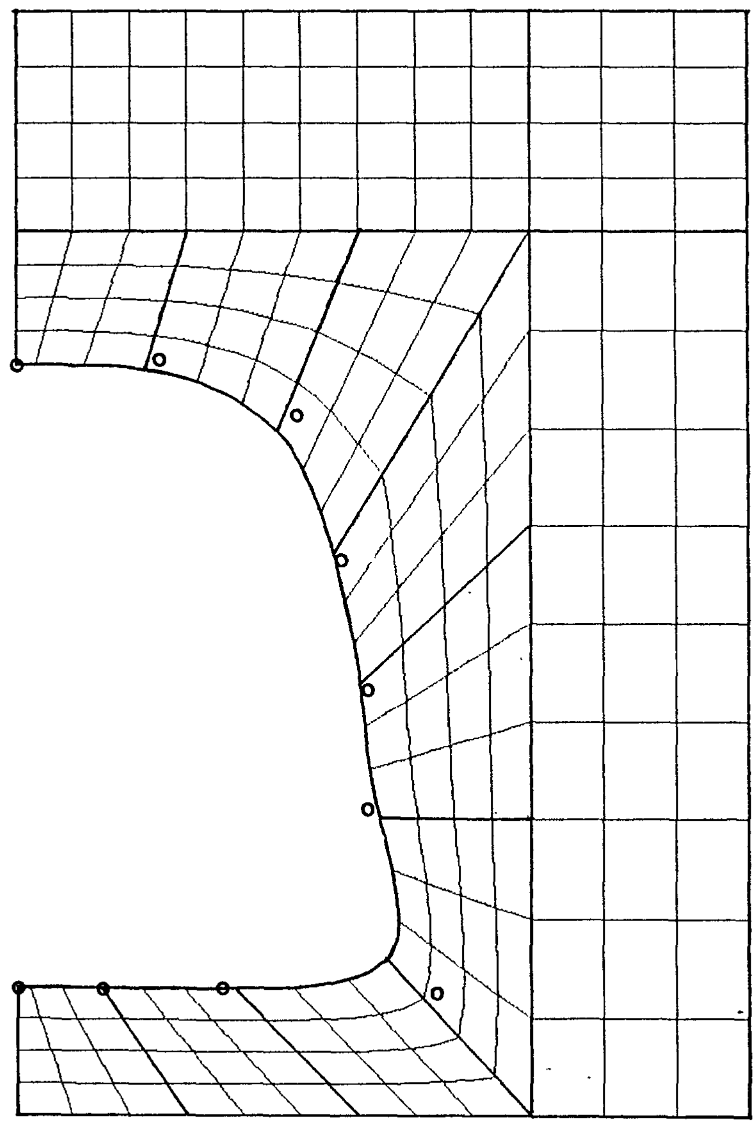


Fig. 7(a) Automatic mesh regeneration. Mesh for a hole shape.



o control points

Fig.7(b)Automatic mesh regeneration.Regenerated mesh for a new hole shape.

finite element model are recorded on a file. The first ten grid points[1] correspond to the control points of the B-spline. When the hole shape changes, the new coordinates of these grid points (control points), supplied by the optimizer, are read into the file in place of the old ones and a new file is created. The rest of the directives remain the same. Automatic mesh regeneration is achieved by simply running the file again.

Fig. 7a shows the finite element mesh for a particular hole shape and fig. 7b shows the regenerated mesh for a new hole shape. Note that the mesh arrangement is the same in both cases. It is important to choose an initial mesh arrangement that will remain accurate for any reasonable shape changes. Note that only the patches which have one of its edges as the B-spline, have changed shape. The other patches remain the same as before since the coordinates of the grid points controlling them have not been changed. The finite element mesh created by the mesh regenerator stays undistorted for a wide range of hole shapes. As the hole grows bigger, though, the fixed edge of the variable patches may have to be moved outward towards the external boundary of the beam, to accommodate the hole and maintain mesh integrity. Designer interaction can be used for this purpose (section 2.1.6).

2.1.4 Finite Element Analysis

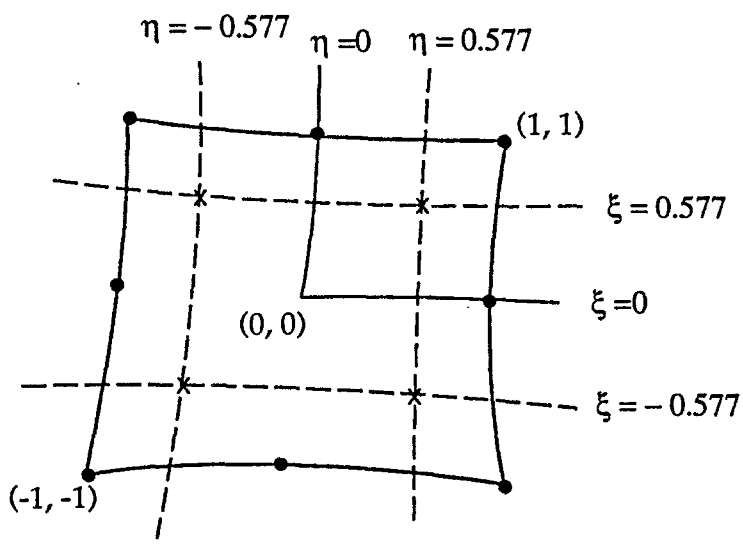
The stiffness or displacement method of finite element analysis is used here. The governing equation is

$$[K]\{U\} = \{F\}$$

[1] After the B-spline is created from these control points, equispaced grid points are created along the spline to define the corners of patches (fig. 7).

where **K** is the global stiffness matrix, **U** is the global nodal displacement vector and F is the vector of global nodal loads. The global nodal displacements are the primary unknowns to be computed. Stress is the secondary unknown, computed from the displacements. On solving the equation, the compatibility requirements of strains and the displacement boundary conditions are exactly satisfied, while the equilibrium equations and force boundary conditions are only approximately satisfied, the accuracy depending on the fineness of the finite element mesh. The displacement method is the most popular form of the finite element method in structural mechanics. Gaussian elimination with Choleski decomposition (Cook, 1981) are used for solution of the equations. The bandwidth scheme is used for storing the global stiffness matrix. The analysis program reads in the nodal coordinates and connectivity information from PATRAN data files. A listing of the finite element analysis code can be found in the appendix.

The in-plane loaded tall beam is a plane stress problem. Isoparametric eight-node quadrilateral plane stress elements (Fig. 8) are chosen. They can model curved boundaries quite well due to the presence of mid-side nodes and are also the simplest such type in the isoparametric family. A reduced 2 x 2 Gaussian integration scheme (Bathe, 1982) is used for the elements to save computer time. It is to be noted that the stresses within the element are of a higher order of accuracy when calculated at the Gaussian integration points (Barlow, 1976). However in most cases (as in the tall beam) the stresses are more critical at the boundary of the structure. The stresses therefore have to be calculated at the nodes of the elements instead of in the interior. Sampling stresses directly at the nodes can lead to errors (Hinton and Campbell, 1974). To overcome this difficulty the stresses are first calculated at the 2 x 2



- ● finite element nodes
- x 2x2 Gaussian integration points

Fig. 8 Eight-node isoparametric plane stress quadrilateral element.

Gaussian integration points and then extrapolated to the nodes using a least squares smoothing technique (Hinton and Campbell, 1974).

Finally, since the displacement method is used, only the displacements are continuous across the boundaries of the elements. The stresses are not continuous. It is therefore necessary to consider all element contributions meeting at any node, to compute the average stresses at that point. This introduces some minor errors. For example, the principal stress normal to a free surface should be zero, but a small stress may be indicated from the analysis, due to extrapolation and averaging.

2.1.5 Optimization

The optimizer is a non-linear programming algorithm to solve a problem of the form

$$\begin{aligned}&\text{Minimize } f(X)\\&\text{Subject to :}\\&\qquad g_j(X) \geq 0 \qquad j=1,m\end{aligned} \qquad \text{...............} \ (1)$$

where X, f, g_j represent design variable vector, objective function and inequality constraint[1] functions. The total number of inequality constraints is m.

A sequential quadratic programming algorithm (Crane et al., 1980) suggested by Powell (1978) is used as the optimizer. Powell's algorithm is designed to converge to a point that satisfies the first-order Kuhn Tucker necessary conditions but does not always satisfy the sufficiency conditions. The algorithm is an iterative procedure which performs two tasks in each iteration. The first is a solution of a positive definite quadratic programming

[1] The 'constraints' referred to here are mathematically formulated constraints and should not be confused with the design constraints of chapter 1.

problem followed by a one-dimensional minimization. The quadratic programming problem, at its solution, provides a search direction S and estimates of the Lagrange multipliers λ.

The Lagrangian function is defined as

$$L(X,\lambda) = f(X) - \sum_{j=1}^{m} \lambda_j g_j(X) \qquad \text{...............} \quad (2)$$

The quadratic programming problem is obtained by forming a quadratic approximation to the Lagrangian function and linearizing the constraints about the current solution estimate for X. It is formulated as

$$\text{Minimize} \quad Q(S) = f(X) - \nabla f(X).\, S + \frac{1}{2} S^T B\, S \qquad \text{...............} \quad (3)$$

$$\text{Subject to:} \quad \nabla g_j\,(X).\, S + \delta_j g_j(X) \geq 0 \qquad j=1,\, m$$

Here the design variables are the components of search direction S, while X contains the current solution estimate of equation (1). The matrix B is a positive definite matrix which is initialized to the identity matrix and later updated to approximate the hessian of the Lagrangian. δ_j is a scalar parameter whose value varies between 0 and 1. The solution of equation (3) yields the new search direction S along which the one-dimensional search is performed.

The one-dimensional minimization problem is formulated as an exterior penalty function,

$$\text{Minimize } \phi(X) = f(X) + \sum_{j=1}^{m} u_j \left\{ \min(0, g_j(X)) \right\} \qquad \text{...............} \quad (4)$$

where, $\quad X = X^{q-1} + \alpha S$

$u_j = |\lambda_j| \qquad j=1, m \qquad$ first iteration

$u_j = \max\left(|\lambda_j|, \frac{1}{2}(u_j^{q-1} + |\lambda_j|) \right) \qquad$ subsequent iterations

Here X^{q-1} and u_j^{q-1} are the values of X and u_j from the previous iteration. The solution of equation (4) yields α^* which minimizes $\phi(X)$. The new estimate for the design variable vector X is then obtained as

$$X = X^{q-1} + \alpha^* S \qquad \text{................ (5)}$$

The matrix B is updated by the Broydon-Fletcher-Goldfarb-Shanno update recommended by Powell (1978).

A convergence test is made on each iteration after the quadratic programming problem is solved. The algorithm is terminated if

$$| \nabla f(X^{q-1}).S | + \sum_{j=1}^{m} | \lambda_j \, g_j(X^{q-1}) | < \varepsilon \qquad \text{................ (6)}$$

where ε is a user supplied error tolerance. X^{q-1} is then the accepted solution of equation(1).

2.1.5.1 Optimization Model for the Tall Beam

The objective function and the constraints have to be formulated for the tall beam. In the case of the tall beam, the design variables are the radii r of the control points. The vector X is therefore a column vector of ten elements. The objective is to find hole shapes with constant tangential stress along the hole boundary. The constraints are that the change in boundary shape should not increase the maximum tensile stress in the beam beyond the level present in the solid case. The other constraints listed in chapter 1 need not be mathematically formulated.

Numerically, the objective function is formulated as follows. The tangential stress[1], $\sigma^t = 2\tau_{max}$, is sampled in the elements adjacent to the hole

---

[1] On load free boundaries the stress normal to the boundary, which is also a principal stress is zero, and the tangential stress which is the other principal stress is twice the magnitude of the maximum shear stress.

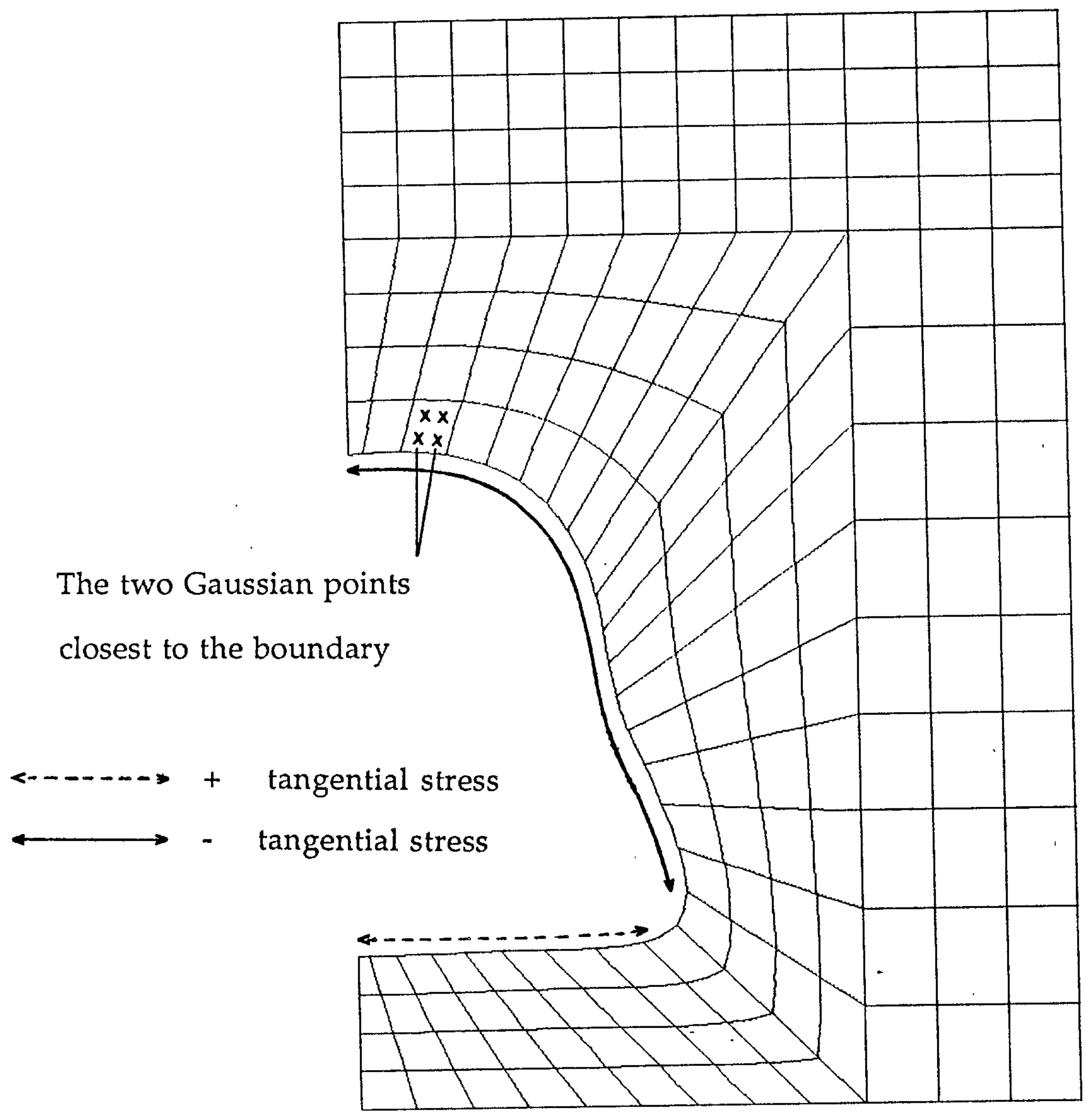


Fig. 9 Formulating the objective function.

boundary, at the two Gaussian integration points[2] closest to the boundary (fig. 9). As explained in chapter 1, the tangential stress may change sign as one moves along the hole boundary. Therefore the mean tangential stress should be calculated for each 'same sign' portion of the boundary. Let this mean stress be denoted as σ_k^t, k=1,n, where the index k indicates a 'same sign' portion of the boundary, and n is the number of same sign boundaries. In fig. 9 the stress changes sign only once, and n=2.

The variance of the stress is then calculated for each 'same sign' portion of the boundary.

$$V_k(X) = \sum_{i=1}^{l} (\sigma_i^t - \sigma_k^t)^2 \qquad k=1,n \qquad \text{................ (7)}$$

where l is the number of sampled Gaussian integration points for that portion of the boundary indicated by index k, and σ_i is the tangential stress at each sampled Gaussian integration point.

The objective function can now be formulated as

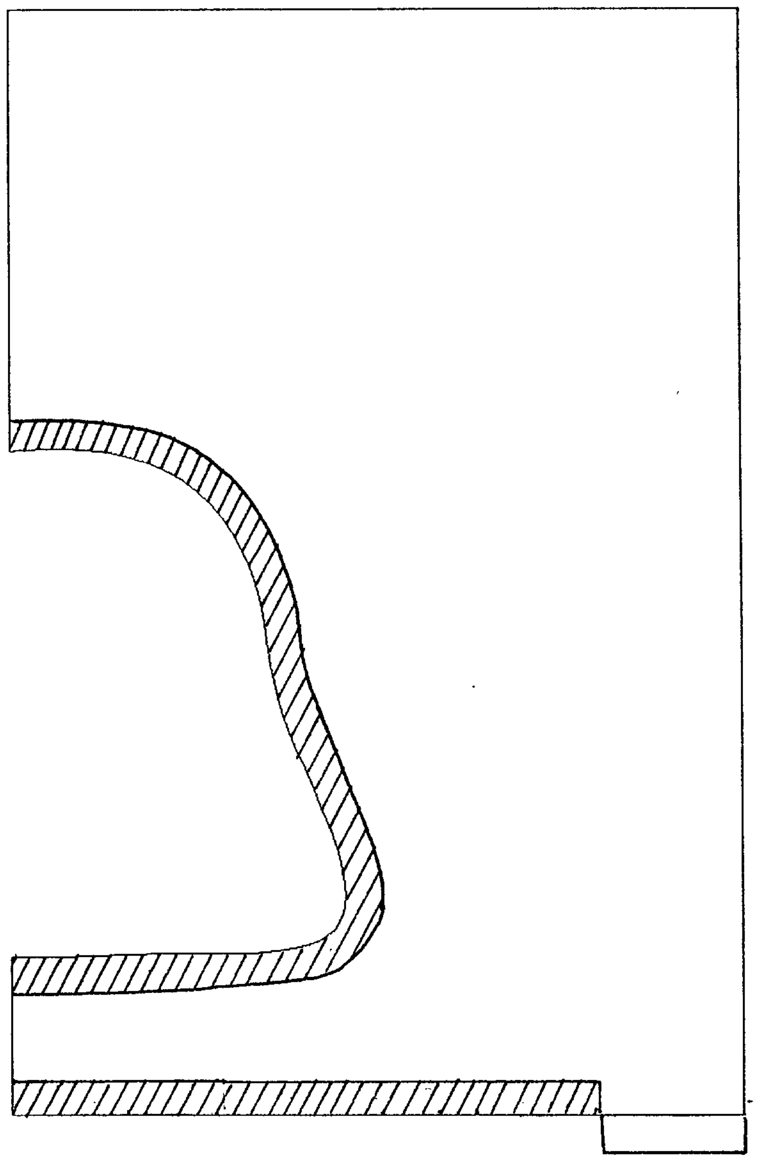
$$\text{Minimize } f(X) = \sum_{k=1}^{n} V_k(X) \qquad \text{................ (8)}$$

From the formulation, it is seen that as f(X) tends to zero, the tensile stresses and compressive stresses are independently kept constant along the hole boundary.

The constraints can be formulated as follows. The tensile stresses can be expected to be maximum along the free edges shown in Fig. 10. This forms

---

2 The relation $\sigma^t = 2\tau_{max}$ holds true only on load free boundaries . But since the boundary stresses are calculated by extrapolating the Gaussian integration point stresses, it is an acceptable approximation to sample τ_{max} at the Gaussian integration points rather than on the boundary nodes.



Free boundaries at which the tensile stress can be maximum

Fig. 10 Formulating the constraints.

the basis of the experimental method (chapter 3). Therefore, in the numerical method too, the tangential stresses will be sampled only at the finite element nodes along the free boundaries marked in fig. 10. The constraints can be written as

$$g_j(X) = \sigma^t_{max} - \sigma^t_j \geq 0 \qquad j=1, m1 \qquad \text{................ (9)}$$

where σ^t_j is the tangential stress sampled at the m1 finite element node points on the free boundary, and σ^t_{max} is the maximum tensile stress occurring in the initial solid beam without the hole (chapter 4).

Besides the constraints on stresses, some side constraints have to be imposed on the shape variables. It is seen that several different hole shapes with constant tangential hole boundary stress can exist. If we are interested in one of these optimum shapes which lies within a particular region, then the shape variables must be constrained to remain within that region. These side constraints can be formulated as

$$\begin{aligned} g_j(X) &= X^U_i - X_i \geq 0 \qquad & j = m1, m2 \qquad \text{................ (10)} \\ g_j(X) &= X_i - X^L_i \geq 0 \qquad & j = m2, m \end{aligned}$$

Here X^L_i and X^U_i are the lower and upper limiting values on the shape variable X_i, which is the i^{th} element of the variable vector X. The number of side constraints for the tall beam is usually twenty, since there is a lower and upper limit for each of the ten shape variables. To examine the hole shapes which lie outside the region defined by the side constraints the optimization can be rerun with the different side constraints.

2.1.5.2 Sensitivity Derivative Calculation

The sensitivity derivatives are calculated by forward differences. Each shape variable X_i is perturbed by an amount ΔX_i and the structural response is calculated for the resulting hole shape. Then the sensitivity derivatives can be computed as

$$\frac{\partial f(X)}{\partial X_i} = \frac{f(X+\Delta X_i) - f(X_i)}{\Delta X_i} \qquad \text{................ (11)}$$

$$\frac{\partial g_j}{\partial X_i} = \frac{g_j(X_i+\Delta X_i) - g_j(X_i)}{\Delta X_i} \qquad j = 1, m$$

This is repeated for each X_i in X. Sensitivity derivative calculations take up the bulk of the computing time used by the optimizer since mesh regeneration and reanalysis has to be performed once for each shape variable. The computing time can be reduced by properly selecting and reducing the number of shape variables as indicated earlier.

2.1.6 Designer Interaction

Designer interaction is a means by which the designer can obtain information about the optimization process, the intermediate designs and can take action to modify and control the optimization process. Designer interaction is an important part of the post-processing stage of analysis. The optimizer can either be run continuously in a batch mode or can be interrupted at the end of each iteration for designer interaction. Interaction allows the designer to make design changes and then lets the optimizer take control again.

Interaction can be divided into two parts. The first consists of displaying the results of the optimization, which may be in the form of contour plots of stresses, displacements and other structural response; iteration history plots and plots of boundary shapes. The second part allows

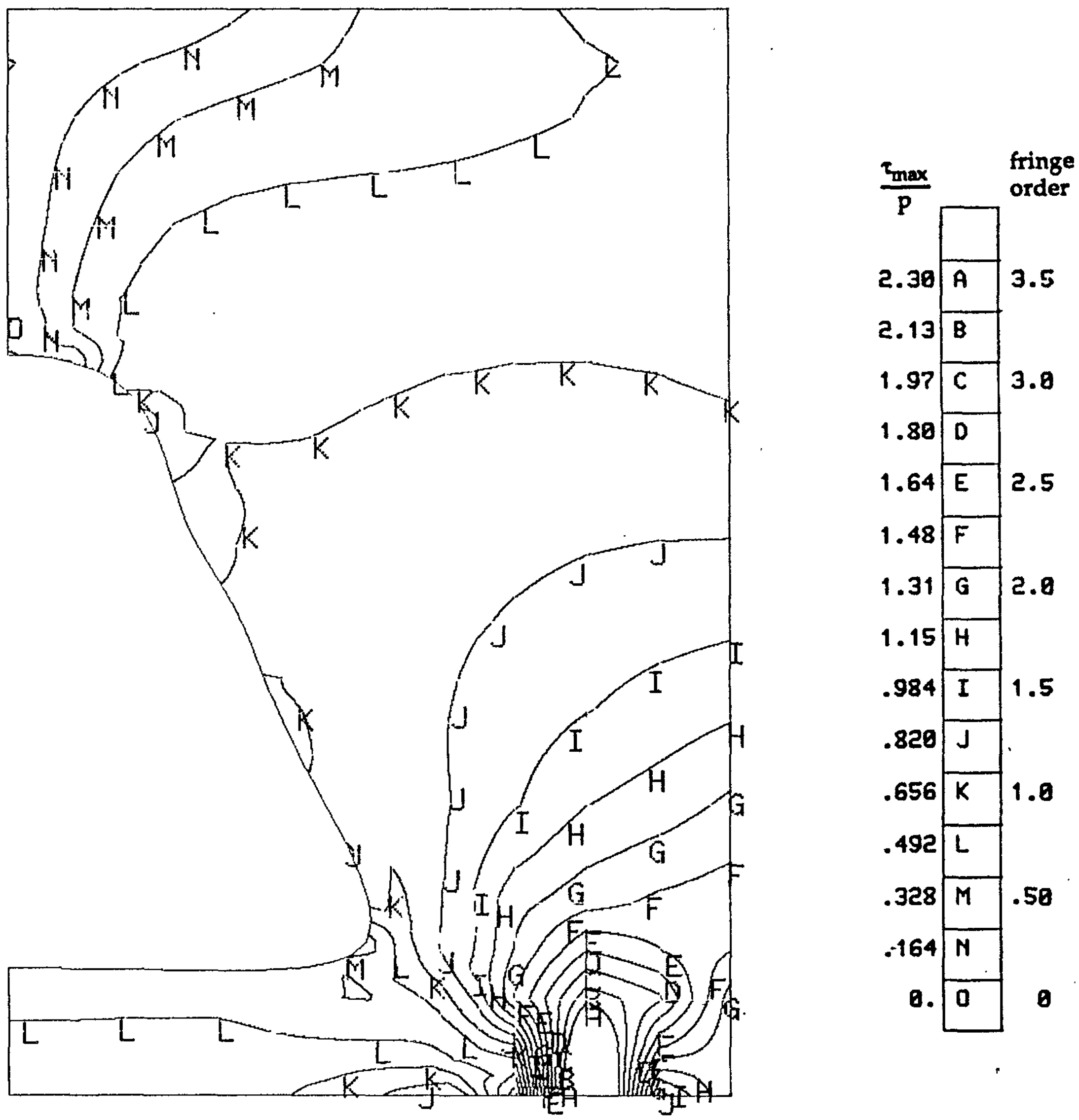


Fig. 11 Contour plots of maximum shear stress τ_{max}

the designer to make design changes based on conclusions drawn from the first part.

The visual display enables the designer to interpret results in physical terms. Fig. 11 shows contour plots of the maximum shear stress (τ_{max}) for a particular hole shape. The plots are actually color-coded but are presented here in black and white. The designer can, at a glance, verify as to what extent the boundary stress is uniform and also if a stress constraint has been violated. The onset of mesh distortion can be easily identified. In the initial stages of the optimization, interaction aids the designer in visually identifying flaws and refining the problem formulation that may not have considered all factors that become important during the iterative optimization process. Fig. 12 shows the progress of the optimization in terms of improvement of the objective function over successive iterations, and fig. 13 shows the different hole shapes after each iteration. These two plots help the designer to evaluate the progress of the optimization and permit termination of the iterative design process. Fig. 14 is a plot of the current values of the design variables (for the hole shape of fig. 11) and the corresponding limits (side constraints) on the shape variables. Note that some of the design variables have reached either their upper or lower limits , thereby restricting the movement of the hole boundary. This prevents the hole boundary of fig. 11 from changing shape in order to attain constant boundary stress.

The designer can take action to control the optimization process. Parameters like step-size and convergence criteria can be adjusted. Side constraints can also be changed. If the mesh is distorted too much, the model can be manually remeshed and the optimization process can be started again. Changing the values of the shape variables gives the designer direct control

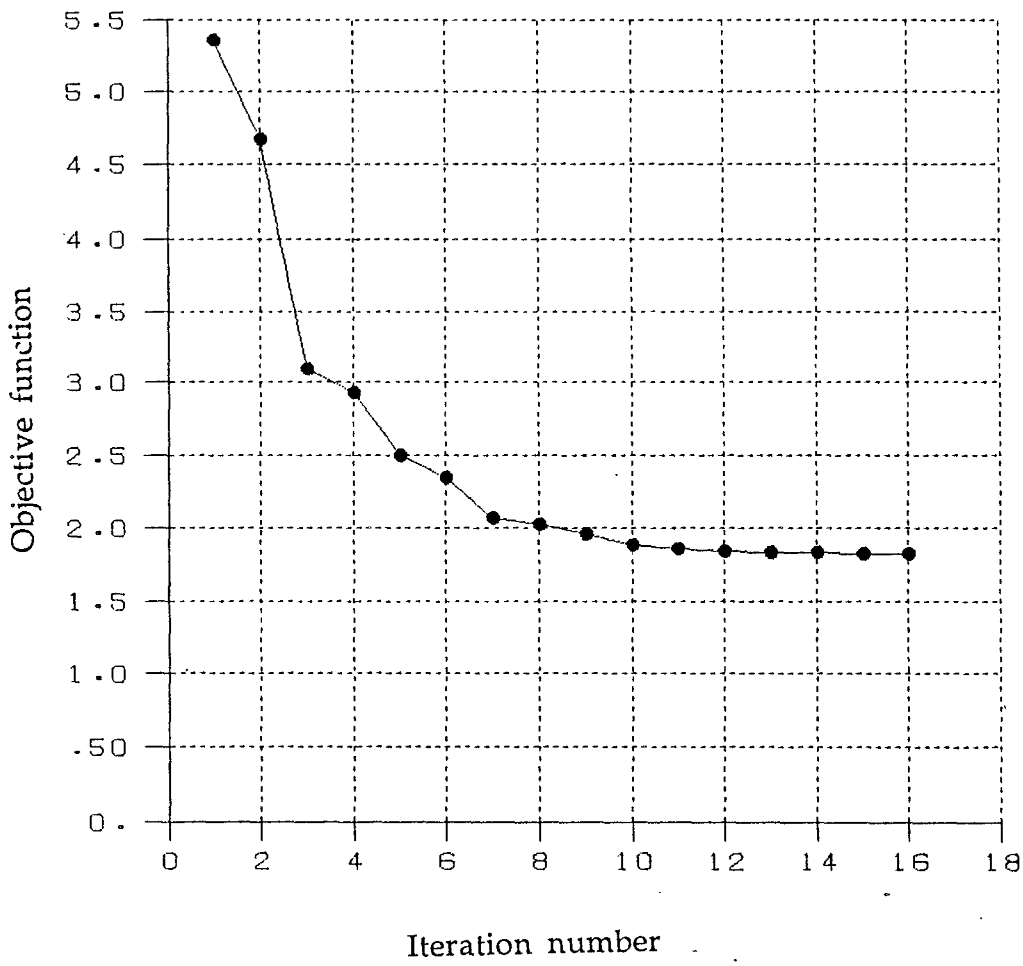


Fig. 12 Plot of objective function values against iteration number.

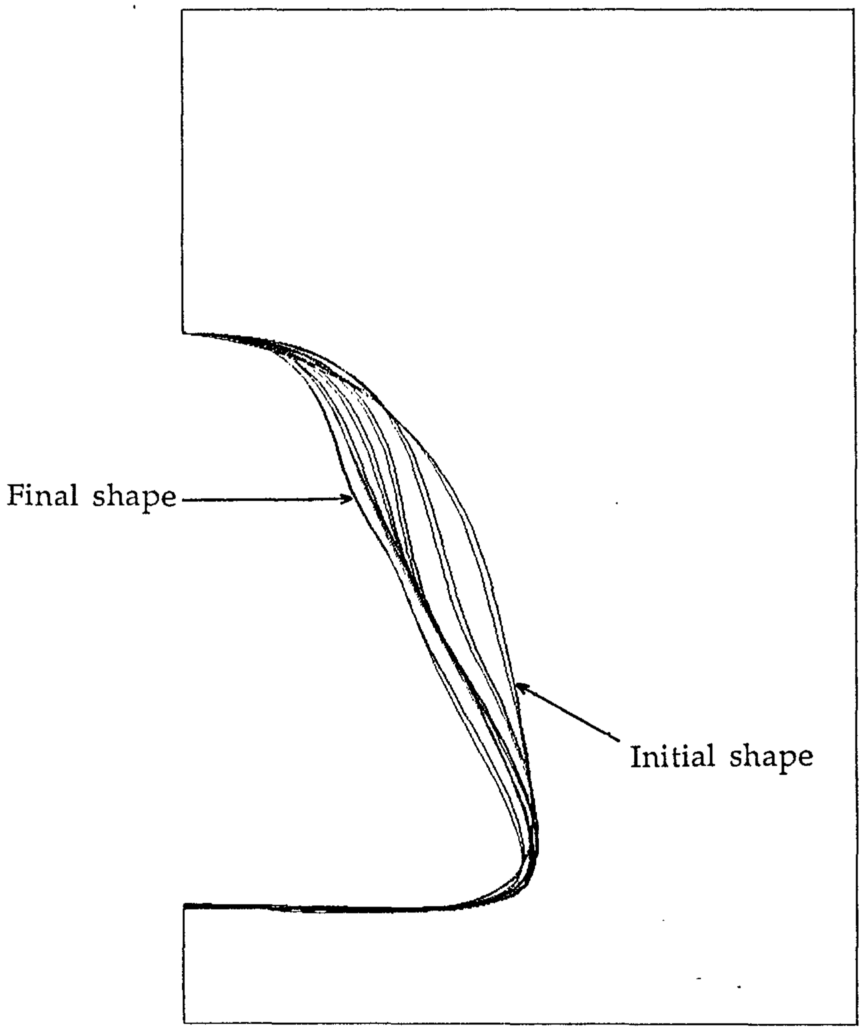


Fig. 13 Superposed plots of hole shapes for successive iterations

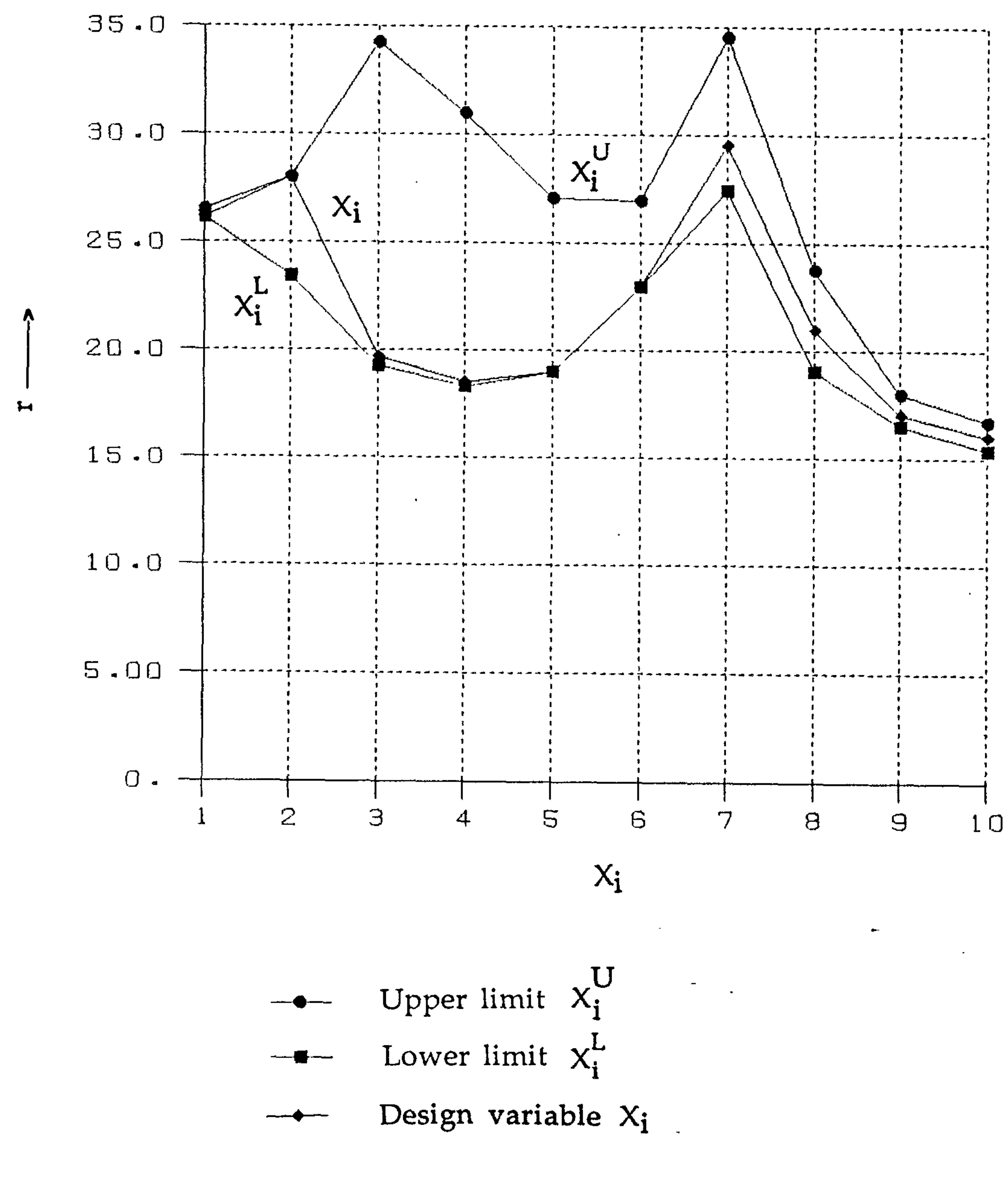


Fig. 14 Plot of design variables and side constraints for the hole shape of fig. 11

over the shape of the hole boundary. This can be done for recovering from errors (e.g., when the hole boundary takes on physically impossible shapes) or to explore several different paths taken by the optimization process starting from an intermediate hole boundary. In both cases the designer essentially inputs a new shape for the hole and then allows the optimization process to take over again. In case of a fatal error, as when the hole boundary crosses the sides of the beam, the finite element analysis will blow up. In such cases the interaction capability can be used to continue the process from a previous iteration after properly constraining the problem.

Finally it should be noted that designer interaction can increase the confidence of a designer in the iterative optimization process and encourage the adoption of the method by skeptics. Without the insight provided by interaction, experienced designers may never give optimization methods a fair evaluation for its ability to predict improved designs in complex situations.

2.2 CAD Approach

The CAD approach described here is a way of simulating, using computer graphics, the photoelasticity method as used by Durelli and described in chapter 3. The CAD approach is similar to the numerical method except that the optimization module of fig. 2 is replaced by a human designer, i.e., the shape variables are updated by the designer instead of being provided by the optimization algorithm.

On starting the CAD program, the data file containing the shape variables is displayed on a full-screen editor. The designer can input the starting values of the shape variables. On exiting the editor, control is transferred to the mesh regenerator. The mesh is automatically regenerated and a finite element analysis is performed. Control is then transferred to

PATRAN for postprocessing the results. A contour plot of the maximum shear stress corresponding to the photoelastic fringes is displayed. The shape variables (control points of the B-spline) show up as red circles (about ten rasters in size). From the plot, regions of low stress on the hole boundary can be easily identified. Plots of principal stresses can also be displayed as also the Von Mises stresses depending on what the analysis program is set up to do. The designer can also zoom in on a particular section of the beam to see magnified plots of the stress contours. This would be particularly useful in regions of high stress concentrations. Hardcopies can be made of any of these plots on a color plotter. Hardcopies can also be made of the data file containing the values of the shape variables corresponding to the current hole shape. The designer can now make a decision as to where to remove material from the hole boundary to obtain a more uniform stress distribution. On exiting PATRAN, the designer is prompted to make changes to the shape variables to effect the desired removal of material. The process is then repeated until a satisfactory boundary shape is obtained.

Appendix A contains information on using the CAD approach and the listing of some important sections of the code.

3 EXPERIMENTAL METHOD

The experimental method requires the manufacturing of transparent models of the tall beam and the use of photoelasticity to determine the stresses. The model is placed in a large-field circular polariscope using diffused light. Durelli was the first to use large size models and diffused light and the details can be found in the book, Durelli and Riley (1965). The optimization process involves the removal of material with a hand-held router and a hand file, while watching the changing fringes through a portable analyzer. The process is mainly interactive in nature.

Most experimental studies in stress distributions in elastic structures have dealt with the determination of the stress associated with a prescribed geometry. The method used here however, determines the geometry that satisfies prescribed conditions. In a sense, the design and analysis are performed simultaneously.

The optimization method using photoelasticity was developed by Durelli and co-workers. Details of the technique used in minimizing stress concentrations around stress raisers are discussed in papers published by in 1978-1980. Application to geometric discontinuities in stress fields were also published (1978). Examples of applications in the solution of central holes in finite and infinite plates subjected to in-plane loading can be found in Durelli and Rajaiah (1979, 1980a). Application to fillets and rings have been presented by the same authors (1980b).

3.1 Solution Process

3.1.1 Theoretical Basis

The arrangement of elements in a circular polariscope using diffused light is shown in fig. 15. The model is made of a transparent birefringent material which is loaded and observed through an analyzer using monochromatic light. When the elements of the polariscope are oriented as shown, alternate light and dark fringes can be seen on the model. The dark fringes are called isochromatics because each fringe shows the same color when white light is used. The condition resulting from the polariscope arrangement in fig. 15 is called the dark field, in which the polarization planes of the polarizer and analyzer are perpendicular. The isochromatics in a dark field can be ordered as 0, 1, 2, 3, 4, ... For a light field, the polarization planes are parallel and the isochromatics are enumerated as 0.5, 1.5, 2.5, 3.5, ... falling in between the fringes of the dark field.

The isochromatic fringes are contours of maximum shear stress (τ_{max}), i.e., all points on a fringe have the same value of maximum shear stress. For points not lying on a fringe, the partial fringe order can be obtained by using compensation techniques (Durelli and Riley, 1965). Once the fringe order for a point is known, the maximum shear stress at that point can be computed from:

$$\tau_{max} = n\, F_{\tau} \qquad \text{where}$$

F_{τ} = model shear stress fringe value (psi/fringe)

n = fringe order

The model shear stress fringe value F_{τ}, is determined by testing a calibration specimen for which the state of stress is known at a point, either theoretically or experimentally (section 3.1.2).

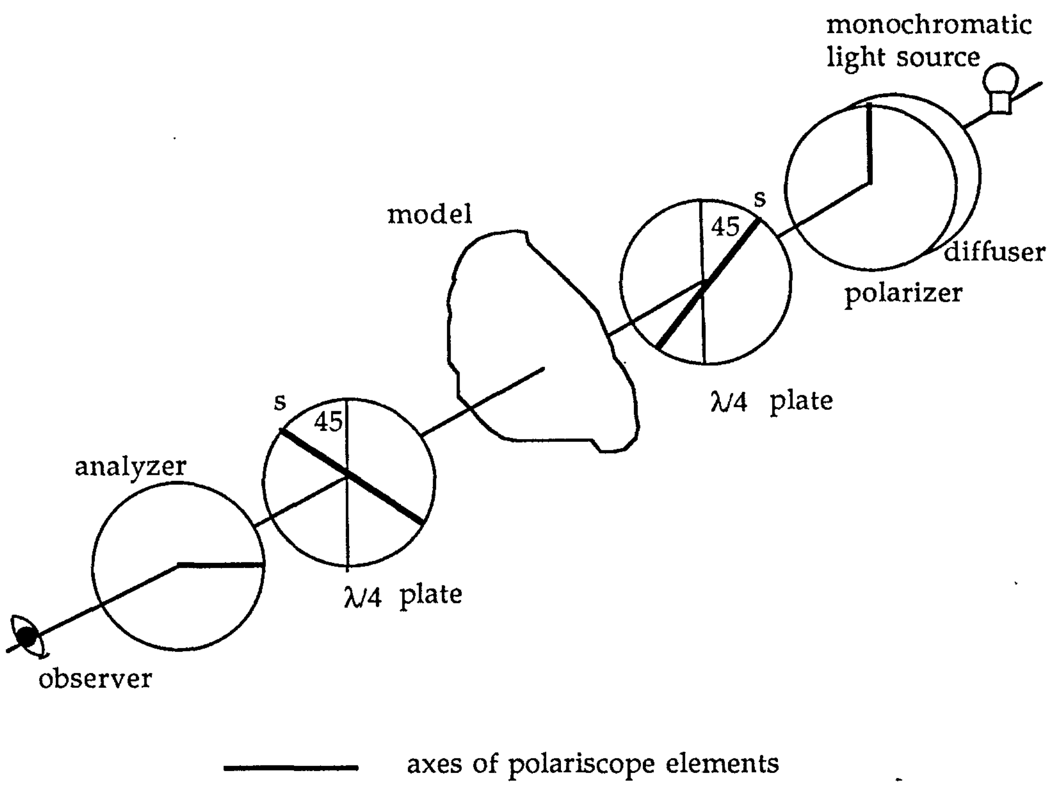


Fig. 15 Circular polariscope using diffused light (dark field)

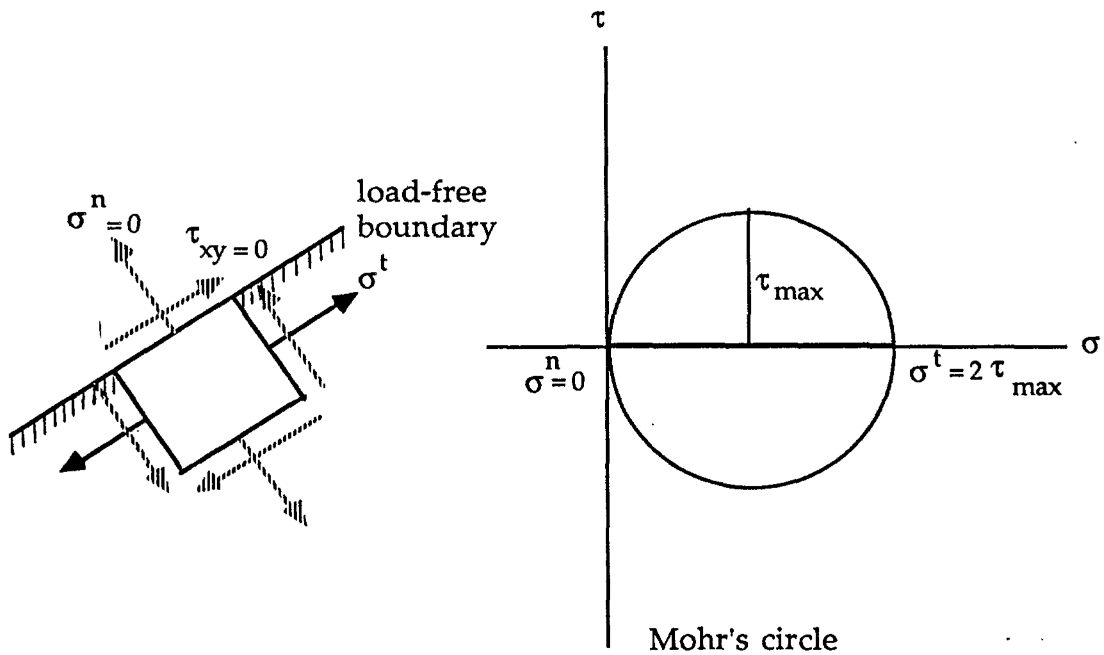


Fig. 16 Stress condition at a point on a load-free boundary.

The experimental method of shape optimization for the tall beam depends on direct viewing of the transformation of the stress patterns while geometric modifications are made. The objective is to obtain a uniform tangential stress distribution along the hole boundary which is load-free. As seen from fig. 16 , at any point along load-free boundaries the stress normal to the boundary σ^n which is a principal stress, is zero, and the tangential stress σ^t which is the other principal stress, is twice the magnitude of the maximum shear stress τ_{max} . Therefore, the fringes along load-free boundaries, give a direct measure of the tangential stress along the load free boundary without need for further computation. In other words, obtaining a uniform tangential stress distribution along the free boundary (the hole boundary in this case) is the same as making a single photoelastic fringe coincide with the boundary. The solution process as explained in the following sections aims at achieving this uniform tangential stress distribution within the constraints of the design.

3.1.2 Experimental Set-up and Preliminaries

The material used for the model is chosen as Homalite 100. Though other materials are available which are more sensitive, Homalite 100 is chosen for its ease of machinability.

The tall beam model is machined as a 3"x 4" specimen. It is placed in a loading frame (fig. 17) in between the two $\lambda/4$ plates (fig. 15) The loading is done using standard weights W (increments of 10 pounds) at one end of the cantilever. The actual load P acting on the model can be obtained by applying moment equilibrium at the pivot, which gives

$$P = \frac{Wy}{(y-x)}$$

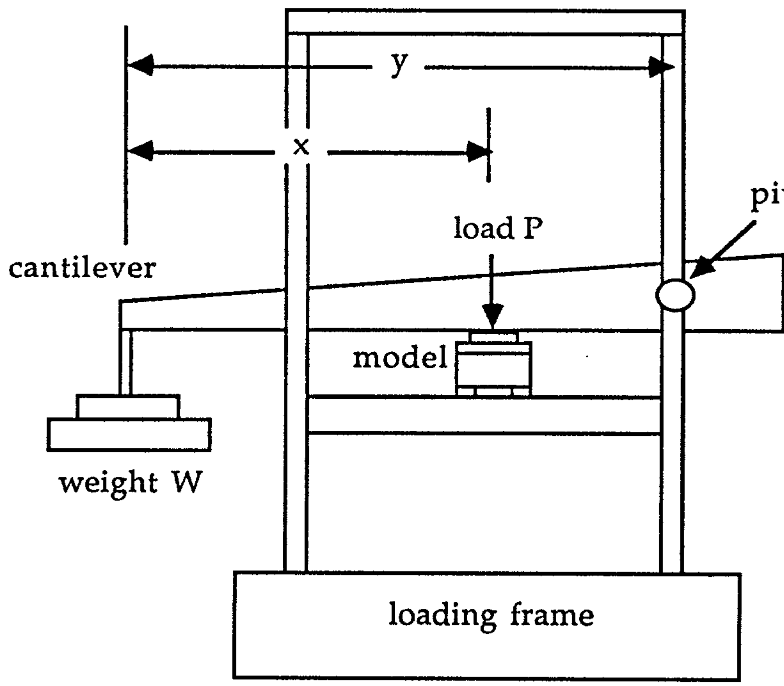


Fig. 17 Loading frame used with the polariscope

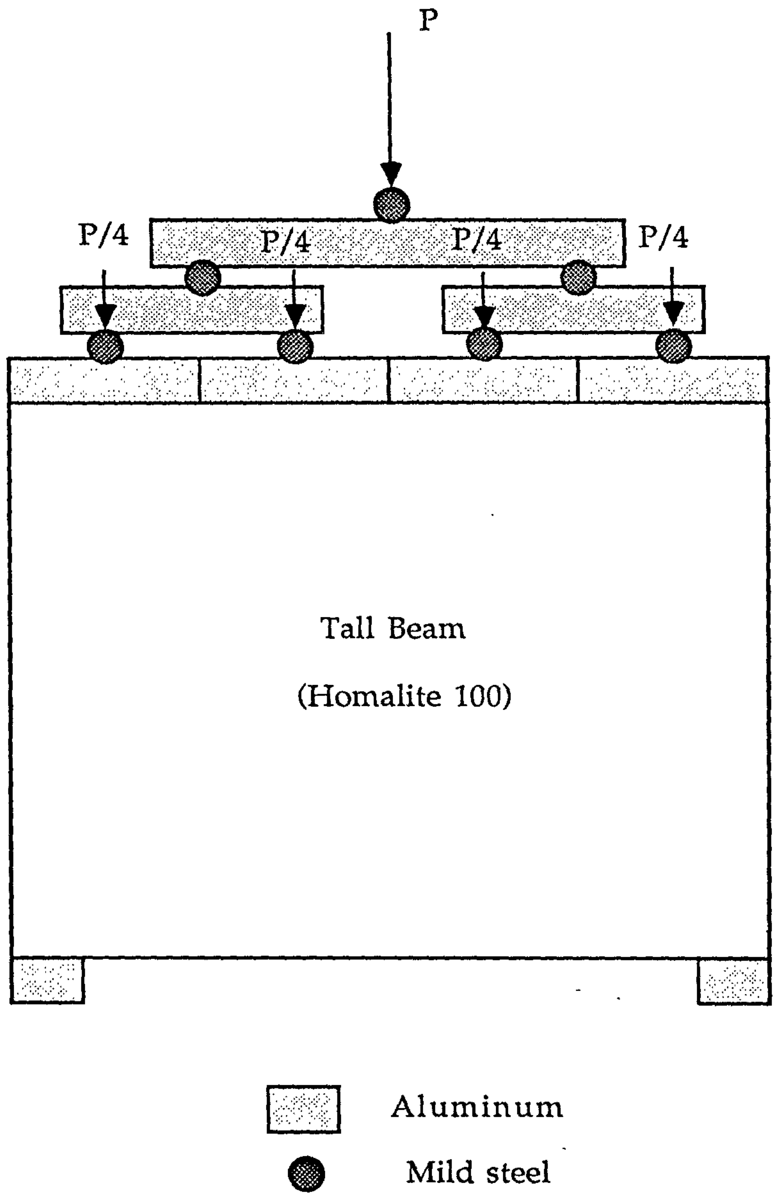


Fig. 18 Approximation to the uniform loading of the tall beam

The uniform loading of the beam is achieved by using an arrangement shown in Fig. 18. The loading is approximately uniform and by Saint Venant's principle the effect of any deviations from the uniform loading are negligible at some distance from the top boundary.

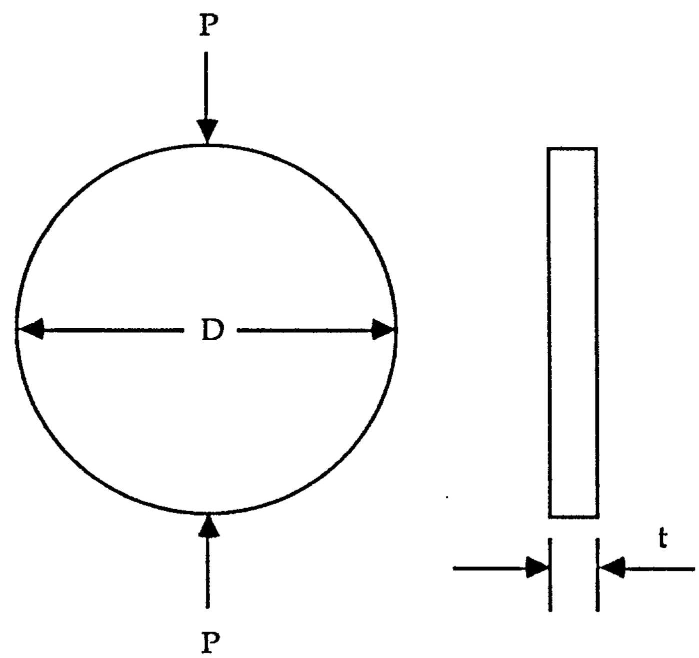
As the load is applied in fig. 17, the cantilever deflects slightly downwards. The load P which should have been vertical is no longer so and the stress distribution is not exactly symmetrical about the vertical axis of the beam. The problem can be solved by slightly tilting the model, i.e., raising the right side of the beam relative to its left side, until the fringes are symmetrical about the central vertical axis of the beam.

Another point to be noted is that frictional forces are present at the supports and the loaded boundary. The frictional forces change as the hole shape and size changes. The presence of frictional forces may affect the stress distribution everywhere in the beam and in particular at the supports and near the loaded top boundary. This makes hard a comparison with numerical results. Another problem is out-of-plane deflection of the beam due to slight eccentricities in the load P along the thickness of the beam. This becomes more pronounced as the hole size increases and the beam loses some of its original rigidity.

The beam is calibrated using a circular disk under diametral compression (fig. 19). The shear stress is known theoretically at the center (Timoshenko and Goodier , 1970) and is given by

$$\tau_{max} = \frac{4P}{\pi t D} = n\, F_{\tau}$$

F_{τ} can be computed when the fringe order n at the center of the disk is found with high precision using compensation techniques. The value was found to



$\tau_{max} = 4P/\pi tD = n F_\tau$ at the center of the disk

$t = 0.244''$

$D = 3.05''$

$P = 417.93$ lbs

$n = 2.333$ at center

$F_\tau = 320.16$ psi/fringe

Fig. 19 Calibration of Homalite 100 using a diametrically compressed disk be F_τ = 320.16 psi/fringe which agrees closely with previously published values for the material.

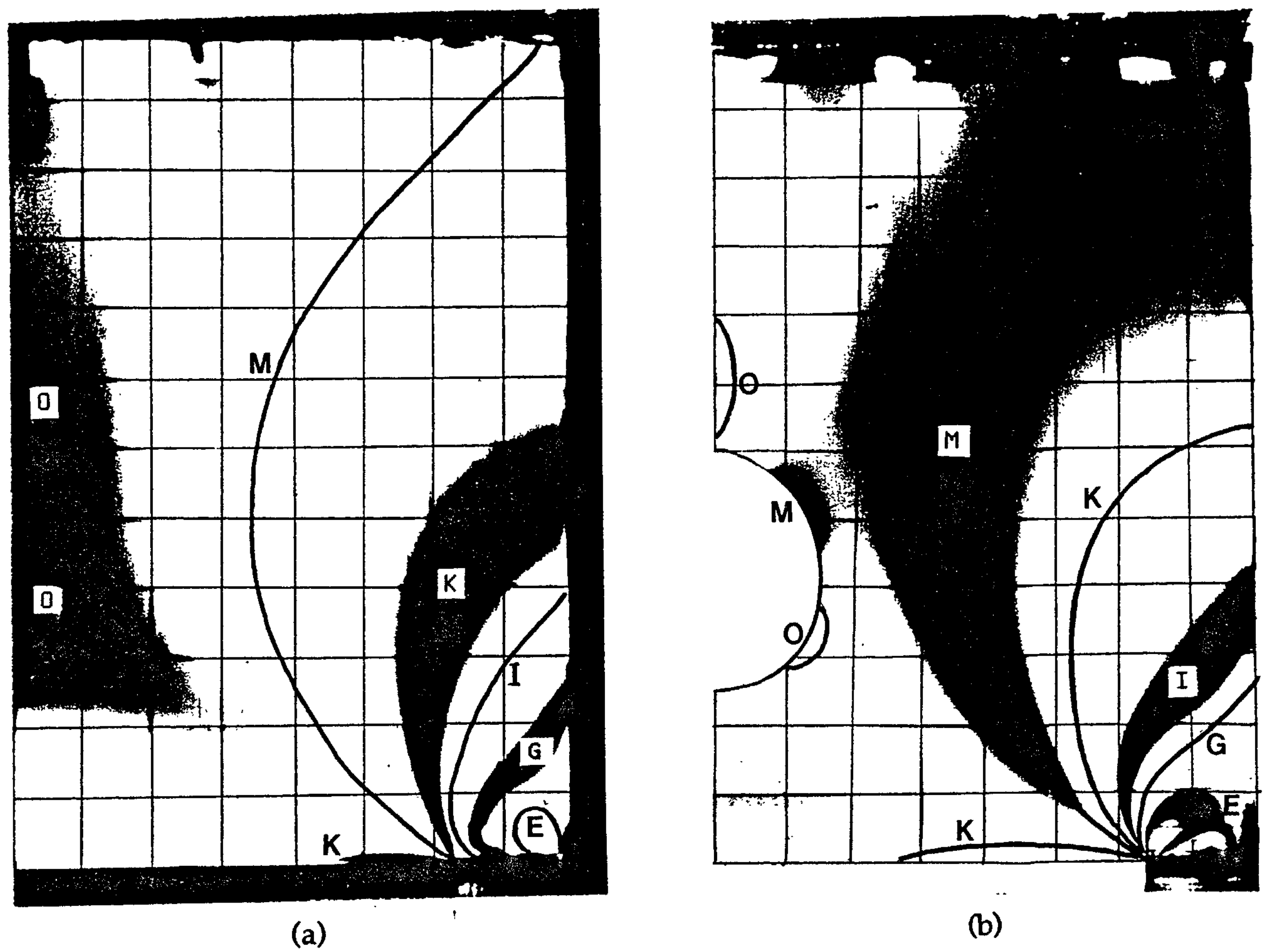
3.1.3 Optimization Process

The shape optimization process consists of systematic removal of material from the low stress regions of the hole boundary. The analyzer of Fig. 15 and the $\lambda/4$ plate next to it are replaced by a portable Polaroid film in the form of spectacles that can be worn by the operator. This allows the operator to watch the changing fringes as he/she modifies the geometry of the beam.

Fig. 20 (a), shows the beam in its initial state without the hole. The hole has to be started at some point within the beam. The presence of a zero isochromatic (zero maximum shear) on the vertical axis of the beam suggests that the hole may be started at this point. But even though the maximum shear stress is zero at this point, the individual stresses can be high, so that the machining of a circular hole could increase the individual stress by a factor of two. But since there is no other feasible point to start, the zero isochromatic is taken as the starting point for machining the hole. Fig. 20 (b) shows the result. It so happened that the two equal stresses on the boundary were relatively small and the presence of the hole did not appreciably increase the stress. The hole is enlarged as shown in fig. 21 (a). Material is removed from the parts of the hole boundary that have lower stress than the rest of the hole boundary. In fig. 21 (a), for example the stress at the vertices of the triangular shaped hole is close to zero and is lower than in the adjacent parts of the boundary. Material is therefore removed from these areas resulting in a hole shape as shown in fig. 21 (b).

Note that the stress along the hole boundary of fig. 21 (b) is almost constant. The lateral sides of the hole boundary are in compression (i.e., the tangential stress at the boundary is negative), while the bottom boundary of the hole is in tension. In this case the tensile and compressive stresses have the same value, but in general they can be different. The compressive and tensile stresses are handled separately by independently making them uniform (i.e., make the tensile and compressive fringes coincide along their respective boundaries). The degree of uniformity of stresses along the boundary can be measured by the coefficient of efficiency K_{eff} (Durelli et al., 1978).

While removing material along the hole boundary, care must be taken not to exceed the limit on the maximum tensile stress. The maximum tensile stress is assumed to occur on the load free boundaries indicated on fig. 10 of chapter 2, and not within the interior of the beam for the present loading conditions. This assumption has been found to be true for all hole shapes via the numerical method. Therefore attention must be kept on the fringe orders occurring on the free bottom edge of the beam next to the support and also on the hole boundary. The fringe orders at these points should not increase beyond the fringe order corresponding to the limiting value of tensile stress. From fig. 20(a), the maximum tensile stress in the solid beam has a fringe order of 0.5. For the hole shapes shown in figs. 20 (b) and 21 (a), (b) the limit on the tensile stress has not yet been reached at any point on the hole and bottom boundaries. On the bottom edge of the beam the limit has been reached but not exceeded. As the hole grows bigger, higher order fringes coincide with the hole boundary and the maximum tensile stress limit is reached even on the hole boundary (chapter 4). Note that the photoelastic figures in this chapter have a lower load p and show a lower response in terms of the number of fringes than the figures of chapter 4.



$\frac{\tau_{max}}{p}$		fringe order
3.47	A	3.5
3.22	B	
2.98	C	3.0
2.73	D	
2.48	E	2.5
2.23	F	
1.98	G	2.0
1.74	H	
1.49	I	1.5
1.24	J	
.992	K	1.0
.744	L	
.496	M	.50
.248	N	
0.	O	0

Fig. 20 Photoelastic photograph of the right half of the solid beam (fig. 20(a)) and with a hole cut out at the zero isochromatic (fig, 20(b))

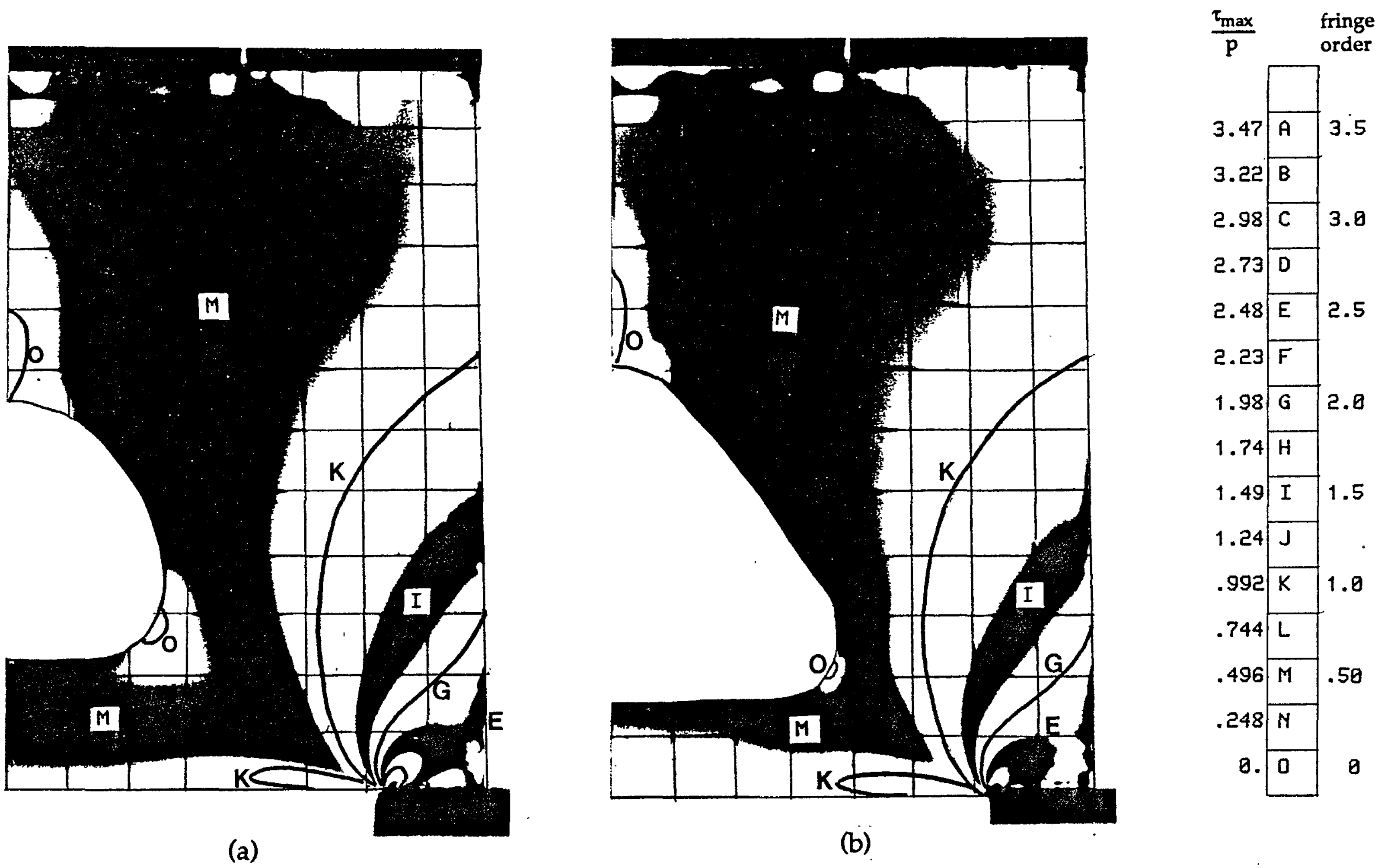


Fig. 21 The beam with an enlarged hole (fig. 21(a)), and with more material removed from the low stressed corners of the hole (fig. 21(b))

During the optimization process, it is observed in general that as material is removed from the low stress regions of the boundary :

1. The maximum value of stress somewhere else on the boundary decreased.
2. The position of the low stress changed.
3. The gradient of stress at the point increased.

When the methodology described above is pursued, several different boundary shapes with coinciding fringes (increasing progressively in fringe order) are obtained (chapter 4). Each of these shapes can be considered as an optimum in terms of obtaining a uniform stress distribution. In terms of efficient material usage, however, the last of the hole shapes is the optimum[1] as it achieves maximum weight reduction without violating any of the constraints. Any further attempt to remove material from the low stress regions increases the maximum tensile stress beyond the acceptable limits.

The results of the experimental approach are presented in chapter 4 and compared with the numerical results.

[1] There may be other optimum holes if multiple discontinuities are considered or if material is removed from the vertical edges or the load free part of the bottom edge of the beam.

4 RESULTS

In this chapter the numerical and experimental results are presented and compared. The solid beam is analyzed first to establish the stress constraints and to identify the position at which to start machining the hole. This is followed by the various optimized hole shapes and comparison and discussion of the results.

4.1 Solid Beam

The photoelastic photograph of the right half of the solid beam is shown in fig. 22(a). The fringes shown are those from the light field, with the dark field fringes superimposed on it. The value of the maximum shear corresponding to a fringe is given by the fringe order times a constant.

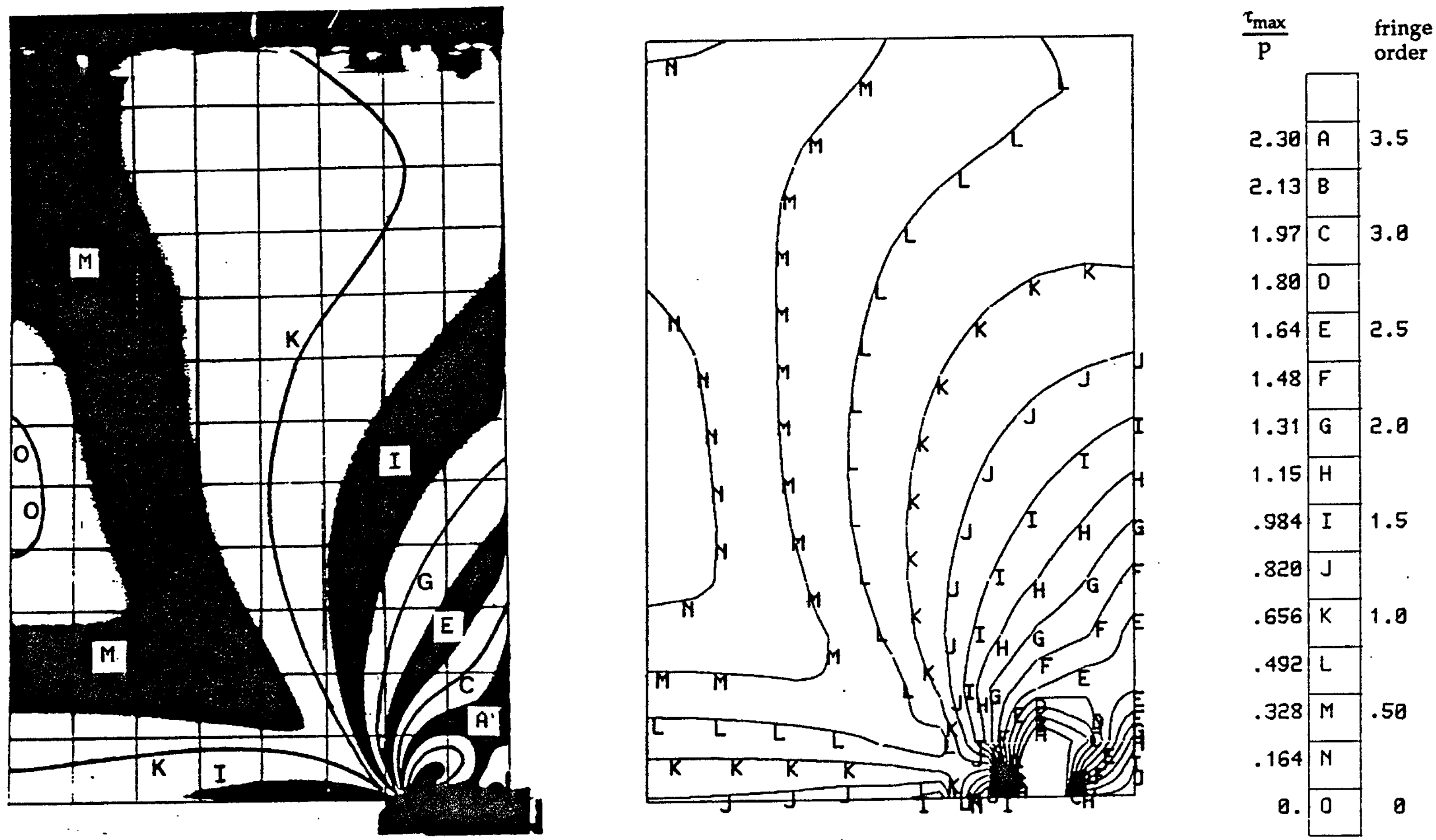
$$\frac{\tau_{max}}{p} = 0.6566\ n$$

where p is the uniform loading in psi and n is the fringe order.

The following points can be noted:

1. The stresses in the beam just above the support are compressive in nature.
2. The maximum tensile stress occurs just to the left of the support on the bottom boundary .
3. The fringe order of the maximum tensile stress is 1.5 ($\sigma^t/p = 1.97$).
4. The zero isochromatic occurs at two points on the vertical axis of the beam near the center.

Experimentally, material removal is started at the position of the zero isochromatic. The constraint that the maximum tensile stress in the beam



(a) (b)

Fig. 22 Photoelastic fringes (fig. 22(a)) and numerical contours (fig. 22(b)) of the maximum shear stress for the right-half solid beam should not increase beyond that in the solid condition, dictates that the fringe order at the tensile portions of the boundary should not exceed a value of 1.5 ($\sigma^t/p = 1.97$) at any stage of optimization. As mentioned earlier, this is under the assumption that the maximum tensile stress occurs only on the load free boundaries, and therefore the fringe order directly gives the magnitude of the maximum tensile stress without need for further calculation. From previous experience the assumption that the maximum tensile stress occurs only on free boundaries, is reasonable with the possible exception of the contact surface at the supports.

The numerical τ_{max} contours for the solid beam are shown in fig. 22(b). The boundary conditions for the finite element model of the right half solid beam is shown in fig. 23(a). Displacements are constrained at the boundary nodes according to fig. 23(a). The loading is achieved by nodal loads calculated from a line integral of the pressure load using the isoparametric shape functions for the element. The equivalent nodal loads for a linearly varying pressure loading of an edge, are shown in fig. 23(b). The displacement boundary conditions are also shown. X direction displacement is constrained for all nodes on the vertical left edge of the beam. Y direction displacement is constrained at the support on the bottom right edge of the beam. A support reaction of 11% of the total load on the top, is applied at the support nodes in the direction shown. This value was obtained by balancing the forces in the photoelastic specimen for X direction equilibrium. The resulting contours of shear stress corresponding to the fringes of the photoelastic model are shown in fig. 22(b). The numerical results match the experimental results quite closely. Several different support reactions were tried out , and the value of 11%, matched the experimental results best.

The following points can be noted:

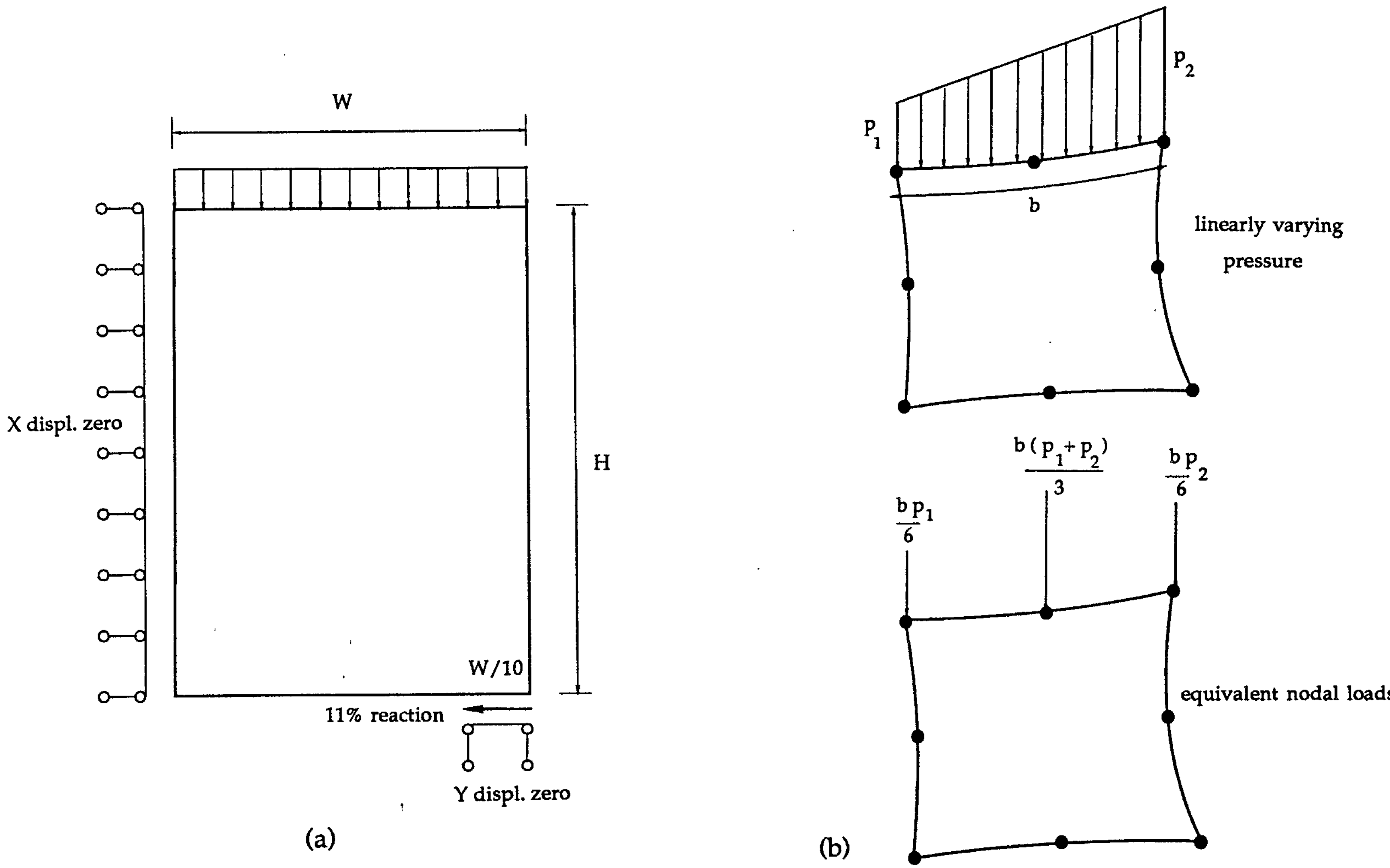


Fig. 23 Boundary conditions for the finite element model of the right-half solid beam (fig. 23(a)) and the equivalent nodal loads corresponding to a pressure load (fig. 23(b))

1. The maximum tensile stress of fringe order 1.5 occurs just to the left of the support.
2. The two zero isochromatics of the experimental results is replaced by small values of stress on the vertical axis.
3. It was verified from a contour plot for principal tensile stress (similar to the plot in fig. 22(b)) that the maximum tensile stress indeed occurs on the bottom boundary of the beam to the left of the support and not in its interior.
4. The shear stress contours do not match the experimental fringes quite precisely at the support and near the loaded top boundary. The numerical fringes are lower by about 0.25 fringe order.

The discrepancies in the two figures can be attributed to :

1. Inability of the finite element method to model the experimental boundary conditions and loads.
2. The presence of frictional forces at the loaded boundaries and supports and out-of-plane bending in the experimental model. Further, these forces change as the hole shape is modified.
3. Inherent inaccuracies and approximations of the finite element method such as extrapolation and averaging of stresses.

4.2 Optimum Hole Shapes

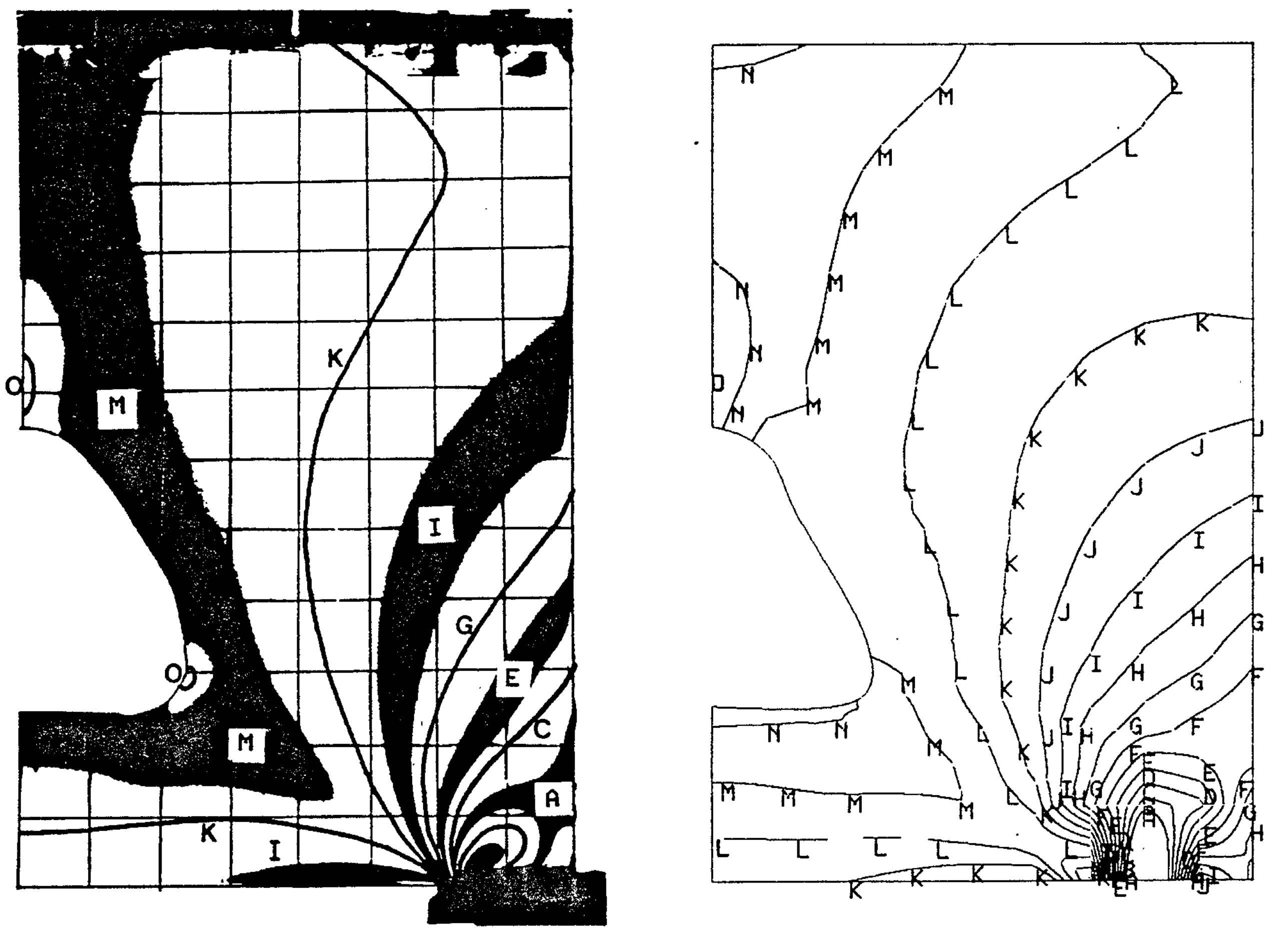
The experimental optimization of the hole shape is performed as described in chapter 3 starting at the zero isochromatic and removing material from the low stress regions of the boundary. The procedure is continued until a single fringe coincides with the boundary. When the tangential stress along the hole boundary changes sign, as is the case here, then each fringe is made to coincide with their respective part of the boundary.

Figs. 24(a), 25(a), 26(a), and 27(a), show four such optimized hole shapes with progressively increasing fringe orders coinciding with the lateral hole boundary. It can be observed that:

1. In all the figures, the tangential stress along the horizontal boundary of the hole is tensile, while the tangential stress along the lateral boundaries is for the most part compressive.
2. The tensile and compressive stresses are separated by a zero isochromatic which can be easily identified as the only black fringe when using white light instead of monochromatic light. (The other fringes are colored).
3. The fringe order of the horizontal hole boundary tensile stress is 0.5 in fig. 24(a), and 0.8 in all the other cases which is below the 1.5 limit.
4. In figs. 25(a), 26(a) and 27(a), there is a brief region of tensile tangential stress at the top edge of the hole, with fringe orders of 0.4, 0.8, and 1.5, respectively. In fig. 27(a), therefore, the tensile stress has reached the 1.5 limit.
5. In all the figures the tangential stress on the bottom edge of the beam to the left of the support has a fringe order of 1.5. In fig. 27(a), any attempt to remove more material along the hole boundary increases the tensile stress beyond the 1.5 limit both at the bottom edge of the beam (to the left of the support) and at the top edge of the hole.

The numerical optimization technique as explained in chapter 2 requires the formulation of the constraints and objective function and a starting hole shape input by the designer.

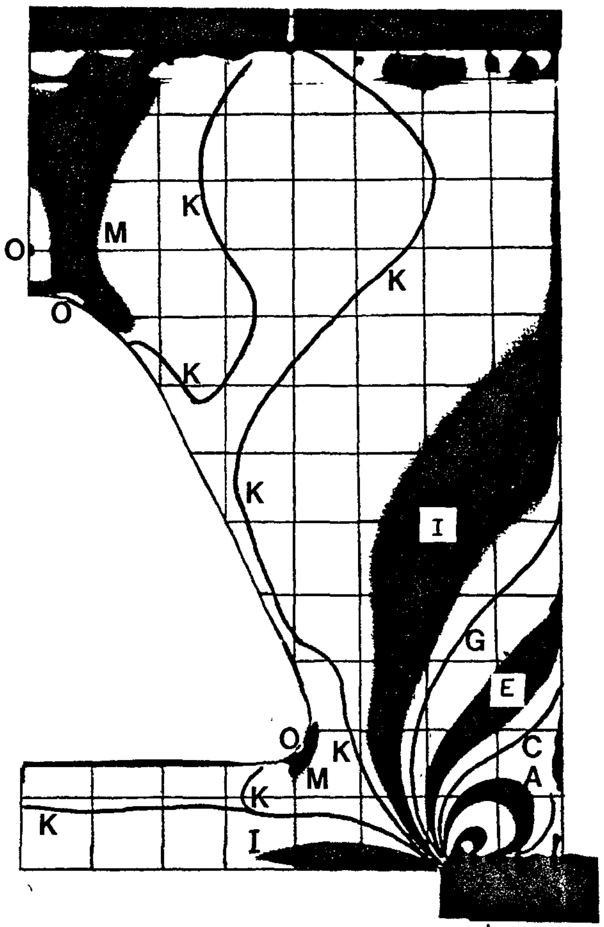
Fig. 24(b), 25(b), 26(b) and 27(b), show the different hole shapes obtained from the numerical technique with a horizontal reaction of 5% the total load applied. These are quite similar to the corresponding experimental



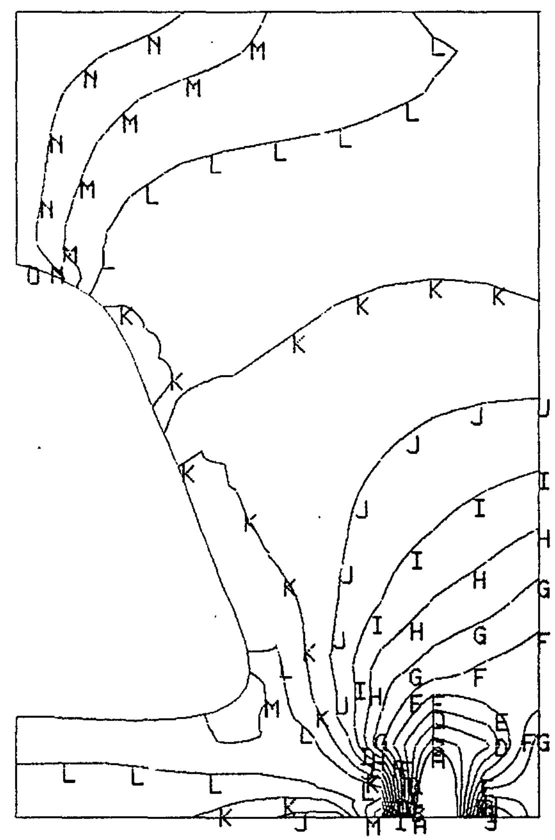
(a) (b)

$\frac{\tau_{max}}{P}$		fringe order
2.30	A	3.5
2.13	B	
1.97	C	3.0
1.80	D	
1.64	E	2.5
1.48	F	
1.31	G	2.0
1.15	H	
.984	I	1.5
.820	J	
.656	K	1.0
.492	L	
.328	M	.50
.164	N	
0.	O	0

Fig. 24 Photoelastic fringes (fig. 24(a)) and numerical contours (fig. 24(b)) for the first of the optimum hole shapes.



(a)



(b)

$\frac{\tau_{max}}{p}$		fringe order
2.30	A	3.5
2.13	B	
1.97	C	3.0
1.80	D	
1.64	E	2.5
1.48	F	
1.31	G	2.0
1.15	H	
.984	I	1.5
.820	J	
.656	K	1.0
.492	L	
.328	M	.50
.164	N	
0.	O	0

Fig. 25 Photoelastic fringes (fig. 25(a)) and numerical contours (fig. 25(b)) for the second optimum hole shape.

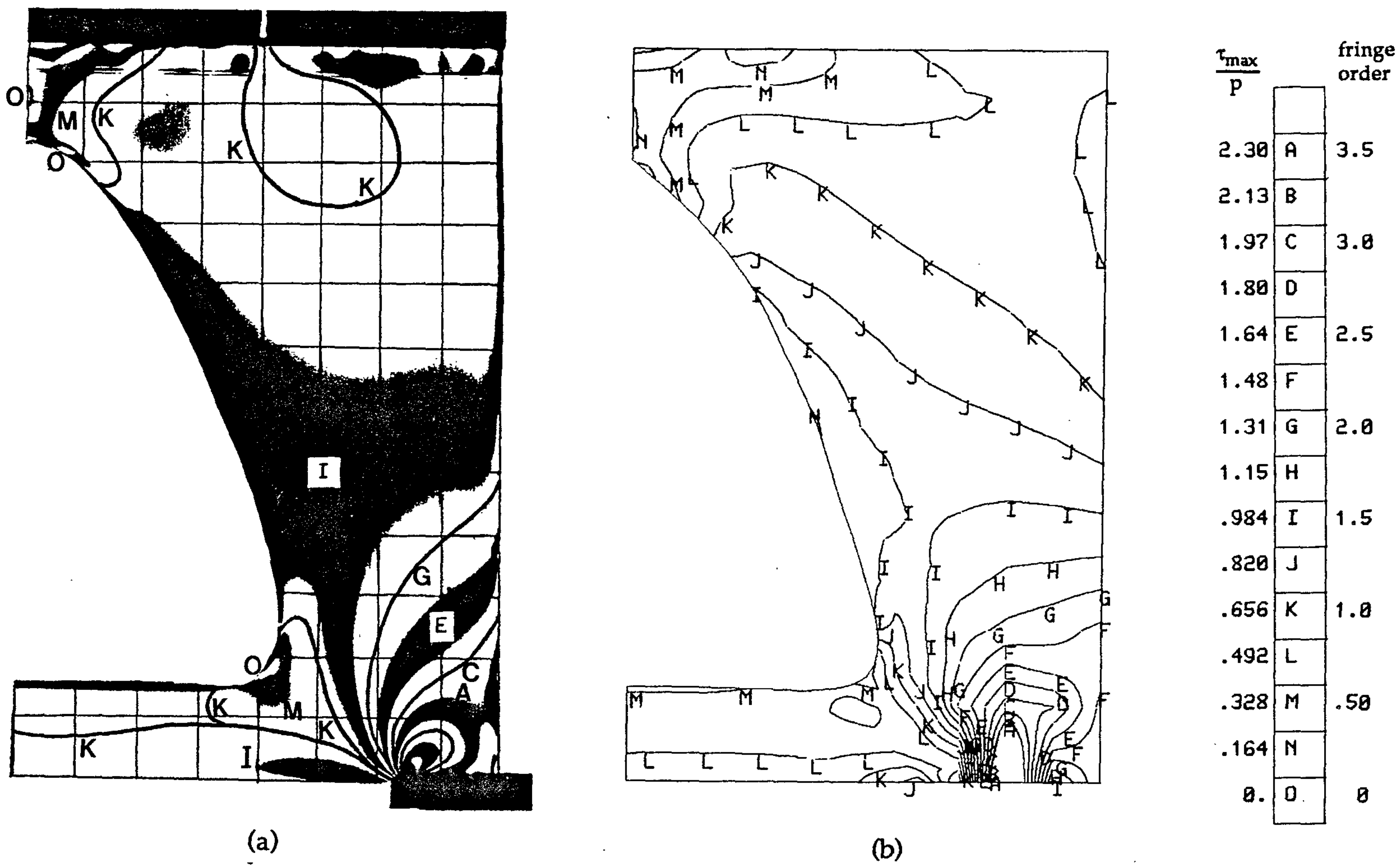
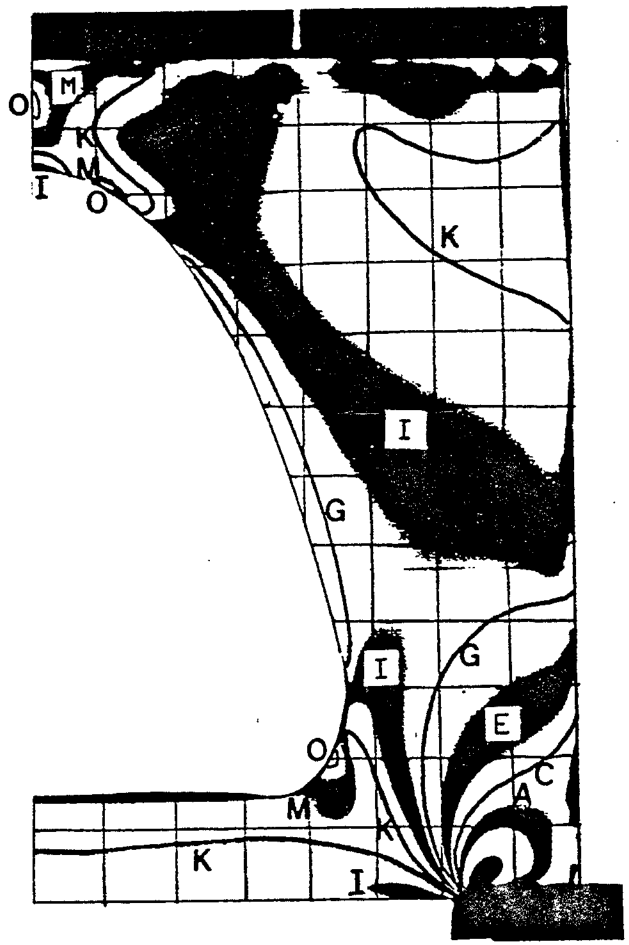
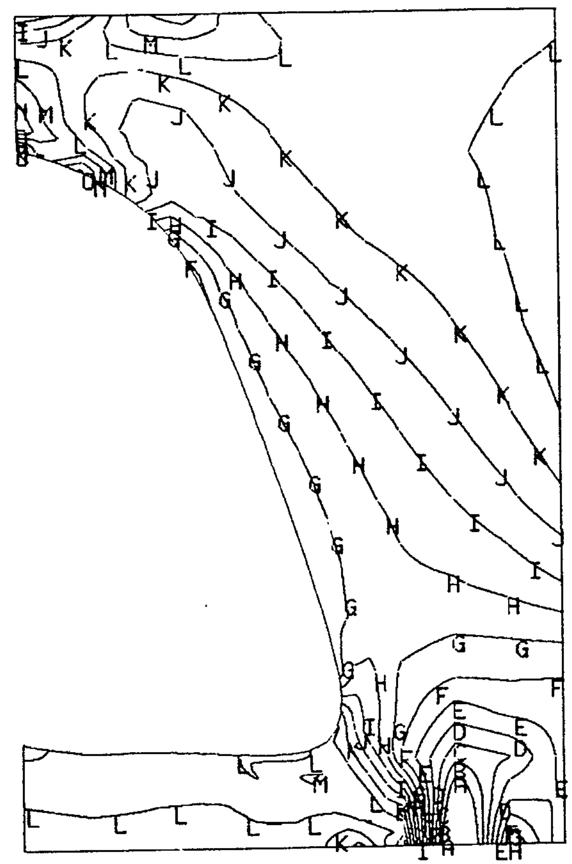


Fig. 26 Photoelastic fringes (fig. 26(a)) and numerical contours (fig. 26(b)) for the third optimum hole shape.



(a)



(b)

$\frac{\tau_{max}}{p}$		fringe order
2.30	A	3.5
2.13	B	
1.97	C	3.0
1.80	D	
1.64	E	2.5
1.48	F	
1.31	G	2.0
1.15	H	
.984	I	1.5
.820	J	
.656	K	1.0
.492	L	
.328	M	.50
.164	N	
0.	O	0

Fig. 27 Photoelastic fringes (fig. 24(a)) and numerical contours (fig. 24(b)) for the fourth and last of the optimum hole shapes.

figures, except for the expected variations at the support and loaded top boundary. The following observations are made:

1. There is close resemblance in the hole shapes and the shear stress contours (fringes) with the experimental results and the observations about the stress patterns for the experimental cases, hold true here as well.
2. Whereas the experimental method only removes material, the numerical method may also add back previously removed material as the hole gradually approaches the optimum hole shape.
3. The zero isochromatics are not as pronounced as in the experimental results due to discrete sampling of the stresses (at the node points only) and due to the extrapolation and averaging mentioned in chapter 2. As a result of this, fringes of a small value ($\approx$ 0.25) replace the zero isochromatics at the hole boundary at some places.
4. The efficiency of the optimizer decreases as the optimum is approached and several further iterations may be required to converge to an optimum. Once a near optimum shape is obtained however, the designer can either use manual interaction or the CAD approach to mimic the experimental approach on the computer, for obtaining a constant stress boundary.
5. It should be noted that the B-spline cannot model sharp curvatures as explained in fig. 6, chapter 2. The top part of the hole in fig. 26(a), for example, will require more control points in the vicinity of the top part for exact numerical modelling.

Fig. 28(a)-(d) shows the iteration history (objective function versus iteration number) of the numerical approach corresponding to each of the figs. 24(b) - 27(b). The objective function is the variance of boundary stress. It is

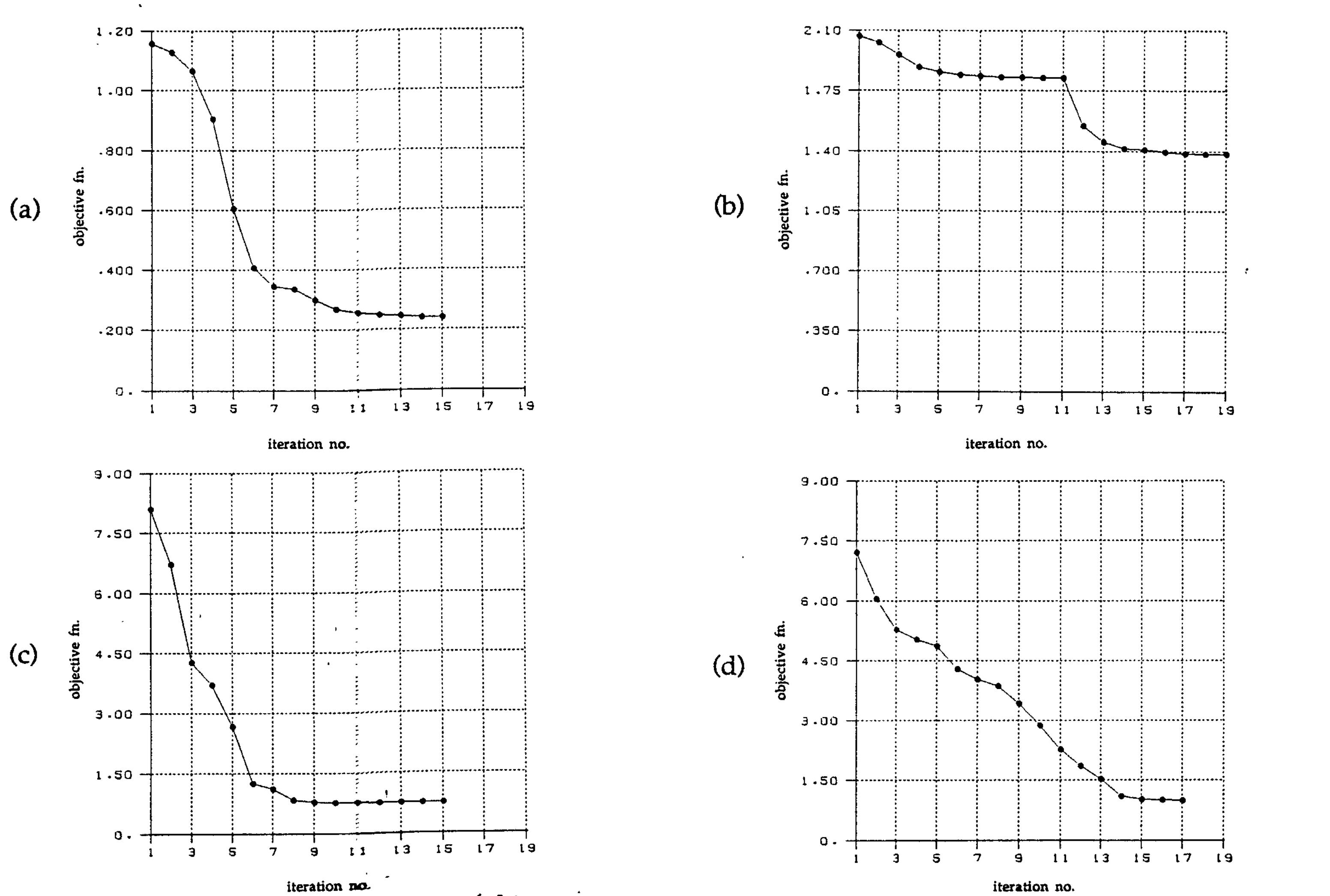


Fig. 28 Plots of iteration histories (objective function versus iteration number) for the numerical method corresponding to the hole shapes of figs.24(b)-27(b)

clearly seen that the optimizer works quite efficiently in the early stages but becomes sluggish towards the end.

Fig. 29(a)-(d) shows the design variables (corresponding to each of the optimum hole shapes of figs. 24(b) - 27(b)) and the side constraints. These side constraints are an essential part of the optimization process. They ensure that the hole does not converge to the optimum shape closest to the starting point, but rather converges to the optimum hole shape in the region[1] defined by the side constraints. From these plots the designer can verify if the design has been artificially bounded by the side constraints. Note that the design variables touch the side constraints only at the two extreme ends of the hole. However if the design variables touch the side constraints anywhere in between, then the side constraints may be preventing the hole from changing shape to attain a constant boundary stress distribution. This is avoided by looking at the side constraint plots at the end of each iteration and changing the side constraints if necessary. In fig. 28(b) a sharp drop in the objective function is observed. This is because the side constraints were preventing the hole shape from changing, causing the objective function to stagnate around a value of 1.8. As soon as the side constraints were modified after iteration 11, the objective function improved again until the optimum hole shape was reached.

Fig. 30 shows the variation of the tangential stress of the hole boundary ($\sigma^t=2\tau_{max}$) with angle θ for the experimental hole shapes. Fig. 31 shows the corresponding results for the numerical hole shapes. In fig. 31 the tangential stress is sampled at 54 node points along the boundary.

---

[1] Here the region is defined so as to match each of the experimental hole shapes. In the absence of experimental results a new region can be defined each time to find different optimum shapes.

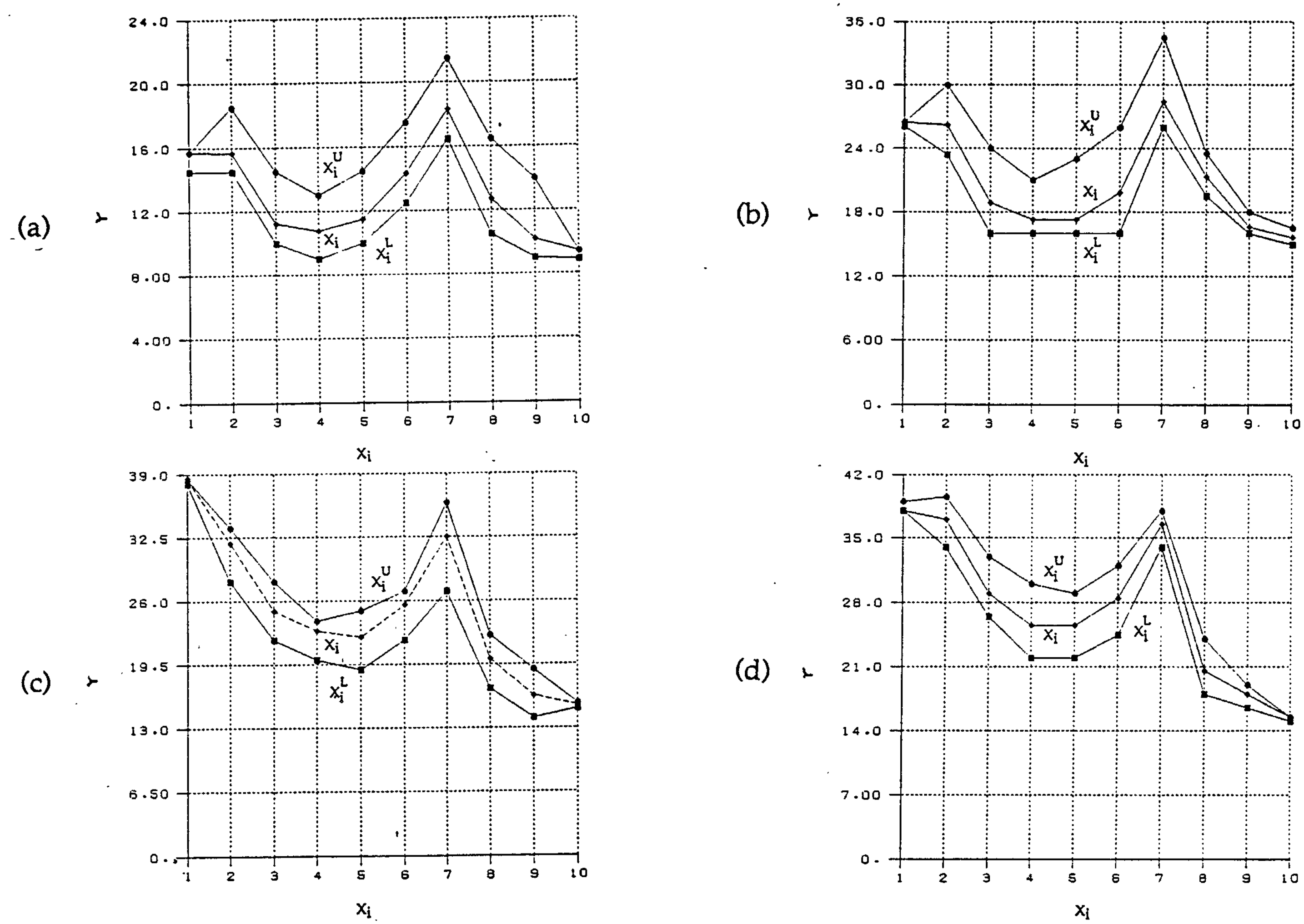


Fig. 29 Plots of design variables and side constraints for the numerical method, corresponding to the hole shapes of figs. 24(b)-27(b)

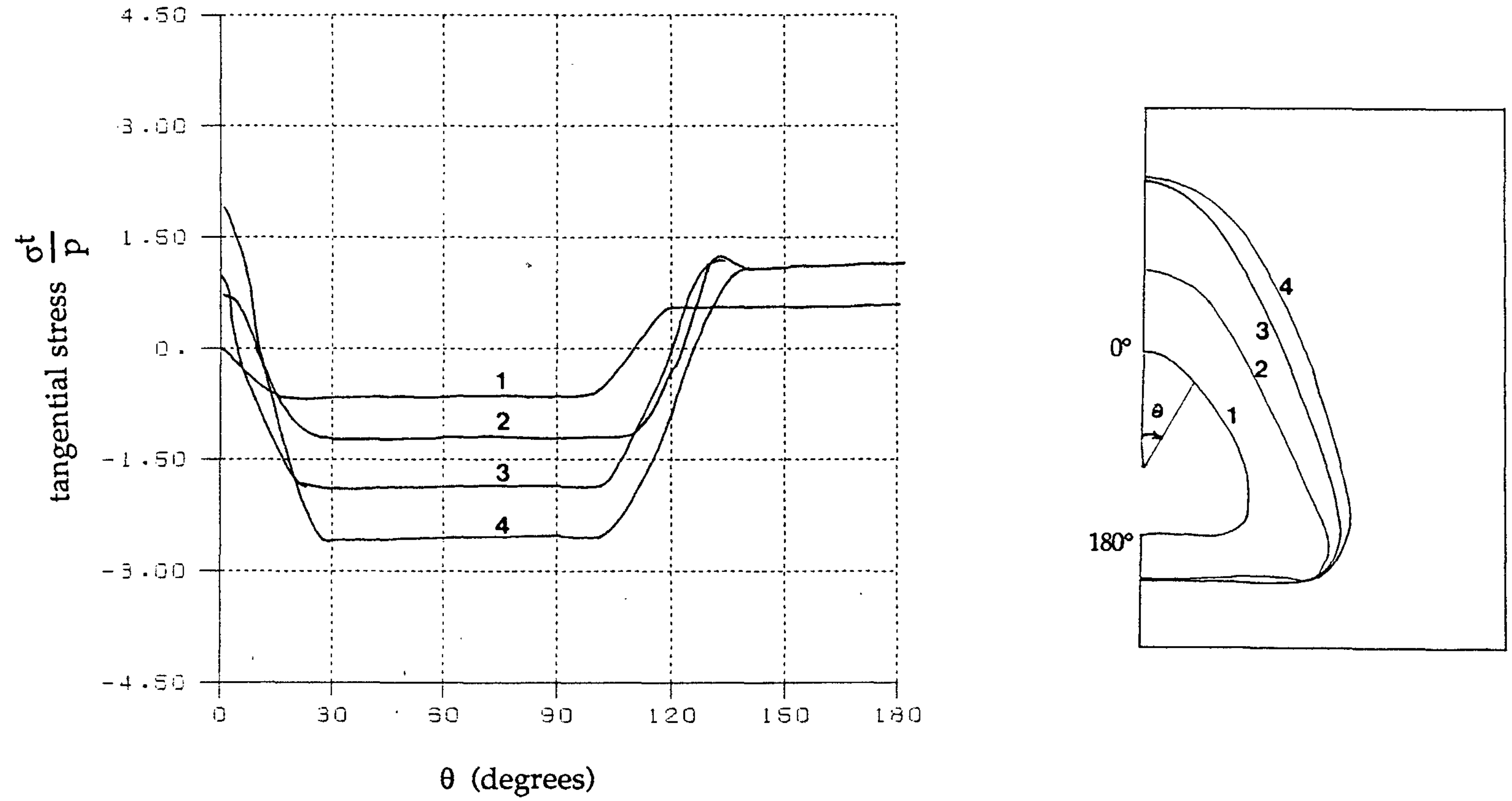


Fig. 30 Variation of the stress tangential to the hole boundary with angle θ for the experimental hole shapes of figs. 24(a) - 27(a).

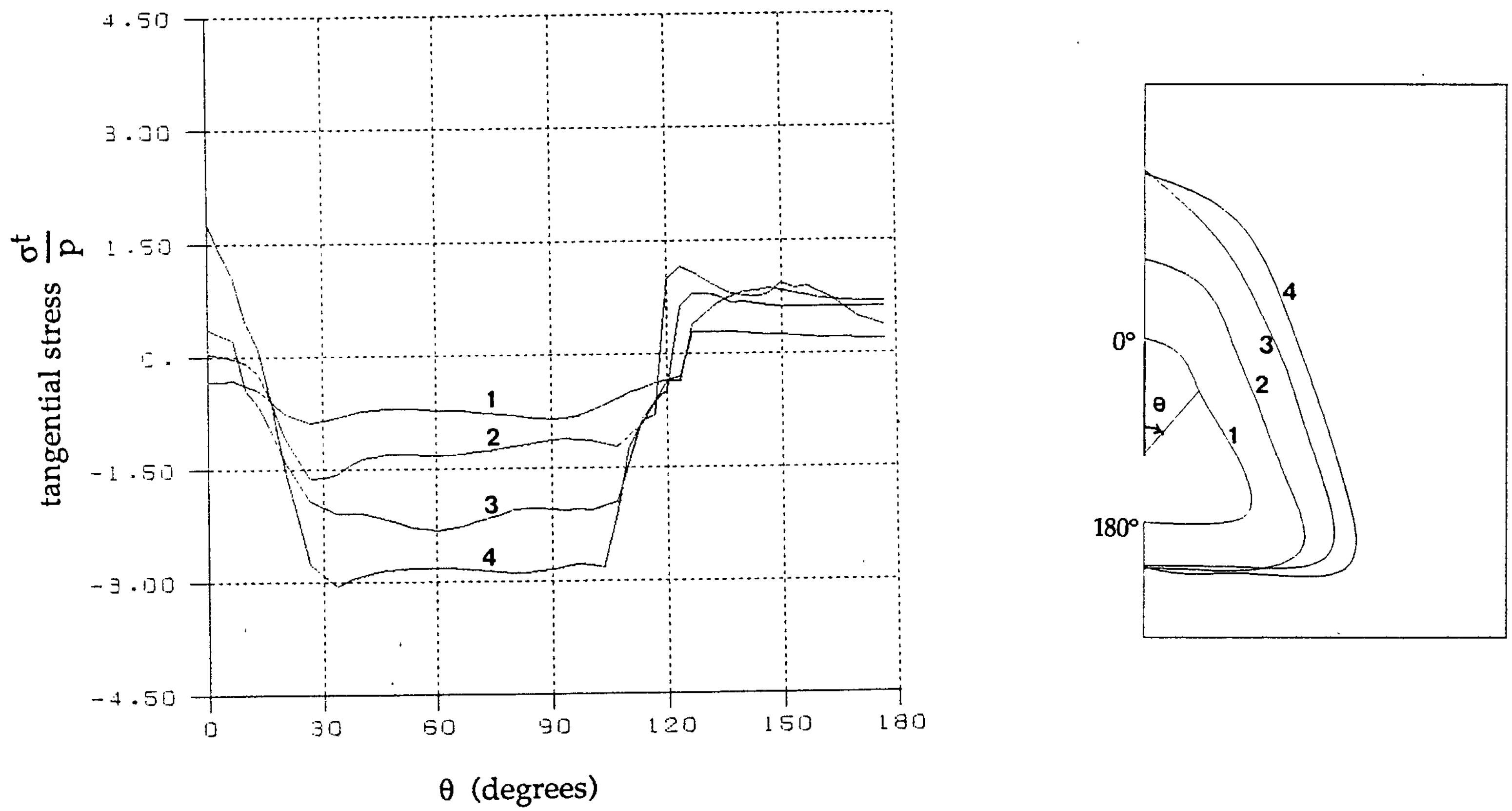
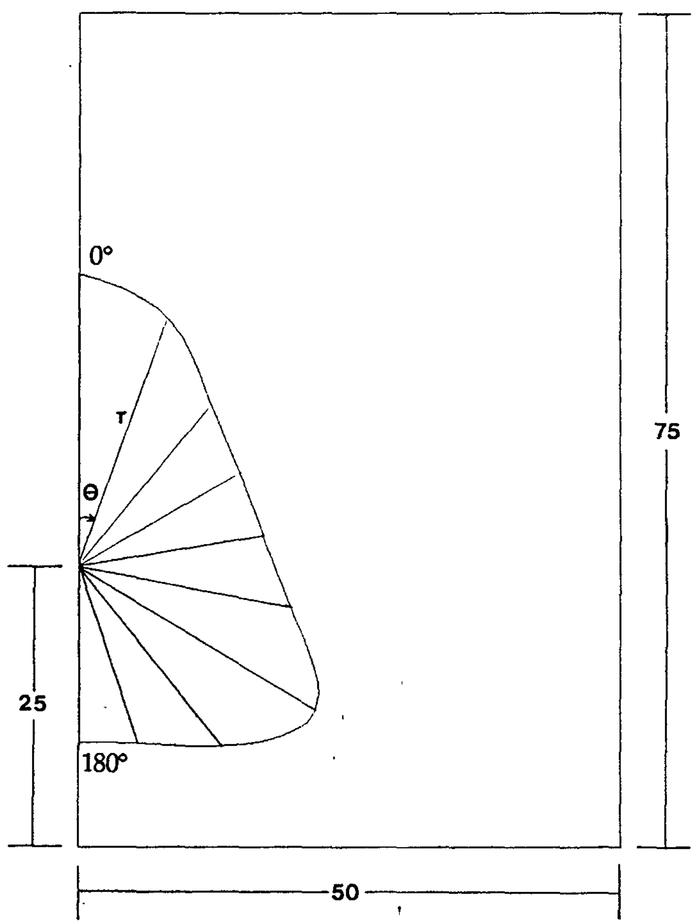
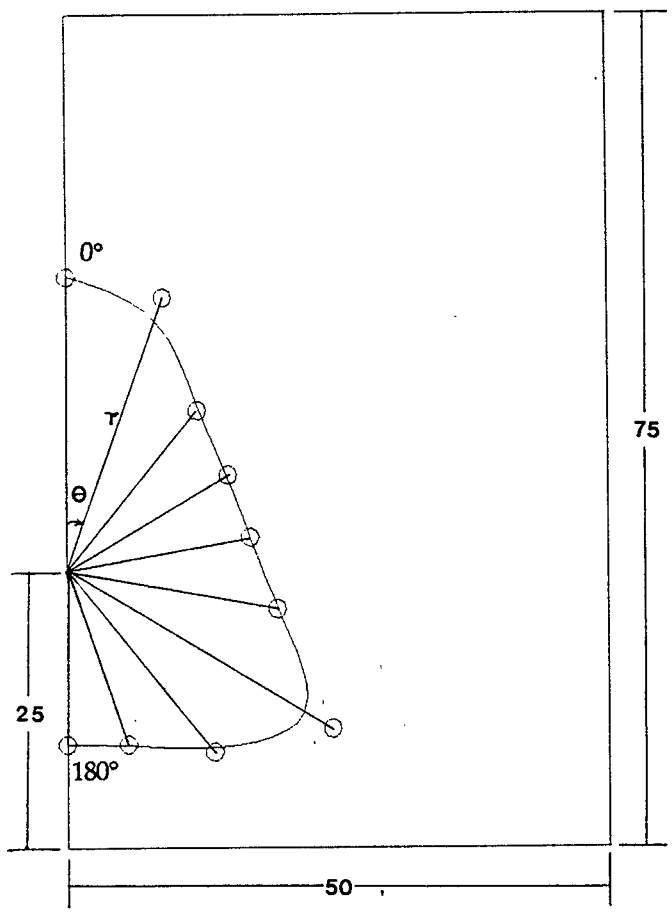


Fig. 31 Variation of the stress tangential to the hole boundary with angle θ for the numerical hole shapes of figs. 24(b) - 27(b).



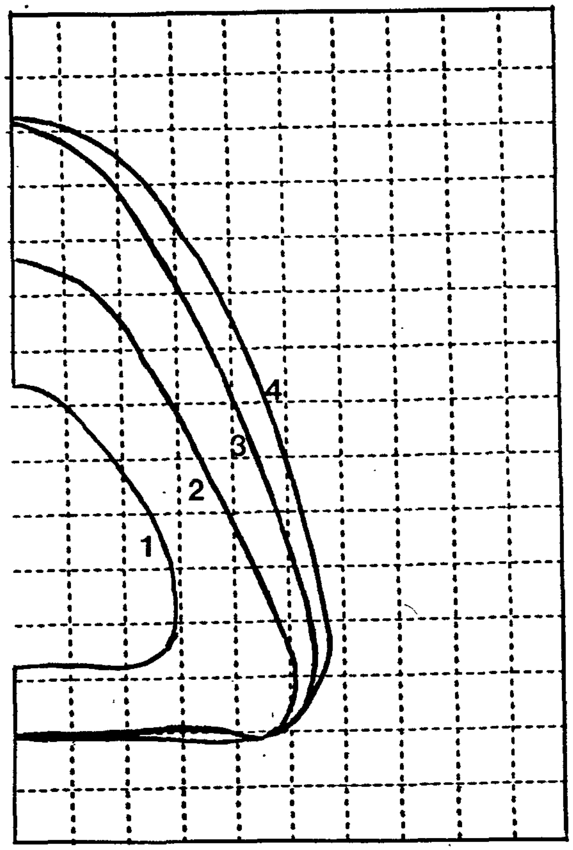
θ	Radius r			
degrees	hole 1	hole 2	hole 3	hole 4
0	15.39	26.99	38.43	38.86
20	14.31	25.82	33.80	35.99
40	12.55	23.04	28.11	31.37
60	11.97	20.53	24.92	27.81
80	12.73	19.57	23.44	26.13
100	14.15	20.43	24.54	26.87
120	15.58	26.77	28.36	29.67
140	12.43	23.96	24.29	24.51
160	9.83	17.12	17.48	17.53
180	9.10	15.04	15.43	15.45

Table 1 Polar coordinates for the experimental hole shapes

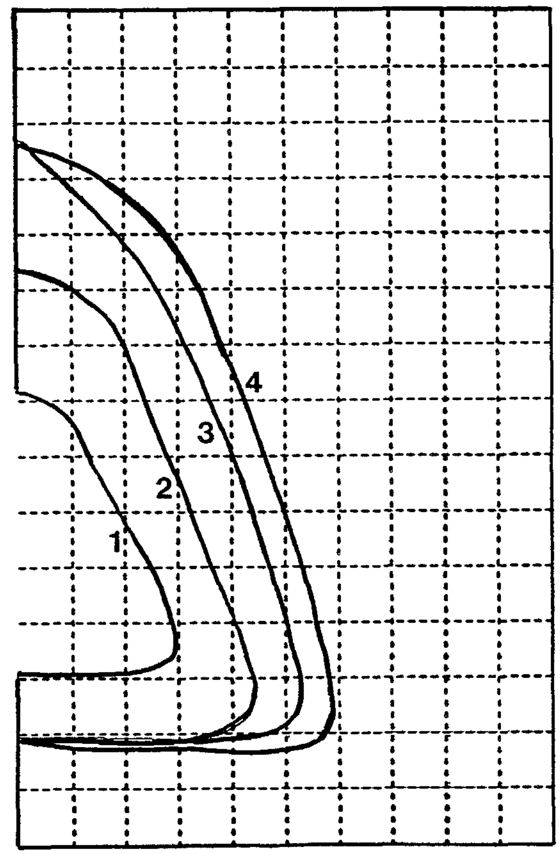


θ degrees	Radius r (of control points)			
	hole 1	hole 2	hole 3	hole 4
0	15.70	26.50	38.48	38.00
20	15.67	26.23	31.91	36.99
40	11.22	18.88	25.02	29.03
60	10.74	17.30	22.98	25.47
80	11.50	17.32	22.34	25.58
100	14.34	19.79	25.62	28.48
120	18.33	28.41	32.53	36.52
140	12.69	21.31	20.05	20.54
160	10.17	16.64	16.31	17.91
180	9.4	15.64	15.32	15.49

Table 2 Polar coordinates (of the control points) for the numerical hole shapes



(a)



(b)

Fig. 32 Superposed experimental(fig. 32(a))and numerical(fig. 32(b))hole shapes

hole	saving in weight (%)
1	7.18
2	20.57
3	28.54
4	32.76

(a)

hole	saving in weight (%)
1	6.85
2	17.41
3	25.58
4	31.10

(b)

Table 3 Percentage saving in weight over the solid beam for (a) experimental hole shapes and (b) numerical hole shapes

Tables 1 and 2 show the hole coordinates for the experimental and numerical approaches at 20° intervals. The experimental coordinates are for points on the hole boundary itself while the numerical coordinates are for the B-spline control points which may not lie on the boundary.

Fig. 32(a) shows all experimental hole shapes on a grid, while fig. 32(b) shows all the numerical hole shapes on a grid. Table 3 gives the percentage saving in weight over the original solid beam for each of the experimental and numerical hole shapes. These two figures together with the tables and fig. 24-27 can help a designer to :

1. Pick out the hole yielding a desired saving in weight.
2. Pick out a hole with a specific boundary stress.
3. Pick out a hole which lies in a given region in the plane.
4. Mark out the region to look for a desired hole shape if the shape satisfying 1, 2 and 3 is not among the four hole shapes. Once the region is marked out the specific hole shape can be easily obtained by starting from one of the four holes and a follow-up optimization.

5 CONCLUSION

In the previous chapters numerical and experimental methodologies were presented to tackle the shape optimal design problem. Four hole shapes were found with constant stress along the boundary without increasing the maximum tensile stress originally present in the solid beam. Each of these holes represents an optimum in the sense that there is an approximately uniform tangential stress along the boundary.

Some of the general conclusions made are listed below. They include the differences and limitations of the two methods which are important in determining their applicability and accuracy. Finally future directions of work have been identified.

5.1 General Conclusions

1. The accuracy of the stress distributions and the accuracy of the solution depends to a large extent on the correct simulation of the boundary conditions (displacements and forces) that actually occur in the physical structure and are largely unknown. The photoelasticity method is more suitable in this respect, as it can realistically simulate these boundary conditions. Of importance are the frictional forces which occur on the top loaded surface and at the supports. The magnitude of these frictional forces changes as the shape of the hole boundary is varied. Also they are often non-linear in nature for large deflections. Modelling these boundary conditions in the finite element method is in general more difficult than in the experimental method.

2. The frictional forces at the supports affect the value of the critical tensile stress just to the left of the support. This contributes to the differences between some of the hole shapes in the experimental and numerical method.
3. Photoelasticity gives directly the in-plane principal stresses on free boundaries and the in-plane maximum shear stresses in the rest of the field. In general photoelasticity is faster and less expensive when used on problems that require knowledge of the stress field only on the free boundaries. However, if structural response in the form of individual stresses is required in the interior of the structure then the finite element method is faster and less expensive. If the maximum tensile stress occurs in the interior of the beam for example, then, the finite element method is the easiest to use.
4. When the loading is more complicated or if modal analysis is involved, then it is faster and more efficient to use the finite element method.
5. In the photoelasticity method the fringe order occurring close to the boundary is often extrapolated to the boundary. The error can be reduced if the response can be improved by increased loading to give more and thinner fringes in the field. Increased loading may however cause the model to buckle and bend out-of-plane. In the finite element method too, the stresses at the Gaussian integration points are extrapolated to the boundary. The error can be reduced by increasing the number of elements, but at the cost of increasing computer time.
6. In regions with high stress gradients the photoelasticity method is more reliable for optimizing the geometry as it gives a more accurate picture of the stress distribution than in the finite element method. It should be noted that the regions of high stress gradients keep moving around the

boundary as the hole shape is modified. To handle this problem adaptive mesh generation (Bennett and Botkin, 1985) can be used, in which, the mesh is refined for each shape in the regions of high stress concentrations, thereby increasing the accuracy of the stress distribution. A robust algorithm is needed for this purpose.

7. The numerical representation of the hole boundary by a B-spline cannot handle some of the possible hole shapes, as mentioned in chapter 2, unless the number of control points is increased (at the cost of increased computer time).
8. Though four 'optimum' hole shapes have been obtained by the two methods, there is no guarantee that even the last of these holes represents the maximum saving in weight possible (global optimum) for the given constraints. It is possible that other hole shapes with a more complicated geometry than the simple triangular hole may yield a higher saving in weight. Also if the constraints were to be modified to allow multiple discontinuities and removal of material from the load free edges of the beam, as shown in fig. 33, then it is reasonable to expect optimal designs which are as effective, if not more, than those obtained with the present design constraints. The experimental method is better for handling multiple discontinuities than the numerical method, since maintaining mesh integrity and the development of robust mesh generators is difficult in the latter case when multiple discontinuities are present.
9. The numerical method was found to be quite expensive in terms of computer time. However, the design converges to a near optimum shape in four or five iterations, (except the last shape) before the optimizer becomes inefficient. To cut down on computer time, the CAD approach or

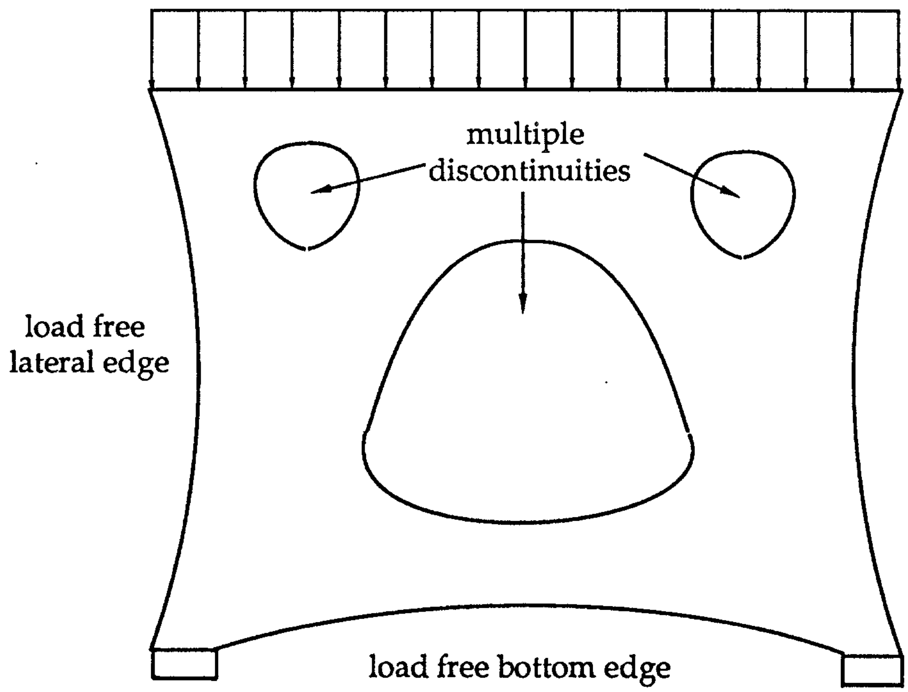


Fig. 33 The tall beam with multiple discontinuities and material removed from the load free edges of the beam.

interaction can be used to manually change the shape after the first few iterations of the optimizer.

Designer interaction was found to be useful for

a. Formulation of the problem -- modifying the finite element mesh, adjusting parameters like step size and convergence criteria in the initial stages of the optimization process.

b. Easy and quick interpretation of results which are presented using computer graphics.

c. Recovery from errors -- if a fatal error occurs during the optimization process, it is possible to restart from a previous iteration after properly constraining the problem to remove the cause of the fatal error.

d. Following the optimization process step by step which provides insight into the optimization mechanism and encourages adoption by skeptics.

e. Overall control over the optimizer.

10. The tensile stress criterion used here would correspond to the use of brittle materials or materials that have a high compressive strength and are more likely to fail in tension than in compression. If other materials are considered then other failure criteria have to be applied.

11. It should be noted that the four hole shapes obtained, are among a family of hole shapes that have a uniform stress distribution along the boundary, each with a different value of tangential stress along the hole boundary.

12. Finally, although described here in relation to the tall beam, the shape optimal design method can be easily applied to a wide variety of structures for weight minimization problems and for minimization of stress concentrations.

5.2 Future Directions

Future work could consist of the following :

1. Solution of the Tall Beam for different values of W/H ratios and support lengths. This would make available a whole range of solutions for the designer to choose from.
2. The inclusion of additional load cases and constraints, and the consideration of dynamics.
3. The use of boundary elements in place of finite elements for the numerical method.
4. The development of a mesh regenerator with adaptive mesh refinement capability instead of using PATRAN.
5. The use of continuum derivatives instead of finite differences to improve accuracy.
6. Generalization of the approach to include a wide variety of shape design problems.

APPENDIX A

CAD APPROACH

The CAD approach is controlled by a DCL (digital command language) procedure contained in the file START.COM. The procedure controls the execution of the necessary programs, sets up the terminal characteristics and performs cleaning up functions to remove unnecessary files after each iteration. Before running the procedure the following files must be prepared :

DATAA.DAT;1   ORIG.SES;1

SIGMA.NOD;1   PATRAN.SES;1   PATRAN.DAT;1

The file DATAA.DAT;1 contains the initial values of the shape variables (radii r of the control points) in the format shown in the attached sample at the end of Appendix A. The file ORIG.SES contains the original PATRAN session file used for creating the finite element mesh. A sample is attached at the end of Appendix A. This file is used as a template for creating the PATRAN session file (PATRAN.SES) for the hole shape defined by DATAA.DAT.

The files SIGMA.NOD;1, PATRAN.SES;1 and PATRAN.DAT;1 are empty initially. Subsequently, when the designer modifies the values of the shape variables in DATAA.DAT;1 (creating a new version DATAA.DAT;2), new versions of these three files are created with a higher version number. Thus SIGMA.NOD;2 will contain the nodal stress values for the modified shape defined by DATAA.DAT;2 and described by the PATRAN files PATRAN.SES;2 and PATRAN.DAT;2. The PATRAN.SES created from

ORIG.SES contains the PATRAN directives used for creating the finite element mesh. PATRAN.DAT is a non-printable data file containing the model created by PATRAN.SES. Note that the version numbers are increased by one, each time the shape is modified. The file ORIG.SES remains as it is.

After preparing the files DATAA.DAT;1 and ORIG.SES;1 the command procedure can be run. These files have to be prepared only once before the first iteration. The procedure is run by typing the following at the $ prompt.

$ @start.com

The following prompt is displayed:

Modify? 1. Previous design 2. Old design 3. Stop :

If 'Previous design' is chosen, the latest DATAA.DAT file is displayed for modification on the full screen editor. If 'Old design' is chosen, then a directory listing is displayed and the designer is prompted for the version number of the DATAA.DAT file he/she wishes to modify. For the very first iteration if the designer wishes to see the stress contours for the initial shape he/she can exit the editor without making any changes to DATAA.DAT;1. On exiting the editor, control is transferred to the DCL procedure.

The new shape is automatically remeshed and a finite element analysis is carried out. Then control is transferred to PATRAN. The designer must type in the terminal type and the following commands inside PATRAN:

> SES

> PIC.SES

Contour plots of maximum shear stress are displayed on the modified shape. PIC.SES contains PATRAN commands to display the third column of the nodal results file SIGMA.NOD which contains the values of the maximum shear stresses. The designer can also see plots of principal stresses or plots of individual stresses in the x and y directions depending on how the analysis program is set-up. The program PROJECT5B.FOR attached in appendix B does the analysis.

On exiting PATRAN the 'Modify ?' prompt shown above is displayed again and the designer can either continue to modify the shape or can terminate the process by choosing 'Stop'. To restart the designer only has to run the DCL procedure again. There is no need to create ORIG.SES;1 or DATAA.DAT;1 again.

The PATRAN.DAT files take up a lot of memory and are erased at the end of each iteration. The files DATAA.DAT and PATRAN.SES are retained with their corresponding version numbers appended to the file name, i.e., DATAA.DAT;3 is saved as DATAA3.DAT and PATRAN.SES;3 is saved as PATRAN3.SES. The shapes can be generated from these files, but this process takes a longer time than when using the PATRAN.DAT file. It is therefore recommended that the designer obtain hardcopies of the contour plots as they are displayed.

At present the CAD approach is setup to run on Tektronix 4107 and 4207 terminals.

An example is now described for demonstration purposes.

1. Create DATAA.DAT;1 and ORIG.SES as shown in the attached example. Create empty SIGMA.NOD;1, PATRAN.SES;1 and PATRAN.DAT;1. The directory listing at this point is attached.

2. Type '@start.com' at the $ prompt. Type '1' at the 'Modify?' prompt. The file DATAA.DAT;1 is displayed for editing on the full screen editor. Exit the editor without modifying any of the numbers in DATAA.DAT. A higher version DATAA.DAT;2 is created. At this point the finite element mesh is created for the hole shape defined in DATAA.DAT;2 and an analysis is carried out. This takes some time.
3. Then PATRAN prompts the designer for the terminal number. Enter 4107 or 4207 depending on the terminal being used. Then enter

   > SES

   > PIC.SES

   The contour plot of τ_{max} (maximum shear stress) is displayed and is attached in the example. A hardcopy of the contour plot can be obtained at this point. Contour plots of other stresses can also be displayed.
4. Observe the τ_{max} contours along the hole boundary. Suppose it is desired that the contour I coincide with the hole boundary. Note that the there is a stress gradient near control point 3 with a higher value of stress than I, whereas the stress is lower than I around the control points 5 and 6. Therefore analogous to the photoelasticity technique, material has to be added at the control point 3 and removed at control points 5 and 6.
5. Exit PATRAN by typing

   > 6

   Type '1' at the 'Modify?' prompt. The file DATAA.DAT;2 is displayed for editing. This time modify the values in the file to create DATAA.DAT;3 as shown in the attached copy. Note that the value of control point 3 has been decreased while the values of control points 5 and 6 have been increased. Exit the editor and repeat steps 3 to 5 until contour I is parallel to the lateral hole boundary. To stop the iterative process type '3' at the modify prompt.

6. The contour plots corresponding to the DATAA.DAT files and the directory listing at the end of three iterations are attached at the end of appendix A. Note that the last contour plot has the contour I parallel to the boundary.

START.COM

```
$ ANSI
$ SET TER/INQUIRE
$ ANSI
$ SETUP PATRAN
$ SET TER/WIDTH=80
$!
$MODIFY:
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT " MODIFY"
$ INQUIRE CHOICE " 1.PREVIOUS DESIGN  2.OLD DESIGN  3.STOP "
$ IF CHOICE .EQS. "1" THEN GOTO PREVIOUS_DESIGN
$ IF CHOICE .EQS. "2" THEN GOTO OLD_DESIGN
$ IF CHOICE .EQS. "3" THEN GOTO STOP
$ GOTO MODIFY
$!
$PREVIOUS_DESIGN:
$ FILE := DATAA.DAT
$ IF F$SEARCH(FILE) .NES. "" THEN GOTO EDIT_FILE
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT " THE FILE ''FILE' DOES NOT EXIST IN YOUR DIRECTORY "
$ GOTO MODIFY
$!
$OLD_DESIGN:
$ DIR
$ WRITE SYS$OUTPUT ""
$ INQUIRE NO " INPUT THE DESIGN NUMBER "
$ FILE := DATAA'NO'.DAT
$ IF F$SEARCH(FILE) .NES. "" THEN GOTO EDIT_FILE
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT " THE FILE ''FILE' DOES NOT EXIST IN YOUR DIRECTORY "
$ GOTO MODIFY
$!
$EDIT_FILE:
$ DEFINE/USER_MODE SYS$INPUT SYS$COMMAND
$ EDIT 'FILE'
$ IF CHOICE .EQS. "2" THEN REN DATAA'NO'.DAT DATAA.DAT
$ RUN CART
$ RUN PASDAT
$ REN PATRAN.SES P.SES
$ DEL CART.DAT;
$ PATRAN
BAT
SES
P.SES
$ ANSI
$ DEL PATRAN.SES;
$ REN P.SES PATRAN.SES
$ SET TER/WIDTH=80
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT " RUNNING THE FINITE ELEMENT ANALYSIS"
$ WRITE SYS$OUTPUT "     THIS WILL TAKE SOME TIME"
$ WRITE SYS$OUTPUT ""
$ RUN PROJECT5B
$ DEL PATRAN.OUT;
$ DEL OUTPUT.DAT;
$!
$ CHVER = F$PARSE(F$SEARCH("PATRAN.SES"),,,"VERSION")
$ VER = F$INTEGER(F$EXTRACT(1,2,CHVER))
$ RVER = VER-1
$ IF F$SEARCH("DATAA.DAT;''RVER'") .NES. "" THEN REN DATAA.DAT;'RVER'-
 DATAA'RVER'.DAT;1
$ IF F$SEARCH("PATRAN.SES;''RVER'") .NES. "" THEN REN PATRAN.SES;'RVER'-
 PATRAN'RVER'.SES;1
$ IF F$SEARCH("SIGMA.NOD;''RVER'") .NES. "" THEN REN SIGMA.NOD;'RVER'-
 SIGMA'RVER'.NOD;1
```

```
$ IF F$SEARCH("PATRAN.DAT;''RVER'") .NES. "" THEN DEL PATRAN.DAT;'RVER'
$!
$ DEFINE/USER_MODE SYS$INPUT SYS$COMMAND
$ PATRAN
$ ANSI
$ DEL PATRAN.SES;
$ SET TER/WIDTH=80
$ GOTO MODIFY
$STOP:
$ SET TER/WIDTH=80
$ EXIT
```

ORIG.SES

```
GO
1
1
2
SET,GRAPHICS,OFF
SET,LABE,OFF
SET,LI,0
GR,1,, 0.000000000E+00/ 0.630000000E+02/0
GR,2,, 0.136808057E+02/ 0.625877048E+02/0
GR,3,, 0.186408407E+02/ 0.472152889E+02/0
GR,4,, 0.216506351E+02/ 0.375000000E+02/0
GR,5,, 0.232818165E+02/ 0.291052124E+02/0
GR,6,, 0.270631853E+02/ 0.202280302E+02/0
GR,7,, 0.285788383E+02/ 0.850000000E+01/0
GR,8,, 0.134985398E+02/ 0.891306669E+01/0
GR,9,, 0.579567766E+01/ 0.907650650E+01/0
GR,10,, 0.000000000E+00/ 0.950000000E+01/0
GR,11,,0/0
GR,12T13,TR,11.66666,11
GR,14,,35/0
GR,15T16,TR,/23.333333,14
GR,17,,35/70
GR,18T19,TR,-11.66666,17
GR,20,,0/70
LI,9#,BS,,10/10/9/8/7/6/5/4/3/2/1/1
LI,9#,ST,,11T19,12T20
GR,29,,0/75
GR,30,,35/75
GR,31,,50/75
GR,32,,50/70
GR,33,,50/0
LI,5#,ST,,33/32/31/30/17,14/17/30/29/20
PA,9#,2L,,1T9,10T18
PA,3#,2L,,19/20/23,20/21/22
PA,1T#,REV
E
2
1
2
GF,1PT9P,,5/4
GF,10P,,10/4
GF,12P,,5/10
GF,11P,,5/4
E
2
2
CF,1PT#,QUAD/8
E
3
N
2
1
Y
7
3
3
2
8
5
1
```

```
1
1
SFFGSDF
N
9
6
```

DATAA.DAT

```
0.3800000000000000D+02
0.3700000000000000D+02
0.2730000000000000D+02
0.2450000000000000D+02
0.2405717925255151D+02
0.2750705474557043D+02
0.3300000000000000D+02
0.2100000000000000D+02
0.1695761248915318D+02
0.1550000000000000D+02
```

PIC.SES

```
GO
2
PATRAN.DAT
SET,GRAPHICS,ON
SET,LABE,OFF
RUN,CONT,COL,3
SIGMA.NOD
3
C
2.296,-.164
WINDOW
4
7
RUN,HI,CONT
```

EXAMPLE

DATAA.DAT;1 (DATAA.DAT;2)

```
0.3850000000000000D+02
0.3193446392647568D+02
0.2800000000000000D+02
0.2300000000000000D+02
0.2034050418131916D+02
0.2362306121502834D+02
0.3252719599568522D+02
0.2005336190617892D+02
0.1630672364079931D+02
0.1532309860737002D+02
```

Initial directory listing

```
Directory STU:[BHANDARKAR.JUNK]

CART.EXE;1          DATAA.DAT;1         FIRST.DAT;1         ORIG.SES;1
PASDAT.EXE;1        PATRAN.DAT;1        PATRAN.SES;1        PIC.SES;1
PROJECT5B.EXE;1     SIGMA.NOD;1         START.COM;41

Total of 11 files.
```

DATAA.DAT;3

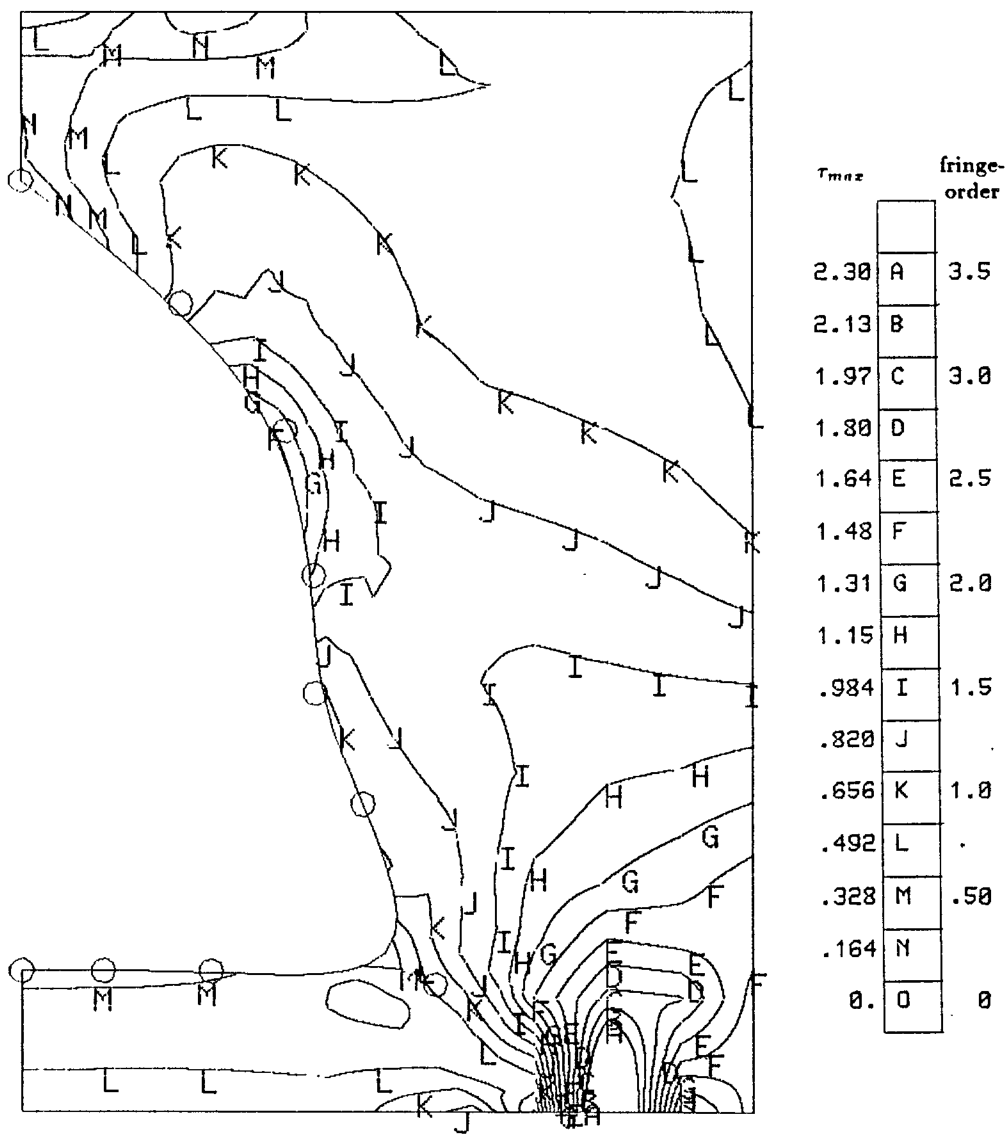
```
0.3850000000000000D+02
0.3193446392647568D+02
0.2600000000000000D+02
0.2300000000000000D+02
0.2150000000000000D+02
0.2450000000000000D+02
0.3252719599568522D+02
0.2005336190617892D+02
0.1630672364079931D+02
0.1532309860737002D+02
```

DATAA.DAT;4

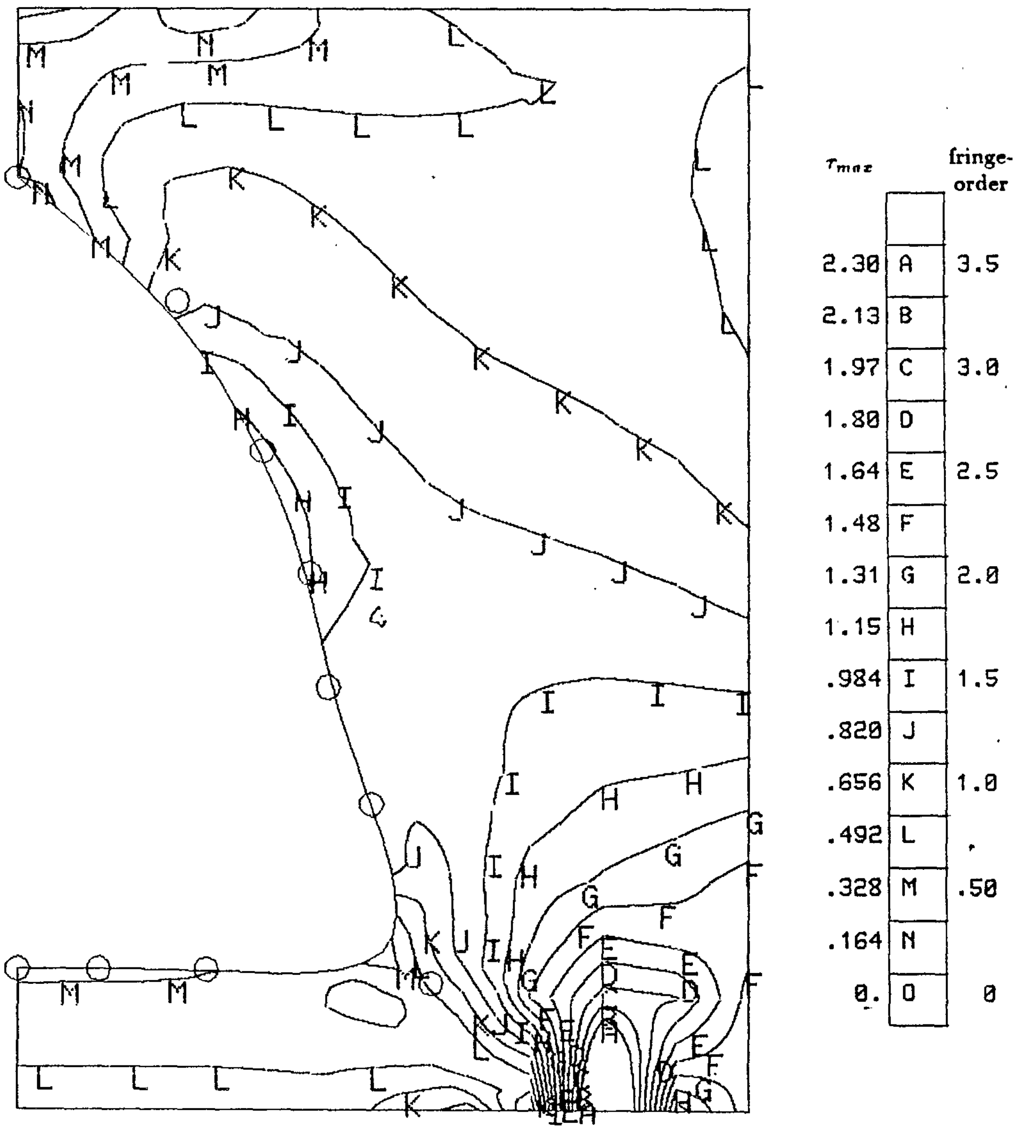
```
0.3850000000000000D+02
0.3193446392647568D+02
0.2500000000000000D+02
0.2300000000000000D+02
0.2230000000000000D+02
0.2550000000000000D+02
0.3252719599568522D+02
0.2005336190617892D+02
0.1630672364079931D+02
0.1532309860737002D+02
```

Final directory listing

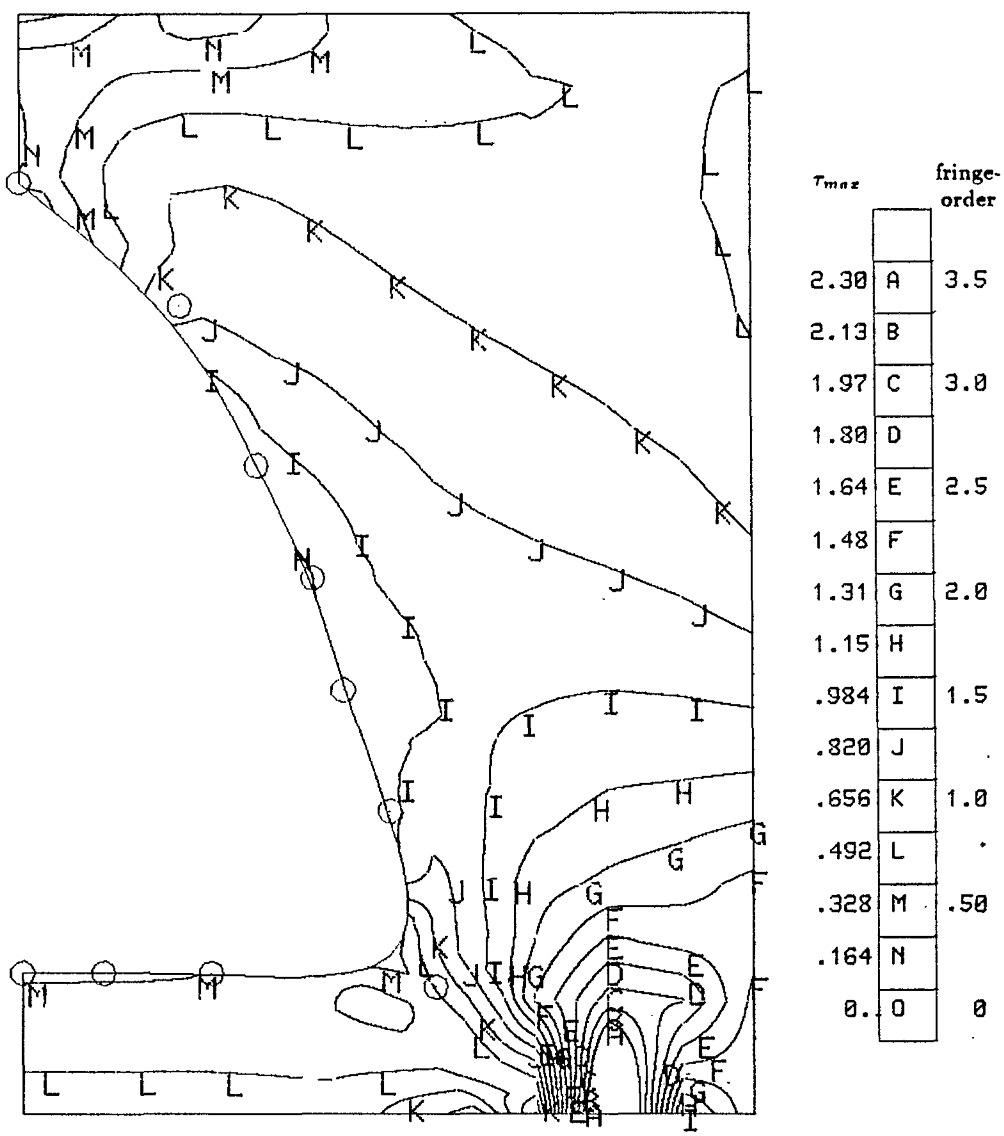
```
Directory STU:[BHANDARKAR.EXAMPLE]

CART.EXE;1          DATAA.DAT;4         DATAA1.DAT;1        DATAA2.DAT;1
DATAA3.DAT;1        LAST.DAT;1          ORIG.SES;1          PASDAT.EXE;1
PATRAN.DAT;4        PATRAN.SES;4        PATRAN1.SES;1       PATRAN2.SES;1
PATRAN3.SES;1       PIC.SES;1           PROJECT5B.EXE;1     SIGMA.NOD;4
SIGMA1.NOD;1        SIGMA2.NOD;1        SIGMA3.NOD;1        START.COM;41

Total of 20 files.
```



Contour plot corresponding to DATAA.DAT;2



Contour plot corresponding to DATAA.DAT;3



Contour plot corresponding to DATAA.DAT;4

APPENDIX B

NUMERICAL METHOD

The numerical approach is run by executing the program VMCON1.FOR. The following files have to be prepared before executing the program.

DATAA.DAT;1	PATRAN.SES;1	SCH33.FOR
ALL_DATAA.DAT;1	ALL_P.SES;1	ALL_SIGMA.NOD;1

DATAA.DAT;1 contains the initial values of the design variables (radii r of the control points) that are input by the designer in the format shown in the attached sample at the end of Appendix A. The file PATRAN.SES;1 is the PATRAN session file that was used to create the original finite element mesh. It is used as a template to remesh the new geometry after the optimizer returns new values for the shape variables. This is similar to ORIG.SES attached at the end of appendix A. The file SCH33.FOR is used for defining the objective functions and constraints for interfacing with the optimizer program VMCON1.FOR. The creation of SCH33.FOR is described in Crane et al., (1980). The objective function and constraints are formulated as described in chapter 2.

The files ALL_DATAA.DAT;1, ALL_P.SES;1 and ALL_SIGMA.NOD;1 are empty initially. At the end of each iteration the new DATAA.DAT, PATRAN.SES and SIGMA.NOD files created are appended to the end of the files ALL_DATAA.DAT;1, ALL_P.SES;1 and ALL_SIGMA.NOD;1 respectively. The file SIGMA.NOD contains the nodal stress data for each shape.

The optimizer can now be run as follows.

```
$ for sch33

$ link vmcon1,sch33,quad8

$ run vmcon1
```

The program QUAD8.FOR is the finite element analysis program that is used for calculating the sensitivity derivatives for the stress constraints and a copy is attached at the end of appendix B. The program has been annotated with comments and is self explanatory. The usage of the program VMCON1.FOR can be found in Crane et al., (1980). The program has been modified to run DCL command procedures to invoke PATRAN using the LIB$SPAWN statement from the run time routine library of the VAX/VMS. A restart option has also been built in. The program however is too bulky to include in this appendix.

The DCL procedures that are used are attached at the end of appendix B. They are GRAD.COM, CLEAN.COM, CLOSE.COM, and SAVE.COM. The other Fortran programs that have been used are CART.FOR for converting the polar coordinates of the shape variables into cartesian coordinates, PASDAT.FOR which rewrites the PATRAN.SES file for the modified shape and PROJECT5B.FOR which does the finite element analysis for creating SIGMA.NOD. Each of these programs are self explanatory. The subroutines used in PROJECT5B.FOR are the same as in QUAD8.FOR but there are some differences in the main program segment.

An example is described below for the setting up the optimization process leading to the fourth optimum hole shape described in chapter 4. The corresponding programs are attached at the end of appendix B.

1. Prepare DATAA.DAT;1 and PATRAN.SES;1 as shown in the attached samples (DATAA.DAT and ORIG.SES) in appendix A.
2. Prepare SCH33.FOR as shown. The program SCH33.FOR is used for
   a. Defining the side constraints
   b. Defining the stress constraints
   c. Setting the convergence parameter and stepsize for gradients.

   The side constraints are defined in subroutine MODEL by setting the values in the array XLIMIT, the ten values of the upper limit, followed by the ten values of the lower limit as shown. The maximum stress values, tensile and compressive, are defined by the values of STRT and STRC respectively. The stress constraints are then computed as shown by calling QUAD8.FOR.

   The convergence parameter is set by HH and the step size is set by H in subroutine DATA.
3. The program QUAD8.FOR is a subroutine which when called by SCH33.FOR returns the values of the stresses at certain finite element nodes and also returns the value of the objective function. The arrays used in the program are described within the program as also are the functions of each segment of the program. The array INDEX is used for defining the node points at which the stresses tangential to the hole boundary will be sampled for calculation of the objective function. The array ICONST is used for defining the element numbers and the Gaussian integration point for the element, at which the stress constraints will be computed. The other arrays are used for carrying out the finite element analysis.
4. The program PROJECT5B.FOR is the same as QUAD8.FOR, except that it does not compute the objective function or the stress constraints. It is

used solely for writing out SIGMA.NOD which is a data file containing the values of the stresses used in contour plotting. The first two columns of SIGMA.NOD contain the principal stresses σ_1 and σ_2 and the third column contains the maximum shear stress τ_{max}.

5. The other programs remain the same as long as the number of control points are not changed. The optimization process can now be started by running VMCON1.FOR as described above. It is preferable to run the program in the batch mode as it takes a lot of computing time.

SCH33.FOR

```
C
      SUBROUTINE MODEL(N,M,X,F,G)
C
C     OBJECTIVE FUNCTION AND CONSTRAINTS:
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(300),SIGMA(500),G(500),XLIMIT(20)
      COMMON /A/H,NF,NG,NDF,NDG,NACTC,COUNT
      DATA XLIMIT/39.0,39.5,32.0,28.5,28.5,31.5,37.5,24.0,
     218.5,15.5,38.0,35.5,28.0,24.5,24.5,27.5,30.5,19.5,
     116.5,15.0/
C     XLIMIT CONTAINS THE SIDE CONSTRAINTS ON THE VARIABLES
C     ALIMIT IS THE LIMIT ON THE "RESIDUAL AREA"
      ALIMIT=265.0
      NQCS=8
      STRT=0.98
      STRC=2.98
C     STRT=2.91
C     STRC=1.90
      WRITE(6,*)NF
C      IF (NF.EQ.(N+1))GO TO 55
      CALL QUAD8(N,M,X,SUM,SIGMA)
C      F=-((50*75)-SUM)
      F=SUM
      DO 50 J=1,NQCS
      G(J)=(SIGMA(J)-STRT)/STRT
 50   CONTINUE
C
C     SIDE CONSTRAINTS
C
      DO 51 J=1,10
      G(NQCS+J)=(X(J)-XLIMIT(J))/XLIMIT(J)
 51   CONTINUE
      DO 52 J=11,20
      G(NQCS+J)=(XLIMIT(J)-X(J-10))/XLIMIT(J)
 52   CONTINUE
C
C     CONSTRAINT ON THE RESIDUAL AREA
C
C      RESAREA=(31.0*19.0)-(50*75)+SIGMA(M-21)
C      G(M-21)=(RESAREA-ALIMIT)/ALIMIT
C
C     CONSTRAINT FOR HORZ BOTTOM EDGE
C
      G(NQCS+21)=(X(10)-X(9)*DCOSD(0.2D+02))/X(10)
   60 NF=NF+1
      IF(COUNT.EQ.21)THEN
      COUNT=11
      ELSE
      COUNT=COUNT+1
      ENDIF
      RETURN
      END
C
C     ................................................
C
      SUBROUTINE DATA(HH,NV,NC,X,F,G)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(300),G(500)
      COMMON/A/H,NF,NG,NDF,NDG,NACTC,COUNT
C
C     HH=TOLERANCE( THE ALGORITHM CONVERGES WHEN A CERTAIN SCALAR IS
C                   LESS THAN THIS VALUE)
C     H=STEPSIZE FOR GRADIENTS
C
      HH=.0001
      H=0.05
      NV=10
      NC=29
      COUNT=11
      REWIND(10)
```

```
C      REWIND(11)
      READ(10,1001) (X(I),I=1,NV)
C      READ(11,1056) F
C      READ(11,1058) NF
C      DO 10 J=1,NC
C 10   READ(11,1057) I,G(J)
 1001 FORMAT(6X,D24.16)
 1056 FORMAT(5X,'AREA',20X,F8.2)
 1057 FORMAT(4X,I4,6X,F9.2)
 1058 FORMAT(5X,'NF',20X,I4)
      RETURN
      END
```

QUAD8.FOR

```
C *****************************************************************
C
C       FEM ANALYSIS PROGRAM FOR EIGHT NODE SERENDIPITY ELEMENT
C
C ******************************************************************
C
C       * PLANE STRESS PROBLEM
C       * 3 x 3 GAUSSIAN INTEGRATION FOR STIFFNESS MATRIX
C       * STRESSES CALCULATED AT 2 x 2 INTEGRATION POINTS
C         LINEAR INTERPOLATION FOR STRESSES AT NODES
C
C       NOMENCLATURE
C       ------------
C
C       N = GLOBAL DEGREES OF FREEDOM
C       M = HALF BAND WIDTH
C       NET = TOTAL NO. OF ELEMENTS
C       NTDOF = N
C       MBAND = M
C       NTOT = NET x 16 ( SIZE OF ICONV )
C       NCON = NO. OF DISPLACEMENT B.C.'s
C       NNP = TOTAL NO. OF NODES.
C       NPE = 8 ( NODES/ELEMENT )
C       NV = NO. OF VARIABLES
C       NC = NO. OF CONSTRAINTS
C       NBS = NO. OF BOUNDARY STRESSES
C       NQCS = NO. OF STRESS CONSTRAINTS
C
C       GK = GLOBAL STIFFNESS MATRIX IN BANDED FORM ( N,M )
C       GL = GLOBAL LOAD VECTOR ( N )
C       EL = ELEMENT LOAD VECTOR
C       E = UPPER TRI. PART OF ELEMENT STIFFNESS MATRIX STORED ROWWISE
C           AS A VECTOR ( 136 )
C       LDOF = VECTOR CONTAINING THE NO. OF DEGREES OF FREEDOM IN EACH
C              ELEMENT ( NET )
C       NBDOF = VECTOR CONTAINING THE D.O.F. NO. OF THE DISPLACEMENT
C                B.C.'s ( NCON )
C       BDIS = VECTOR CONTAINING THE VALUES OF THE DISPLACEMENT B.C.'s
C       Q = GLOBAL DISPLACEMENT ( NTDOF )
C       IBC = WORKING INTEGER VECTOR ( NTDOF )
C       YY = WORKING VECTOR ( NTDOF )
C       ICONV = CONNECTIVITY VECTOR ( NTOT )
C       XX = X COORD. OF ALL THE NODES ( NNP )
C       XY = Y COORD. OF ALL THE NODES ( NNP )
C       NOD = NODE ORDER IN ELEMENT READ IN FROM PATRAN.OUT ( NPE )
C       CX = X COORD. OF ELEMENT NODES ( NPE )
C       CY = Y COORD. OF ELEMENT NODES ( NPE )
C       CE = ELASTIC CONSTANT MATRIX ( 3,3 )
C       EK = ELEMENT STIFFNESS MATRIX ( 16,16 )
C       QI = ELEMENT DISPLACEMENT VECTOR ( 16 )
C       SIGMA = STRESSES AT 2 x 2 GAUSS POINTS ( 4,3 )
C       SIGMAN = STRESSES AT NODES (8,3)
C       ASIGMA = CONSTRAINT ARRAY (500)or (NQCS)
C                CONTAINS STRESSES,TOTAL AREA OF THE BEAM
C       ICONST = ARRAY WHICH SPECIFIES THE CONSTRAINTS IN ASIGMA(NQCS,2)
C                FIRST COLUMN = NODE NUMBER
```

```
C              SECOND COLUMN = TYPE OF STRESS (2,3 OR 4) DEPENDING
C                              ON THE STRESS CONTAINED IN THE CORRES-
C                              PONDING COLUMN OF COLOR
C       COLOR = ARRAY CONTAINING THE AVERAGE SIGMA1,SIGMA2 AND TAU_MAX
C               EACH NODE (4,NNP).THE FIRST ROW IS USED FOR AVERAGING
C       TOTAREA = TOTAL AREA OF THE BEAM
C       X = VARIABLE VECTOR (300)or (NV)
C       SUM = OBJECTIVE FUNCTION ( VARIANCE IN THIS CASE )
C       INDEX = INTEGER ARRAY CONTAINING THE LOCATION OF THE
C               BOUNDARY STRESSES ( MAX. SHEAR )(NBS)
C       BSTR = CONTAINS THE VALUES OF THE BOUNDARY STRESS (NBS)
C       NBS = DIMENSION OF INDEX AND BSTR (NO. OF BOUNDARY STRESSES)
C       BSIGMA = ARRAY FOR STORING MAX SHEAR AT GAUSS POINTS (4*NET)
C       NOTE: IF THE NUMBER OF VARIABLES IS CHANGED THEN FNMBR
C             INDEX, BSTRESS ETC. ALSO CHANGE.
C
C *******************************************************************
      SUBROUTINE QUAD8(NV,NC,X,SUM,ASIGMA)
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(N=1252,M=92,NET=183,NTDOF=N,MBAND=M,NTOT=2928,NCON=31,
     1 NNP=626,NPE=8,NBS=54,NQCS=8)
      COMMON/A/H,NF,NG,NDF,NDG,NACTC,COUNT
      DIMENSION GK(N,M),GL(N),EL(16),E(136),LDOF(NET),NBDOF(NCON),
     1 BDIS(NCON),Q(NTDOF),IBC(NTDOF),YY(NTDOF),ICONV(NTOT),
     2 XX(NNP),XY(NNP),NOD(NPE),CX(NPE),CY(NPE),CE(3,3),EK(16,16),
     3 QI(16),SIGMA(4,3),SIGMAN(8,3),COLOR(4,NNP),ASIGMA(NQCS),X(NV),
     4 INDEX(NBS),BSTR(NBS),BSIGMA(4*NET),ICONST(NQCS,2)
      CHARACTER*33 FNMBR
      DATA FNMBR/'123456789101112131415161718192021'/
      DATA LDOF/NET*16/
      DATA NBDOF/987,991,1039,1065,1105,1127,1159,1181,1213,1233,1243,
     1 1251,1247,1235,1221,1191,1189,1,3,7,11,13,17,19,25,23,284,288,
     2 290,298,294/
      DATA BDIS/NCON*0.0/
      DATA INDEX/418,417,402,401,386,385,370,369,354,353,338,337,322,
     1 321,306,305,290,289,274,273,258,257,242,241,226,225,210,209,
     2 194,193,178,177,162,161,146,145,130,129,114,113,98,97,82,81,66,
     3 65,50,49,34,33,18,17,2,1/
      DATA ICONST/84,92,98,106,112,120,126,135,4,4,4,4,4,4,4,4/
      SUM=0.0
      TOTAREA=0.0
C
C
C       INITIALIZE THE ELASTIC CONSTANT MATRIX
C
C
      XEM=400000.0
      XNU=.36
      A=XEM/(1.-XNU**2)
      CE(1,1)=A
      CE(1,2)=XNU*A
      CE(2,1)=CE(1,2)
      CE(2,2)=CE(1,1)
      CE(3,3)=XEM/(2*(1+XNU))
      CE(3,1)=0.
      CE(3,2)=0.
      CE(1,3)=0.
      CE(2,3)=0.
C
C
C       READ IN CONNECTIVITY AND NODAL PHYSICAL COORDINATES
C
C
      ICOUNT=COUNT
      JFL=ICOUNT-10
      IF (JFL.LE.9) THEN
      OPEN(UNIT=ICOUNT,FILE='PATRAN'//FNMBR(JFL:JFL)//'.OUT',STATUS='OLD')
      ELSE
      IFL=2*JFL-10
      OPEN(UNIT=ICOUNT,FILE='PATRAN'//FNMBR(IFL:IFL+1)//'.OUT',STATUS='OLD')
      ENDIF
```

```
C
      READ(ICOUNT,1500)IJUNK
      DO 10 II=1,NNP
 10   READ(ICOUNT,1000) XX(II),XY(II),Z
      KK=1
      DO 12 II=1,NET
      READ(ICOUNT,2000)(NOD(J),J=1,8)
      DO 11 JJ=1,(NPE/2)
      ICONV(KK)=2*NOD(JJ)-1
      ICONV(KK+1)=2*NOD(JJ)
      ICONV(KK+2)=2*NOD(JJ+4)-1
      ICONV(KK+3)=2*NOD(JJ+4)
      KK=KK+4
 11   CONTINUE
 12   CONTINUE
C
      CLOSE(ICOUNT)
C
C     CALL COORD TO MODIFY XX,XY FOR PARTIAL DERIVATIVES
C
C     CALL COORD(NV,NC,NNP,X,XX,XY)
C
C
C     INITIALIZE THE GK AND GL MATRICES
C     INPUT GLOBAL NODAL LOAD VECTOR
C
C
      CALL INITM(N,M,GK)
      CALL INITV(N,GL)
      GL(988)=-0.64813333
      GL(990)=-2.59253333
      GL(984)=-1.29626666
      GL(982)=-2.59253333
      GL(978)=-1.29626666
      GL(976)=-2.59253333
      GL(972)=-1.29626666
      GL(970)=-2.59253333
      GL(966)=-1.29626666
      GL(964)=-2.59253333
      GL(960)=-1.29626666
      GL(958)=-2.59253333
      GL(938)=-1.29626666
      GL(906)=-2.59253333
      GL(886)=-1.29626666
      GL(854)=-2.59253333
      GL(834)=-1.29626666
      GL(802)=-2.59253333
      GL(782)=-1.48146666
      GL(750)=-3.33333333
      GL(728)=-1.66666666
      GL(688)=-3.33333333
      GL(662)=-1.66666666
      GL(614)=-3.33333333
      GL(610)=-0.83333333
      GL(283)=-0.41666666
      GL(287)=-1.66666666
      GL(289)=-0.83333333
      GL(297)=-1.66666666
      GL(293)=-0.41666666
C
C
C     CALL ELEMENT STIFFNESS , GLOBAL STIFFNESS ASSEMBLY, BOUNDARY
C     CONDITION , GAUSS-CHOLESKY DECOMPOSITION, AND DISPLACEMENT
C     SOLUTION SUBROUTINES. THIS GIVES THE GLOBAL NODAL DISPLACEMENT.
C
C
      DO 1 LNUM=1,NET
      DO 13 JJ=1,NPE
      KK=(LNUM-1)*16+(2*JJ)
      NN=ICONV(KK)/2
      CX(JJ)=XX(NN)
      CY(JJ)=XY(NN)
 13   CONTINUE
```

```
      CALL D24(CX,CY,CE,EK)
      II=0
      DO 14 JJ=1,16
      DO 14 KK=JJ,16
      II=II+1
      E(II)=EK(JJ,KK)
 14   CONTINUE
      CALL ASEMBM (NTDOF,MBAND,NET,NTOT,LNUM,GK,E,ICONV,LDOF)
C     CALL ASEMBV (NTDOF,NET,NTOT,LNUM,GL,EL,ICONV,LDOF)
 1    CONTINUE
      CALL BOUND (GK,GL,NTDOF,MBAND,NCON,NBDOF,BDIS,Q)
      CALL DECOM (NTDOF,MBAND,IBC,NCON,NBDOF,GK)
      CALL SOLV (NTDOF,MBAND,IBC,GK,GL,Q,YY)
C
C
C     CALCULATE STRESSES
C
C
      DO 25 LNUM=1,NET
      DO 23 JJ=1,NPE
      KK=(LNUM-1)*16+(2*JJ)
      NN=ICONV(KK)/2
      CX(JJ)=XX(NN)
      CY(JJ)=XY(NN)
 23   CONTINUE
C
C     INITIALIZE QI(I)
C
      NNN=(LNUM-1)*16
      DO 24 KK=1,16
      QI(KK)=Q(ICONV(NNN+KK))
 24   CONTINUE
C
C     STRESS (AT GAUSS POINTS) AND AREA
C
      CALL STRESS(CX,CY,CE,QI,SIGMA)
      CALL AREA(CX,CY,ARRE)
C
      TOTAREA=TOTAREA+ARPE
C
      DO 51 I=1,4
      CENTER=(SIGMA(I,1)+SIGMA(I,2))/2.
      RAD=SQRT((SIGMA(I,1)-SIGMA(I,2))**2+(4.*(SIGMA(I,3)**2)))/2.
      SIGMA(I,1)=CENTER+RAD
      SIGMA(I,2)=CENTER-RAD
      SIGMA(I,3)=RAD
 51   CONTINUE
      J=(LNUM-1)*4
      DO 55 I=1,4
      BSIGMA(J+I)=SIGMA(I,3)
 55   CONTINUE
C
C     STRESS AT NODES
C
      CALL STRESSN(SIGMA,SIGMAN)
C
C     CALCULATE COLOR(4,NNP)
C
      DO 58 II=1,NPE
      KK=(LNUM-1)*16+(2*II)
      NN=ICONV(KK)/2
      COLOR(1,NN)=COLOR(1,NN)+1
      DO 57 JJ=1,3
 57   COLOR(JJ+1,NN)=COLOR(JJ+1,NN)+ SIGMAN(II,JJ)
 58   CONTINUE
C
 25   CONTINUE
C
C     END OF STRESS CALCULATION
C
C     CALCULATION OF OBJECTIVE FUNCTION (VARIANCE)
C
```

```
      DO 60 I=1,NBS
 60   BSTR(I)=BSIGMA(INDEX(I))
      ADD=0.0
      DO 70 I=1,NBS-20
 70   ADD=ADD+BSTR(I)
      AVG1=ADD/(NBS-20.)
      ADD=0.0
      DO 75 I=NBS-19,NBS
 75   ADD=ADD+BSTR(I)
      AVG2=ADD/20.
      VAR=0.0
      DO 80 I=1,NBS-20
 80   VAR=VAR+(BSTR(I)-AVG1)**2
      DO 85 I=NBS-19,NBS
 85   VAR=VAR+(BSTR(I)-AVG2)**2
      SUM=VAR
      WRITE(6,*)SUM
C
C     CALCULATE ASIGMA
C
      DO 87 I=1,NQCS
 87   ASIGMA(I)= COLOR(ICONST(I,2),ICONST(I,1))/COLOR(1,ICONST(I,1))
C
C     THE LAST NON-ZERO ELEMENT OF ASIGMA (i.e. THE  (NET*4)+1 th ELEMENT)
C     IS THE AREA OF THE BEAM
C
C     I=NET*4+1
C     ASIGMA(I)=TOTAREA
C     WRITE(6,*) ASIGMA(I)
C
 99   FORMAT(/,4X,3(F10.5,2X))
 1000 FORMAT(3E16.9//)
 1500 FORMAT(I2////)
 2000 FORMAT(/8I8/)
      CLOSE(9)
      RETURN
      END

*****************************************************
C
C     INITM     INITIALIZATION OF THE GLOBAL STIFF MATRIX TO 0.0
C     INITV        "          "    "   GLOBAL LOAD VECTOR "    "
C
C ****************************************************************

      SUBROUTINE INITM(N,M,GK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GK(N,M)
      DO 1 I=1,N
      DO 1 J=1,M
 1    GK(I,J)=0.
      RETURN
      END
C
      SUBROUTINE INITV (N,GL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GL(N)
      DO 1 I=1,N
 1    GL(I)=0
      RETURN
      END

C ****************************************************************
C
C     ASEMBM     ASSEMBLY OF THE GLOBAL STIFF MATRIX IN BANDED FORM
C
C ****************************************************************

      SUBROUTINE ASEMBM (NTDOF,MBAND,NET,NTOT,LNUM,GK,E,ICONV,LDOF)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GK(NTDOF,MBAND),ICONV(NTOT),LDOF(NET),E(1)
      L1=LNUM-1
      NSUM=0
```

```
      IF(L1.EQ.0) GO TO 2
      DO 1 I=1,L1
1     NSUM=NSUM+LDOF(I)
2     NEL=LDOF(LNUM)
      L=0
      DO 10 I=1,NEL
      II=ICONV(NSUM+I)
      DO 10 J=I,NEL
      L=L+1
      JJ=ICONV(NSUM+J)
      IF (JJ.LT.II) GO TO 20
      NC=JJ-II+1
      GK(II,NC)=GK(II,NC)+E(L)
      GO TO 10
20    NC=II-JJ+1
      GK(JJ,NC)=GK(JJ,NC)+E(L)
10    CONTINUE
      RETURN
      END
C ******************************************************************
C
C     ASEMBV      ASSEMBLY OF THE GLOBAL LOAD VECTOR
C
C ******************************************************************
      SUBROUTINE ASEMBV(NTDOF,NET,NTOT,LNUM,GL,EL,ICONV,LDOF)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GL(NTDOF),ICONV(NTOT),LDOF(NET),EL(2)
      L1=LNUM-1
      NSUM=0
      IF (L1.EQ.0) GO TO 2
      DO 1 I=1,L1
1     NSUM=NSUM+LDOF(I)
2     NEL=LDOF(LNUM)
      DO 10 I=1,NEL
      II=ICONV(NSUM+I)
10    GL(II)=GL(II)+EL(I)
      RETURN
      END
C ********************************************************************
C
C     BOUND        APPLICATION OF THE DISPLACEMENT B.C.'s
C
C ********************************************************************
      SUBROUTINE BOUND(GK,GL,NTDOF,MBAND,NCON,NBDOF,BDIS,Q)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GK(NTDOF,MBAND),GL(NTDOF),NBDOF(NCON),BDIS(NCON),Q(NTDOF)
      DO 1 I=1,NTDOF
      Q(I)=0.
      DO 2 J=1,NCON
      K=NBDOF(J)
      IF (K.EQ.I.OR.K.GT.I) JJ=K-I+1
      IF (JJ.GT.MBAND) GO TO 2
      IF (K.EQ.I.OR.K.GT.I) GL(I)=GL(I)-GK(I,JJ)*BDIS(J)
      IF (K.LT.I) JJ=I-K+1
      IF (JJ.GT.MBAND) GO TO 2
      IF (K.LT.I) GL(I)=GL(I)-GK(K,JJ)*BDIS(J)
2     CONTINUE
1     CONTINUE
      DO 3 I=1,NCON
      J=NBDOF(I)
3     Q(J)=BDIS(I)
      RETURN
      END
C ******************************************************************
C
C     DECOM      CHOLESKY DECOMPOSITION
C
C ******************************************************************
```

```
      SUBROUTINE DECOM (NTDOF,MBAND,IBC,NCON,NBDOF,U)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION U(NTDOF,MBAND),IBC(NTDOF),NBDOF(NCON)
      DO 20 I=1,NTDOF
20    IBC(I)=0
      DO 21 I=1,NCON
      J=NBDOF(I)
21    IBC(J)=1
      N=MBAND
      NN=NTDOF-MBAND+1
      KK=0
      DO 11 I=1,NTDOF
      IF (IBC(I).EQ.1) GO TO 11
      KK=KK+1
      IF (I.LE.NN) GO TO 2
      N=N-1
2     DO 1 J=1,N
      ID1=I+J-1
      IF (IBC(ID1).EQ.1) GO TO 1
      SUM=0.
      IF (KK.EQ.1) GO TO 4
      L1=I-1
      DO 5 L=1,L1
      L2=L1-L+1
      JJ=J+L
      IF (JJ.GT.MBAND) GO TO 4
      II=L+1
      ID1=II+L2-1
      ID2=JJ+L2-1
      IF (IBC(ID1).EQ.1.OR.IBC(ID2).EQ.1.OR.IBC(L2).EQ.1) GO TO 5
      SUM=SUM+U(L2,II)*U(L2,JJ)
5     CONTINUE
4     IF (J.NE.1) GO TO 3
10    U(I,1)=SQRT(U(I,1)-SUM)
      GO TO 1
3     U(I,J)=(U(I,J)-SUM)/U(I,1)
1     CONTINUE
11    CONTINUE
      RETURN
      END

C ***********************************************************************
C
C     SOLV      ITERATIVE GAUSSIAN SOLUTION FOR THE GLOBAL DISPL.
C
C ***********************************************************************

      SUBROUTINE SOLV(NTDOF,MBAND,IBC,U,F,Q,YY)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION U(NTDOF,MBAND),IBC(NTDOF),F(NTDOF),YY(NTDOF),Q(NTDOF)
      IN=0
5     IN=IN+1
      IF (IBC(IN).EQ.1) GO TO 5
      YY(IN)=F(IN)/U(IN,1)
      NN=IN+1
      DO 3 I=NN,NTDOF
      IF(IBC(I).EQ.1) GO TO 3
      I1=I-1
      SUM=0
      N=1
      II=I
      IF (I.LT.MBAND) GO TO 2
      N=I-MBAND+1
      II=MBAND
2     DO 4 J=N,I1
      IF (IBC(J).EQ.1) GO TO 6
      SUM=SUM+YY(J)*U(J,II)
6     II=II-1
4     CONTINUE
      YY(I)=(F(I)-SUM)/U(I,1)
3     CONTINUE
      IN=NTDOF+1
1     IN=IN-1
```

```
      IF (IBC(IN).EQ.1) GO TO 1
      Q(IN)=YY(IN)/U(IN,1)
      NN=IN-1
      DO 12 I=1,NN
      JJ=IN-I
      IF (IBC(JJ).EQ.1) GO TO 12
      SUM=0
      II=1
      N=IN
      J1=JJ+1
      IF (I.LT.MBAND) GO TO 15
      N=MBAND+JJ-1
15    DO 14 J=J1,N
      II=II+1
      IF (IBC(J).EQ.1) GO TO 14
      SUM=SUM+Q(J)*U(JJ,II)
14    CONTINUE
      Q(JJ)=(YY(JJ)-SUM)/U(JJ,1)
12    CONTINUE
      RETURN
      END

C *********************************************************************
C
C       D24         ELEMENT STIFFNESS MATRIX SUBROUTINE
C
C *********************************************************************

      SUBROUTINE D24(CX,CY,CE,EK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION CX(8),CY(8),CE(3,3),EK(16,16)
      DIMENSION CHZ(3),ETA(3),WT(3),NDE(16),DJ(2,8),YK(2,2),
     1 YL(2,2),DE(3,16),CM(3,3),DC(16,3),AN(8),DP(2,8)
      DATA NDE/1,3,5,7,9,11,13,15,2,4,6,8,10,12,14,16/
      DATA CHZ/.774596669241483D0,.0D0,-.774596669241483D0/
      DATA ETA/.774596669241483D0,.0D0,-.774596669241483D0/
      DATA WT/.555555555555556D0,.888888888888889D0,.555555555555556D0/
      DATA NODE,NE,NDOF,NGAUSS/8,3,16,3/
      DO 487 I=1,NDOF
      DO 487 J=1,NDOF
487   EK(I,J)=0.
      DO 488 I=1,NE
      DO 488 J=1,NDOF
488   DE(I,J)=0.
      DO 151 II=1,NGAUSS
      DO 151 JJ=1,NGAUSS
      A=1.+CHZ(II)
      B=1.-CHZ(II)
      C=1.+ETA(JJ)
      D=1.-ETA(JJ)
      E=CHZ(II)+ETA(JJ)-1.
      F=-CHZ(II)+ETA(JJ)-1.
      G=-CHZ(II)-ETA(JJ)-1.
      H=CHZ(II)-ETA(JJ)-1.
      AN(1)=A*C*E/4.
      AN(2)=B*C*A/2.
      AN(3)=B*C*F/4.
      AN(4)=B*C*D/2.
      AN(5)=B*D*G/4.
      AN(6)=A*B*D/2.
      AN(7)=A*D*H/4.
      AN(8)=A*C*D/2.
      DJ(1,1)=((C*E)+(A*C))/4.
      DJ(1,2)=(-(A*C)+(B*C))/2.
      DJ(1,3)=(-(C*F)-(B*C))/4.
      DJ(1,4)=(-(D*C))/2.
      DJ(1,5)=(-(D*G)-(B*D))/4.
      DJ(1,6)=(-(A*D)+(B*D))/2.
      DJ(1,7)=((D*H)+(A*D))/4.
      DJ(1,8)=(C*D)/2.
      DJ(2,1)=((A*E)+(A*C))/4.
      DJ(2,2)=(A*B)/2.
      DJ(2,3)=((B*F)+(B*C))/4.
      DJ(2,4)=(-(C*B)+(D*B))/2.
```

```
      DJ(2,5)=(-(B*G)-(B*D))/4.
      DJ(2,6)=(-(A*B))/2.
      DJ(2,7)=(-(A*H)-(A*D))/4.
      DJ(2,8)=(-(C*A)+(D*A))/2.
      DO 155 I=1,2
      DO 155 J=1,2
155   YK(I,J)=0.
      DO 601 I=1,2
      DO 601 K=1,NODE
      YK(I,1)=YK(I,1)+DJ(I,K)*CX(K)
601   YK(I,2)=YK(I,2)+DJ(I,K)*CY(K)
      DTJ=YK(1,1)*YK(2,2)-YK(1,2)*YK(2,1)
      YL(1,1)=YK(2,2)/DTJ
      YL(1,2)=-YK(1,2)/DTJ
      YL(2,1)=-YK(2,1)/DTJ
      YL(2,2)=YK(1,1)/DTJ
      DW=DTJ
      DO 499 I=1,2
      DO 499 J=1,NODE
      DP(I,J)=0.
      DO 499 K=1,2
499   DP(I,J)=DP(I,J)+YL(I,K)*DJ(K,J)
      DO 3 I=1,NODE
      DE(1,NDE(I))=DP(1,I)
      DE(2,NDE(I+NODE))=DP(2,I)
      DE(3,NDE(I))=DP(2,I)
3     DE(3,NDE(I+NODE))=DP(1,I)
      DO 11 I=1,NE
      DO 11 J=1,NE
11    CM(I,J)=CE(I,J)*DW*WT(II)*WT(JJ)
      DO 6 I=1,NDOF
      DO 6 J=1,NE
      DC(I,J)=0.
      DO 6 K=1,NE
6     DC(I,J)=DC(I,J)+DE(K,I)*CM(K,J)
      DO 7 I=1,NDOF
      DO 7 J=I,NDOF
      DO 7 K=1,NE
7     EK(I,J)=EK(I,J)+DC(I,K)*DE(K,J)
151   CONTINUE
      RETURN
      END
C *********************************************************************
C
C     STRESS      GAUSS POINT STRESS CALCULATION SUBROUTINE
C
C *********************************************************************

      SUBROUTINE STRESS(CX,CY,CE,QI,SIGMA)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION CX(8),CY(8),CE(3,3)
      DIMENSION CHZ(2),ETA(2),NDE(16),DJ(2,8),YK(2,2),QI(16)
      DIMENSION YL(2,2),DE(3,16),CM(3,3),DC(3,16),AN(8),DP(2,8)
      DIMENSION SG(3),SIGMA(4,3)
      DATA NDE/1,3,5,7,9,11,13,15,2,4,6,8,10,12,14,16/
      DATA CHZ/.577350269189826D0,-.577350269189826D0/
      DATA ETA/.577350269189826D0,-.577350269189826D0/
      DATA NODE,NE,NDOF,NGAUSS/8,3,16,2/
      DO 488 I=1,NE
      DO 488 J=1,NDOF
488   DE(I,J)=0.
      KK=0
      DO 151 II=1,NGAUSS
      DO 151 JJ=1,NGAUSS
      A=1.+CHZ(II)
      B=1.-CHZ(II)
      C=1.+ETA(JJ)
      D=1.-ETA(JJ)
      E=CHZ(II)+ETA(JJ)-1.
      F=-CHZ(II)+ETA(JJ)-1.
      G=-CHZ(II)-ETA(JJ)-1.
      H=CHZ(II)-ETA(JJ)-1.
```

```
      AN(1)=A*C*E/4.
      AN(2)=B*C*A/2.
      AN(3)=B*C*F/4.
      AN(4)=B*C*D/2.
      AN(5)=B*D*G/4.
      AN(6)=A*B*D/2.
      AN(7)=A*D*H/4.
      AN(8)=A*C*D/2.
      DJ(1,1)=((C*E)+(A*C))/4.
      DJ(1,2)=(-(A*C)+(B*C))/2.
      DJ(1,3)=(-(C*F)-(B*C))/4.
      DJ(1,4)=(-(D*C))/2.
      DJ(1,5)=(-(D*G)-(B*D))/4.
      DJ(1,6)=(-(A*D)+(B*D))/2.
      DJ(1,7)=((D*H)+(A*D))/4.
      DJ(1,8)=(C*D)/2.
      DJ(2,1)=((A*E)+(A*C))/4.
      DJ(2,2)=(A*B)/2.
      DJ(2,3)=((B*F)+(B*C))/4.
      DJ(2,4)=(-(C*B)+(D*B))/2.
      DJ(2,5)=(-(B*G)-(B*D))/4.
      DJ(2,6)=(-(A*B))/2.
      DJ(2,7)=(-(A*H)-(A*D))/4.
      DJ(2,8)=(-(C*A)+(D*A))/2.
      DO 155 I=1,2
      DO 155 J=1,2
155   YK(I,J)=0.
      DO 601 I=1,2
      DO 601 K=1,NODE
      YK(I,1)=YK(I,1)+DJ(I,K)*CX(K)
601   YK(I,2)=YK(I,2)+DJ(I,K)*CY(K)
      DTJ=YK(1,1)*YK(2,2)-YK(1,2)*YK(2,1)
      YL(1,1)=YK(2,2)/DTJ
      YL(1,2)=-YK(1,2)/DTJ
      YL(2,1)=-YK(2,1)/DTJ
      YL(2,2)=YK(1,1)/DTJ
      DW=DTJ
      DO 499 I=1,2
      DO 499 J=1,NODE
      DP(I,J)=0.
      DO 499 K=1,2
499   DP(I,J)=DP(I,J)+YL(I,K)*DJ(K,J)
      DO 3 I=1,NODE
      DE(1,NDE(I))=DP(1,I)
      DE(2,NDE(I+NODE))=DP(2,I)
      DE(3,NDE(I))=DP(2,I)
3     DE(3,NDE(I+NODE))=DP(1,I)
C
C     CALCULATE C x B
C
      DO 6 I=1,NE
      DO 6 J=1,NDOF
      DC(I,J)=0.
      DO 6 K=1,NE
6     DC(I,J)=DC(I,J)+CE(I,K)*DE(K,J)
C
C     CALCULATE STRESSES
C
      DO 15 I=1,NE
      SG(I)=0.
      DO 15 K=1,NDOF
15    SG(I)=SG(I)+DC(I,K)*QI(K)
      KK=KK+1
      DO 16 I=1,NE
16    SIGMA(KK,I)=SG(I)
151   CONTINUE
      RETURN
      END

C *********************************************************************
C
C     STRESSN       NODAL STRESS CALCULATION SUBROUTINE
C
C *********************************************************************
```

```
      SUBROUTINE STRESSN(SIGMA,SIGMAN)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SIGMA(4,3),SIGMAN(8,3)
      A=2.7320508
      B=-.7320508
      DO 10 I=1,3
      SIGMAN(1,I)=(A*A*SIGMA(1,I)+A*B*SIGMA(2,I)+A*B*SIGMA(3,I)
     1 +B*B*SIGMA(4,I))/4
      SIGMAN(2,I)=(A*SIGMA(1,I)+B*SIGMA(2,I)+A*SIGMA(3,I)
     1 +B*SIGMA(4,I))/4
      SIGMAN(3,I)=(A*B*SIGMA(1,I)+B*B*SIGMA(2,I)+A*A*SIGMA(3,I)
     1 +A*B*SIGMA(4,I))/4
      SIGMAN(4,I)=(B*SIGMA(1,I)+B*SIGMA(2,I)+A*SIGMA(3,I)
     1 +A*SIGMA(4,I))/4
      SIGMAN(5,I)=(B*B*SIGMA(1,I)+A*B*SIGMA(2,I)+A*B*SIGMA(3,I)
     1 +A*A*SIGMA(4,I))/4
      SIGMAN(6,I)=(B*SIGMA(1,I)+A*SIGMA(2,I)+B*SIGMA(3,I)
     1 +A*SIGMA(4,I))/4
      SIGMAN(7,I)=(A*B*SIGMA(1,I)+A*A*SIGMA(2,I)+B*B*SIGMA(3,I)
     1 +A*B*SIGMA(4,I))/4
      SIGMAN(8,I)=(A*SIGMA(1,I)+A*SIGMA(2,I)+B*SIGMA(3,I)
     1 +B*SIGMA(4,I))/4
 10   CONTINUE
      RETURN
      END
C *******************************************************************
C
C       AREA       CALCULATES THE AREA OF THE ELEMENT
C
C*******************************************************************
      SUBROUTINE AREA(CX,CY,ARRE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION CX(8),CY(8)
      DIMENSION CHZ(2),ETA(2),DJ(2,8),YK(2,2),AN(8)
      DATA CHZ/.577350269189826D0,-.577350269189826D0/
      DATA ETA/.577350269189826D0,-.577350269189826D0/
      DATA NODE,NE,NDOF,NGAUSS/8,3,16,2/
      ARRE=0.0
      DO 151 II=1,NGAUSS
      DO 151 JJ=1,NGAUSS
      A=1.+CHZ(II)
      B=1.-CHZ(II)
      C=1.+ETA(JJ)
      D=1.-ETA(JJ)
      E=CHZ(II)+ETA(JJ)-1.
      F=-CHZ(II)+ETA(JJ)-1.
      G=-CHZ(II)-ETA(JJ)-1.
      H=CHZ(II)-ETA(JJ)-1.
      AN(1)=A*C*E/4.
      AN(2)=B*C*A/2.
      AN(3)=B*C*F/4.
      AN(4)=B*C*D/2.
      AN(5)=B*D*G/4.
      AN(6)=A*B*D/2.
      AN(7)=A*D*H/4.
      AN(8)=A*C*D/2.
      DJ(1,1)=((C*E)+(A*C))/4.
      DJ(1,2)=(-(A*C)+(B*C))/2.
      DJ(1,3)=(-(C*F)-(B*C))/4.
      DJ(1,4)=(-(D*C))/2.
      DJ(1,5)=(-(D*G)-(B*D))/4.
      DJ(1,6)=(-(A*D)+(B*D))/2.
      DJ(1,7)=((D*H)+(A*D))/4.
      DJ(1,8)=(C*D)/2.
      DJ(2,1)=((A*E)+(A*C))/4.
      DJ(2,2)=(A*B)/2.
      DJ(2,3)=((B*F)+(B*C))/4.
      DJ(2,4)=(-(C*B)+(D*B))/2.
      DJ(2,5)=(-(B*G)-(B*D))/4.
      DJ(2,6)=(-(A*B))/2.
      DJ(2,7)=(-(A*H)-(A*D))/4.
      DJ(2,8)=(-(C*A)+(D*A))/2.
```

```
      DO 155 I=1,2
      DO 155 J=1,2
155   YK(I,J)=0.
      DO 601 I=1,2
      DO 601 K=1,NODE
      YK(I,1)=YK(I,1)+DJ(I,K)*CX(K)
601   YK(I,2)=YK(I,2)+DJ(I,K)*CY(K)
      DTJ=YK(1,1)*YK(2,2)-YK(1,2)*YK(2,1)
      ARRE=ARRE+DTJ
151   CONTINUE
      RETURN
      END

C ********************************************************************
C
C      COORD       MODIFY XX,XY FOR PARTIAL DERIVATIVES
C                  NOT USED ANY MORE
C ********************************************************************
      SUBROUTINE COORD(NV,NC,NNP,X,XX,XY)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(300),XX(NNP),XY(NNP)
      XX(195)=X(1)
      XY(195)=X(2)
      XX(177)=X(3)
      XY(177)=X(4)
      XX(155)=X(5)
      XY(155)=X(6)
      XX(133)=X(7)
      XY(133)=X(8)
      XX(111)=X(9)
      XY(111)=X(10)
      XX(89)=X(11)
      XY(89)=X(12)
      XX(67)=X(13)
      XY(67)=X(14)
      XX(45)=X(15)
      XY(45)=X(16)
      XX(23)=X(17)
      XY(23)=X(18)
      XX(1)=X(19)
      XY(1)=X(20)
      RETURN
      END
```

GRAD.COM

```
$ RUN CART
$ RUN PASDAT
$ DEL CART%.DAT;*
$ DEL CART%%.DAT;*
$!
$ SETUP PATRAN
$ PATRAN
BAT
SES
PATRAN1.SES;1
$ DEL PATRAN.DAT;*
$ REN PATRAN1.SES P.SES
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN1.OUT
$!
$ PATRAN
BAT
SES
PATRAN2.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN2.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN2.OUT
$!
```

```
$ PATRAN
BAT
SES
PATRAN3.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN3.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN3.OUT
$!
$ PATRAN
BAT
SES
PATRAN4.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN4.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN4.OUT
$!
$ PATRAN
BAT
SES
PATRAN5.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN5.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN5.OUT
$!
$ PATRAN
BAT
SES
PATRAN6.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN6.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN6.OUT
$!
$ PATRAN
BAT
SES
PATRAN7.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN7.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN7.OUT
$!
$ PATRAN
BAT
SES
PATRAN8.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN8.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN8.OUT
$!
$ PATRAN
BAT
SES
PATRAN9.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN9.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN9.OUT
$!
$ PATRAN
BAT
SES
PATRAN10.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN10.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN10.OUT
$!
```

```
$ PATRAN
BAT
SES
PATRAN11.SES;1
$ DEL PATRAN.DAT;*
$ DEL PATRAN11.SES;*
$ DEL PATRAN.SES;2
$ REN PATRAN.OUT PATRAN11.OUT
$ ANSI
$ LOGOUT
```

CLEAN.COM

```
$ DEL DATAA.DAT;
$ DEL SIGMA.NOD;
$ DEL P.SES;
$ LOGOUT
```

CLOSE.COM

```
$ RUN PROJECT5B
$ DEL OUTPUT.DAT;*
$ DEL PATRAN%.OUT;*
$ DEL PATRAN%%.OUT;*
$ LOGOUT
```

SAVE.COM

```
$ APPEND DATAA.DAT; ALL_DATAA.DAT
$ APPEND P.SES; ALL_P.SES
$ APPEND SIGMA.NOD; ALL_SIGMA.NOD
$ LOGOUT
```

CART.FOR

```
C
C       THIS PROGRAM CONVERTS THE POLAR COORDINATES IN DATAA.DAT
C       TO CARTESIAN COORDS FOR THE PURPOSE OF WRITING INTO PATRAN.SES
C
C       DOES IT ONCE FOR THE UNPERTURBATED GEOMETRY AND 8 TIMES FOR EACH
C       PERTURBATED VARIABLE
C
C       VARIABLE PARAMETERS:
C
C       NV = NUMBER OF VARIABLES (AFFECTS FNMBR,LIMITS ON LOOP COUNTERS
C            AND THE BASIC ALGORITHM IN GENERAL)
C       H = STEP SIZE
C       T = POLAR COORD THETA (AFFECTS THE BASIC ALGORITHM )
C
C

        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION X(30),XI(30),XX(30),YY(30),T(30)
        CHARACTER*33 FNMBR
        DATA FNMBR/'123456789101112131415161718192021'/
        NV=10
        H=0.05
        CEN=25.0
        T(1)=90.00
        T(2)=70.00
        T(3)=50.00
        T(4)=30.00
```

```
      T(5)=10.00
      T(6)=T(5)
      T(7)=T(4)
      T(8)=T(3)
      T(9)=T(2)
      T(10)=T(1)
C
C
      OPEN(UNIT=10,FILE='DATAA.DAT',STATUS='OLD')
      READ(10,1001)(X(I),I=1,NV)
      CLOSE(10)
C
C     FOR THE UNPERTURBATED GEOMETRY
C
      OPEN(UNIT=9,FILE='CART1.DAT',STATUS='NEW')
      DO 10 I=1,5
      XX(I)=X(I)*DCOSD(T(I))
      YY(I)=CEN+X(I)*DSIND(T(I))
 10   CONTINUE
      DO 20 I=6,NV
      XX(I)=X(I)*DCOSD(T(I))
      YY(I)=CEN-X(I)*DSIND(T(I))
 20   CONTINUE
      DO 30 I=1,NV
 30   WRITE(9,1002)(XX(I),YY(I))
      CLOSE(9)
C
C
      DO 70 J=1,NV
      KK=J+1
      IF (KK.LE.9) THEN
      OPEN(UNIT=9,FILE='CART'//FNMBR(KK:KK)//'.DAT',STATUS='NEW')
      ELSE
      JJ=2*KK-10
      OPEN(UNIT=9,FILE='CART'//FNMBR(JJ:JJ+1)//'.DAT',STATUS='NEW')
      ENDIF
C
C
      X(J)=X(J)+H
      DO 40 I=1,5
      XX(I)=X(I)*DCOSD(T(I))
      YY(I)=CEN+X(I)*DSIND(T(I))
 40   CONTINUE
      DO 50 I=6,NV
      XX(I)=X(I)*DCOSD(T(I))
      YY(I)=CEN-X(I)*DSIND(T(I))
 50   CONTINUE
      DO 60 I=1,NV
 60   WRITE(9,1002)(XX(I),YY(I))
      CLOSE(9)
      X(J)=X(J)-H
 70   CONTINUE
 1001 FORMAT(6X,D24.16)
 1002 FORMAT(6X,E16.9,6X,E16.9)
      STOP
      END
```

PASDAT.FOR

```
C
C     THIS PROGRAM MODIFIES PATRAN.SES;1 ACCORDING TO THE CARTESIAN
C     COORDS. OF THE VARIABLES (  CALCULATED BY CART.FOR )
C     ONE NEW PATRAN.SES FILE IS CREATED FOR EACH VARIABLE + ONE FOR
C     THE UNMODIFIED GEOMETRY ( WITHOUT ADDING THE STEP SIZE )
C
C     VARIABLE PARAMETERS:
C     NV = NUMBER OF VARIABLES ( AFFECTS FNMBR )
C
C
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      CHARACTER*(16) X(11),Y(11)
      CHARACTER*80 IBUF
      CHARACTER*33 FNMBR
      DATA FNMBR/'123456789101112131415161718192021'/
      NV=10
      IDEV1=11
      IDEV2=12
C
C
      DO 25 KK=1,NV+1
      OPEN(UNIT=IDEV1,FILE='PATRAN.SES;1',STATUS='OLD')
C
      IF (KK.LE.9)THEN
      OPEN(UNIT=IDEV2,FILE='PATRAN'//FNMBR(KK:KK)//'.SES',STATUS='NEW')
      OPEN(UNIT=9,FILE='CART'//FNMBR(KK:KK)//'.DAT',STATUS='OLD')
      ELSE
      JJ=2*KK-10
      OPEN(UNIT=IDEV2,FILE='PATRAN'//FNMBR(JJ:JJ+1)//'.SES',
     1STATUS='NEW')
      OPEN(UNIT=9,FILE='CART'//FNMBR(JJ:JJ+1)//'.DAT',STATUS='OLD')
      ENDIF
C
      J=1
      DO 100 I=1,1000
      READ(IDEV1,2,END=23)IBUF
      IF(IBUF(1:2).EQ.'GR')THEN
      L1=LEN(IBUF)
      N0=INDEX(IBUF(1:L1),',')
      READ(9,1,END=22)X(J),Y(J)
    1 FORMAT(6X,A16,6X,A16)
    2 FORMAT(A80)
      IBUF(N0+2:N0+17)=X(J)
      IBUF(N0+18:N0+18)='/'
      IBUF(N0+19:N0+34)=Y(J)
      IBUF(N0+35:N0+35)='/'
      IBUF(N0+36:N0+51)='0'
      ELSE
      ENDIF
   22 CONTINUE
      WRITE(IDEV2,*)IBUF
  100 CONTINUE
   23 CONTINUE
      CLOSE(IDEV1)
      CLOSE(IDEV2)
      CLOSE(9)
   25 CONTINUE
      STOP
      END
```

PROJECT5B.FOR

```
C **********************************************************************
C
C        FEM ANALYSIS PROGRAM FOR EIGHT NODE SERENDIPITY ELEMENT
C
C ***********************************************************************
C
C        * PLANE STRESS PROBLEM
C        * 3 x 3 GAUSSIAN INTEGRATION FOR STIFFNESS MATRIX
C        * STRESSES CALCULATED AT 2 x 2 INTEGRATION POINTS
C          LINEAR INTERPOLATION FOR STRESSES AT NODES
C
C        NOMENCLATURE
C        ------------
C
C        N = GLOBAL DEGREES OF FREEDOM
C        M = HALF BAND WIDTH
C        NET = TOTAL NO. OF ELEMENTS
C        NTDOF = N
C        MBAND = M
```

```
C       NTOT = NET x 16 ( SIZE OF ICONV )
C       NCON = NO. OF DISPLACEMENT B.C.'s
C       NNP = TOTAL NO. OF NODES.
C       NPE = 8 ( NODES/ELEMENT )
C       NCOL = NO. OF NODES USED FOR COLOR CODING
C
C       GK = GLOBAL STIFFNESS MATRIX IN BANDED FORM ( N,M )
C       GL = GLOBAL LOAD VECTOR ( N )
C       EL = ELEMENT LOAD VECTOR
C       E = UPPER TRI. PART OF ELEMENT STIFFNESS MATRIX STORED ROWWISE
C           AS A VECTOR ( 136 )
C       LDOF = VECTOR CONTAINING THE NO. OF DEGREES OF FREEDOM IN EACH
C              ELEMENT ( NET )
C       NBDOF = VECTOR CONTAINING THE D.O.F. NO. OF THE DISPLACEMENT
C               B.C.'s ( NCON )
C       BDIS = VECTOR CONTAINING THE VALUES OF THE DISPLACEMENT B.C.'s
C       Q = GLOBAL DISPLACEMENT ( NTDOF )
C       IBC = WORKING INTEGER VECTOR ( NTDOF )
C       YY = WORKING VECTOR ( NTDOF )
C       ICONV = CONNECTIVITY VECTOR ( NTOT )
C       XX = X COORD. OF ALL THE NODES ( NNP )
C       XY = Y COORD. OF ALL THE NODES ( NNP )
C       NOD = NODE ORDER IN ELEMENT READ IN FROM PATRAN.OUT ( NPE )
C       CX = X COORD. OF ELEMENT NODES ( NPE )
C       CY = Y COORD. OF ELEMENT NODES ( NPE )
C       CE = ELASTIC CONSTANT MATRIX ( 3,3 )
C       EK = ELEMENT STIFFNESS MATRIX ( 16,16 )
C       QI = ELEMENT DISPLACEMENT VECTOR ( 16 )
C       SIGMA = STRESSES AT 2 x 2 GAUSS POINTS( 4,3 )
C       SIGMAN= STRESSES AT NODES (8,3)
C       COLOR = STRESS MATRIX USED FOR COLOR CODING( 4,NNP )
C
C
C
C ******************************************************************

        IMPLICIT REAL*8(A-H,O-Z)
        PARAMETER(N=1252,M=92,NET=183,NTDOF=N,MBAND=M,NTOT=2928,NCON=31,
     1  NNP=626,NPE=8)
        DIMENSION GK(N,M),GL(N),EL(16),E(136),LDOF(NET),NBDOF(NCON),
     1  BDIS(NCON),Q(NTDOF),IBC(NTDOF),YY(NTDOF),ICONV(NTOT),
     2  XX(NNP),XY(NNP),NOD(NPE),CX(NPE),CY(NPE),CE(3,3),EK(16,16),
     3  QI(16),SIGMA(4,3),SIGMAN(8,3),COLOR(4,NNP)
        DATA LDOF/NET*16/
        DATA NBDOF/987,991,1039,1065,1105,1127,1159,1181,1213,1233,1243,
     1  1251,1247,1235,1221,1191,1189,1,3,7,11,13,17,19,25,23,284,288,
     2  290,298,294/
        DATA BDIS/NCON*0.0/
        OPEN(UNIT=10,FILE='OUTPUT.DAT',STATUS='NEW')
        OPEN(UNIT=9,FILE='PATRAN1.OUT',STATUS='OLD')
        OPEN(UNIT=8,FILE='SIGMA.NOD',STATUS='NEW')
        NCOL=NNP
C
C
C       INITIALIZE THE ELASTIC CONSTANT MATRIX
C
C
        XEM=400000.0
        XNU=.36
        A=XEM/(1.-XNU**2)
        CE(1,1)=A
        CE(1,2)=XNU*A
        CE(2,1)=CE(1,2)
        CE(2,2)=CE(1,1)
        CE(3,3)=XEM/(2*(1+XNU))
        CE(3,1)=0.
        CE(3,2)=0.
        CE(1,3)=0.
        CE(2,3)=0.
C
C
C       READ IN CONNECTIVITY AND NODAL PHYSICAL COORDINATES
C
```

```
C
      READ(9,1500)IJUNK
      DO 10 II=1,NNP
 10   READ(9,1000) XX(II),XY(II),Z
      KK=1
      DO 12 II=1,NET
      READ(9,2000)(NOD(J),J=1,8)
      DO 11 JJ=1,(NPE/2)
      ICONV(KK)=2*NOD(JJ)-1
      ICONV(KK+1)=2*NOD(JJ)
      ICONV(KK+2)=2*NOD(JJ+4)-1
      ICONV(KK+3)=2*NOD(JJ+4)
      KK=KK+4
 11   CONTINUE
 12   CONTINUE
C
C
C     INITIALIZE THE GK AND GL MATRICES
C     INPUT GLOBAL NODAL LOAD VECTOR
C
C
      CALL INITM(N,M,GK)
      CALL INITV(N,GL)
      GL(988)=-0.64813333
      GL(990)=-2.59253333
      GL(984)=-1.29626666
      GL(982)=-2.59253333
      GL(978)=-1.29626666
      GL(976)=-2.59253333
      GL(972)=-1.29626666
      GL(970)=-2.59253333
      GL(966)=-1.29626666
      GL(964)=-2.59253333
      GL(960)=-1.29626666
      GL(958)=-2.59253333
      GL(938)=-1.29626666
      GL(906)=-2.59253333
      GL(886)=-1.29626666
      GL(854)=-2.59253333
      GL(834)=-1.29626666
      GL(802)=-2.59253333
      GL(782)=-1.48146666
      GL(750)=-3.33333333
      GL(728)=-1.66666666
      GL(688)=-3.33333333
      GL(662)=-1.66666666
      GL(614)=-3.33333333
      GL(610)=-0.83333333
C
      GL(283)=-0.41666666
      GL(287)=-1.66666666
      GL(289)=-0.83333333
      GL(297)=-1.66666666
      GL(293)=-0.41666666
C
C     CALL ELEMENT STIFFNESS , GLOBAL STIFFNESS ASSEMBLY, BOUNDARY
C     CONDITION, GAUSS-CHOLESKY DECOMPOSITION, AND DISPLACEMENT
C     SOLUTION SUBROUTINES. THIS GIVES THE GLOBAL NODAL DISPLACEMENT.
C
C
      DO 1 LNUM=1,NET
      DO 13 JJ=1,NPE
      KK=(LNUM-1)*16+(2*JJ)
      NN=ICONV(KK)/2
      CX(JJ)=XX(NN)
      CY(JJ)=XY(NN)
 13   CONTINUE
      CALL D24(CX,CY,CE,EK)
      II=0
      DO 14 JJ=1,16
      DO 14 KK=JJ,16
      II=II+1
      E(II)=EK(JJ,KK)
 14   CONTINUE
```

```
      CALL ASEMBM (NTDOF,MBAND,NET,NTOT,LNUM,GK,E,ICONV,LDOF)
C     CALL ASEMBV (NTDOF,NET,NTOT,LNUM,GL,EL,ICONV,LDOF)
 1    CONTINUE
      CALL BOUND (GK,GL,NTDOF,MBAND,NCON,NBDOF,BDIS,Q)
      CALL DECOM (NTDOF,MBAND,IBC,NCON,NBDOF,GK)
      CALL SOLV (NTDOF,MBAND,IBC,GK,GL,Q,YY)
      WRITE(10,91)
C
C
C     CALCULATE STRESSES
C     WRITE OUTPUT TO OUTPUT.DAT AND SIGMA.NOD
C
C
      DO 25 LNUM=1,NET
      DO 23 JJ=1,NPE
      KK=(LNUM-1)*16+(2*JJ)
      NN=ICONV(KK)/2
      CX(JJ)=XX(NN)
      CY(JJ)=XY(NN)
 23   CONTINUE
C
C     INITIALIZE QI(I)
C
      NNN=(LNUM-1)*16
      DO 24 KK=1,16
      QI(KK)=Q(ICONV(NNN+KK))
 24   CONTINUE
C
C     CALL STRESS AND STRESSN
C
      CALL STRESS(CX,CY,CE,QI,SIGMA)
C
      DO 51 I=1,4
      CENTER =(SIGMA(I,1)+SIGMA(I,2))/2.
      RAD=SQRT((SIGMA(I,1)-SIGMA(I,2))**2+(4.*(SIGMA(I,3)**2)))/2.
      SIGMA(I,1)=CENTER+RAD
      SIGMA(I,2)=CENTER-RAD
      SIGMA(I,3)=RAD
 51   CONTINUE
C
      CALL STRESSN(SIGMA,SIGMAN)
C
C     WRITE  OUTPUT.DAT
C
      WRITE(10,92)LNUM
      WRITE(10,93)
      DO 50 I=1,8
 50   WRITE(10,99)(SIGMAN(I,J),J=1,3)
C
C     PREPARE DATA FOR SIGMA.NOD
C
C
      DO 34 II=1,NPE
      KK=(LNUM-1)*16+(2*II)
      NN=ICONV(KK)/2
      COLOR(1,NN)=COLOR(1,NN)+1
      DO 32 JJ=1,3
 32   COLOR(JJ+1,NN)=COLOR(JJ+1,NN)+ SIGMAN(II,JJ)
 34   CONTINUE
 25   CONTINUE
C
C     WRITE SIGMA.NOD
C
      NUMNOD=NCOL
      MAXNOD=NNP
      DEFMAX=1.
      NODMAX=1
      NWIDTH=3
      WRITE(8,'(80A1)')TITLE
      WRITE(8,'(2I5,E15.6,2I6)')NUMNOD,MAXNOD,DEFMAX,NODMAX,NWIDTH
      WRITE(8,'(80A1)')TITLE1
      WRITE(8,'(80A1)')TITLE2
      DO 33 II=1,NNP
      FREQ=COLOR(1,II)
```

```
      IF (FREQ.EQ.0.0) GO TO 33
      WRITE(8,'(I8,(3E13.7))')II,(COLOR(J,II)/FREQ,J=2,NWIDTH+1)
 33   CONTINUE
C
C     END OF STRESS CALCULATION
C
 99   FORMAT(/,4X,3(F10.5,2X))
 91   FORMAT (/,5X,' 8 NODE ELEMENT ')
 92   FORMAT (//,5X,'STRESS IN ELEMENT ',I5)
 93   FORMAT(/,10X,'XX',10X,'YY',10X,'XY')
      CLOSE(10)
      CLOSE(9)
 1000 FORMAT(3E16.9//)
 1500 FORMAT(I2////)
 2000 FORMAT(/8I8/)
      STOP
      END
```

BIBLIOGRAPHY

Barlow, J., 1967, "Optimal Stress locations in Finite Element Models," International Journal for Numerical Methods in Engineering, Vol. 10, pp. 243-251.

Bathe, K. J., 1982, "Finite Element Procedures in Engineering Analysis," 1st ed., Prentice-Hall, New Jersey.

Bennett, J. A., and Botkin, M. E., 1985, "Structural Shape Optimization with Geometric Description and Adaptive Mesh Refinement, "AIAA Journal, Vol. 23, No. 3, pp. 458-464.

Braibant, V., and Fleury, C., 1985, "An Approximation Concepts Approach to Shape Optimal Design," Computer Methods in Applied Mechanics and Engineering, Vol. 53, No. 2, pp. 119-148.

Cook, R. D., 1981, "Concepts and Applications of Finite Element Analysis," 2nd ed., Wiley, New York.

Crane, R. L., Hillstrom, K. E., Minkoff, M., 1980, "Solution of the General NLP Problem with Subroutine VMCON, ANL-80-64, Argonne National Lab.

Durelli, A. J., and Riley, W. F., 1965, "Introduction to Photomechanics," Prentice-Hall Inc, Englewood Cliffs, New Jersey.

Durelli, A. J., Brown, K., and Yee, P., 1978, "Optimization of Geometric Discontinuities in Stress Fields," Experimental Mechanics, Vol. 18, No. 8, pp. 303-308.

Durelli, A. J., Rajaiah, K., Hovanesian, J. D., and Hung, Y. Y., 1979, "General Method to Directly Design Stress-Wise Optimum Two-Dimensional Structures," Mechanics Research Communications, Vol. 6, No. 3, pp. 159-165.

Durelli, A. J., and Rajaiah, K., 1979, "Optimum Hole Shapes in Finite Plates Under Uniaxial Load," ASME Journal of Applied Mechanics, Vol. 46, No. 3, pp. 691-695.

Durelli, A. J., and Rajaiah, K., 1980a, "Quasi-Square Hole with Optimum Shape in an Infinite Plate Subjected to In-Plane Loading," ASME Paper No. 80-WA/DE-3.

Durelli, A. J., and Rajaiah, K., 1980b, "Lighter and Stronger," Experimental Mechanics, Vol. 20, No. 11, pp. 369-380.

Fleury, C., 1987, "Computer Aided Optimal Design of Elastic Structures," Computer Aided Optimal Design: Structural and Mechanical Systems, C. A. Mota Soares, ed., Springer Verlag, Berlin, pp. 831-900.

Galileo, G. L., 1638, "Discorsie dimonstrazioni matematiche intomo a due nuove scienze attenenti alla mecanica et i movimenti locali, " Leida.

Haftka, R. T., and Grandhi, R., 1986, "Structural Shape Optimization - a Survey," Computer methods in Applied Mechanics and Engineering, Vol.57, pp. 91-106.

Haftka, R. T., and Kamat, M. P., 1987, "Finite Elements in Structural Optimization, " Computer Aided Optimal Design: Structural and Mechanical Systems, C. A. Mota Soares, ed., Springer Verlag, Berlin, pp. 241-270.

Haug, E. J., Choi, K. K., Hou, J. W., and Yoo, Y. M., 1984, "A variational Method for Shape Optimal Design of Elastic Structures, New Directions in Optimum Structural Design, E. Atrek, R. H. Gallagher, K. M. Ragsdell and O. C. Zienkiewicz, eds. , Wiley , New York, pp. 105-137.

Hinton, E., and Campbell, J. S., 1974, "Local and Global Smoothing of Discontinuous Finite Element Functions Using a Least Squares Method," International Journal for Numerical Methods in Engineering, Vol. 8, pp. 461-480.

Juvinall, R. C., 1967, "Engineering Considerations of Stress, Strain and Strength," Mc. Graw-Hill, New York.

Mota Soares, C. A., Rodrigues, H. C., Oliveira Faria, L. M., and Haug, E. J., 1984, "Optimization of the Geometry of Shafts Using Boundary Elements," ASME Journal of Mechanisms, Transmissions and Automation in Design," Vol. 106, No. 2, pp. 199-202.

PDA Engineering, 1984, "PATRAN-G User's Guide," Vol. I and II

Powell, M. J. D., 1978, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," Proceedings of the 1977 Dundee Conference on Numerical Analysis, Lecture Notes in Mathematics, Springer-Verlag, Berlin, Vol. 630, pp. 144-157.

Schmit, L. A., 1960, "Structural design by systematic synthesis," Proceedings of the Second ASCE Conference on Electronic Computation, Pittsburgh, Pennsylvania, pp. 105-122.

Timoshenko, S. P., and Goodier, J. N., 1970, "Theory of Elasticity," 3rd ed., Mc-Graw Hill Inc., New York.

Topping, B. H. V., 1983, "Shape Optimization of Skeletal Structures : A Review," ASCE Journal of Structural Engineering, Vol. 109, No. 8, pp. 1933-1951.

CURRICULUM VITAE

Name: Sarvotham M. Bhandarkar

Permanent address: 43, Luz Avenue
Madras, Tamilnadu 600004
India

Degree and date to be conferred: M.S., 1988

Date of birth: March 22, 1965

Place of birth: Mangalore, Karnataka, India

Secondary education:	Graduation date
Vidya Mandir Senior Secondary School Madras, Tamilnadu 600004	May 1986

Collegiate institutions attended:	Dates	Degree	Date of Degree
Birla Institute of Technology and Science Pilani, Rajasthan 333031, India	1982 - 86	B.E.	June 1986
University of Maryland at College Park College Park, MD 20742	1986 - 88	M.S.	August 1988

Major: Computer Integrated Manufacture and Design

Minor: Solid Mechanics