

SYNTHESIS OF A NEAR-OPTIMUM

FINITE ELEMENT MESH

by

J. Jara-Almonte

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University

MASTER OF SCIENCE

in

Mechanical Engineering

APPROVED:

C. E. Knight, Chairman

L. D. Mitchell

R. G. Mitchiner

December, 1982

Blacksburg, Virginia

TABLE OF CONTENT

Section	page
List of Figures.....	iii
List of Tables.....	iv
Introduction.....	1
Literature Review.....	4
Theory.....	13
Implementation.....	27
Test Cases and Synthesis of a Near-Optimum Mesh.....	37
Conclusions.....	99
Recomendations for Future Work.....	102
References.....	103
Vita.....	105
Abstract	

LIST OF FIGURES

Figure	Content	page
2-1a.	Schematic of a Square Plate Under Tension.....	10
2-1b.	Near-Optimum Mesh Arrived at by Turcke.....	10
2-1c.	Near-optimum Mesh Obtained with a Mathematical Optimizer.	10
3-1.	An Element in an Intrinsic Coordinate System.....	14
3-2.	The Previous Element in a Cartesian Coordinate System...	16
3-3.	Notation Used for the Extrapolation of Stresses.....	20
4-1.	Flowchart of the Original MESS Program.....	28
4-2.	Flowchart of the Modified MESS Program.....	31
5-1a.	Schematic of the Thick Wall Cylinder, in Case 1.....	43
5-1b.	Stresses in a Thick Wall Cylinder	43
5-2.	Plot of the Stresses Within the cylinder wall.....	45
5-3.	First Mesh Used to Model the Thick Wall Cylinder.....	47
5-4a.	Plot of the Tensile X Stresses.....	49
5-4b.	Plot of the Shear Stresses.....	49
5-5a.	Plot One of Method One.....	51
5-5b.	Plot Two of Method One.....	51
5-6.	Method One Mesh	52
5-7a.	Plot One of Method Two.....	54
5-7b.	Plot Two of Method Two.....	54
5-8.	Method Two Mesh.....	55
5-9a.	Plot One of Method Three	56
5-9b.	Plot Two of Method Three.....	56
5-10.	Method Three Mesh.....	57
5-11a.	Plot One of Method One for New Mesh.....	59
5-11b.	Plot Two of Method Two for New Mesh.....	59
5-12.	Plot of the Tensile X Stresses.....	61
5-13.	Plot of the Tensile X Stresses.....	63
5-14a.	Plot One of Method Two for New Mesh.....	64
5-14b.	Plot Two of Method Two for New Mesh.....	64
5-15a.	Plot One of Method Three for New Mesh.....	65
5-15b.	Plot Two of Method Three for New Mesh.....	65
5-16.	Plot of the Tensile X Stresses.....	67
5-17.	Schematic of the Platform Studied in Case 2.....	69
5-18.	Mesh Used to Model the Platform.....	70
5-19a.	Plot One of Method One for Coarse Mesh.....	72
5-19b.	Plot Two of Method One for Coarse Mesh.....	72
5-20.	Method One Mesh.....	74
5-21a.	Plot One of Method Two for Coarse Mesh.....	75
5-21b.	Plot Two of Method Two for Coarse Mesh.....	75
5-22.	Method Two Mesh.....	77
5-23a.	Plot One of Method Three for Coarse Mesh.....	78
5-23b.	Plot Two of Method Three for Coarse Mesh.....	78
5-24.	Method Three Mesh.....	79
5-25a.	Plot One of Method One for New Mesh.....	81
5-25b.	Plot Two of Method One for New Mesh.....	81
5-26a.	Plot of the Tensile X Stresses.....	83

5-26b.	Plot of the Tensile Y Stresses.....	83
5-26c.	Plot of the Shear Stresses.....	84
5-27.	Second New Mesh based On Method One.....	86
5-28a.	Plot One of Method One for Second New Mesh.....	88
5-28a.	Plot Two of Method One for Second New Mesh.....	89
5-29.	Plot of the Tensile X Stresses.....	90
5-30a.	Plot One of Method Two for New Mesh.....	91
5-30b.	Plot Two of Method Two for New Mesh.....	91
5-31.	Plot of the Tensile X Stresses.....	93
5-32a.	Plot One of Method Three for New Mesh.....	94
5-32b.	Plot Two of Method Three for New Mesh.....	94
5-33.	Plot of the Tensile X Stresses.....	95

LIST OF TABLES

Table	Content	page
4-1	Interfacing Subroutines.....	35
5-1	Postulated Optimization Criteria.....	38
5-2	Postulated Optimization Methods.....	41
	Table 4-1 Interfacing Subroutines	

Subroutine

INTRODUCTION

Since its introduction in the 1950's, the finite element method has become a commonly accepted problem solving tool in many different fields of study. The basic premise behind this method is that a large continuous physical system can be modeled with a finite set of pieces, each of which exhibits the same physical material properties, as the continuous system. Hence, the large continuous system is broken down to a set of small elements. Theoretically, as the elements become infinitesimally small, the solution obtained with the finite element method will be the same as the solution obtained by solving the differential equations of the continuous system.

Although in theory an infinitely small mesh will give results that are the same as the analytical results, in reality this is presently not feasible. Most of the computers available today have some sort of numerical accuracy limit. The average computers are only able to carry a dozen or so significant digits in each numerical operation, resulting in a loss of accuracy with each additional operation. Furthermore, an infinitely small mesh would require an almost infinite amount of storage capacity. Although the random access memories in computers are becoming faster and more compact, it is impractical to allocate that much memory just to solve a finite element problem. Therefore, it is desirable to have some concept of what would constitute an optimal mesh for the specific problem. That is, what type of mesh will give accurate results and at the same time not exceed the physical resources available.

The goal of this study was to develop a procedure for generating a

near-optimum mesh. A near-optimum mesh is one which gives fairly accurate results with a minimum number of elements. Hence, a near-optimum mesh will reduce the amount of computer memory and time needed to solve the problem. This characteristic of near-optimum meshes is especially useful for large finite element problems, such as the ones found in industry.

The quest for an optimal finite element mesh as such did not start until the 1970's, after the finite element method had become an established solution method in industry. In the literature review section of this work, some of the different approaches taken to arrive at an optimal mesh are presented, as well as some typical examples of each approach. Although all the methods presented in the literature review will arrive at a near-optimum mesh, not all will do so with the same degree of difficulty, or amount of computation. Therefore, the method chosen for this study was the one that seemed simple to implement, and inexpensive to execute. This method is presented as the contour analysis method in the literature review section.

The theory section includes the development of the finite element displacement method, and the formulation of the energy functions that are postulated as mesh optimizing functions. The actual development is carried out for a four noded, quadrilateral, isoparametric element. It is assumed that the physical system to be modeled is made of an isotropic material, which is true for most materials.

Finally, the test case section shows how the contour analysis method would arrive at the near-optimum mesh for two different test cases. For each test case, the results obtained with each of the

methods are compared among themselves to determine which gave more reasonable results. Speed is not the only characteristic of the desirable mesh refinement method. A combination of accuracy, ease of implementation, and speed determined which of the contour analysis functions used was the better optimizing function, for the isoparametric element used in this study.

LITERATURE REVIEW

The search for an optimal mesh has been a topic of study since the early 1970's. It was a logical development once the finite element method had been proven to be a valid form of solution for many problems. The procedure of mesh refinement is a complicated process, since there are many factors to consider in an optimal mesh. In some instances the factors can be contradictory. In an optimal mesh one would expect to find accurate values. It is known that as the elements in a mesh get smaller the mesh approximates the continuous system better. This requires more elements in the mesh, thus, increasing the number of computations. However, an optimal mesh would have a good accuracy with a minimum number of computations.

Shephard (1) presents a review of the brief history of mesh optimization. Shephard's work is a chronological review of approximately 40 different studies and attempts to develop an optimal mesh. Early investigators postulated a system of equations, which had to be solved simultaneously. McNiece and Marcal (2) is one of these studies. They proposed finding the optimal mesh by including the nodal coordinates of the mesh as unknowns, along with the nodal displacements, in the development of the finite element system equations. Using the classical potential energy formulation, they had to find a displacement field with a stable and minimum potential energy. The correct displacement field of the continuum of an elastic body under load and at equilibrium, makes the potential energy a stationary value. Moreover, if the stationary value is a minimum then the equilibrium is stable (3).

Therefore, for a grid with only nodal degrees of freedom, McNiece and Marcal had to satisfy the following conditions to find a stable potential energy level:

$$\frac{\partial \pi}{\partial u_j} = 0 \quad j = 1, 2, 3, \dots, n \quad (2.1)$$

$$\frac{\partial \pi}{\partial d_i} = 0 \quad i = 1, 2, 3, \dots, m \quad (2.2)$$

where,

π is the potential energy function,
 u is the nodal coordinate of node j ,
 d is the displacement of node i ,
 m is the number of unrestrained nodal displacements,
 n is the number of unconstrained nodal coordinates.

Equation (2.2) is the basis for the standard finite element equation,

$$[K]\{D\} - \{P\} = 0 \quad (2.3)$$

where $[K]$ is the stiffness matrix, $\{D\}$ is the global displacement vector, and $\{P\}$ is the global load vector. While Eq. (2.3) is a set of linear simultaneous equations, Eq. (2.1) leads to the following non-linear equation,

$$\frac{1}{2} D \begin{bmatrix} K \\ u_j \end{bmatrix} D - \frac{P}{u_j} = 0, \quad j = 1, 2, \dots, m \quad (2.4)$$

Because of the non-linearity of Eq. (2.4), a direct solution was not feasible. Instead, McNiece and Marcal resorted to a numerical optimization solution of the above equations. The numerical method used was the direct search technique, also known as the gradient technique. However, even after every possible numerical efficiency was incorporated into the algorithm, the method proved to be extremely slow, and expensive. This same observation was made by other investigators in the field, according to Shephard (1).

In contrast to the non-linear system of equations developed by early investigators, later studies have proposed using iterative schemes to find the optimal mesh. In this approach, the problem is first solved with a relatively coarse mesh. Then, based on the values obtained with the coarse mesh, a new, better mesh is developed. After several iterations, the new mesh should eventually approach the near optimum mesh. Obviously, the criteria used to define the new mesh is critical, since it will determine the new mesh, and the optimum mesh. Consequently, the main difference among the different iterative algorithms, is the criteria used to find the near optimum grid, and the way in which the mesh transitions between the larger and smaller elements. For this reason, the iterative method studies can be broadly classified into those which use error functions as a criteria and those that use contour analysis. These categories can be broken down further by which error or contours are analyzed, and the method of mesh refinement. Hence, the two categories presented above will be described below, and some examples within each category will be outlined.

The methods that use an error or residual function as an optimizing criteria attempt to reduce the error estimate until it is below a predefined maximum. Once the criteria has been met, the mesh is said to be the near optimum mesh. Babuska and Rheinboldt (4,5) have studied error estimates in one- and two-dimensional problems. In particular, the two dimensional problem they investigated was a Dirichlet problem (constant boundary conditions) on an L shaped region. The study was based on the pointwise error ϵ ,

$$\epsilon = u - u^* \quad (2.5)$$

where, u is the finite element solution and u^* is the theoretical solution. After applying the Laplacian operator on the above equation, the following equation is obtained:

$$-K = \nabla^2 \epsilon \quad (2.6)$$

where, K is the Dirac function, since $\nabla^2 u^* = 0$. Based on this, they go on to obtain the pointwise error ϵ as a function of two addends; one a well understood harmonic function, and the other an unknown function. However this unknown function was bounded by considering the adjoining nodes and writing difference equations between the node at which the error is being computed and the neighboring nodes. The use of difference equations to bound the unknown function gave rise to the concept of nodal cells. Furthermore, they found that within a cell the dominant error term was given by the bounds of the unknown function, and not the harmonic function. Therefore, if the error within a cell is computed, by using the difference equations, then the cells which exceed the predefined maximum error are the ones which must be refined. The refinement chosen by Babuska and Rheinboldt was the addition of more degrees of freedom, additional nodes. Note that the mathematical derivations of this method have been deliberately omitted, because of the complexity -- they used Besov, Sobolev, and Hilbert spaces in the derivation. The interested reader should see references 4 and 5.

Other authors tried to improve the residual method proposed by Babuska, so as to arrive at the near optimum mesh faster. Carey and Humphrey (6) proposed normalizing the residuals with respect to the

element size such that an error per unit length was obtained. Then, the elements in the grid were ranked statistically; that is, based on the standard deviation from the average residual, they deduced in which areas the elements needed to be either refined or enlarged. Thus, an area that is three standard deviations away from the average had more nodes added, or taken away, than an area that was only one standard deviation away.

The second category of iteration schemes is the one based on the contour analysis of certain functions. This approach is not as mathematically involved as the previously discussed methods. While the two approaches above started from the equations of the finite element method and developed numerical techniques to perform the optimization, this method starts by asking, "What are the characteristics of an optimal mesh?" Several authors have attempted to answer this question for certain specific elements, by analyzing different functions which are the result of finite element solution. Turcke and McNiece (7) looked at one and two dimensional problems. Their results indicated that as one approaches the optimal mesh configuration, the amount of strain energy in each element is the same. Furthermore, Turcke concluded that in a two-dimensional problem the nodes of the optimal mesh were roughly aligned along the contours of the strain energy density function.

Further studies have used the conclusions of Turcke and McNiece, and applied them to other types of elements. Melosh and Marcal (8) studied both plate and shell element meshes. They measured the efficiency of a mesh by taking the difference between the strain energy at any point from that at the center of the element, which provides a

measure of the strain energy contributed by higher modes of displacement. The resulting algorithm was quite simple; they looked at the strain energy contours and added more degrees of freedom where the contours were closer. Shephard, et al. (9) used the same procedure as Melosh and Marcal(8). However, Shephard, et al. included an interactive post processor with their contouring routines. This post-processor worked as a pre-processor to create a second mesh based on the results of a first mesh.

This process of adding additional degrees of freedom is independent of the number of nodes originally in the mesh. Therefore, while the amount of optimization in each iteration of a mathematical optimizer, like Babuska's, was dependent on the original mesh, the contour analysis optimizer is practically independent of the original grid. Moreover, mathematical optimizers tended to overconstrain the optimal mesh, whereas in contour analysis, this is entirely up to the user. For example, the problem shown in Fig. 2-1a shows a square plate being loaded by four equal loads acting along the two diagonals of the square. This problem was modeled with a mathematical mesh optimizer by Turcke (7), and with a contour analysis scheme by Shephard (1). Because of symmetry, only the upper right-hand quadrant was modeled. In both cases the optimized meshes were made up of ideal triangular elements. Turcke's final mesh is shown in Fig. 2-1b, and Shephard's final mesh is shown in Fig. 2-1c. The mathematical mesh optimizer overconstrained the problem, by adding more elements, and it did not refine the mesh as much around the point load as Shephard did in his optimized mesh, Fig 2-1c.

As presented above, much of the current work in mesh optimization

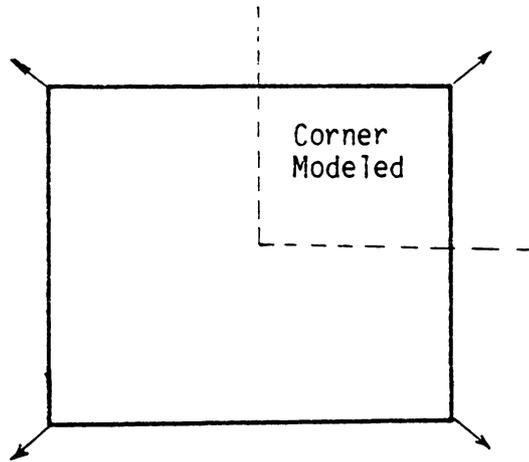


Figure 2-1a. Test problem.

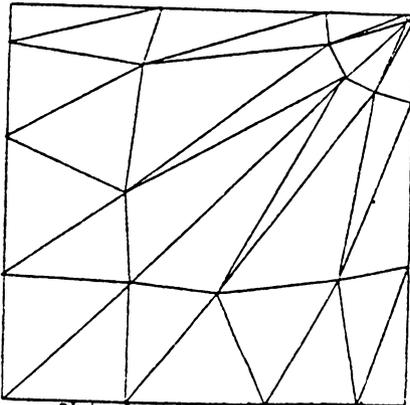


Figure 2-1b. Near-Optimum Obtained with a Mathematical Optimizer.

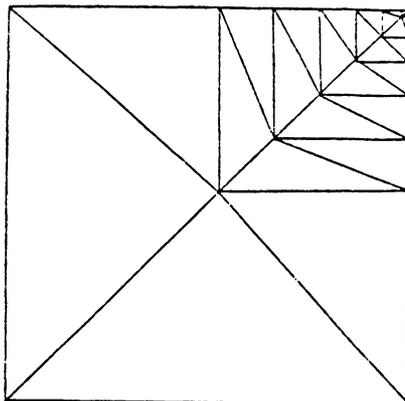


Figure 2-1c. Near-optimum Obtained with a Contour analysis.

is being done with iterative schemes. One of the objectives is the study of various scalar functions of stress that can be displayed in contours, and how these scalar functions can be used to determine areas needing refinement. A second objective of this study is to develop a simple iterative scheme that a first time user would find easy to follow. Obviously, such a scheme would be well suited for educational purposes. Therefore, the iterative scheme chosen for study was the contour analysis scheme. But, in order to perform any contour analysis, the contours must first be displayed either on a terminal screen or on a pen plotter. This can be a problem when the function to be plotted is very discretized; that is, there are big jumps in value from one element to the next. Experienced finite element users are quite familiar with this phenomena when stresses are reported at the center of the elements. Therefore, some way had to be found to extrapolate the stresses to the nodes, and ,thus, cut down on the large jumps in values.

In the finite element method a continuous system is modeled as a set of piecewise continuous functions. In particular, the displacements are generally forced to have inter-element continuity. This is called compatibility. Nevertheless, this does not guarantee that the derivatives of the piecewise continuous displacement function will be continuous across element boundaries, and in many cases they are not. Thus, one might have one element in compression with the adjacent one in tension. This effect is sometimes compounded by the way in which element stresses are reported. Usually, the Gauss point stresses are averaged, and then the average is said to be the element stress. Hinton and Campbell (10) proposed using a least square plane on each

element. With the plane, the Gauss point stresses of an element are extrapolated to the nodes, and are then averaged with the values contributed by the surrounding elements at a node. Hinton and Campbell's smoothing plane can be expressed by the matrix presented in the theory section that follows.

THEORY

A. ISOPARAMETRIC ELEMENT FORMULATION

The mesh optimization studied here was performed on an existing finite element code at Virginia Polytechnic Institute and State University. This code was known as the MESS program, which contained several elements. Two of these elements were chosen for this study. The elements were a two-dimensional, plane-stress element and an axisymmetric element, which were formulated based on an isotropic, isoparametric element formulation. The theoretical development presented here will be for a two-dimensional, plane-stress, isoparametric, quadrilateral element. The reader may wish to refer to Cook (3) and Zienkiewicz (11) for a more detailed explanation on the other members of the isoparametric element family, and on other element formulation techniques.

Since their introduction in the 1960's, isoparametric elements have been successfully used in various problem areas. They are widely employed in the areas of plates, general shells, and two- and three-dimensional elastic problems. The basis of the isoparametric formulation lies in the choice of functions used to describe the element shape, as well as the element displacement field. The element is named isoparametric because the same set of functions (parameters) are used for both shape and displacement.

Isoparametric elements are formulated in an intrinsic coordinate system ξ, η , as shown in Fig. 3-1. Figure 3-1 is a square, whose center is located at the origin of the ξ, η coordinate axes. If this element

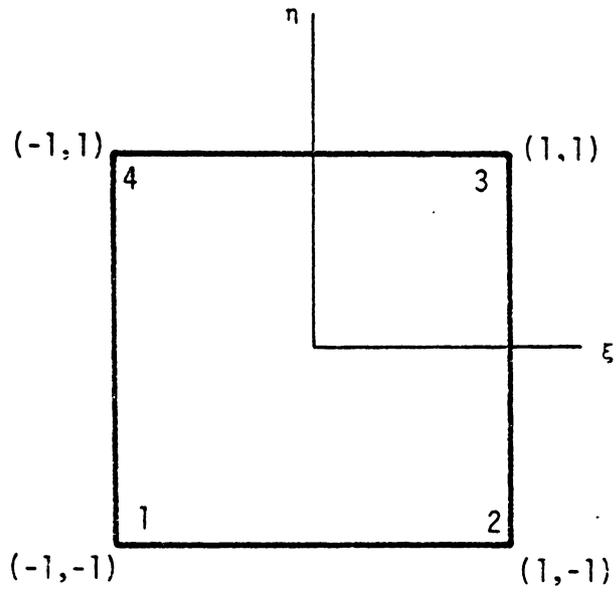


Figure 3-1. An Element in an Intrinsic Coordinate System.

were mapped into a Cartesian coordinate system, the result would be as shown in Fig. 3-2. Moreover, with the notation found in Cook (3), the displacement field, $\{f\}$, within the element can be expressed as:

$$\{f\} = [N]\{d\} \quad (3.1)$$

or in expanded form,

$$\begin{Bmatrix} U \\ V \end{Bmatrix} = \begin{bmatrix} N1 & 0 & N2 & 0 & N3 & 0 & N4 & 0 \\ 0 & N1 & 0 & N2 & 0 & N3 & 0 & N4 \end{bmatrix} \begin{Bmatrix} U1 \\ V1 \\ \cdot \\ \cdot \\ U4 \\ V4 \end{Bmatrix} \quad (3.1)$$

where, U and V are the cartesian field displacement components, the vector d is the nodal point displacement vector for the element, $U1$ through $V4$ are the nodal displacements, and $[N]$ is the matrix containing the shape functions for the element. The matrix $[N]$ is the same matrix used to describe the element shape,

$$\{X\} = [N]\{X_i\} \quad (3.2)$$

or in expanded form,

$$\begin{Bmatrix} X \\ Y \end{Bmatrix} = \begin{bmatrix} N1 & 0 & N2 & 0 & N3 & 0 & N4 & 0 \\ 0 & N1 & 0 & N2 & 0 & N3 & 0 & N4 \end{bmatrix} \begin{Bmatrix} X1 \\ Y1 \\ \cdot \\ \cdot \\ X4 \\ Y4 \end{Bmatrix} \quad (3.2)$$

where, X and Y represent the coordinates of any point within the element boundaries; $X1, Y1$ through $X4, Y4$ are the global coordinates of the node points; and $N1$ through $N4$ are the shape functions, which are polynomial functions of ξ, η . The shape functions, arranged in the manner shown in Eqs. (3.1) and (3.2) make up the $[N]$ matrix.

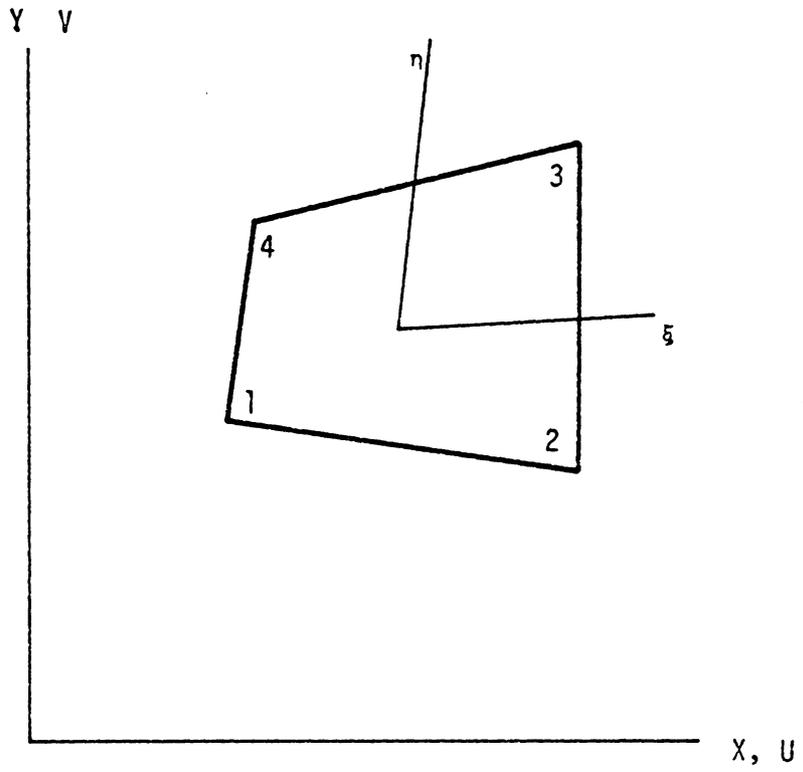


Figure 3-2. The Element of Fig. 3-1 in a Cartesian Coordinate System.

Since the quantities of interest, such as stresses and strains, are differential functions of the displacement field, a relation between the derivatives in both coordinate systems must be found. This relationship is commonly known as the Jacobian matrix, [J]. The notation that will be used below for the partial derivatives of functions also comes from Cook (3). In particular, a comma subscript will be used to indicate a partial derivative with respect to the variable following the comma.

$$\begin{Bmatrix} ()_{,\xi} \\ ()_{,\eta} \end{Bmatrix} = [J] \begin{Bmatrix} ()_{,x} \\ ()_{,y} \end{Bmatrix} \quad (3.3)$$

Equation (3.3) says that the partial derivative of a function taken in the Cartesian coordinate system can be related to the partial derivative of the same function in the intrinsic coordinate system through the Jacobian matrix. Obviously, if a partial derivative in the intrinsic coordinate must be related to a partial derivative in the Cartesian coordinate then the inverse of the Jacobian matrix is needed. The Jacobian matrix for a quadrilateral element can be expressed as:

$$[J] = \begin{bmatrix} N1,_{\xi} & N2,_{\xi} & N3,_{\xi} & N4,_{\xi} \\ N1,_{\eta} & N2,_{\eta} & N3,_{\eta} & N4,_{\eta} \end{bmatrix} \begin{bmatrix} X1 & Y1 \\ X2 & Y2 \\ X3 & Y3 \\ X4 & Y4 \end{bmatrix} \quad (3.4)$$

As can be seen from Eqs. (3.3) and (3.4), the Jacobian matrix is a two by two matrix for the element under study. The strains obtained in the intrinsic coordinate system must be related to the strains in the Cartesian coordinate system below. Thus, the inverse Jacobian matrix must be found to carry this out. Let us denote the inverse of the

Jacobian matrix as follows:

$$[J]^{-1} = \begin{bmatrix} J11^* & J12^* \\ J21^* & J22^* \end{bmatrix} \quad (3.5)$$

With this notation the partial derivatives in the Cartesian system are found from the derivative values in the intrinsic coordinate system.

$$\begin{Bmatrix} u_{,x} \\ u_{,y} \\ v_{,x} \\ v_{,y} \end{Bmatrix} = \begin{bmatrix} J11^* & J12^* & 0 & 0 \\ J21^* & J22^* & 0 & 0 \\ 0 & 0 & J11^* & J12^* \\ 0 & 0 & J21^* & J22^* \end{bmatrix} \begin{Bmatrix} u_{,\xi} \\ u_{,\eta} \\ v_{,\xi} \\ v_{,\eta} \end{Bmatrix} \quad (3.6)$$

where, the u and v in the equation above are the displacements in the X and Y directions, respectively at any point inside the element. Hence,

$$\{f\} = \begin{Bmatrix} u \\ v \end{Bmatrix} \quad (3.7)$$

The strain-displacement relation in matrix form is,

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{Bmatrix} u_{,x} \\ u_{,y} \\ v_{,x} \\ v_{,y} \end{Bmatrix} \quad (3.8)$$

Finally, combining Eqs. (3.1), (3.6) and (3.8), a relation between the element strain vector, in the Cartesian coordinate system, and the displacements, in the intrinsic coordinate system, can be derived,

$$\{\epsilon\} = [B]\{d\} \quad (3.9)$$

where, the $[B]$ matrix contains said relation. For the plane-stress, quadrilateral element used here, the $[B]$ matrix is a three by eight matrix. With the $[B]$ matrix known, the element stiffness matrix developed by Cook (3) is computed by the following integral over the element volume:

$$[k] = \int_{vol} [B]^T [E] [B] dV \quad (3.10)$$

where, $[k]$ is the element stiffness matrix, an eight by eight matrix for the element considered here; and $[E]$ is the stress-strain relation matrix. The matrix $[E]$, a three by three matrix, is as shown below for an isotropic material,

$$[E] = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \quad (3.11)$$

where, E is the Young's modulus for the element, and ν is Poisson's ratio. The integral in Eq. (3.10) is rather difficult to compute analytically because of the inherent nonlinearities in matrix $[B]$. Therefore, in practice it is computed by a numerical Gauss integration technique. For the plane-stress element the $[k]$ matrix may be computed with 2X2 Gauss integration. The Gauss points for the integration are shown in Fig. 3-3. From Eq. (3.10) the global stiffness matrix for the entire model can be assembled by addition of the element stiffness matrices, and inverted to get:

$$\{D\} = [K]^{-1} \{F\} \quad (3.12)$$

where,

$\{D\}$ is the global nodal displacement vector,
 $\{F\}$ is the external load vector,
 $[K]$ is the global stiffness matrix.

B. STRESS COMPUTATIONS

Having computed the displacements of each degree of freedom with Eq. (3.12), the next step is the calculation of stresses. This can be done

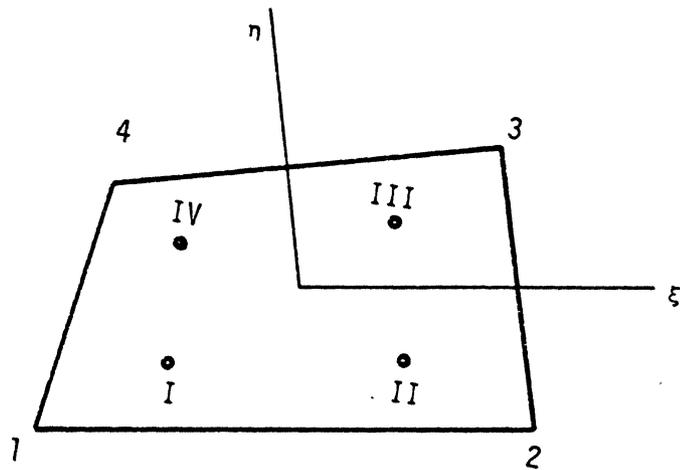


Figure 3-3. Notation Used for the Extrapolation of Stresses.

by using Eq. (3.9) along with the stress-strain relations, which result in the following equation:

$$\{\sigma\} = [E][B]\{d\}. \quad (3.13)$$

For accuracy the stresses obtained with Eq. (3.13) are computed at the Gauss points. Normally, the Gauss point stresses, which result from Eq. (3.13), are averaged for each element. The ensuing average represents the stress at the center of the element. This averaging magnifies the discrete jumps in the stresses reported at adjacent elements. When the stress gradient is fairly flat across two elements, the stress jump is not as evident as when the gradient is quite steep. Unfortunately, in many instances the locations of interest lie in an area where the stress gradient is fairly steep. For example, the regions adjacent to a constrained boundary will see steep stress gradients, as well as the regions around a stress raiser, such as holes or sharp angles. Therefore, if the stresses are to be plotted as smooth contours some way must be found to smooth the discrete jumps. Hinton and Campbell (10) have proposed a method to smooth stresses by extrapolating the Gauss point stresses out to the nodes, where they are then averaged with the extrapolated values from the other elements that use the node. This method was adopted for the MESS post-processor, not only for the smoothing capabilities, but also because the plotting routines used to display the stresses required that the plotted functions be defined at the nodes.

Hinton and Campbell (10) suggested passing a plane through the Gauss point stresses to reduce the discrete jumps in element stresses. This plane extrapolates the Gauss point stresses to the nodes by a least

squares fit through the Gauss point values. Thus, the stresses at the nodes of the element shown in Fig. 3-3 would be:

$$\{\sigma_n\} = [S] \{\sigma_g\} \quad (3.14)$$

where,

$\{\sigma_n\}$ is the smoothed nodal stress vector,

$\{\sigma_g\}$ is the unsmoothed Gaussian stress vector, and

$[S]$ is the least square matrix.

The least square matrix is dependent on the number of nodes and integration technique used in each element. For the quadrilateral element and the 2 X 2 Gaussian integration used in this study, the $[S]$ matrix is:

$$[S] = \begin{bmatrix} 1+\sqrt{3}/2 & 1/2 & 1-\sqrt{3}/2 & -1/2 \\ -1/2 & 1+\sqrt{3}/2 & -1/2 & 1-\sqrt{3}/2 \\ 1-\sqrt{3}/2 & -1/2 & 1+\sqrt{3}/2 & -1/2 \\ -1/2 & 1-\sqrt{3}/2 & -1/2 & 1+\sqrt{3}/2 \end{bmatrix} \quad (3.15)$$

This matrix has the property of extrapolating the stresses out to the nodes. However, several elements may share one node, and not all of them may have the same value for the extrapolated stress at the shared node. Thus, the stresses that have been extrapolated out to a particular node, by the surrounding elements, are averaged, and the average is called the smoothed nodal stress value.

C. SCALAR ENERGY FUNCTIONS

In the literature review section, the mesh optimization process based on contour analysis was introduced. Among the published contour analysis studies, the commonly used function was the strain-energy density. But, the strain-energy density is not the only function based on the internal energy of an element. The object of this study was to examine other energy-based functions that are not mentioned in

literature, to determine which energy-based function is a better optimizing function. Hence, a total of six different functions were programmed into the MESS code to see what type of optimized mesh would result by using each one of these functions as optimization criteria. The first function considered was the strain-energy density, since it seems to be the popular one in the literature. Among the other functions considered was the strain-energy per element, which is a normalized version of the strain-energy density. The strain-energy, as presented below, is computed with the normal stresses of the element. The same equation can be used with the deviatoric stresses, which are a measure of the stress variation within the element. This deviatoric-stress-based, strain-energy density can also be normalized by multiplying its value by the element area, resulting in a deviatoric-stress-based strain-energy per element. Two other functions studied as optimization functions were the Von Mises stresses and the Von Mises stress times the element area. The six functions just presented are described below.

The strain-energy density (strain-energy per unit volume), U_0 , can be obtained with following relation (3):

$$U_0 = 1/2 \{\epsilon\}^T [E] \{\epsilon\} \quad (3.16)$$

where, the initial strains and stresses are assumed to be zero. Recalling that $\{\sigma\} = [E] \{\epsilon\}$, and that the $[C]$ matrix is the inverse of the $[E]$ matrix, the following relationship can be derived:

$$\{\sigma\}^T = \{\epsilon\}^T [C]^T. \quad (3.17)$$

The $[C]$ matrix is symmetric for an elastic material. Also, the

transpose of a symmetric matrix is the original matrix. Thus, substituting Eq. (3.17) back into Eq. (3.16) one can see that the strain-energy density can also be computed by:

$$U_0 = 1/2 \{\sigma\}^T [C] \quad (3.18)$$

where, the stresses in Eq. (3.18) are taken to be the stresses at the Gauss points, since the program MESS already computes stresses at the Gauss points. The [C] matrix for a plane-stress quadrilateral, isotropic element is:

$$[C] = 1/E \begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{bmatrix} \quad (3.19)$$

Since the Gauss point stresses are used to compute the strain-energy density, the energy density must be extrapolated to the nodes using Eq. (3.15), and smoothed there by averaging.

To obtain the strain-energy per element, one must multiply the strain-energy density of Eq. (3.18) by the volume of the element. But, since the strain-energy density defined by Eq. (3.16) was for an element of unit volume, the strain-energy per element can be converted to the strain-energy per unit thickness by multiplying the strain-energy by the element area only. Therefore, an expression for the area of a quadrilateral, in terms of the nodal coordinates is needed,

$$\text{AREA} = 1/2 \{ [(X1)(Y2)+(X2)(Y3)+(X3)(Y4)+(X4)(Y1)] \\ - [(Y1)(X2)+(Y2)(X3)+(Y3)(X4)+(Y4)(X1)] \} \quad (3.20)$$

where, the variables X1 through Y4 are the node point coordinates.

As pointed out earlier, previous work in contour analysis had determined that a near optimal mesh resembled the strain-energy density

contour for that grid. If this is the case, it implies that if the strain-energy density is multiplied by the area of the element in which it was computed, the optimal mesh should have a constant strain-energy per element.

One of the principle reasons for looking at the strain-energy per element and the strain-energy density as mesh optimizing functions is that they constitute a measure of the energy within an element. They are also easy to compute from Eqs. (3.18) and (3.20), since most finite element codes compute stresses. However, there are other functions that are also a measure of the energy content of an element. Probably the best known is the Von Mises equivalent stress of an element, which is based on the distortion-energy theory. The distortion-energy theory is a commonly known method used to predict failure. This theory is normally used to compute the equivalent Von Mises stress, which for a plane-stress problem is:

$$s' = \sqrt{S1^2 - 2(S1)(S2) + S2^2 + 3 \tau12^2} \quad (3.21)$$

where, $S1$ and $S2$ are the normal stress components in the orthogonal 1 and 2 directions, and $\tau12$ is the shear stress. The stresses used in the right side of Eq. (3.21) are the stresses computed by most finite element codes. Therefore, only a few added computations are required to find a value for the Von Mises stress. Furthermore, the Von Mises stress can be multiplied by the area of Eq. (3.20) to obtain a normalized Von Mises stress per element. If the Von Mises equivalent stresses are a measure of the how the optimum mesh must be transitioned then the normalized Von Mises equivalent stresses could be used as a criteria for mesh optimization. If the Von Mises equivalent contours

values should have a flat contour for an optimal mesh.

The deviatoric strain-energy density is computed in the same way as the strain-energy density, Eq. (3.18). However, instead of the normal stresses, the deviatoric stresses are employed in Eq. (3.18). The deviatoric stresses are defined as the normal stress minus the average or hydrostatic stress for the element. For example, the deviatoric stress in the X direction would be expressed as:

$$S_x' = S_x - (S_x + S_y)/3, \quad (3.22)$$

since the normal Z stress for a plane-stress element is zero. The deviatoric Z stress will not be zero; it will be equal to the negative of the hydrostatic stress. This implies that one of the columns that was canceled in the overall isotropic [C] matrix must be brought back, as well as one of the canceled rows. Hence the [C] matrix for the deviatoric strain-energy density is a four by four matrix,

$$[C] = 1/E \begin{bmatrix} 1 & -\nu & -\nu & 0 \\ -\nu & 1 & -\nu & 0 \\ -\nu & -\nu & 1 & 0 \\ 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \quad (3.23)$$

As with the strain-energy density contour, the deviatoric strain-energy contour should give a guide to the mesh transition, while the normalized deviatoric strain-energy density per element should give a constant value for an optimum mesh. Notice, that the normal stresses used in Eq. (3.22) are taken to be the Gauss point stresses. Thus, the deviatoric based strain-energy density is computed at each Gauss point within an element. Then, it is extrapolated to the nodes with Eq. (3.15), and smoothed by averaging.

IMPLEMENTATION

As previously mentioned, the present study was performed on a modified version of an existing code at Virginia Polytechnic Institute and State University. The code is commonly known within the Mechanical Engineering department as the MESS program (Mechanical Engineer's Stress program). The logic of this program is presented in the flowchart of Fig. 4-1, which was presented by Bathe and Wilson (12). This flowchart has three distinct areas in the flow of the logic. First, the user's data is read into the main memory. Then, certain data, such as the different load cases and the element connectivity, is stored on tape. Second, the stiffness matrix of each element is calculated, and incorporated into the structure stiffness matrix. Third, the structure stiffness matrix is used to solve for the nodal displacements, which in turn are used to compute the Gauss stresses for each element. During this step, the element connectivity and load case data that was stored in step one, are brought back into main memory. Obviously, this third step must be executed once for each load case; where, a load case is defined as a certain combination of loads that are applied to the structure at the same time.

As presented in their book, the program of Bathe and Wilson contained only one type of finite element, a truss element. Two other finite elements were added to the subroutines contained in reference 12, at Virginia Polytechnic Institute and State University. Since most elements based on assumed displacement fields use basically the same algorithm to solve for the stiffness matrix, the logic flow presented in

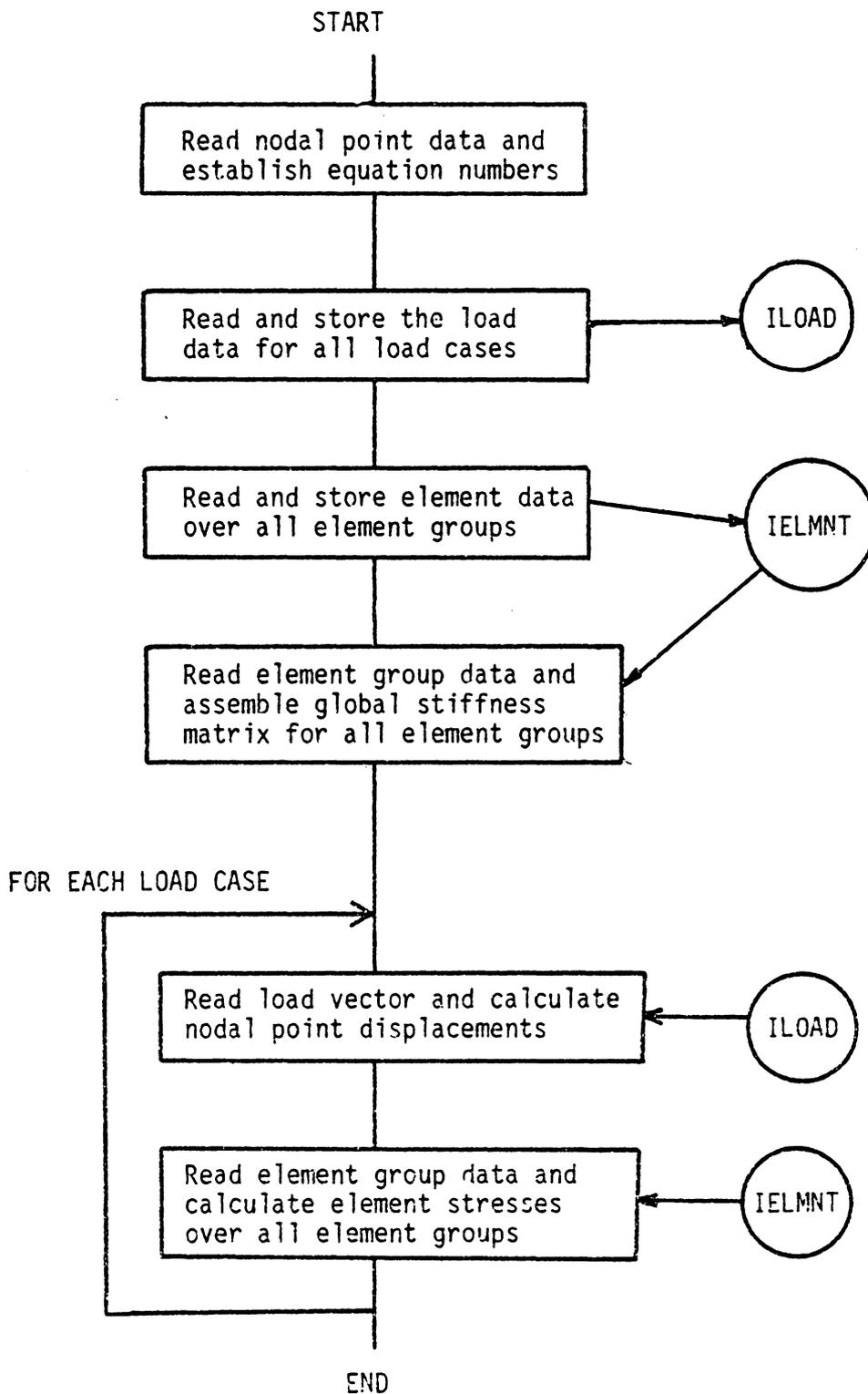


Figure 4-1. Flowchart of the Original MESS Program.

Fig. 4-1 remained the same. The two elements added to the MESS program were: first, a quadrilateral, axisymmetric, isotropic, isoparametric element; and second, a quadrilateral, plane-stress, isotropic, isoparametric element that could be degraded to a triangular element. The code for the first element resides in subroutines TRING and RING, while that for the second resides in TISOPARAMETERIC and ISOPARAMETERIC, within MESS.

At the time of this study, program MESS had primitive interactive graphic capabilities. Only the original finite element mesh and the displaced mesh could be obtained on the screen. However, neither the element numbers or node numbers could be obtained in any graphic form. Furthermore, stresses computed in MESS could not be displayed interactively on the terminal, neither in contour plot nor tabulated form. They could only be obtained through the hard-copy output of MESS. For these reasons, it was decided to add these interactive post-processing capabilities, since they would be needed for this study. Granted that the stresses could be obtained at the element centers from the hard copy output, but when comparing different meshes, with different element sizes, it is difficult to make accurate stress comparisons when the stresses are not at precisely the same location. A contour plot of the stresses would make this comparison almost trivial since the stresses at exactly the same location in two different meshes can be compared, and general stress trends can also be analyzed. On the other hand, being able to check the nodal numbering and element numbering can be quite helpful in the development of meshes, since it makes it simple to determine where a mistake has been made.

As well as modifying the capabilities of the original MESS subroutines for post-processing ease, the energy functions of interest were added to the subroutines ISOPARAMETERIC and RING. Figure 4-2 is a flowchart of the logic of the modified MESS program. It can be seen that the energy computations required for this study were computed within the original MESS subroutines, instead of separate post-processing routines. This way the storage requirements are reduced, since the Gauss stresses need not be stored, or recomputed. Also, this way the stiffness matrix of each element is computed only twice -- once for the structural stiffness matrix calculation, and once for the stress computations. However, it is obvious that unneeded computations are made if the user is not interested in obtaining all six energy contours. Nevertheless, the approach taken here allows new elements to be added to the MESS program in an easier fashion, because all the programmer needs to do is compute the specific energies in his element subroutines and write them out to a tape. Therefore, the logic of the post-processing subroutines need not be modified with the addition of new elements to MESS.

The resulting code greatly enhanced the post-processing features of MESS. The user of MESS now has the features listed below available to him or her. Note that in this list the features that the user is required to see are indicated by the word "required", while the other the features are presented if they are requested by the user. The interactive post-processing features in MESS are:

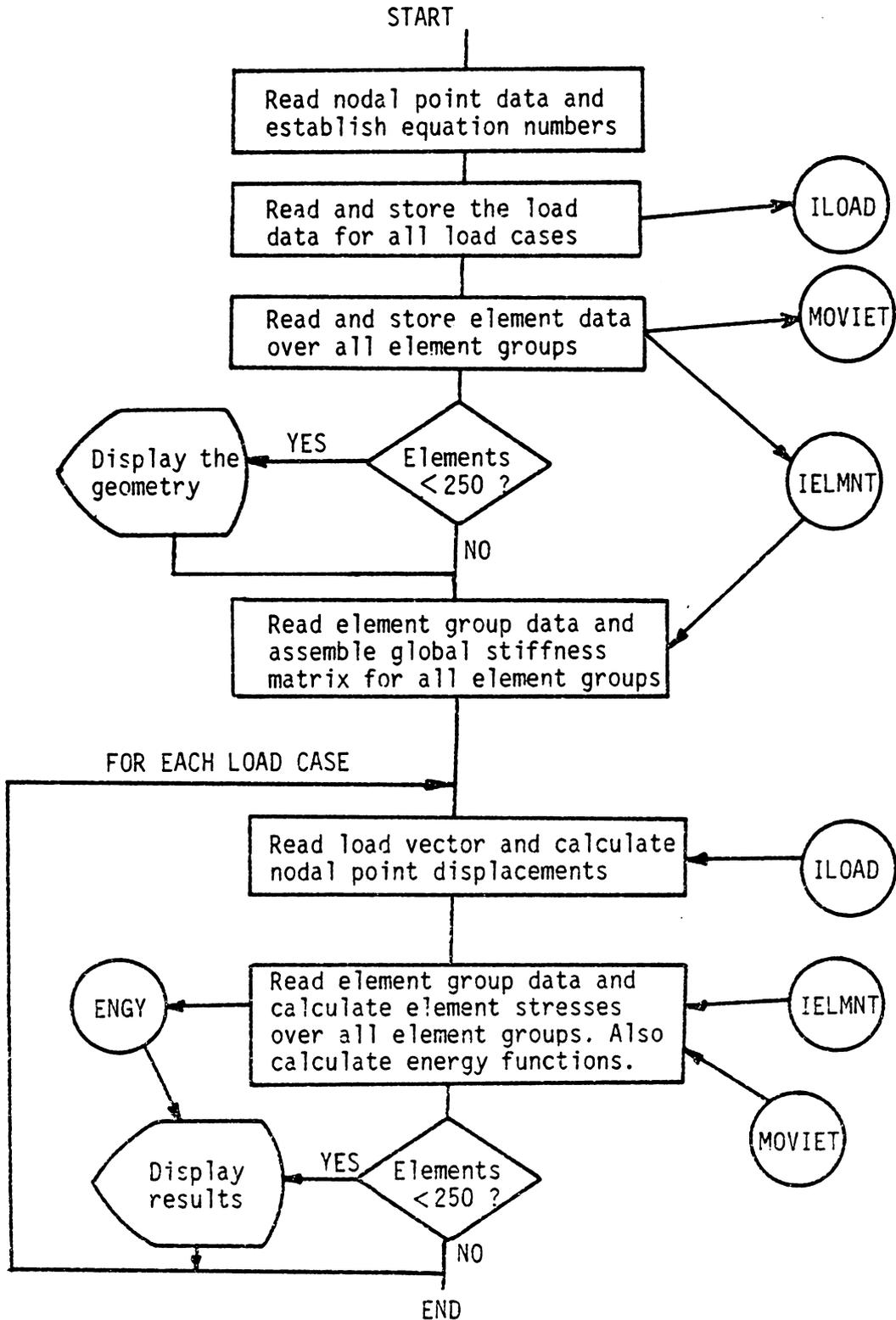


Figure 4-2. Flowchart of the Modified MESS Program.

1. A plot of the mesh (required),
2. Node number plot,
3. Element number plot, with or without zoom,
4. Displaced mesh (required),
5. Strain energy density (required),
6. Strain energy per element,
7. Strain energy density computed with the deviatoric stresses
8. Strain energy per element computed with the deviatoric stresses
9. Von Mises stresses times the element area,
10. Stress plots.

Many of the post-processing functions explained above were done by passing certain parameters to the package MOVIE.BYU, which was interfaced with MESS. The MOVIE.BYU package is distributed by Brigham Young University, as the "Movie, 1981 version". This package consists of seven independent programs, which either generate or display polygonal images. The actual arrangement and use of the seven programs will not be covered here; the reader may wish see reference 13 for this information. The important thing to note, however, is that this package was used only to show the results obtained in MESS.

The only MOVIE.BYU program of interest in this study was the DISPLAY program, because it can display polygonal models on a terminal screen or on a plotter. DISPLAY is an executable image (program) that consists of three source module libraries, COMMAND.FOR, HIDDEN.FOR, and DEVICE.FOR. As the name of the first module indicates, COMMAND.FOR contains all the subroutines necessary for interactive command processing, as well as the MAIN for program DISPLAY. Library HIDDEN.FOR contains the hidden line algorithm for three-dimensional plotting, and DEVICE.FOR contains all device dependent functions -- graphic commands for moving or drawing with the cursor or pen.

At the time of interfacing it was decided to make the post-

processor such that any novice user would find it easy to employ. For this reason all of the interactive command processing of program DISPLAY was eliminated, and replaced with questions requiring a 'yes' or 'no' answer. This entailed removing the MAIN (the part of DISPLAY that calls the different subroutines) of program DISPLAY, and modifying two of its subroutines. The MAIN was replaced with subroutine MOVIE, which is one of the interfacing routines. Subroutine MOVIE retains the essence of the MAIN -- calling the appropriate subroutines -- but does not ask any questions. Instead, the necessary questions are asked through subroutine DISPLY, another interfacing subroutine. Depending on the user's response, DISPLY passes the appropriate values to MOVIE. Although the present version of the post-processor uses only a few of the available commands in program DISPLAY, it has the capability of executing any of the interactive commands, which are summarized in subroutine MOVIE. Therefore, if a future version of the post-processor must support a color terminal, then the only modifications that would be needed would be a couple of calls to MOVIE from DISPLY with the appropriate values for the variable LIT. The possible values for the variable LIT and the corresponding interactive command are summarized in subroutine MOVIE.

Two subroutines in the DISPLAY program were modified. The first subroutine modified was SUMCEN. In the interactive version of program DISPLAY, SUMCEN presents a summary of the parameters set by the user on the screen. Since all the DISPLAY parameters are kept transparent to the MESS users, all the WRITE statements in SUMCEN were deleted, as well as the accompanying FORMAT statements. The second subroutine modified

was VIEDRA. VIEDRA contains the necessary calls to draw the indicated model on the screen. The only modifications made to VIEDRA were the addition of titles for the different plots made through VIEDRA, and the change of the FORMAT statement on line number 240. Finally, note that these two subroutines must be linked ahead of program DISPLAY so that they may replace the respective routines in program DISPLAY.

As mentioned above, subroutine DISPLY determines what values to pass to the DISPLAY program. For this reason DISPLY is the heart of the interfacing set of routines, which are presented in Table 4-1. DISPLY asks the user certain 'yes' or 'no' questions that help it determine which set of instructions to invoke. For example, some typical questions are: "Do you want node numbers? (Y/N)", "Do you want the X stresses? (Y/N)". Although it is not documented within the post-processor, any time DISPLY asks a question, the user may respond with a letter 'q' (for quit), which instructs DISPLY to quit the post-processing. DISPLY accomplishes this by setting the variable MOVIEF to FALSE in common block BMOVIE. In sum, DISPLY determines which of the subroutines in Table 4-1 to call, based on the MESS user's answers; if the MOVIE variable is set to FALSE by DISPLY or by any other means, then all post-processing will stop.

The maximum size of the problem that MESS can solve is determined by the dimension of the A array, which is stored in the only unnamed common block. If MESS requires more storage capacity, then the dimension of the A array must be increased, and the variable MTOT in MESS must be set to the new dimension. This is a simple modification, since only two lines need to be changed in the MAIN of program MESS.

Table 4-1 Interfacing Subroutines

<u>Subroutine</u>	<u>Description</u>
CLEA3D	This subroutine cleans the arrays used by the three dimensional plot of the stresses, and restores the viewing frustrum angle and other related constants.
DISPLY	This subroutine is the one which calls the other subroutines in this table. It determines which one to call depending on the user's response to 'yes' or 'no' type questions. DISPLY is set up to also accept a 'q' for an answer to these questions, in which case it will quit post-processing the MESS output.
MOVIE	MOVIE performs the calls to the subroutines in the DISPLAY program. It determines which sequence of subroutines to call by the value of the variable LIT passed to it. The possible values of LIT are summarized in Appendix B.
QUADRT	QUADRT computes new frustrums of vision to display certain portions of the model. It displays the selected portions and restores the original frustrum of vision when through.
READ3D	It reads a stress tape and converts the stresses to Z coordinates, making the stress field appear as a surface when plotted on the screen.
READG	This subroutine reads the geometry of a model and sets the appropriate values for color, viewing distance, frustrum angle, and other default constants.
READS	This subroutine reads stresses stored on tape by MESS. It displays them on the screen as contour plots.
ZOOM	ZOOM is the subroutine in charge of magnifying certain portions of the model geometry according to what the user wishes to see.

Note that any other dimension specifications of array A, in other subroutines, need not be changed because the pointers to array A will still be passed correctly. On the other hand, the maximum problem that DISPLAY can handle is limited by the dimension specifications of the arrays stored in common block QFORIO. In its original release, the maximum problem size was limited to 1,000 nodes and 250 elements. Subroutine DISPLY checks to see if the problem in MESS can be handled by the DISPLAY subroutines. If the problem is too large then the post-processing is not invoked, and variable MOVIE is set to FALSE. Changing the maximum problem size in the DISPLAY subroutines is not as simple as changing the problem size in MESS. The post-processing routines do not use a dynamic array to store the nodal data. Thus, each individual reference to the size of the arrays in common block QFORIO must be altered to the desired values. For a complete list of the subroutines involved see reference 13, and include the following routines from Table 4-1: DISPLY, READS, and READ3D.

CASE STUDIES AND SYNTHESIS OF A NEAR-OPTIMUM MESH

As previously stated, the objective of this study was to synthesize a near-optimum mesh, based on certain energy functions, in as few iterations as possible. The energy functions were presented in the theory section, and are summarized in Table 5-1. In this table, a postulated optimization criteria is presented alongside each function. A true optimum mesh based on any of these functions would have to meet the criteria in Table 5-1. However, in practice it may not be feasible or efficient to meet the criteria exactly.

In practice, an engineer has certain limitations which may not allow him to reach the optimum mesh. Some of these limitations could be, for example, the accuracy of the computer, time, and money. Each criteria of Table 5-1 assumes that the theoretically exact solution is obtained. The accuracy of the computer being used will ultimately determine the overall accuracy of the final mesh. Therefore, the best solution an engineer can obtain will be close estimate of the theoretically exact solution. In practice, the user will, also, be limited by the time he has available to solve a problem. Thus, the user may not be able to perform more than one or two iterations. Furthermore, each iteration will require a new computer solution; and, although computers are becoming less expensive to use, it will still require more money in computer time and engineering effort. Therefore, in practice, the engineer must determine what is the objective; is it to develop an optimum mesh, or to obtain reasonably accurate displacements

Table 5-1. Postulated Optimization Criteria

<u>Criteria Name</u>	<u>Optimization Criteria</u>
Strain energy density	The optimum mesh transition should resemble this function's contour plot.
Strain energy per element	The value of this function should be constant throughout the optimum grid.
Strain energy density, based on the deviatoric stresses	The optimum mesh transition should resemble this function's contour plot.
Deviatoric stress based strain energy per element	The value of this function should be constant throughout the optimum grid.
Von Mises equivalent stress	The optimum mesh transition should resemble this function's contour plot.
Von Mises Equivalent stress times element area	The value of this function should be constant throughout the optimum grid.

and stresses with a near-optimum mesh? In many cases the objective will be to obtain a fairly accurate solution based on a near-optimum mesh.

In the development of a near-optimum mesh, an engineer must work within the limitations outlined above, and other more subtle limitations. The optimization criteria, for generating a near-optimum mesh, based on the strain-energy density contours is that the near-optimum mesh should transition in the same manner as the contour plot. Thus, smaller elements are located where the contour lines are more closely spaced, and larger elements where they are far apart. This criteria sounds straightforward, until the engineer asks how many contour lines are needed? Obviously, if one contour plot has twice as many contour lines as another contour plot, then a near-optimum mesh based on the first contour plot will appear to need more elements than a near-optimum mesh based on the second contour plot. Therefore, how does the engineer select the number of contour lines to use in developing the near-optimum mesh? The answer to this question is that it need not be answered at all, because one can use the strain-energy per element criteria to determine where the near-optimum mesh is over or under refined. This criteria, summarized in Table 5-1, states that the strain-energy per element should be constant throughout the optimum mesh. This translates to some desirable ratio between the maximum and minimum contour values in a near-optimum mesh. Obviously, the closer the ratio is to unity, the closer the near-optimum mesh is to the optimum mesh. Obtaining the actual value of the desirable ratio was another objective of this study, and is discussed later in this section. But, even if everyone was to use the same ratio criteria for the near-

optimum mesh, not everyone would refine the mesh in exactly the same way. Hence, not all near-optimum meshes will look alike; however, they should give nearly the same solution.

In the discussion above, the strain-energy per element criteria in conjunction with the strain-energy density criteria are implicitly postulated as a method to determine a near-optimum mesh. In a similar way, the equivalent Von Mises stress and the equivalent Von Mises stress times element area can be postulated as an optimization method based on the distortion-energy theory. The deviatoric-stress-based strain-energy density and the strain-energy per element can also be combined to give a third mesh refinement method. Hence the six energy functions summarized in Table 5-1 were grouped into three distinct methods, as shown in Table 5-2. The postulated optimization scheme for each method is included in Table 5-2. Each method has two energy functions or contour plots associated with it. The first plot of each method is postulated to resemble the near-optimum mesh based on that particular method. The second plot of each method must have a constant value throughout the model for the optimum mesh based on that method.

The three postulated methods will lead to three different near-optimum meshes based on the results of a parent analysis. The mesh of the parent analysis is called the coarse mesh in this work. Based on the results of the coarse mesh, the three methods will be used to generate new refined meshes, and each one of these meshes will be analyzed. Then, the particular method which generated the refined mesh will be applied to the new mesh to determine whether or not the refined mesh is near-optimum. For example, once a refined mesh based on method

Table 5-2. Postulated Optimization Methods

<u>Method Number</u>	<u>Plot Number</u>	<u>Function Name</u>	<u>Criteria for Use</u>
1	1	Strain-Energy Density	The near-optimum mesh should resemble the contour plot of this function.
	2	Strain-Energy per element	The ratio of the maximum to minimum values of this function should be as close to one as possible.
2	1	Deviatoric-Stress-based Strain-Energy density	The near-optimum mesh should resemble the contour plot of this function.
	2	Deviatoric-Stress-based Strain-Energy per element	The ratio of the maximum to minimum values of this function should be as close to one as possible.
3	1	Von Mises Equivalent Stress	The near-optimum mesh should resemble the contour plot of this function.
	2	Von Mises Equivalent Stress times Element area	The ratio of the maximum to minimum values of this function should be as close to one as possible.

one has been generated and solved with program MESS, the same method one is applied to the results of the refined mesh to determine if the refined mesh is an improved mesh over the original mesh. The same scheme will be used for the other two methods of Table 5-2. These will be discussed within the test cases that follow. These test cases will also be used to establish whether or not any of the criteria in Table 5-1, or methods in Table 5-2, or a combination thereof, will lead to a near-optimum mesh faster. If there is a such a combination then the objective is to establish simple guidelines that a novice user can follow to generate near-optimum meshes efficiently.

The first test case presented below is a rather simple problem. A thick-wall cylinder is subjected to an internal pressure. The solution to this problem can be found through classical elastic theory. Therefore, it is not likely that an engineer would need to solve this problem with a finite element code. However, it was solved this way to gain an understanding of the effect and relationship of the criteria presented in Table 5-1. The conclusions obtained with this test case were then applied to a more complex problem. In the second test case, a bridge beam that is simply supported is subjected to a point load at the midspan. The point load introduces the concept of stress raiser or stress concentration, which was not present in case one.

TEST CASE 1: A thick-wall cylinder subjected to internal pressure.

Figure 5-1a is a schematic of the short cylinder studied in this test case. The cylinder has an inside radius of 5 in (127 mm), and an outside radius of 10 in (254 mm). The pressure differential between the

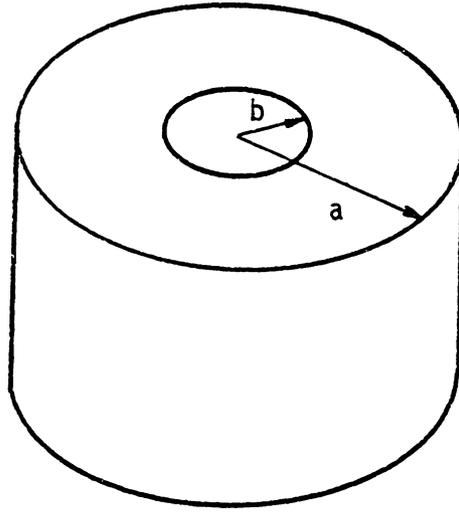


Figure 5-1a. Schematic of the Thick-Wall Cylinder, in Case 1.

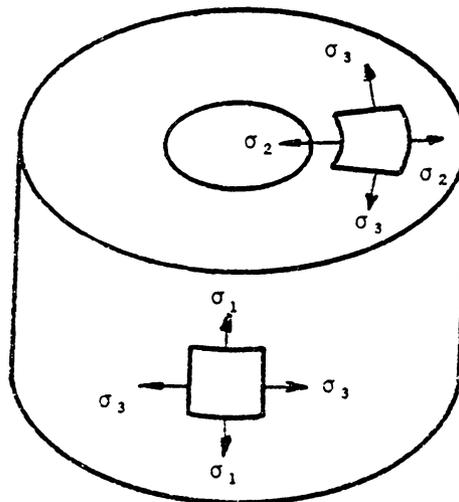


Figure 5-1b. Stresses in a Thick Wall Cylinder Under Internal Pressure.

inside and outside of the cylinder is 1,000 psi (6.89 MPa). The cylinder is made of steel, with a Young's modulus of 30 million psi (207 GPa), and a Poisson's ratio of 0.3. Furthermore, the cylinder is so short that a plane-stress state exists at the location of interest; that is, the longitudinal stress is zero.

This problem was chosen as the first test case for two reasons. First, it is free of stress concentration areas, stress raisers, which will bias the energy computations. Sharp corners and high point loads usually generate high localized stresses, which in turn generate high energy levels. These energy levels can be high enough at times to overshadow the remaining lower energy levels in the mesh. Thus, the energy levels tend to gather around the stress raiser. This will be addressed in more detail in the next test case. Second, the solution of this problem can be obtained analytically, as reported by Roark and Young (14). They found the following relation for the radial stress, σ_2 , shown on Fig. 5-1b,

$$\sigma_2 = \frac{Pb^2(a^2 - r^2)}{r^2(a^2 - b^2)} \quad (5.1)$$

where,

P is the internal pressure,
 b is the inside radius,
 a is the outside radius, and
 r is the radius of interest.

Substituting the problem parameters into equation (5.1), the lower curve in Fig. 5-2 was obtained. This curve shows that the largest, compressive, radial stress occurs at the inner radius, 1,000 psi. Then, the magnitude of the compressive, radial stress decreases to zero at the outer radius. Roark and Young (14), also found the relation for the

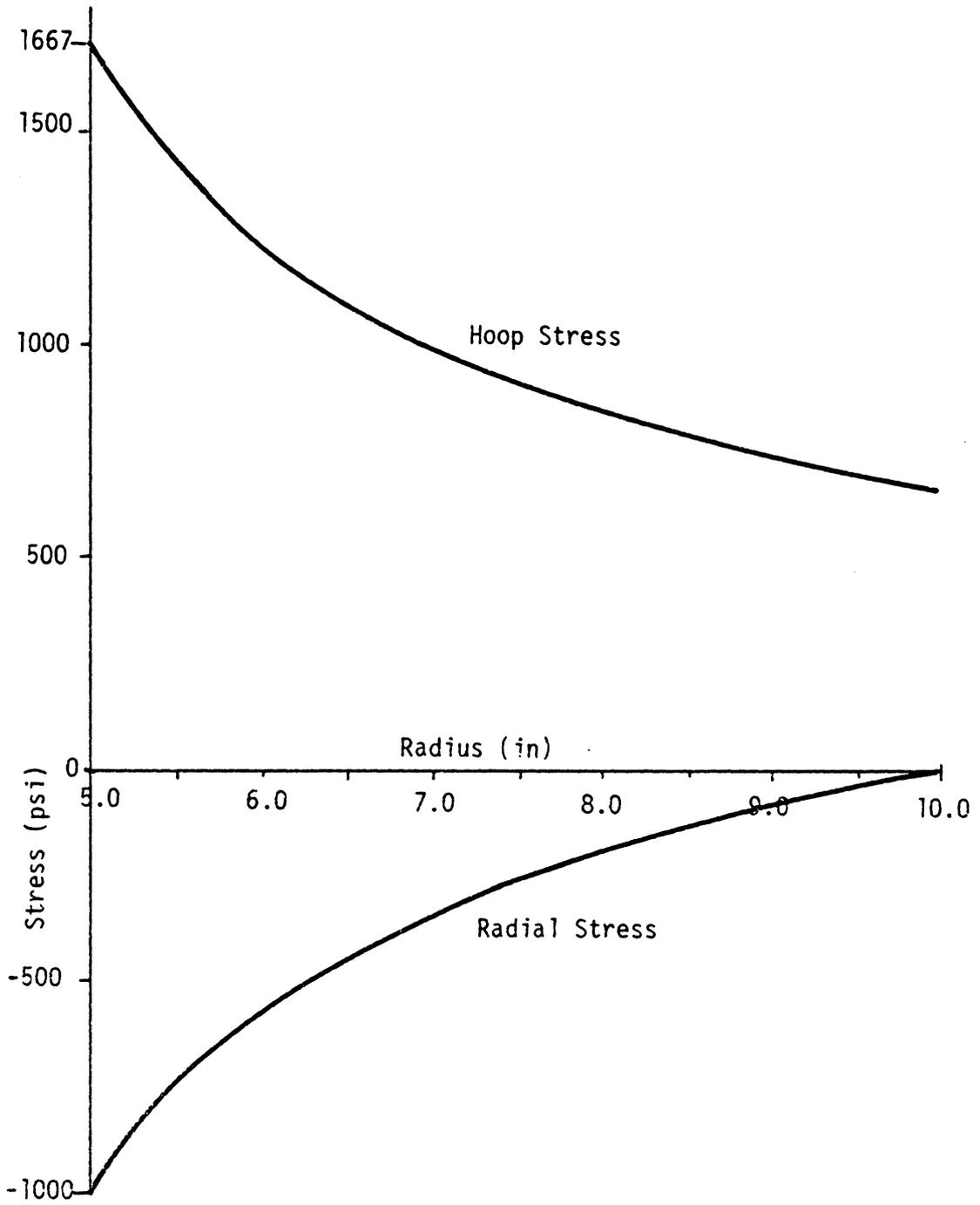


Figure 5-2. Plot of the Stresses Within the cylinder wall.

tangential, or hoop, stress, σ_s ,

$$\sigma_s = \frac{Pb^2(a^2 + r^2)}{r^2(a^2 - b^2)} \quad (5.2)$$

Upon substituting the problem parameters into equation (5.2), the upper curve in Fig. 5-2 was obtained. From this curve, it is easy to see that the highest tensile hoop stress occurs at the inner radius, 1,667 psi (11.5 MPa), then it tapers to 1,000 psi (6.89 MPa) at the outer radius. Finally, the longitudinal stress is known to be zero from the problem statement. Therefore, according to Fig 5-2, the maximum normal stress is the tensile hoop stress at the inner radius, 1,667 psi (11.5 MPa). Now, the task is to find a near-optimum mesh that will give accurate results with as few iterations as possible. The accuracy will be determined by how close the finite element stresses at the inside surface come to the analytical results, presented above.

This example problem is a one-dimensional problem. The stresses in this problem are independent of the angle, as can be seen in equations (5.1) and (5.2). Hence, this problem can be modeled with an axisymmetric finite element, which is available in MESS. Instead, it was modeled with a plane-stress, quadrilateral element. The plane-stress, quadrilateral, isoparametric element was chosen to gain a better understanding of the optimization process with this element; especially since this element was used in the next test case.

Figure 5-3 shows the fairly coarse and even mesh used in this problem. Twenty-five quadrilateral elements and thirty-six nodes were used to model a quarter of the thick-wall cylinder. This mesh is enough

WANT NODE Nos. ?
(Y,N) >

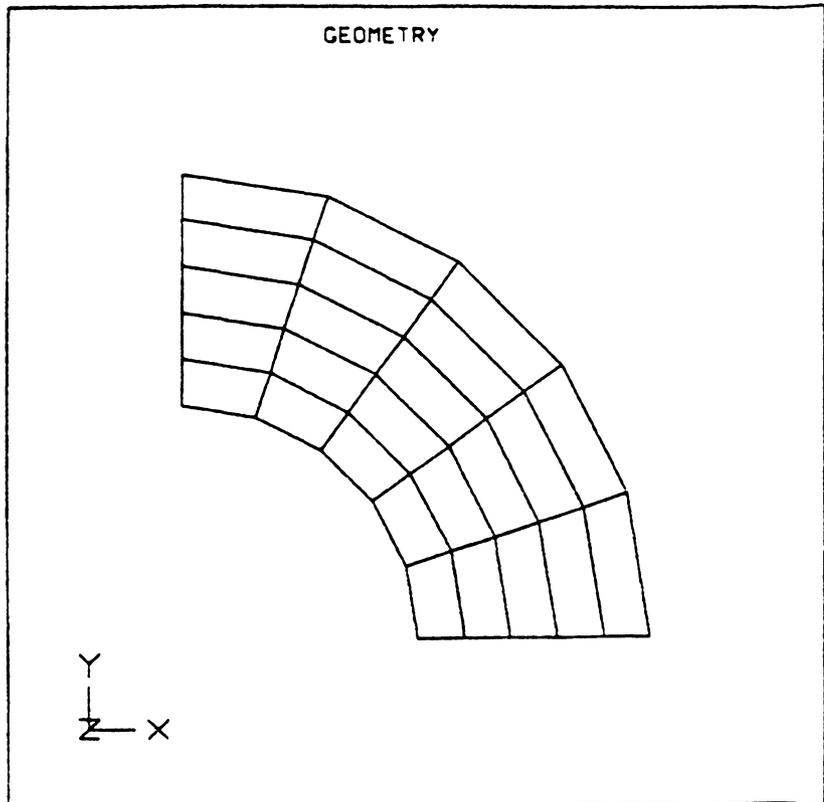


Figure 5-3. First Mesh Used to Model the Thick Wall Cylinder.

to model the entire cylinder, since the geometry and loading are symmetric. In this model, the horizontal line of symmetry was constrained in the Y direction; that is, the nodes on this line could only displace in the X direction. The vertical line of symmetry was constrained in the X direction. The internal pressure loading the structure was applied as forces in the X and Y directions on the nodes lying along the inner circumference of the quarter cylinder.

The results of interest, from the model described above, are shown in Figs. 5-4a through 5-9b. In particular Fig. 5-4a is a contour plot of the horizontal X stresses, and Fig. 5-4b is a contour plot of the shear stresses. The Y stress plot was omitted because it was a mirror image of the X stress plot about a 45 degree line. The tensile X stress at the vertical line of symmetry in Fig. 5-4a is the hoop stress. Thus, the X stress at the inner radius and on the vertical line of symmetry should be approximately equal to the maximum normal stress obtained using the formula of Roark and Young, 1,667 psi (11.5 MPa). According to the contour sequence in Fig. 5-4a, the stress level at the point of interest is slightly larger than the G contour, which is 1,665 psi (11.5 MPa). Knowing that the stress range plotted by the MESS post-processor is ninety percent of the actual stress range, the maximum tangential stress can be found to be 1,749 psi (12.0 MPa). The maximum radial stress is slightly less than the A contour, -680 psi (4.68 MPa), or a compressive 680 psi (4.68 MPa). The radial stress obtained with equation (5.1) at the inner radius was a compressive 1,000 psi (6.89 MPa). In summary, the coarse mesh of Fig. 5-3 provided a fairly accurate hoop stress, but a poor estimate of the radial stress.

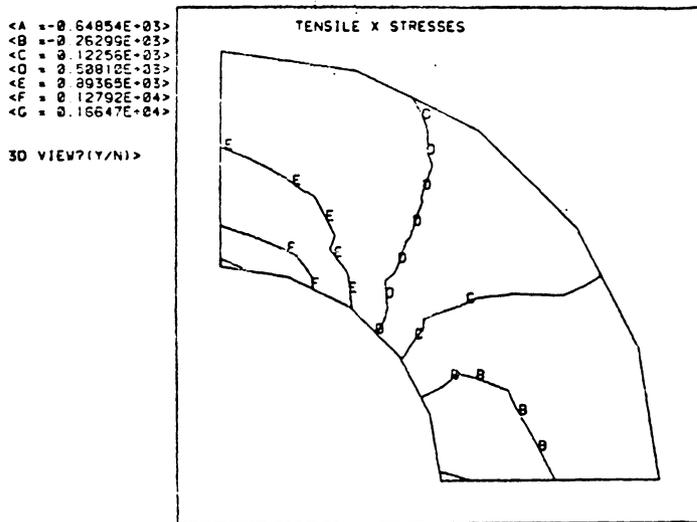


Figure 5-4a. Plot of the Tensile X Stresses

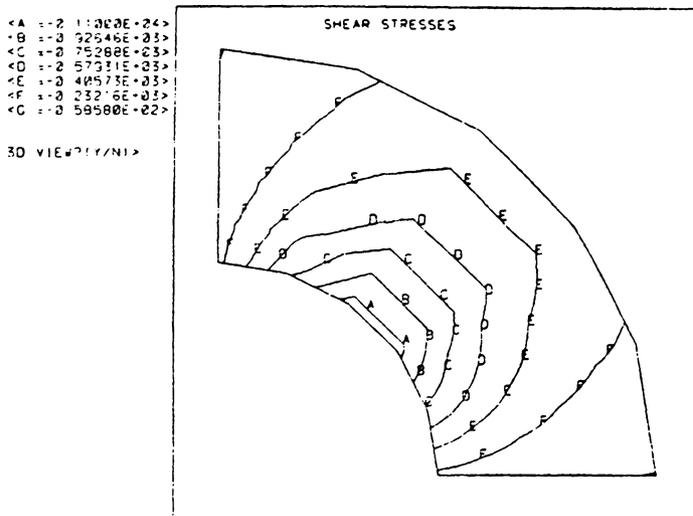


Figure 5-4b. Plot of the Shear Stresses

Therefore, the effectiveness of the new meshes will be evaluated by the value of the radial stress obtained at the inner radius.

The strain-energy density contour, for the mesh in Fig. 5-3, is shown in Fig. 5-5a, while the strain-energy per element contour is shown in Fig. 5-5b. These plots are the first and second plots of method one. According to the criteria in Table 5-1 and the plot in Fig. 5-5a, the elements of the coarse mesh should be enlarged around the outer surface of the cylinder, and reduced around the inner surface, to generate a better mesh. Obviously, there are several ways to accomplish this. Elements could be placed between the contour lines, such that two of each of the element's four sides lie on a contour of Fig. 5-5a, except the A and G contours. Instead of the A and G contours, the outer and inner boundaries would be used as element sides. On the other hand, instead of inserting just one element between each contour, two or more elements could be inserted between contours. The final choice would be up to the engineer. In this test case the number of nodes and elements were kept constant, because a measure of the effect of the grid transition on the final solution was desired. Hence, the elements in the coarse mesh were resized to give a mesh that was proportional to the maximum to minimum strain-energy density ratio of Fig. 5-5a. Moreover, they were also proportioned to give a linear mesh; that is, the amount by which the element size changed from one element to another was approximately constant. This new mesh is shown in Fig. 5-6.

Figure 5-5b shows the second plot of method one, presented in Table 5-2. According to the outline for method one, this plot should be constant for an optimum mesh. The contours in Fig. 5-5b show that the

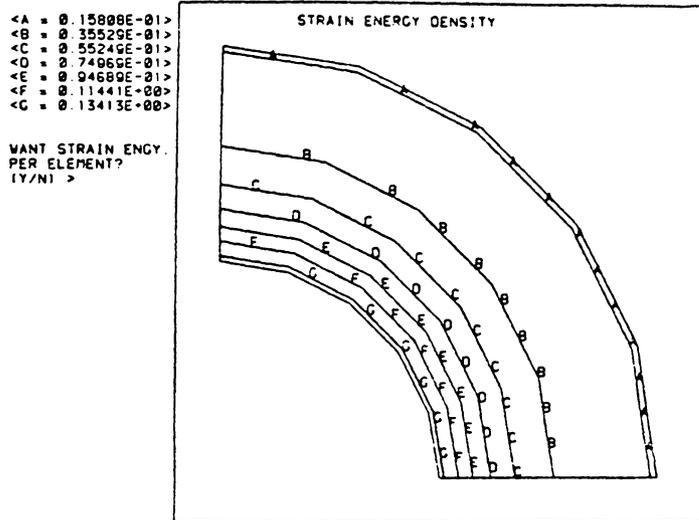


Figure 5-5a. Plot One of Method One.

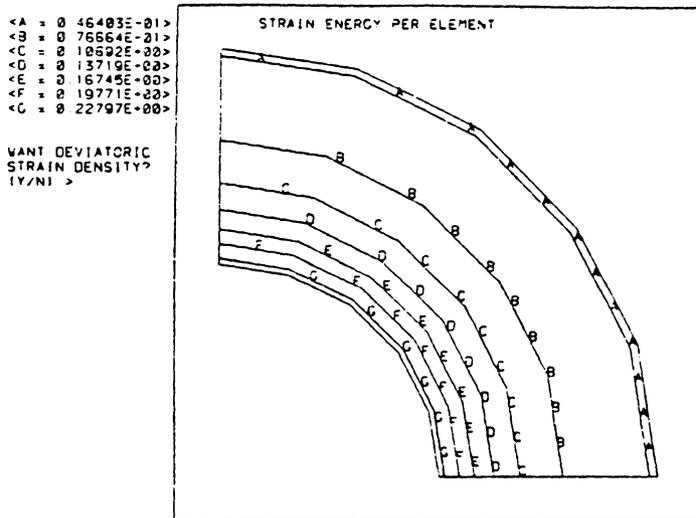


Figure 5-5b. Plot Two of Method One.

WANT NODE Nos. ?
(Y,N) >

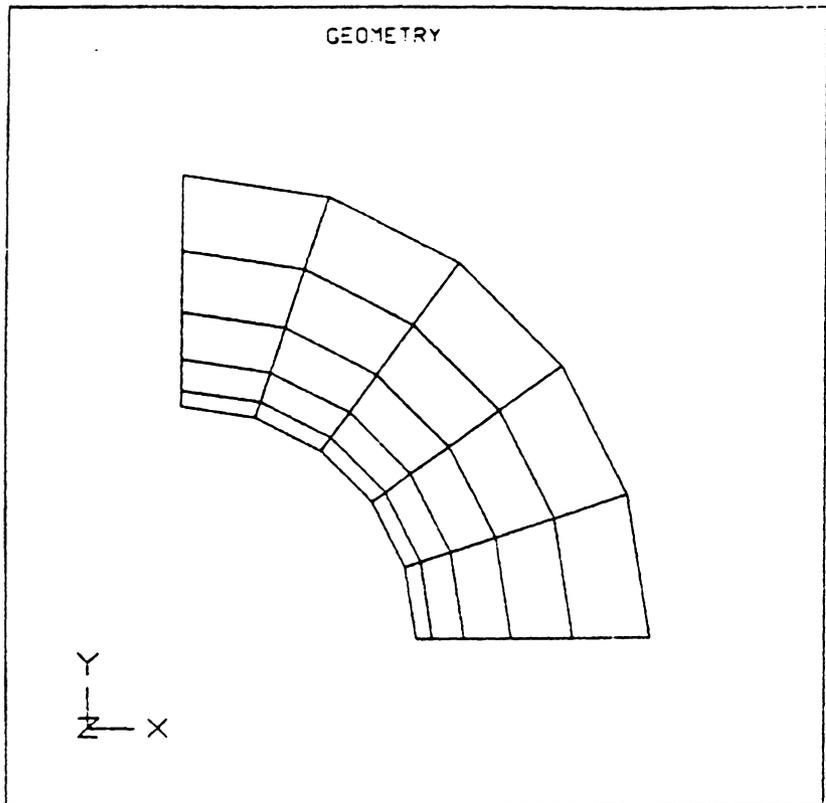


Figure 5-6. Method One Mesh

values vary from 0.04, for contour A, to 0.20, for contour G. The units on these values are not important, since only a constant value is desired. Thus, the size of the elements around contour A must be increased to raise the value of the function, while the size of the elements around the G contour must be reduced to decrease the value. This is exactly what was done by following the first plot of method one, above.

Figure 5-7a shows the first plot of method two for the coarse mesh. The contours in this plot are very similar to the ones in the first plot of method one. In fact, if the two plots are overlaid, it can be seen that the contour lines almost coincide. Thus, the mesh based on method two was very similar to the mesh from method one. This mesh is presented in Fig. 5-8. Note that the element size was arranged to vary linearly from the inner to the outer elements, as was done for the mesh of method one. Figure 5-7b shows the second plot of method two. This plot is also very similar to the corresponding plot of method one, as can be observed in Figs. 5-7b and 5-5b.

The first and second plots for the third method are presented in Figs. 5-9a and b, respectively. Like the four other energy functions reviewed above, these two contour plots of the coarse mesh indicated that the new mesh had to have larger elements around the periphery and smaller elements around the inner edge. The new mesh based on method three is presented in Fig. 5-10. Again, the number of elements and nodes was kept constant, and the mesh was transitioned linearly from the smallest to the largest elements. Although this mesh is similar to the meshes in Fig. 5-6, method one, and Fig. 5-8, method two, its grading is

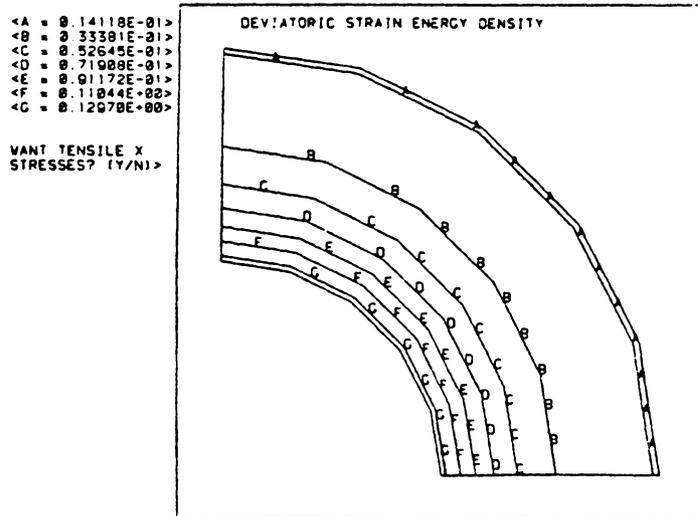


Figure 5-7a. Plot One of Method Two

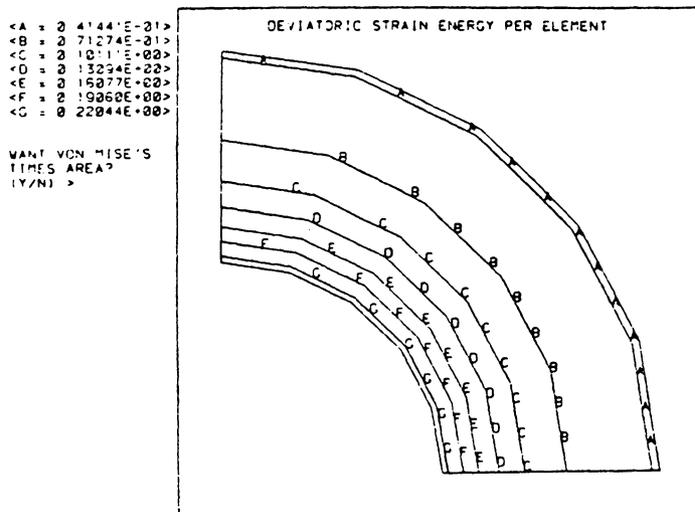


Figure 5-7b. Plot Two of Method Two

WANT NODE Nos. ?
(Y,N) >

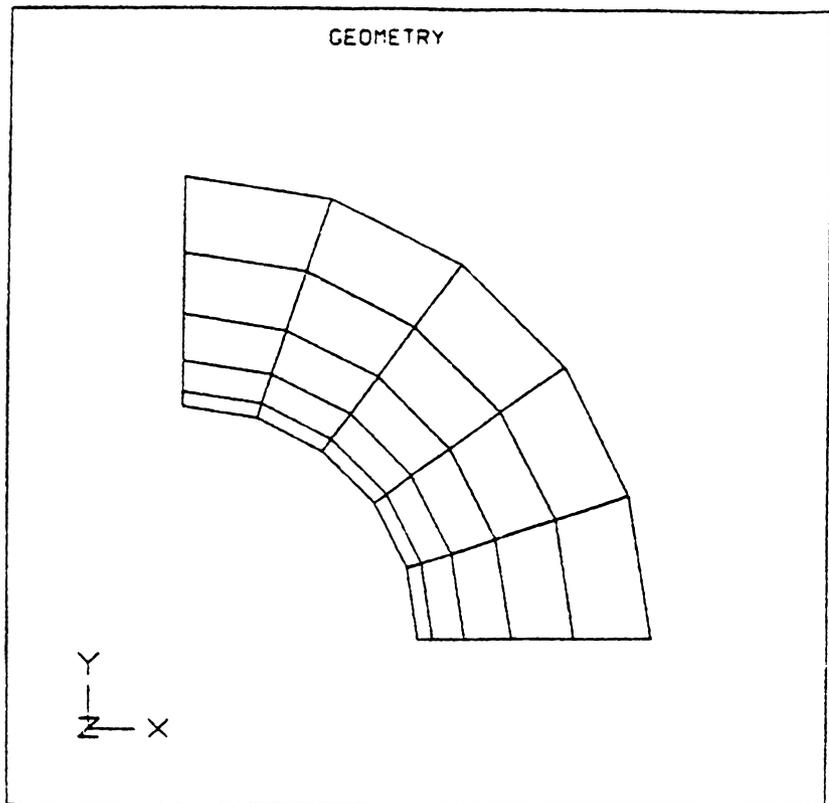


Figure 5-8. Method Two Mesh.

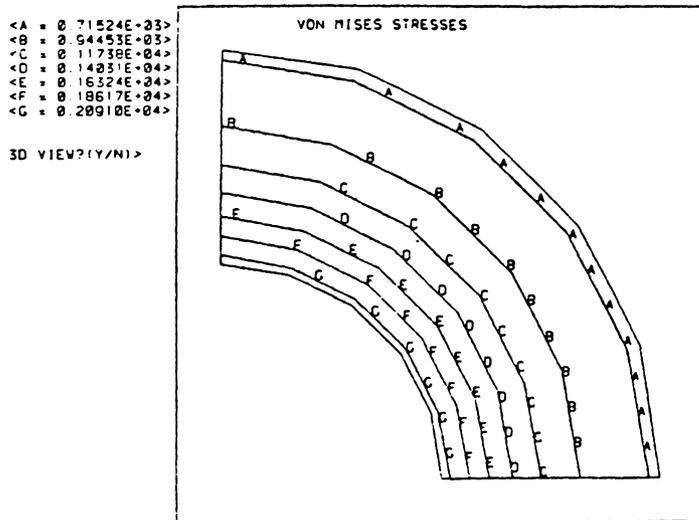


Figure 5-9a. Plot One of Method Three

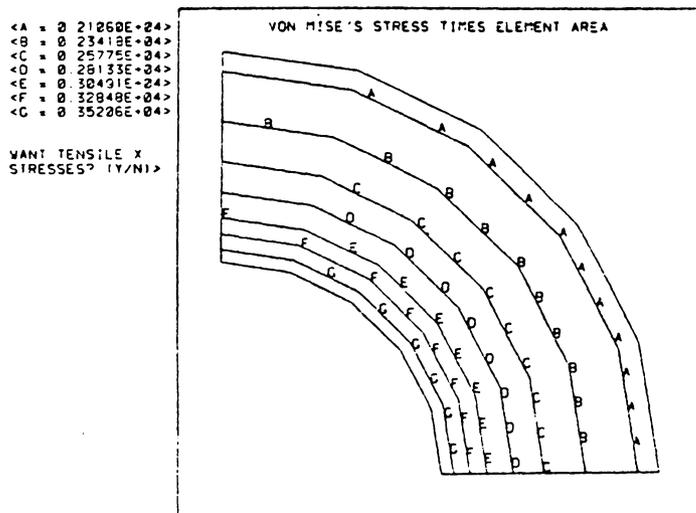


Figure 5-9b. Plot Two of Method Three

WANT NODE Nos. ?
(Y,N) >

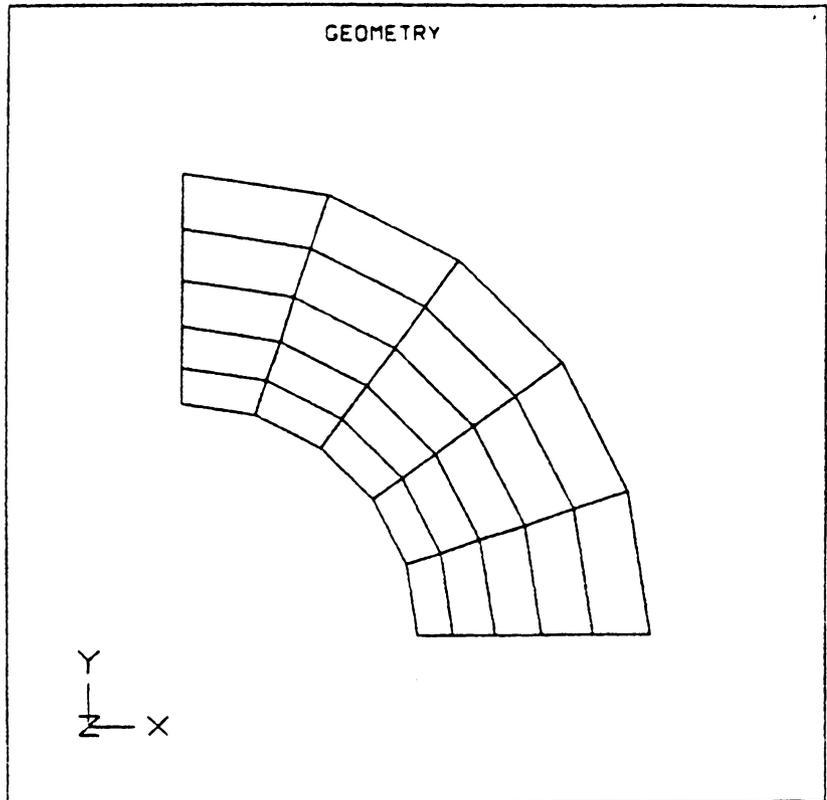


Figure 5-10. Method Three Mesh.

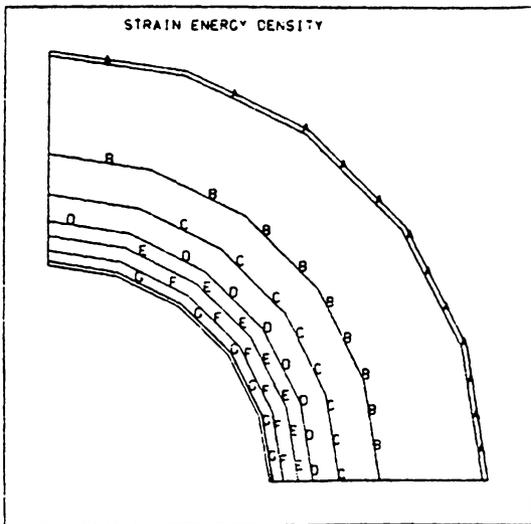
not as pronounced as that for the two other meshes, because the maximum to minimum ratio in plot two of method three was not as large as for the second plot of the other two methods.

Summarizing the results of the coarse mesh, all six energy functions that were obtained for the coarse mesh indicated that the mesh had to be refined more around the inner radius. Since it was decided beforehand to keep the number of elements and nodes constant throughout this test case, the only way to accomplish this was to enlarge the outer elements, while reducing the inner ones. This was carried out by increasing the size of the elements linearly from the inner radius to the outer radius. At the same time, the large to small element ratio in the new mesh was made equal to the ratio of the high to low contour value of the function of interest. This strategy gave rise to the meshes presented in Figs. 5-6, 5-8, and 5-10, which were based on methods one, two and three, respectively. Each one of these meshes was then loaded in the same manner as the coarse mesh, and solved with program MESS. Recall that in the new meshes only the plots of the method that generated that particular mesh will be analyzed.

Figure 5-11a presents the first plot of method one obtained with the new mesh of Fig. 5-6. The corresponding second plot is shown in Fig. 5-11b. Each one of these plots is accompanied by a reduced frame of the corresponding plot in the coarse mesh. This was done to facilitate comparisons between the two meshes and the plots they produced. The first plot of method one for the new mesh has the inner contours more closely spaced than the plot of the coarse mesh. In the coarse mesh, the maximum to minimum contour ratio was 8.5, in the new

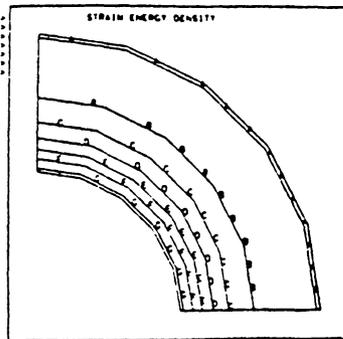
<A = 0.15693E-01>
 <B = 0.37512E-01>
 <C = 0.50343E-01>
 <D = 0.81168E-01>
 <E = 0.10299E+00>
 <F = 0.12482E+00>
 <G = 0.14664E+00>

WANT STRAIN ENGY.
 PER ELEMENT?
 (Y/N) >



<A = 0.15000E-01>
 <B = 0.30000E-01>
 <C = 0.45000E-01>
 <D = 0.60000E-01>
 <E = 0.75000E-01>
 <F = 0.90000E-01>
 <G = 0.10000E+00>

WANT STRAIN ENGY
 PER ELEMENT?
 (Y/N) >

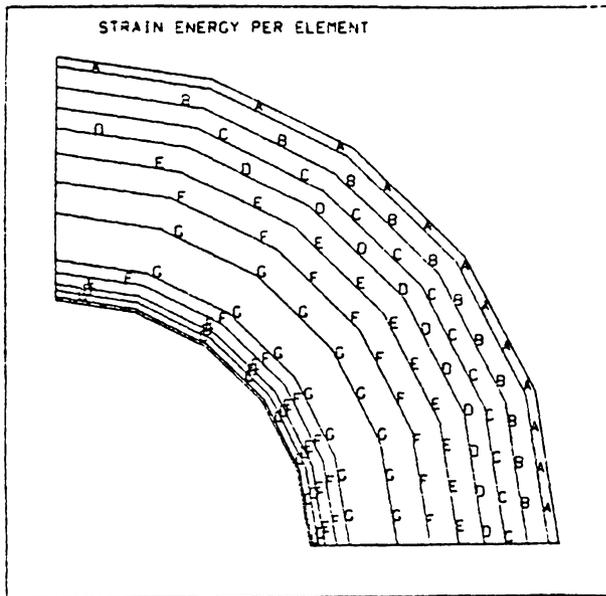


Coarse Mesh Result

Figure 5-11a. Plot One of Method One for New Mesh.

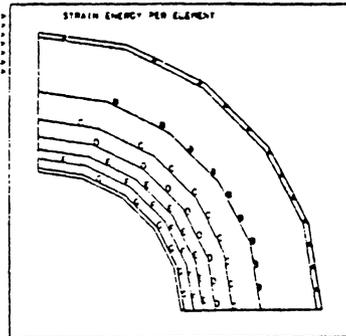
<A = 0.73615E-01>
 <B = 0.80959E-01>
 <C = 0.88300E-01>
 <D = 0.95542E-01>
 <E = 0.10298E+00>
 <F = 0.11033E+00>
 <G = 0.11767E+00>

WANT DEVIATORIC
 STRAIN DENSITY?
 (Y/N) >



<A = 0.46403E-01>
 <B = 0.76364E-01>
 <C = 0.10602E+00>
 <D = 0.13710E+00>
 <E = 0.16743E+00>
 <F = 0.19710E+00>
 <G = 0.22707E+00>

WANT DEVIATORIC
 STRAIN DENSITY?
 (Y/N) >



Coarse Mesh Result

Figure 5-11b. Plot Two of Method Two for New Mesh.

mesh it is 9.3. The near-optimum mesh should have a maximum to minimum element size ratio equal to the maximum to minimum contour ratio. This implies that the new mesh should be refined further. The next refined mesh should have a outside to inside element ratio equal to 9.3. However, the maximum contour value in the second plot of method one, Fig. 5-11b, is no longer at the inner boundary; it is inside the cylinder wall. Thus, Fig.5-11b implies that the size of the elements underneath the G contour should be decreased, to obtain a more nearly constant value for the second plot of method one. The discrepancy between the two plots of method one for the new mesh was caused by the linear distribution of element sizes. Obviously, the next iteration on the mesh should be done with a nonlinear mesh.

The new mesh, presented in Fig. 5-6, can be called a near-optimum mesh because the values in the second plot of method one improved during this iteration. The coarse mesh had a maximum to minimum ratio equal to 4.9, whereas, for the new mesh it was 1.6, much closer to one. The new mesh ratio is not quite unity, which it would be for an optimum mesh, but the improvement in this plot was almost a factor of 3 in just one iteration. Therefore, the stresses obtained with the new mesh should be more accurate than the stresses obtained with the coarse mesh. Figure 5-12 shows the X stresses obtained with this mesh. The maximum hoop stress, according to Fig. 5-12 is 1,700 psi (11.7 MPa), which is slightly less than the 1,747 psi (12.0 MPa), hoop stress obtained with the coarse mesh. This should not cause any alarm, since the analytical solution was 1,667 psi (11.5 MPa). On the other hand, the maximum radial stress, which can be obtained from the horizontal line of

```
<A = -0.83716E+03>  
<B = -0.42697E+03>  
<C = -0.16773E+02>  
<D = 0.39342E+03>  
<E = 0.80361E+03>  
<F = 0.12138E+04>  
<C = 0.16240E+04>
```

```
3D VIEW?(Y/N)>
```

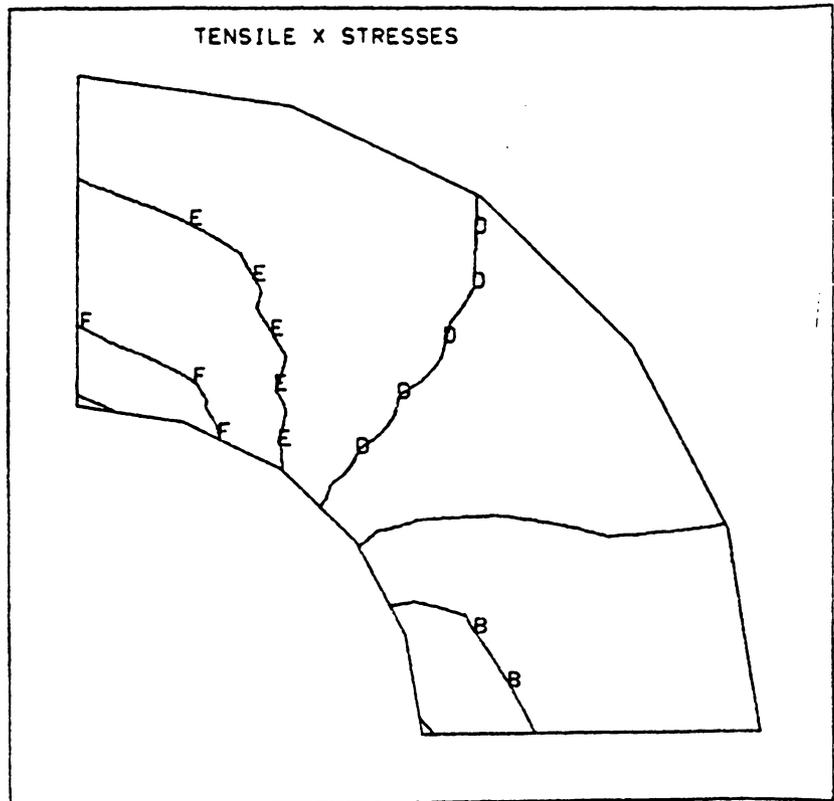


Figure 5-12. Plot of the Tensile X Stresses

symmetry of Fig. 5-12, increased with the refined mesh to a compressive 879 psi (6.06 MPa). It had been 680 psi (4.68 MPa) with the coarse mesh, and the analytical solution gave 1,000 psi (6.89 MPa). Hence, the refined mesh gave better estimates for both the maximum radial and tangential stresses. This concludes the first iteration for the mesh of method one.

The refined mesh based on the second method was presented in Fig. 5-8. As was pointed out then, it was very similar to the mesh based on the first method, and so were the results. The tangential stress at the inner radius was 1,700 psi (11.7 MPa), and the radial stress at the inner radius was compressive 886 psi (6.11 MPa), Fig. 5-13. As with the method one mesh results, the maximum hoop and maximum radial stresses were closer to the analytical solution, than the coarse mesh.

The other results of interest for the method two mesh are presented in Figs. 5-14a and b. These two figures show the first and second plots of method two for the new mesh. The new mesh resembles the first plot of method two, shown in Fig. 5-14a. Also, the coarse mesh had a maximum to minimum ratio of 5.3 for the second plot of method two, and the refined mesh of has a ratio of 1.7, much closer to one. Hence, both, the criteria of Table 5-2, and the actual stress results presented above indicate that the refined mesh based on the method two is a better mesh than the original coarse mesh.

The third new mesh was based on method three, and was presented in Fig. 5-10. The stresses from this mesh were used to produce the first and second plots of method three, shown in Figs. 5-15a and b. The first plot of method three resembles the new mesh. Furthermore, the maximum

<A = -0.84403E+03>
<B = -0.43300E+03>
<C = -0.21964E+02>
<D = 0.38907E+03>
<E = 0.80010E+03>
<F = 0.12111E+04>
<G = 0.16222E+04>

3D VIEW?(Y/N)>

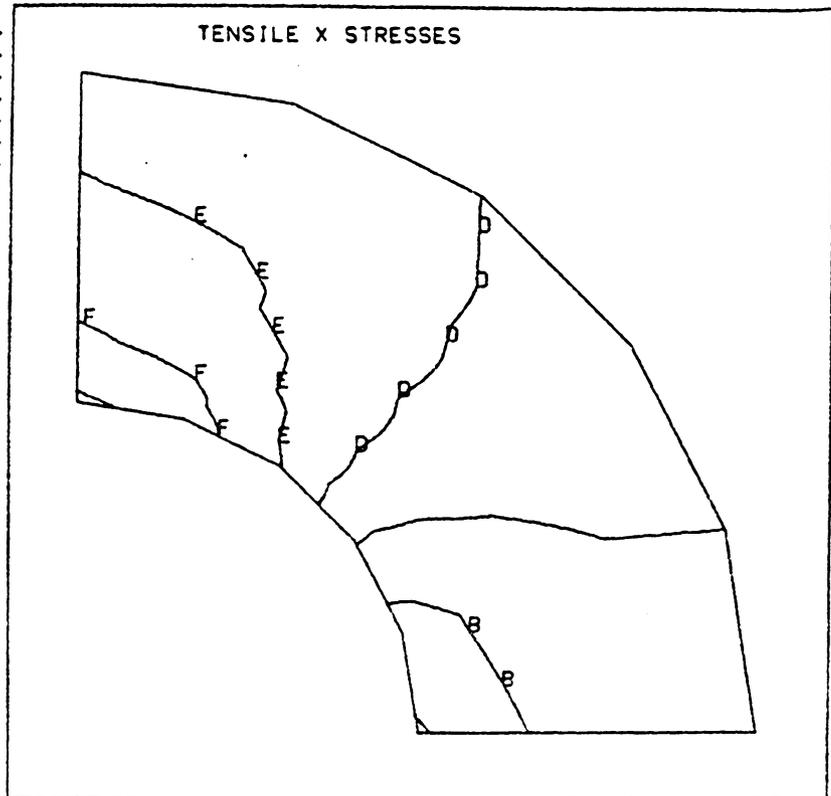
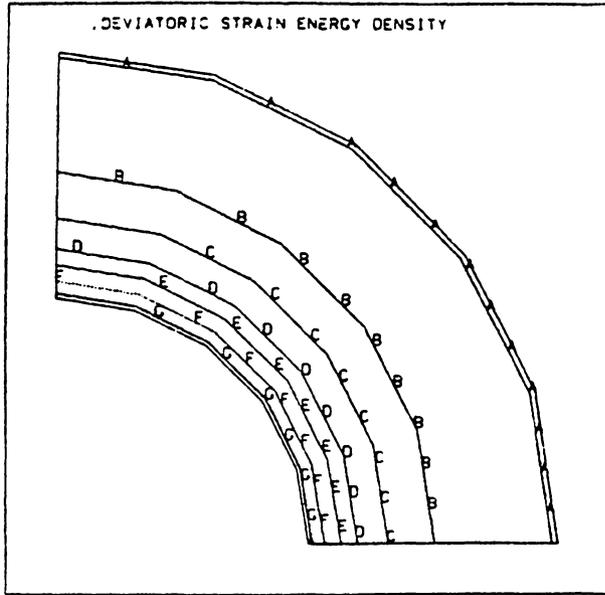


Figure 5-13. Plot of the Tensile X Stresses.

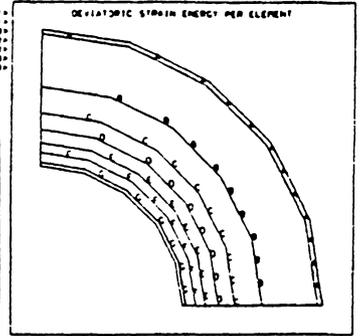
<A = 0.14350E-01>
 <B = 0.36007E-01>
 <C = 0.57664E-01>
 <D = 0.79321E-01>
 <E = 0.10098E+00>
 <F = 0.12264E+00>
 <G = 0.14429E+00>

WANT DEVIATORIC
 PER ELEMENT?
 (Y/N) >



*A = 0.0100E-01
 *B = 0.71274E-01
 *C = 0.10111E+00
 *D = 0.18274E+00
 *E = 0.16871E+00
 *F = 0.10040E+00
 *G = 0.22041E+00

WANT VON MISE'S
 TIMES AREA?
 (Y/N) >

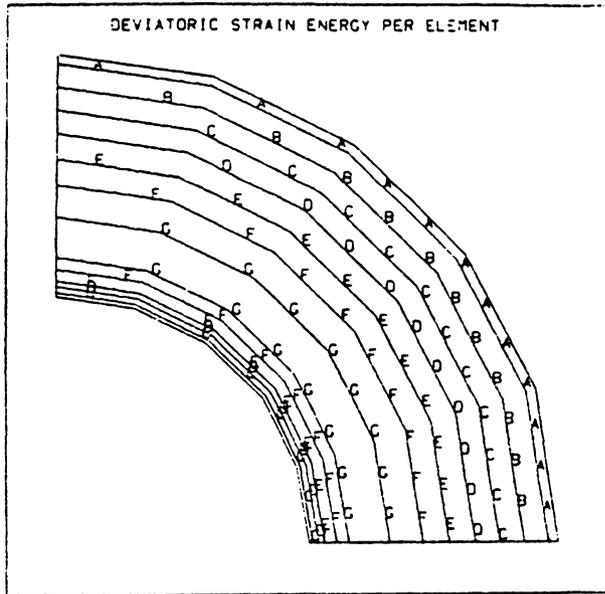


Coarse Mesh Result

Figure 5-14a. Plot One of Method Two for New Mesh.

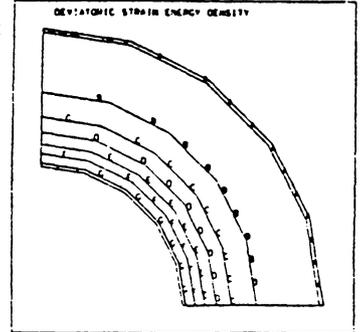
<A = 0.68109E-01>
 <B = 0.75893E-01>
 <C = 0.83677E-01>
 <D = 0.91461E-01>
 <E = 0.99245E-01>
 <F = 0.10703E+00>
 <G = 0.11481E+00>

WANT VON MISE'S
 TIMES AREA?
 (Y/N) >



*A = 0.14110E-01
 *B = 0.33381E-01
 *C = 0.52652E-01
 *D = 0.71923E-01
 *E = 0.91174E-01
 *F = 0.11044E+00
 *G = 0.12974E+00

WANT TENSILE 2
 STRESSES? (Y/N) >



Coarse Mesh Result

Figure 5-14b. Plot Two of Method Two for New Mesh.

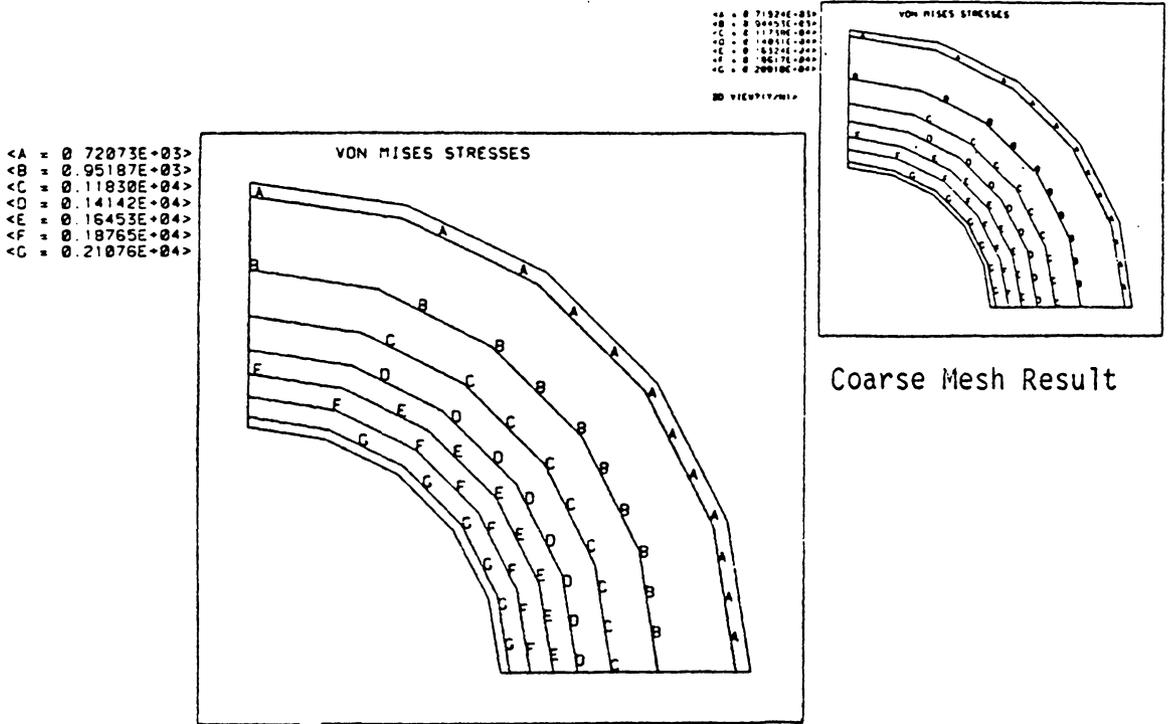


Figure 5-15a. Plot One of Method Three for New Mesh.

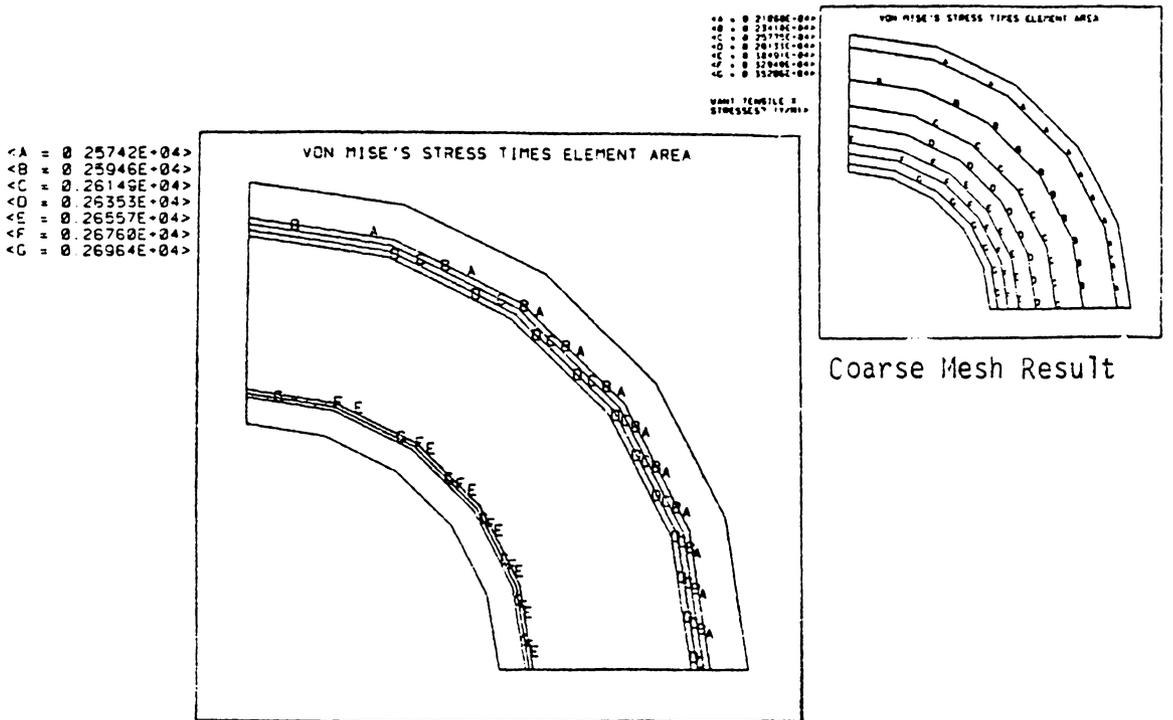


Figure 5-15b. Plot Two of Method Three for New Mesh.

to minimum ratio in the second plot decreased from 1.7 to 1.05, after only one iteration with a linear mesh. Thus, the third method produced an almost optimum mesh quicker than the other two methods considered in Table 5-2. This may indicate that this method is not as sensitive to mesh size as methods one or two. This will be investigated further in test case two.

If the method three mesh is truly an almost optimum mesh, then the stresses obtained with this mesh should approximately equal the ones obtained analytically. Figure 5-16 presents the tensile X stresses for this mesh. The maximum hoop stress for the third new mesh can be obtained from the inner radius at the vertical line of symmetry. The maximum hoop stress is 1,737 psi (12.0 MPa). The maximum radial stress can be obtained from the horizontal line of symmetry. It is a compressive 746 psi (5.13 MPa). The maximum hoop stress for this mesh is slightly lower than the hoop stress obtained from the coarse mesh, like the two previous new meshes analyzed. But, the stresses obtained with this mesh were more inaccurate than the ones obtained with the previous two meshes. Therefore, the third method converged on a near-optimum mesh faster than the other two methods considered here. However, the stresses were not nearly as accurate as the stresses of the other two near-optimum meshes.

Summarizing this test case, several things were observed during the development of the near-optimum grids, using the quadrilateral, isoparametric, plane-stress element. First, a linear mesh helped the method three, in Table 5-2, reach a near-optimum mesh faster, while at the same time it hindered the other methods presented in the same table.

<A = -0.71020E+03>
<B = -0.31609E+03>
<C = 0.78010E+02>
<D = 0.47211E+03>
<E = 0.86622E+03>
<F = 0.12603E+04>
<G = 0.16544E+04>

3D VIEW?(Y/N)>

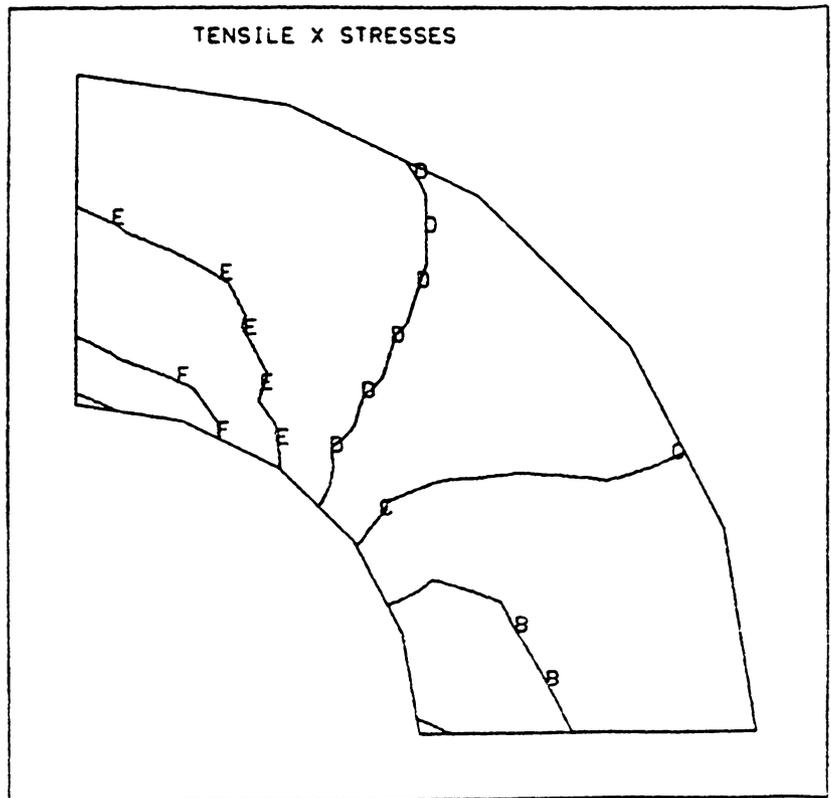


Figure 5-16. Plot of the Tensile X Stresses.

Second, the maximum stresses obtained with the method three near-optimum mesh were not as accurate as the maximum stresses obtained with the first and second method near-optimum meshes. Accuracy was defined as how the hoop and radial stresses compared to the analytical solution, computed with formulas of reference 13. Third, for this isoparametric finite element, methods one and two gave almost the same results; that is, the meshes obtained by applying the strain-energy function based on the normal stresses, and the deviatoric stresses were extremely similar.

CASE 2. Tapered platform subjected to a midspan line load.

The second case analyzed in this study was a tapered platform. A schematic of this platform is presented as Fig. 5-17. Here, it can be seen that the platform is simply supported on the outer edges, and a 10,000 lb/in (1.75 MN/m) load is applied along the midspan line. The tapered platform is made of steel with a Young's modulus of 30 million psi (207 MPa), and a Poisson's ratio of 0.3. Obviously, this problem is a symmetric problem, since both the geometry and the loading are symmetric. Because of the symmetry of this problem, only one half of the platform was modeled, as shown in Fig. 5-18.

The mesh presented in Fig. 5-18 was intentionally kept uniform, although it was determined by inspection of the problem that the highest stresses would occur at the line of symmetry. The highest stresses were expected at the line of symmetry because this is where the bending will be highest. Moreover, this is also where the mesh was constrained along an entire line of nodes. Nevertheless, the mesh was kept uniform to determine which of the methods from Table 5-2 would lead to a refined

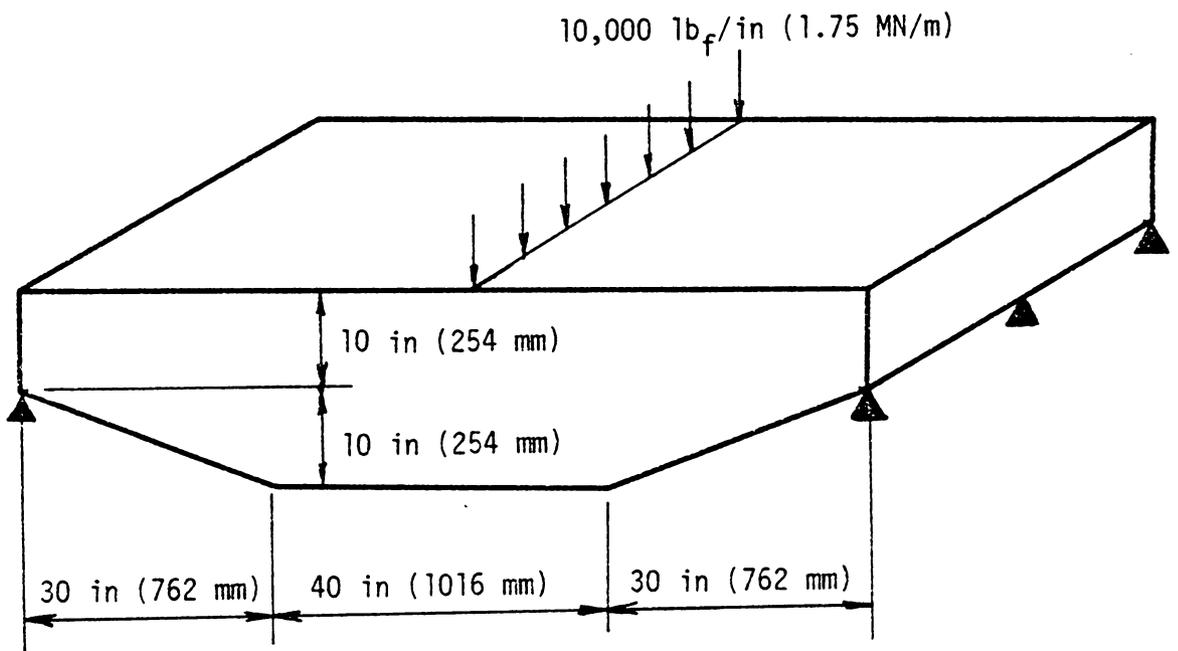


Figure 5-17. Schematic of the Platform Studied in Case 2.

WANT NODE Nos.?
(Y,N) >

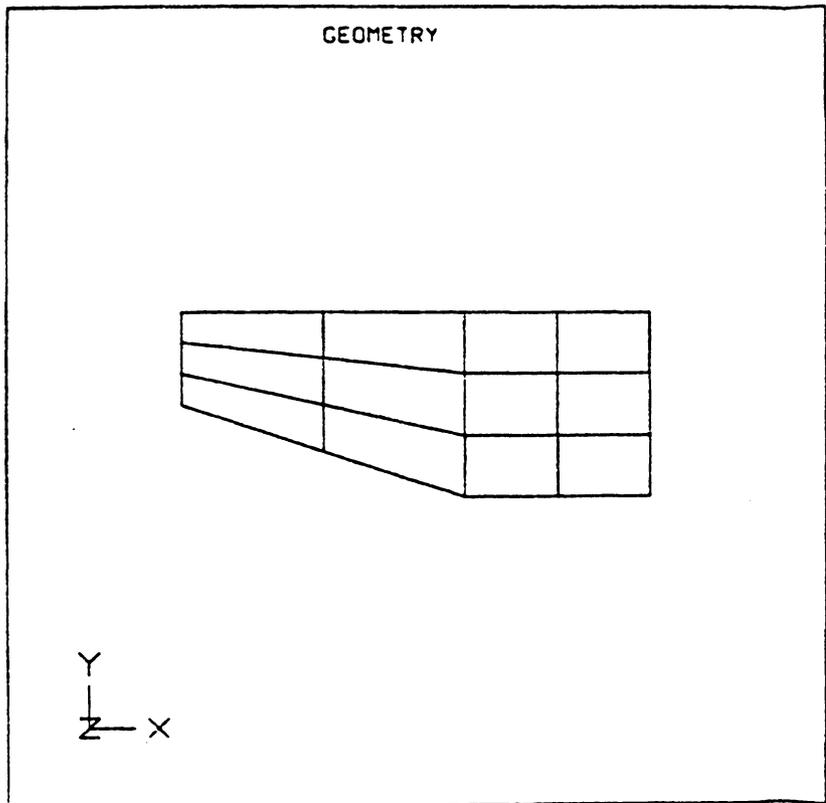


Figure 5-18. Mesh Used to Model the Platform.

mesh faster. Thus, the mesh in Fig. 5-18 consisted of twelve elements and sixteen nodes. As mentioned above, the nodes along the line of symmetry were constrained in the X direction; that is, they were only allowed to move in the Y direction. The only other node that was constrained was the one on the lower left-hand corner, which is the one that is simply supported; therefore, it was constrained in the X and Y directions.

The mesh described above was loaded and solved with program MESS. Figure 5-19a shows the first plot of method one. It shows that the contours come closer together at the right side of the model, especially toward the upper and lower right-hand corners. From the first plot of method one, it is obvious that the mesh should be refined in such a way that the new mesh has smaller elements around the right-hand corners. The plot presented in Fig. 5-19b is the second plot of method one. According to this plot, the size of the elements between the two B contours of Fig. 5-19b should be increased to raise the contour values there. Meanwhile, the size of the elements in the right-hand corners of the model should be decreased to reduce the high contour values there. This agrees with the conclusions obtained from the first plot of method one.

In the previous test case, the first mesh, or what was called the coarse mesh, gave a maximum to minimum ratio on the order of 5 for the second plot of method one. In this test case the corresponding ratio for the second plot of method one is 265. Clearly, this is approximately 50 times greater than for the previous test case. If this ratio was to be brought down to approximately one, then the mesh had to

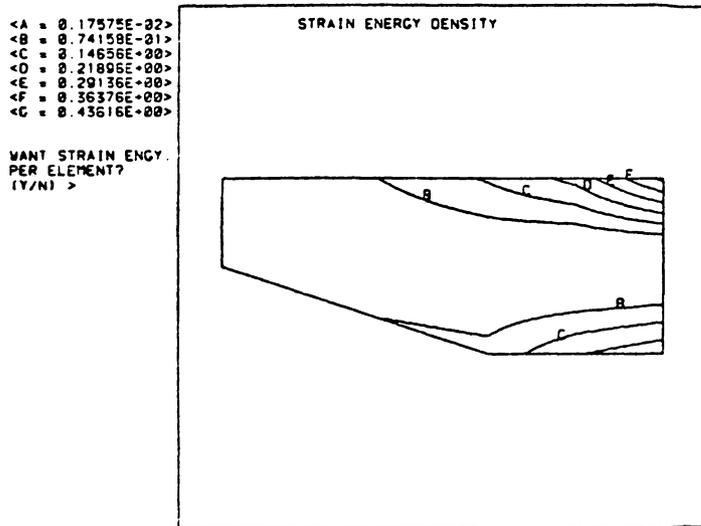


Figure 5-19a. Plot One of Method One for Coarse Mesh.

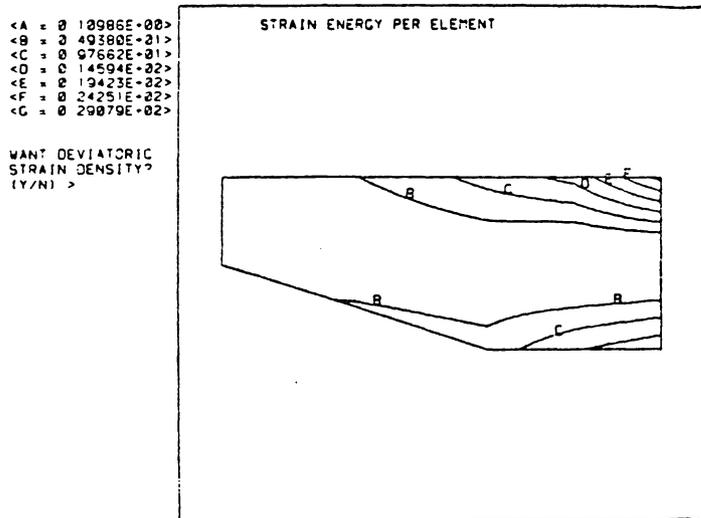


Figure 5-19b. Plot Two of Method One for Coarse Mesh.

be refined efficiently. In the first test case, the linear mesh restriction hindered the optimization process, because the near-optimum mesh was nonlinear. Nonetheless, the restriction was kept to determine how it would affect the final result. In this case, the new mesh was allowed to take any shape. Therefore, the contour lines of Fig 5-19a were used as element boundaries, inserting one element between each contour line.

Because of the nonlinearity of the mesh, an automatic mesh generator could not be used to specify the new meshes. Instead, a digitizer was used to locate the coordinates of the new nodes, which were placed on the contour lines of Fig. 5-19a. This made the process of mesh creation considerably faster. The new mesh, based on method one, is shown in Fig. 5-20. The element boundaries running from left to right resemble the original contour lines quite closely. The element lines running up and down were chosen to make individual elements as square as possible, without drastically increasing the number of elements. A square element will give better accuracy in the two by two Gauss integration used in program MESS.

The second mesh generated was the one based on the method two plots. The first plot of method two is presented in Fig. 5-21a, and the second is shown in Fig. 5-21b. Both of these contour plots indicate that the mesh should be refined more around the right-hand corners of the grid. The first plot shows the contour lines getting closer as they approach either of the two corners. Hence, according to the criteria of the first plot of method two, the elements in the mesh should get smaller as they approach either corner on the right side of the mesh.

WANT NODE Nos. ?
(Y,N) >

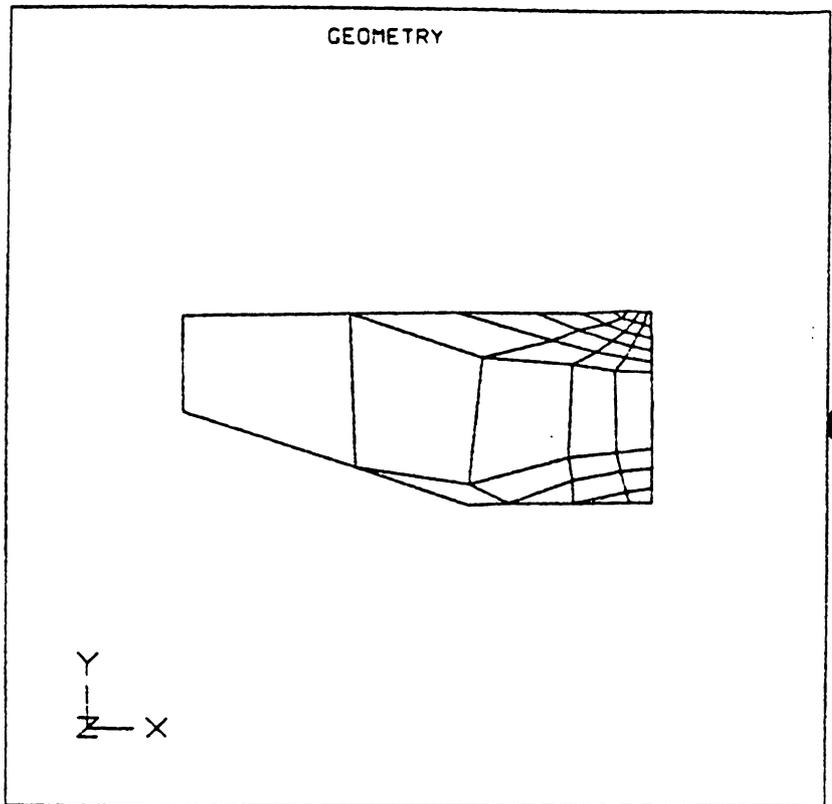


Figure 5-20. Method One Mesh.

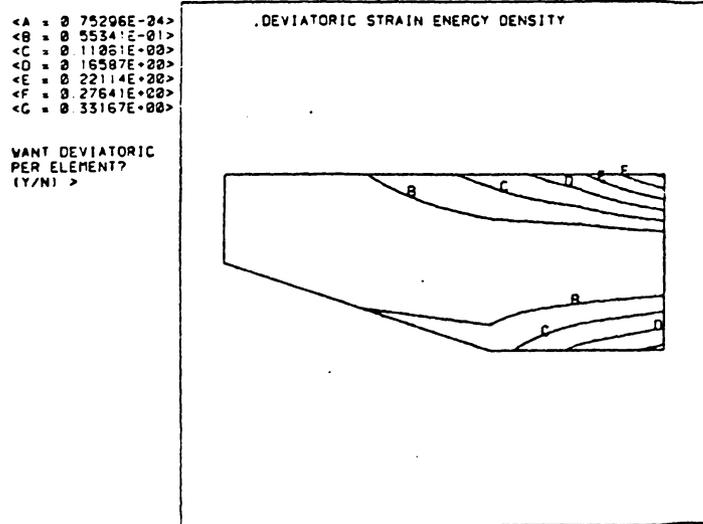


Figure 5-21a. Plot One of Method Two for Coarse Mesh.

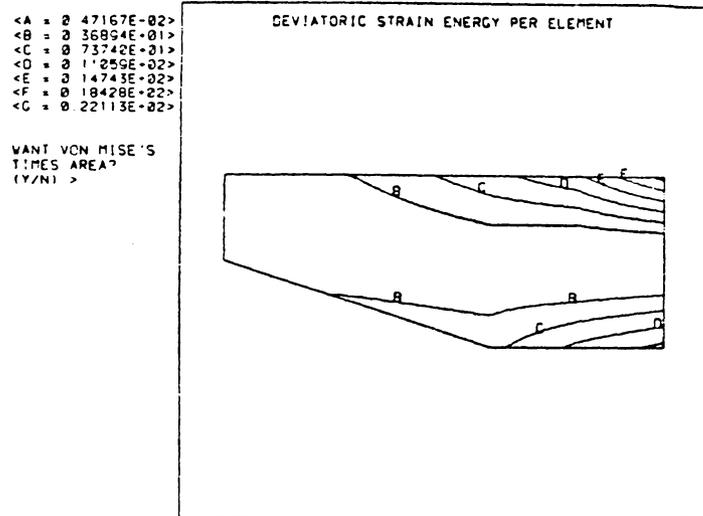


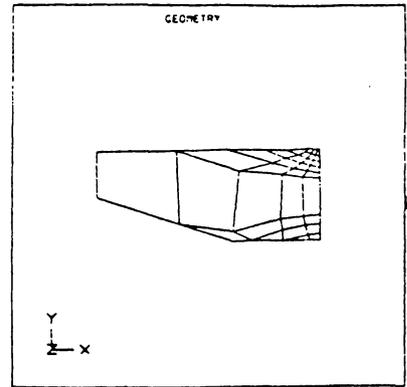
Figure 5-21b. Plot Two of Method Two for Coarse Mesh.

In the same manner the second plot shows that the elements between the two B contours should be enlarged, whereas the elements underneath the E, F, and G contour levels should be reduced. Hence, both plots of method two indicate that the refined mesh should have smaller elements around the right-hand corners.

As with the new mesh based on method one, the contour lines of the first plot of method two were used as element boundaries to generate the new mesh based on method two. This new mesh is shown in Fig. 5-22. It is similar to the mesh obtained with method one that was presented in Fig. 5-20 -- see the reduced frame inset in Fig. 5-22. However, the mesh of method two has less elements than the mesh of method one. The reason for this is that the first plot of method one, in Fig. 5-19a, had contour lines that showed a slightly steeper gradient as the contour approached the right side. This can be seen by overlaying the first plot of method one over the first plot of method two.

The first plot of method three obtained with the mesh of Fig. 5-18 is presented in Fig. 5-23a. This contour could have been placed on a digitizer to generate nodal points along the contour lines shown. However, such a grid would have had a rather large, pointed element along the contour level B. To avoid this, the contours were not used to generate the element boundaries. Instead, the transition of the new grid was chosen to resemble the contour gradient of this plot. This mesh is shown in Fig. 5-24.

With the three new meshes generated, each one was solved to see which method arrived at a near-optimum mesh fastest. First, the mesh based on method one was solved. This mesh was presented in Fig 5-20.



Method one Mesh

WANT NODE Nos.?
(Y,N) >

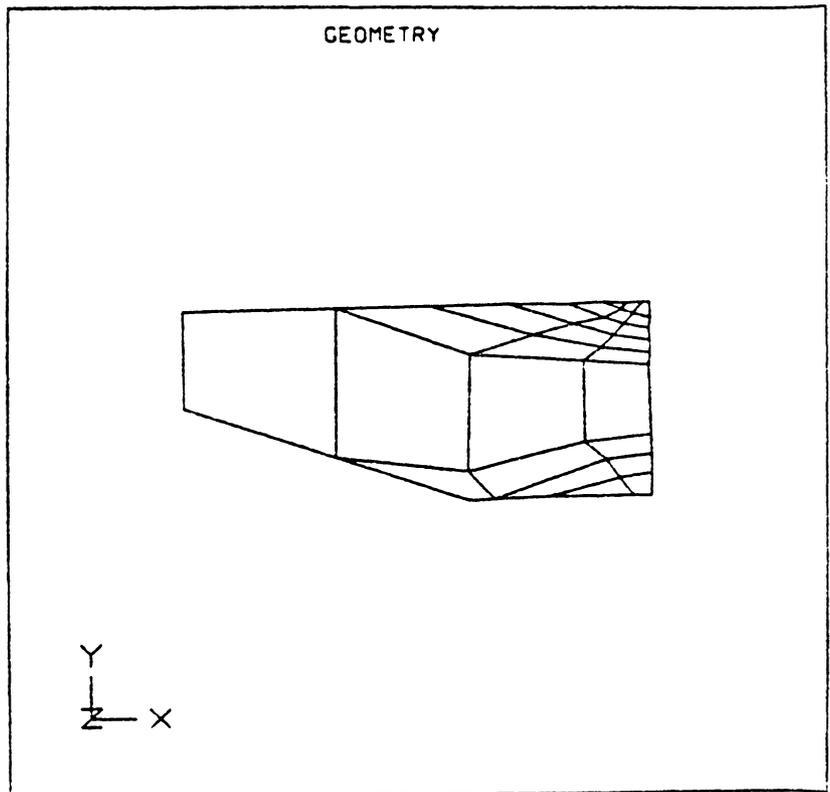


Figure 5-22. Method Two Mesh.

```

<A = 0.31819E+03>
<B = 0.83416E+03>
<C = 0.13581E+04>
<D = 0.18661E+04>
<E = 0.23821E+04>
<F = 0.28888E+04>
<G = 0.34148E+04>

```

```
3D VIEW? (Y/N) >
```

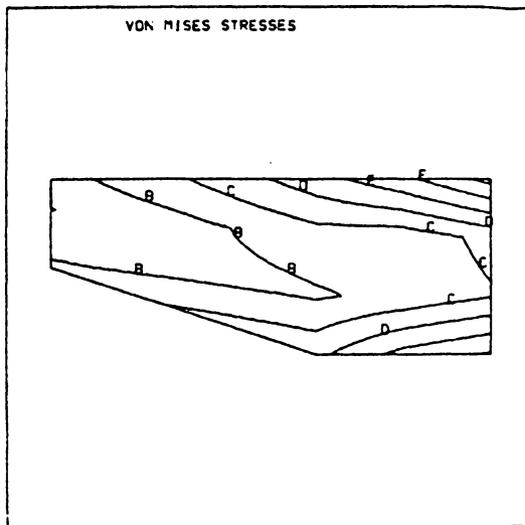


Figure 5-23a. Plot One of Method Three for Coarse Mesh.

```

<A = 0.19889E+05>
<B = 0.54588E+05>
<C = 0.89129E+05>
<D = 0.12375E+06>
<E = 0.15837E+06>
<F = 0.19299E+06>
<G = 0.22761E+06>

```

```
WANT TENSILE X
STRESSES? (Y/N) >
```

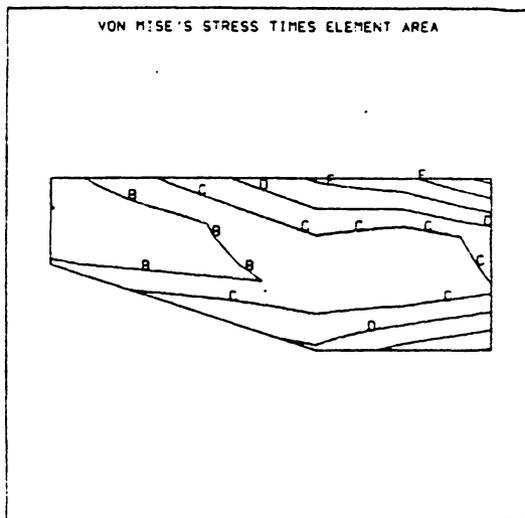


Figure 5-23b. Plot Two of Method Three for Coarse Mesh.

WANT NODE Nos. ?
(Y,N) >

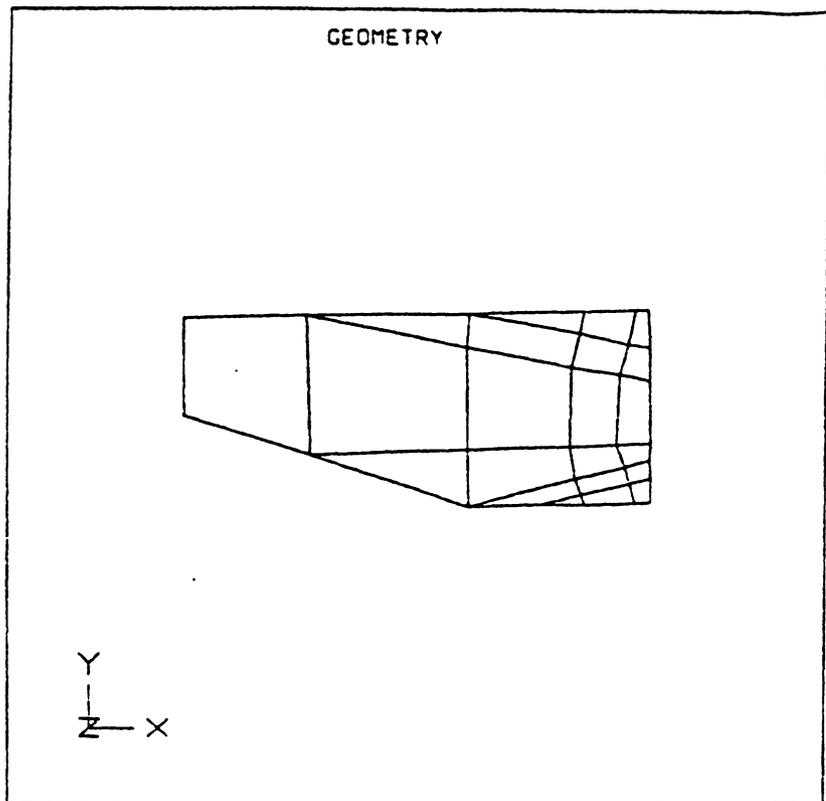


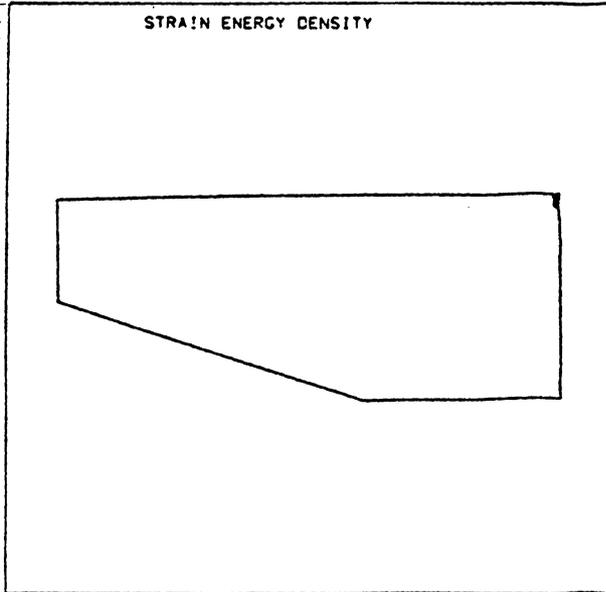
Figure 5-24. Method Three Mesh.

The first plot of method one obtained with this new mesh is presented in Fig 5-25a. It seems that there are no contours present, because the contours are all in the upper right-hand corner of the grid. The reason for this is that the point load applied at the upper right-hand corner introduced an infinite stress raiser, and biased the stress values, which in turn biased the energy levels. Thus, the contour lines gathered around the node at which the point load is applied. This was not seen in the plot of the coarse mesh, which is inset in Fig. 5-25a, because that mesh was coarse and the effect of the point load was localized, within a large element. Thus, as the grid used to model the problem is refined about the upper right-hand corner then the localized effect of the point load becomes more apparent. Therefore, it is important to start out with a coarse mesh for the first solution of the problem; this will ensure that if there any localized stress raisers they will not affect the different plots used for the future iterations. Furthermore, if a coarse mesh is used for the first attempt, then the overall computer and engineering time used to solve the problem is reduced, because coarse meshes have less nodes, and they are easier to generate with a mesh generator.

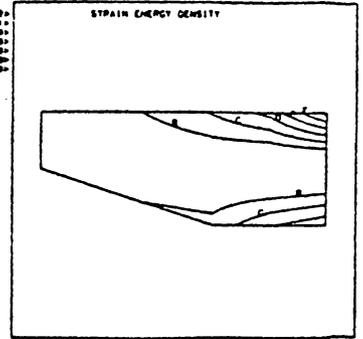
The second plot of method one for the new mesh in Fig. 5-20 is shown in Fig. 5-25b. In this contour, the effect of the localized stress introduced by the point load is also quite apparent. Up in the right-hand corner of the mesh the value of the contour levels seem to be decreasing as they approach the corner where the load is applied. It seems that when they reach the corner, the value of the function should be the value associated with the A level. In reality, when the function

<A = 0.16241E-02>
 <B = 0.13580E+01>
 <C = 0.27345E+01>
 <D = 0.41000E+01>
 <E = 0.54674E+01>
 <F = 0.68330E+01>
 <G = 0.82002E+01>

WANT STRAIN ENCY.
 PER ELEMENT?
 (Y/N) >



<A = 0.17075E-00>
 <B = 0.74150E-01>
 <C = 0.14650E-00>
 <D = 0.21850E-00>
 <E = 0.20150E-00>
 <F = 0.58375E-00>
 <G = 0.42475E-00>

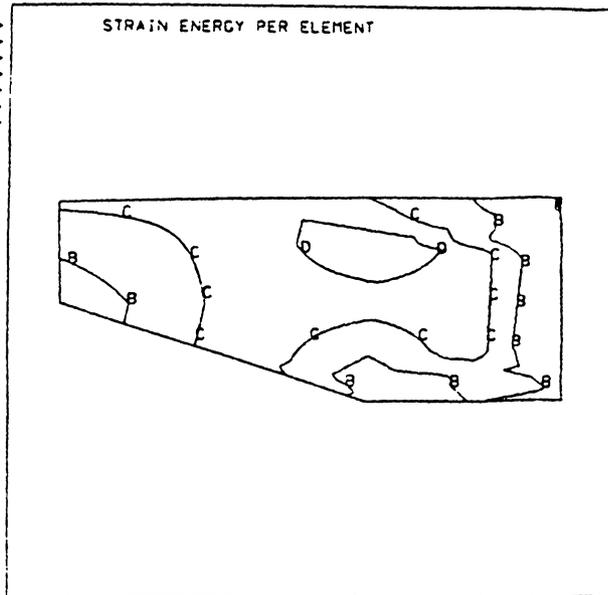


Coarse Mesh Result

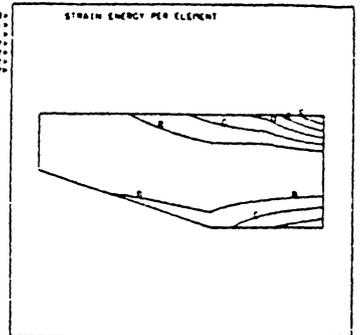
Figure 5-25a. Plot One of Method One for New Mesh.

<A = 0.25741E+00>
 <B = 0.14679E+01>
 <C = 0.25664E+01>
 <D = 0.35689E+01>
 <E = 0.52694E+01>
 <F = 0.62699E+01>
 <G = 0.74704E+01>

WANT DEVIATORIC
 STRAIN DENSITY?
 (Y/N) >



<A = 0.18200E-00>
 <B = 0.48100E-01>
 <C = 0.74675E-01>
 <D = 0.15750E-02>
 <E = 0.10215E-02>
 <F = 0.25275E-02>
 <G = 0.20070E-02>



Coarse Mesh Result

Figure 5-25b. Plot Two of Method One for New Mesh.

reaches the corner it will reach the G level, implying that a very small element size is required to model this region correctly. Hence, the results of both the first and second plots suggest that the size of the elements in the upper right-hand corner should be decreased to reach the optimum mesh.

From the material just presented, the method one mesh of Fig. 5-20 is a near-optimum mesh compared to the original coarse mesh. The maximum to minimum ratio in the second plot of method one for the coarse mesh was 265, whereas, the same ratio for the new mesh is approximately 28. Hence, this mesh is better than the coarse mesh. But, what do the stresses indicate about this mesh? As mentioned before, the point load in the upper right-hand corner will theoretically give an infinite stress; and as the elements around this corner are made smaller the stress will continue to increase without bound, until the computer can no longer handle the large numbers. Therefore, it is best to compare the stress at the lower right-hand corner of the platform, since it is also a high stress region, but finite. One way to see whether or not the accuracy of the mesh is increasing with each new mesh is to see if the stress at the lower corner is asymptotically reaching a certain value. The final stress state at the corner will have stress components in the X and Y directions, plus a shear stress. As can be seen in Figs. 5-26a, b, and c, the point load has also biased the Y stress, and shear stress contours. Therefore, it would be difficult to compare these stresses from mesh to mesh. Fortunately, the X tensile stress contour is not affected by the point load, since the point load acts in the Y direction. Hence, the value of the X stress at the lower right-hand

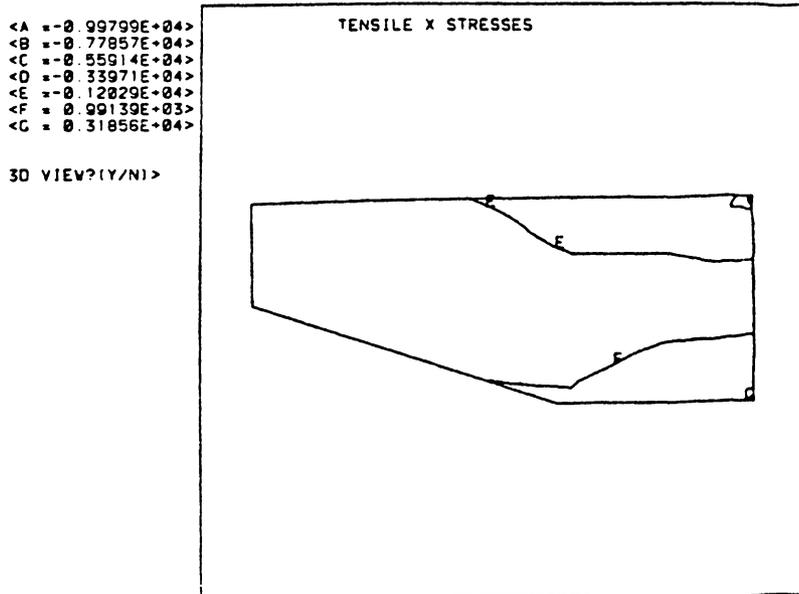


Figure 5-26a. Plot of the Tensile X Stresses

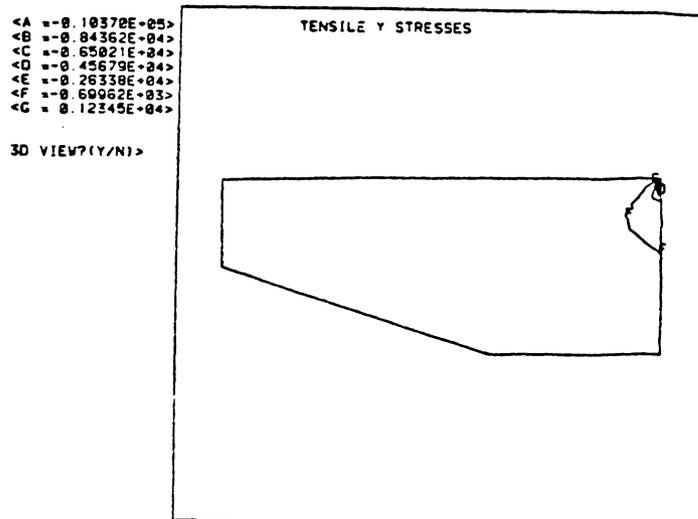


Figure 5-26b. Plot of the Tensile Y Stresses

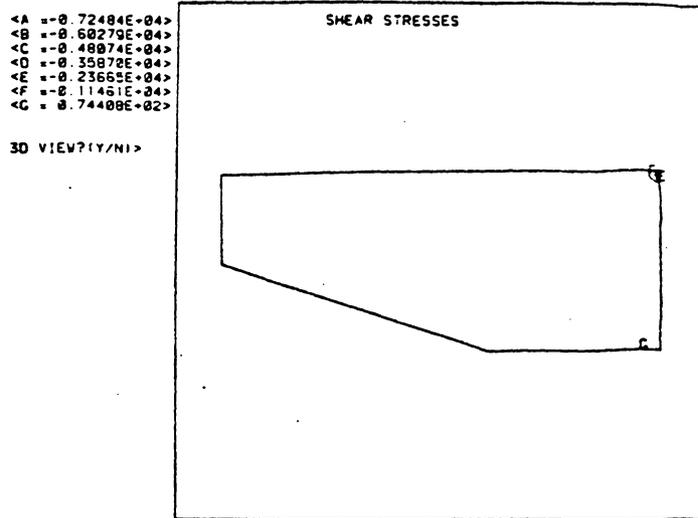


Figure 5-26c. Plot of the Shear Stresses

corner can be used to determine whether the accuracy of the overall stress state of the mesh is improving. In the original coarse mesh the value of the stress at this corner was approximately 2,840 psi (19.5 MPa), whereas, in the refined mesh it is almost 3,360 psi (23.1 MPa). Obviously, the stress level has increased in the refined mesh.

Although the results presented above indicate that the mesh in Fig. 5-20 is a near-optimum mesh compared to the original one, the value of the second plot of method one was not constant. The ratio of the maximum to minimum contour ratio in this plot was 28, which implies that further refinements can be made to the mesh in Fig. 5-20, to make it more "near-optimum"; that is, reduce the contour ratio further. Obviously, the first plot of method one is useless for further iterations, since the stress concentration region has biased the contours in Fig. 5-25a. Therefore, only the second plot was used to generate the next near optimum mesh. The second plot, Fig. 5-25b, shows that the size of the elements underneath the A, B and C contour levels should be increased, since this is where contour values are low. The size of the elements under the E, F, and G levels should be increased; but these levels occur at the stress concentration region, where the elements are already very small. Hence, for the next near-optimum mesh the number of elements was reduced, because this was the simplest way to fulfill the requirements presented above. The new near-optimum mesh is presented in Fig. 5-27. Thirty elements were used in this mesh, whereas thirty-six had been used for the previous method one mesh, in Fig. 5-20.

The new near-optimum mesh based on method one shows the same characteristics as the previous method one mesh. The results for this

WANT NODE Nos.?
(Y,N) >

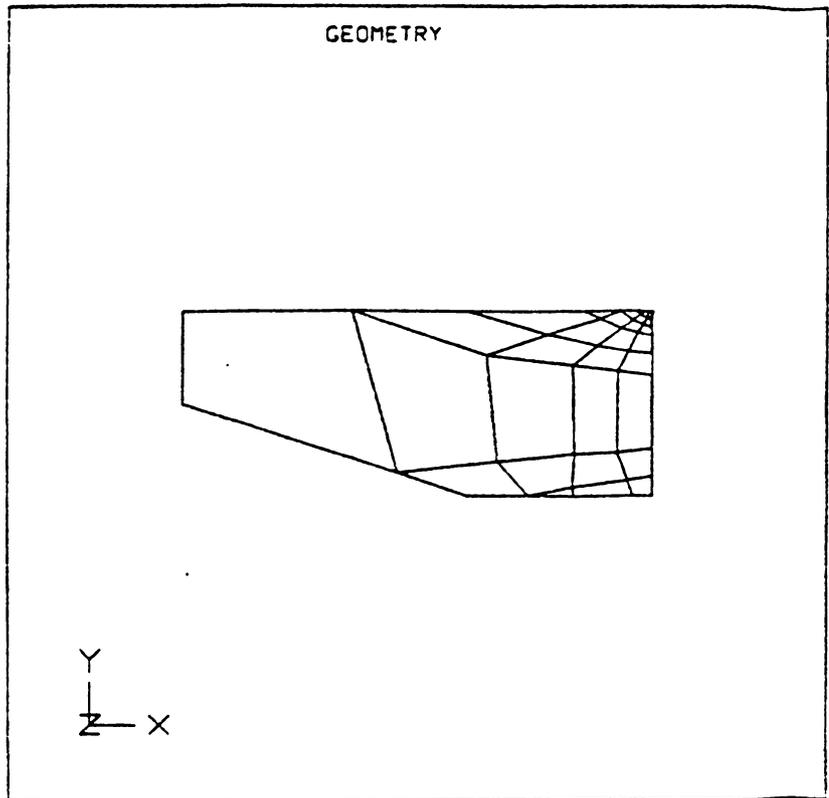
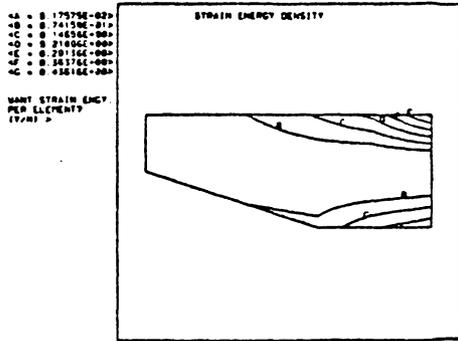


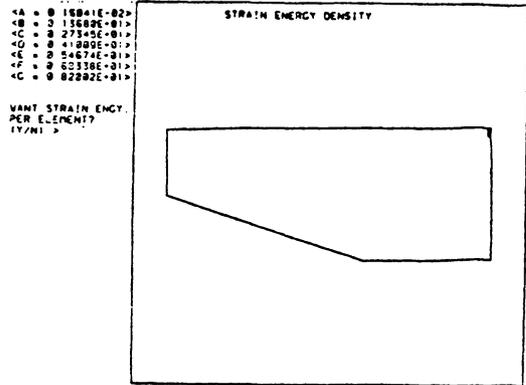
Figure 5-27. Second New Mesh based On Method One.

new mesh are presented in Figs. 5-28a, 5-28b, and 5-29, along with insets of the results of the coarse and previous near optimum meshes. The first plot of method one, shown in Fig. 5-28a, is again not very meaningful because of the stress concentration biasing. On the other hand, the ratio of the maximum to minimum values in the second plot, Fig. 5-28b, increased from 28 for the previous near-optimum mesh to 33 for this new mesh. According to the outline of method one the second new mesh is farther from being an optimum mesh than the previous one, although they are both better than the original coarse mesh, which had a 265 ratio. Finally, the lower right-hand corner tensile X stress obtained with this new mesh was 3,300 psi (22.7 MPa), whereas it was 3,200 psi (22.0 MPa) for the previous mesh, a mere 100 psi (0.69 MPa) improvement. Thus, the time and effort invested in the new near-optimum mesh was not justified by the meager improvement in the tensile X stress seen at the lower right-hand corner.

The method two mesh was presented in Fig. 5-22. As may be expected by now, the results from this mesh were very similar to the results obtained with the method one mesh. The first contour plot of method two, for the new mesh, is presented in Fig. 5-30a. This plot of the new mesh is meaningless since all the contour levels are gathered about the point load. On the other hand, the second plot of method two, Fig 5-30b, shows that the new mesh is better than the original coarse mesh. In the coarse mesh the maximum to minimum ratio in the second plot of method two was approximately 5,300. In the new mesh this ratio is 19, which is a sizeable improvement. This also shows that this plot is more sensitive to the mesh coarseness than the corresponding plot of method



Coarse Mesh Result



Previous Mesh Result

```

<A = 0.22230E+02>
<B = 0.31239E+01>
<C = 0.62456E+01>
<D = 0.93672E+01>
<E = 0.12489E+02>
<F = 0.15611E+02>
<G = 0.18732E+02>
  
```

WANT STRAIN ENGY.
PER ELEMENT?
(Y/N) >

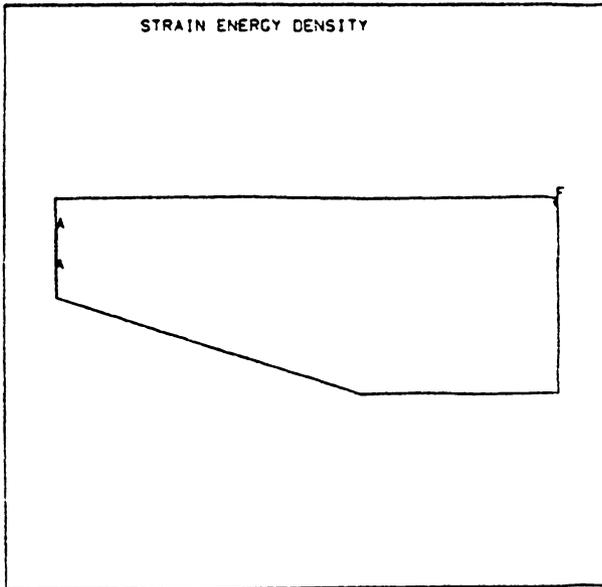
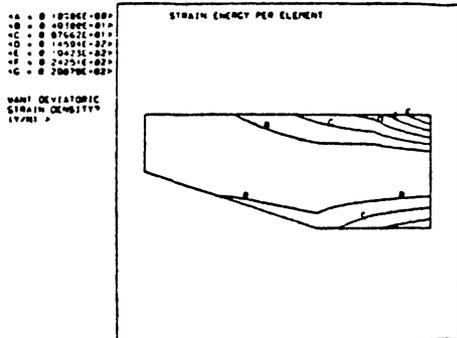
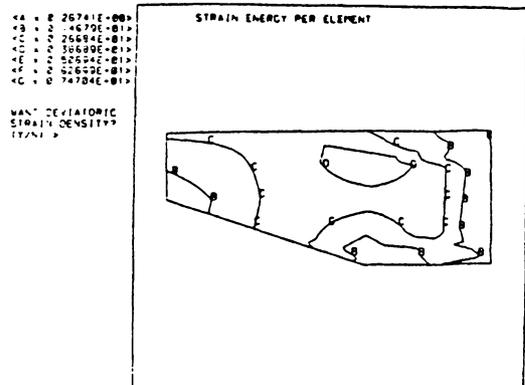


Figure 5-28a. Plot One of Method One for Second New Mesh.



Coarse Mesh Result



Previous Mesh Result

```

<A = 0.20554E+00>
<B = 0.12937E+01>
<C = 0.23618E+01>
<D = 0.34400E+01>
<E = 0.45181E+01>
<F = 0.55963E+01>
<G = 0.66744E+01>
  
```

WANT DEVIATORIC STRAIN DENSITY? (Y/N) >

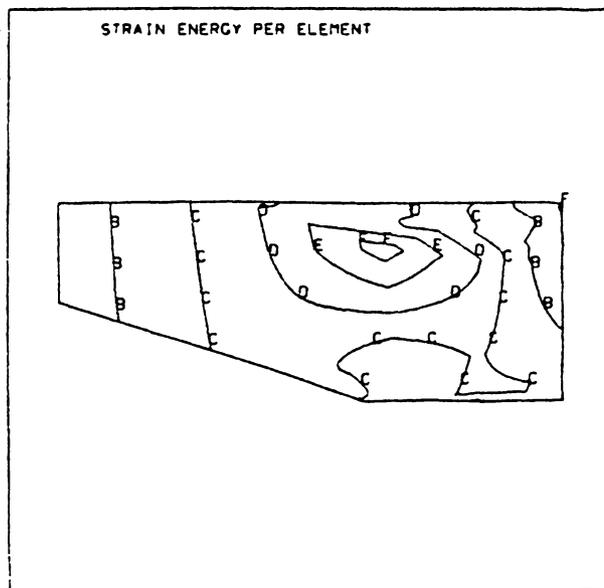
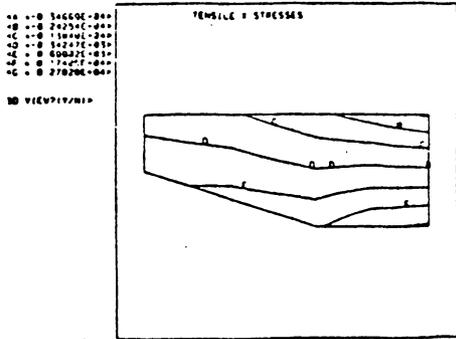
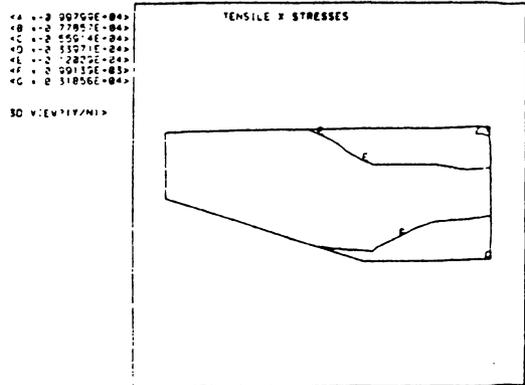


Figure 5-28b. Plot Two of Method One for Second New Mesh.



Coarse Mesh Result



Previous Mesh Result

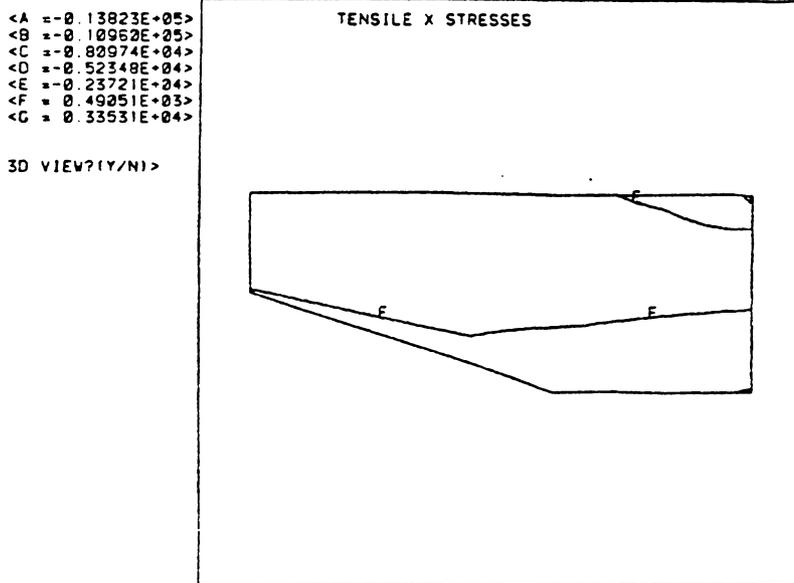


Figure 5-29. Plot of the Tensile X Stresses

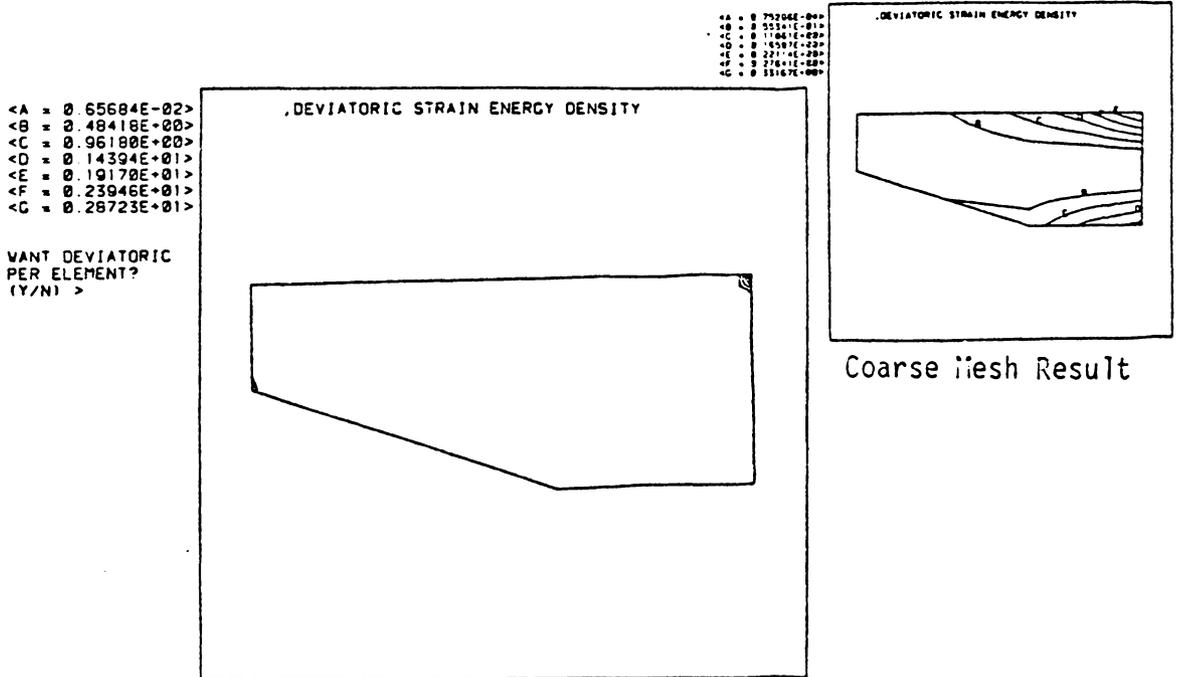


Figure 5-30a. Plot One of Method Two for New Mesh.

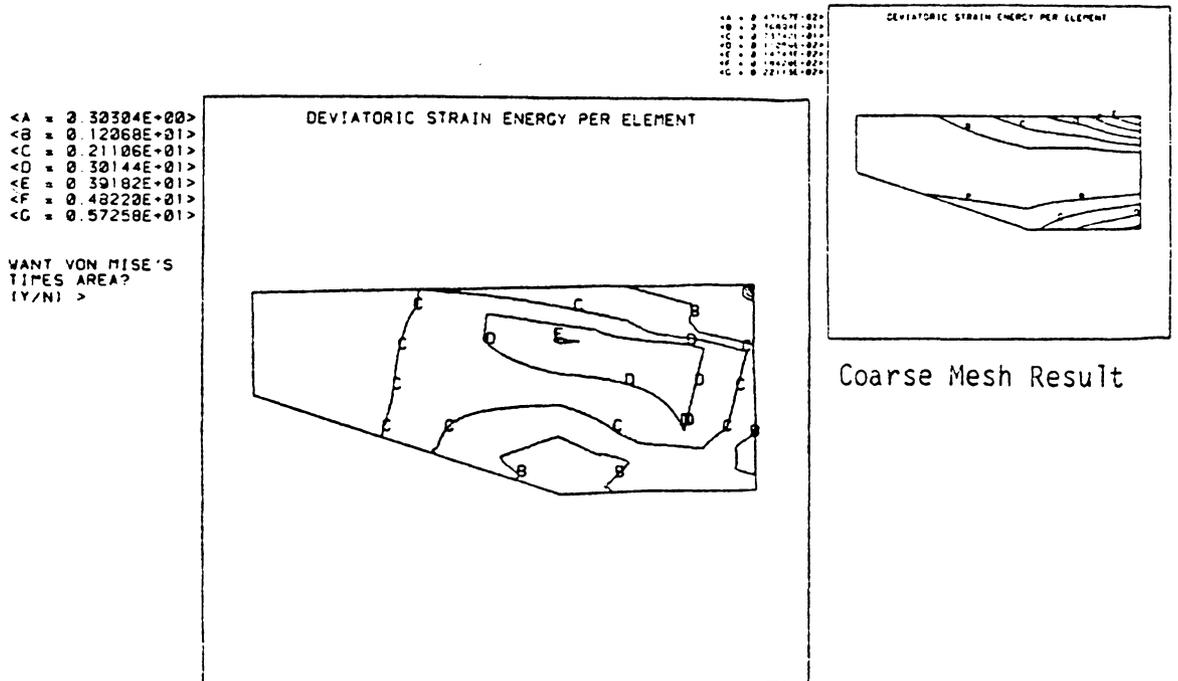


Figure 5-30b. Plot Two of Method Two for New Mesh.

one. The second plot of method one had a 265 ratio for the coarse mesh, while the second plot of method two had a 5,300 ratio for the same mesh. Finally, the tensile X stresses obtained with this new mesh are presented in Figs. 5-31. In this figure, the maximum tensile stress occurs at the lower right-hand corner, the corner of interest, and has a value of approximately 3,460 psi (23.8 MPa). This is almost the same result obtained with the first near-optimum mesh of method one.

The grid based on method three was presented in Fig. 5-24. The results from this mesh are shown in Figs. 5-32a, 5-32b, and 5-33. Figure 5-32a presents the first plot of method three. It is apparent that the grid was not refined enough in the first iteration, because the point load at the upper right-hand corner has not biased the contours yet. Therefore, both method one and method two refined the mesh faster in the first iteration. However, one may think that this is not a fair comparison because those two meshes were generated by strictly following the specified contours. As explained above, this was not done for method three mesh, for fear of creating too sharp an angle on some quadrilateral elements. This reasoning may give one the inclination to go back to the plot in Fig. 5-23a and generate a first iteration mesh by strictly following the contours of the first plot of method three. However, this would not affect the conclusion presented here. Plot one of method three for the coarse mesh does not call for smaller elements around the right-hand corner than either method one or method two. Moreover, the optimum mesh will have extremely small elements around those corners, because of the stress concentration there. Therefore, method three will lead to a refined mesh. However, it will not reach

```
<A =-0.76488E+04>  
<B =-0.58280E+04>  
<C =-0.40071E+04>  
<D =-0.21862E+04>  
<E =-0.36532E+03>  
<F = 0.14556E+04>  
<G = 0.32764E+04>
```

```
3D VIEW?(Y/N)>
```

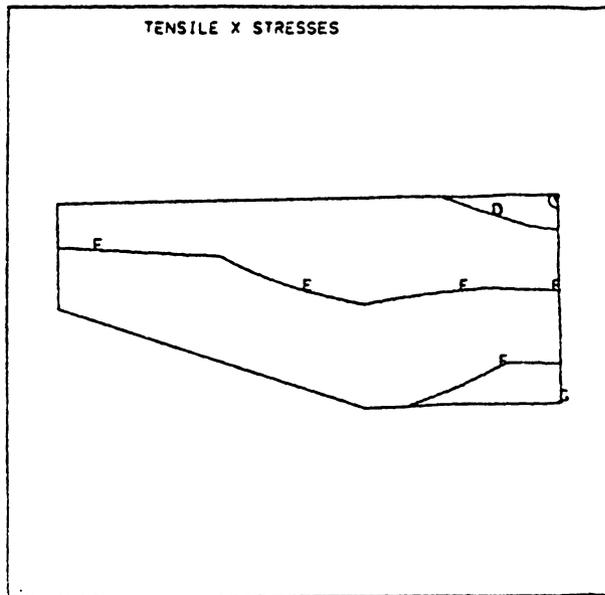
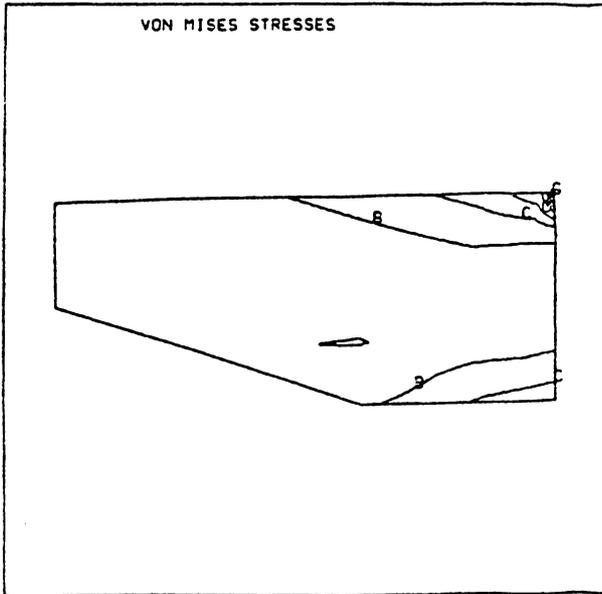


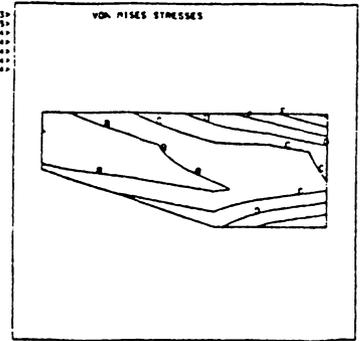
Figure 5-31. Plot of the Tensile X Stresses.

<A = 0.60480E+03>
 <B = 0.15250E+04>
 <C = 0.24452E+04>
 <D = 0.33654E+04>
 <E = 0.42856E+04>
 <F = 0.52059E+04>
 <G = 0.61261E+04>

3D VIEW?(Y/N)>



<A = 0.31010E+03>
 <B = 0.03416E+03>
 <C = 0.13501E+04>
 <D = 0.19661E+04>
 <E = 0.23821E+04>
 <F = 0.28006E+04>
 <G = 0.34140E+04>

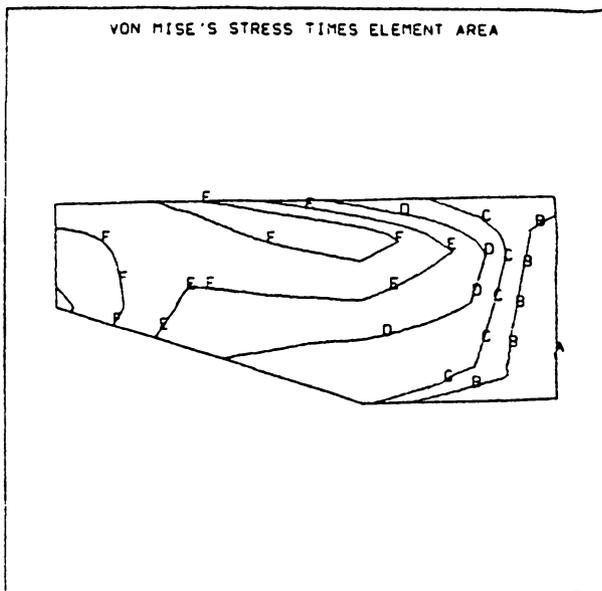


Coarse Mesh Result

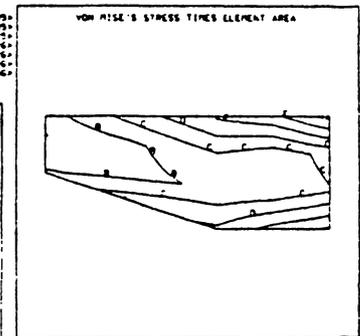
Figure 5-32a. Plot One of Method Three for New Mesh.

<A = 0.99464E+04>
 <B = 0.29566E+05>
 <C = 0.49186E+05>
 <D = 0.68806E+05>
 <E = 0.88426E+05>
 <F = 0.10805E+06>
 <G = 0.12767E+06>

WANT TENSILE X STRESSES? (Y/N)>



<A = 0.0000E+00>
 <B = 0.5058E+05>
 <C = 0.82721E+05>
 <D = 0.12171E+06>
 <E = 0.15857E+06>
 <F = 0.19209E+06>
 <G = 0.22767E+06>



Coarse Mesh Result

Figure 5-32b. Plot Two of Method Three for New Mesh.

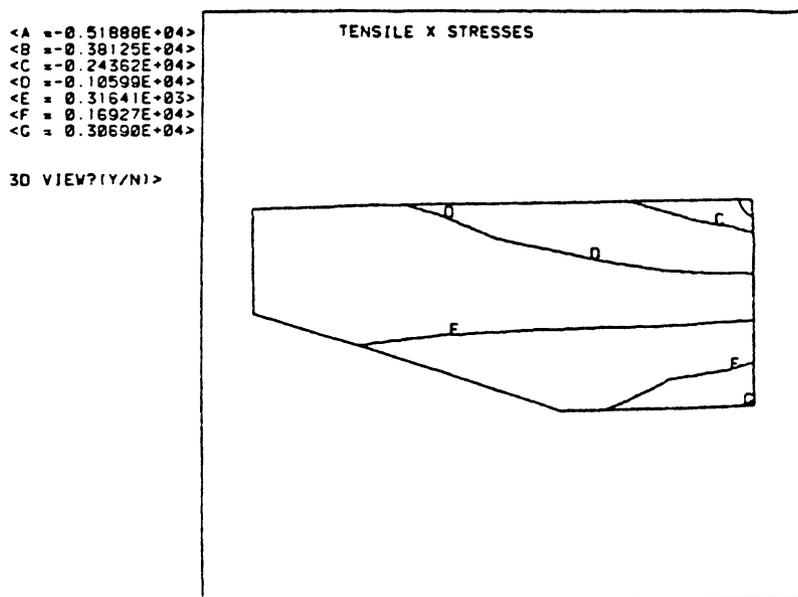


Figure 5-33. Plot of the Tensile X Stresses.

accurate values for the stresses as quickly as the grids made with methods one and two.

The results from these two test cases indicate that method three will also lead to a near-optimum mesh. However, in test case two method three did not lead to a near-optimum mesh as quickly as methods one and two. In the first test case method three lead to a near-optimum mesh faster; where, the comparisons between near-optimum meshes generated by different methods is based on the maximum to minimum ratios of the second plots of each method. The stresses obtained with this near-optimum mesh were not as close to the analytical solution as the stresses obtained with the not so near-optimum meshes based on methods one and two. For these reasons, it seems that for the element considered in this study, a quadrilateral, isoparametric, plane stress element, method three is not the best optimizing algorithm. Instead, methods one or two give a more accurate mesh optimizing algorithms. Recall that method three consisted of the Von Mises equivalent stresses contour plot (first plot) and the Von Mises equivalent stresses times element area (second plot). Method one consisted of the strain energy density function (first plot) and the strain energy per element (second plot). Finally, method two consisted of the same functions as method one, except that instead of using the normal stresses, the deviatoric stresses were used. In the remainder of this discussion, the following abbreviations will be used for the plots of methods one and two:

- SED - Strain energy density, first plot of method one
- SE - Strain energy per element, second plot of method one
- DSSED - Strain energy density computed with deviatoric stresses
first plot of method two
- DSSE - Strain energy per element computed with deviatoric
stresses, second plot of method two.

The near-optimum results observed with the normal and deviatoric-stress-based strain-energy function were extremely close. In fact, it was not determined which of the two would be a better basis for mesh optimizing contour plot. The procedure followed in the two test cases above seems to be the most appropriate for a first time finite element user. A relatively coarse even mesh is used for the first solution of the problem. The first mesh must be coarse for three reasons. First, a coarse mesh is easy to generate using any of the commonly available mesh generators. Second, it will not take much computer time to solve such a mesh. Third, if there are any stress concentration regions within the mesh, either from the geometry or the loads, the results from the coarse mesh will not be biased. If they are biased, the contours of the SED, or DSSED functions appear as a blob in a one region. Once this coarse mesh has been solved, the user must observe the DSSED or SED contour plots and determine whether the overall plot resembles the original coarse mesh. If the mesh does not look like the SED or DSSED contour plots, then the user should use the contour lines as guides in developing the next mesh for the problem. In the first iteration it will probably be easier to follow the SED and DSSED contour lines, rather than determine which areas to refine and which to enlarge based on the SE per element or the DSSE per element function plots. Once this new mesh is solved, the user should again look at the SED or DSSED contour plots to see if they resemble the mesh used to compute the contours. If the contour lines are gathered about a certain region of the mesh, while other regions have almost no contours in them, then there is a stress concentration area where all the contours are

gathered. If, on the other hand, the contour plots look similar to the mesh, then a near-optimum mesh of sorts has been developed. To determine the adequacy of this near-optimum mesh the user must study the SE or DSSE per element contour plots -- the user must also do this if there is stress concentration region in the problem. In particular, the user must examine the maximum to minimum ratio of these contours. The closer this ratio is to one the closer the near-optimum mesh is to the optimum mesh.

A single desirable ratio for the DSSE and SE contour plots was not established. In this study, the coarse mesh of the first test case had a ratio in the order of 10, while in the second test case the refined mesh had a ratio on the order of 30. A possible recommendation is that if there are stress concentration regions, the ratio should be brought down to below 30 before the mesh is called a near-optimum mesh; and if there are no stress concentration regions, then the ratio should be brought below 5 before calling the mesh a near-optimum mesh. Note that to bring the ratio within these recommended values the user will have to reduce the size of the elements which have the higher values for the SE or DSSE per element function, and increase the size of the elements which have low values for the same functions.

CONCLUSIONS

Six energy functions were postulated as possible mesh optimizing functions. These functions were grouped together based on the stresses used in each function. The grouping resulted in three different methods for mesh optimization. Each one of the groups or methods had two plots associated with it. The first plot of each method was the one which resembled the optimum mesh. The second plot in each method was the one which had to be constant throughout the mesh for the mesh to be optimum.

Two test cases were analyzed to determine whether the six postulated functions generated near optimum meshes. The test cases were conducted on a finite element code at Virginia Polytechnic Institute and State University. This finite element code was interfaced with one of the programs in the MOVIE package distributed by Brigham Young University, to display the contour plot of the desired functions interactively. The finite element used in the test cases was an isoparametric, quadrilateral, plane-stress element. The results of the test cases showed that all six postulated optimization functions lead to a refined meshes which were closer to the optimum mesh than the coarse mesh used for the parent analysis.

Having established that all the postulated functions lead to a near-optimum mesh, the test cases were used to determine which function was the better one. The postulated method three was the first one to be eliminated, in the search for the best method. In one instance, method three did not lead to a near-optimum mesh as fast as methods one and two. In the other, method three arrived at an almost optimum mesh in

just one iteration; however, the stresses obtained from that mesh were not as accurate as the stresses from method one and two. It was more difficult to determine which of the two remaining methods was the best one to use because they were so much alike.

Methods one and two use the strain energy function to determine the values to be plotted. Method one used the normal stresses in conjunction with the strain energy function. Method two uses the deviatoric stresses in conjunction with the strain energy function to compute the values to be plotted. Both methods gave extremely similar results for the isoparametric element used in this study, although method two seemed to be more sensitive. Therefore, the following recommendation is based on computational speed rather than on accuracy. Method one is method recommended as an optimizing function, because program MESS computes the normal stresses for the available isoparametric elements. Thus, these stresses can be directly passed to the strain energy formula. If method two were to be used, then the deviatoric stresses would have to be computed first from the normal stresses before being passed to the strain energy formula.

The test cases provided a means to determine which approach was the best one to use in developing a near-optimum mesh. A list of the steps to follow was included along with the results of the test cases. A desirable ratio for the second plot of method one was not determined. Instead, two target ratios were given for the strain energy per element contour plot (method one, plot two). In the presence of an infinite stress raiser the ratio should be brought down to below 30, in the absence of an infinite stress raiser then the ratio should be brought

down to below 5. Once these requirements have been met, then the mesh can be called a near-optimum mesh. If by chance the ratio is determined to be one then the mesh is the optimum mesh for that problem.

RECOMENDATIONS FOR FUTURE WORK

What follows is a list of the possible future work based on the present findings and developments. The recommendations have been organized by their degree of complexity.

- (1) Investigate the sensitivity of the strain-energy per element and the deviatoric-stress-based strain-energy per element. It seems that the latter is more sensitive to the coarseness of the mesh; that is, it has a larger maximum to minimum contour ratio for the same mesh. The investigation should determine whether or not the added sensitivity is needed to develop a near-optimum mesh, taking into account the extra computations needed to calculate the deviatoric-stress-based strain-energy.
- (2) Verify whether or not the methods and criteria presented here will lead to the same conclusions for other finite elements. Note that the isoparametric, axisymmetric element in the MESS program has been setup to provide the plots used in this study.
- (3) Enhance the post-processor to serve as a pre-processor; that is, help the user develop the new mesh. There are several approaches that may be taken to accomplish this. A quick and easy way may be to provide cursor movements to the user. The user can use the cursor, instead of a digitizing tablet, to generate the node points of the new mesh.
- (4) Make the mesh refinement process totally automatic. This will require more mesh refinement studies. The studies will have to determine several of the criteria needed to make the optimization process automatic. For example:
 - (a) Which contour plots are to be used?
 - (b) How will degrees of freedom be added or deleted?
 - (c) What areas are to be refined first?
 - (d) When does the mesh refinement process stop?

REFERENCES

1. Shephard M. S., "Finite Element Grid Optimization - A Review," Publication of the American Society of Mechanical Engineers, PVP-38, pp. 1-15, 1979.
2. McNiece, G. M., and Marcal, P. V., "Optimization of Finite Element Grids Based on Minimum Potential Energy", Journal of Engineering for Industry, Trans. ASME, Vol. 95, 1973, No. 1, pp. 186-190.
3. Cook, R. D., Concepts and Applications of Finite Element Analysis, John Wiley and Sons, New York, 1974.
4. Babuska, I., and Rheinboldt, W. C., "A Posteriori Error Estimates For The Finite Element Method". International Journal for Numerical Methods in Engineering, Vol. 12, 1978, pp. 1597-1615.
5. Babuska, I., and Rheinboldt, W. C., "On the Reliability and optimization of the Finite Element Method", Computers and Structures, Vol 10, 1974, pp. 87-94.
6. Carey, F. C., and Humphrey, D. L., "Mesh Refinement and Iterative Solution Methods for Finite Element Computations", International Journal for Numerical Methods in Engineering, Vol 17, 1981, pp. 1717-1734.
7. Turcke, D. J., and McNeice, G. M. "Guidelines for Selecting Finite Element Grids Based on an Optimization Study", Computers and Structures, Vol. 4, 1974, pp. 499-519.
8. Melosh, R. J., and Marcal, P. V., "An Energy Basis For Mesh Refinement of Structural Continua", International Journal for Numerical Methods in Engineering, Vol. 11, 1976, pp. 1083-1091.
9. Shephard, M. S., Gallagher, R. H., and Abel, J. F., "The Synthesis of Near-Optimum Finite Element Meshes With Interactive Computer Graphics", International Journal for Numerical Methods in Engineering, Vol. 15, 1979, pp. 1021-1039.
10. Hinton, E., and Campbell, J. S., "Local and Global Smoothing of Discontinuous Finite Element Functions Using a Least Squares Method", International Journal for Numerical Methods in Engineering, Vol. 8, 1973, pp. 461-480.
11. Zienkiewicz, O. C., The Finite Element Method, McGraw-Hill Book Co. (UK) London, 1979.

12. Bathe, K. J., and Wilson, E. L., Numerical Methods in Finite Element Analysis, Prentice-Hall Incorporated, Englewood Cliffs, New Jersey, 1976.
13. Brigham Young University, MOVIE.BYU -- A General Purpose Computer Graphics System, Brigham Young University Computer Science Department, Provo Utah, 1981.
14. Roark, R. J., and Young, W. C. Formulas for Stress and Strain -- Fifth Edition, McGraw-Hill Book Company, New York, 1975.

**The vita has been removed from
the scanned document**

SYNTHESIS OF A NEAR-OPTIMUM FINITE ELEMENT MESH

by

Jaime Jara-Almonte

(ABSTRACT)

A finite element program post-processor is developed to help synthesize a near-optimum mesh. This post-processor displays, interactively, six different energy functions, along with the usual stress functions. The six energy functions are functions of the energy within each element of the mesh. Since it is intended to be an educational package, the post-processor does not automatically generate the new mesh for the user. Instead, guidelines for near-optimum mesh generation are presented. The guidelines are based on the test cases included in the study. These test cases were also used to select the combination of functions that seemed to be more desirable for optimization. A list of possible future work is included.