

AD-772 508

ASPECTS OF MECHANICAL BEHAVIOR OF
ROCK UNDER STATIC AND CYCLIC LOADING.
PART A: MECHANICAL BEHAVIOR OF ROCK
UNDER STATIC LOADING

Jesus E. Basas, et al

Wisconsin University

Prepared for:

Advanced Research Projects Agency
Bureau of Mining

April 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

ASPECTS OF MECHANICAL BEHAVIOR OF ROCK UNDER
STATIC AND CYCLIC LOADING

PART A: MECHANICAL BEHAVIOR OF ROCK UNDER STATIC LOADING

Final Technical Report
April, 1973

by

R.W. Heins (Co-Principal Investigator
with B.C. Haimson)

Department of Metallurgical & Mineral Engineering
and the
Engineering Experiment Station
College of Engineering
The University of Wisconsin
Madison, Wisconsin 53706

ARPA Order No. 1579, Amendment 3
Program Code No. 2F10. Contract No. H0220041
Contract Period: March 1972 through April 1973
Total Amount of Contract: \$50,000
Sponsored by ARPA and Monitored by U.S. Bureau of Mines

Disclaimer:

The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

AD 772508

3200.8 (Att 1 to Encl 1)

Mar 7, 66

| Security Classification | | |
|--|--|---|
| DOCUMENT CONTROL DATA - R & D | | |
| <small>(Security classification of text, body of abstract and index of annotations must be entered when the overall report is classified)</small> | | |
| 1. ORIGINATING ACTIVITY (Corporate author) Engineering Experiment Station University of Wisconsin Madison, Wisconsin 53706 | | 2a. REPORT SECURITY CLASSIFICATION Unclassified |
| 3. REPORT TITLE Aspects of Mechanical Behavior of Rock under Static and Cyclic Loading Part A - Mechanical Behavior of Rock under Static Loading | | 2b. GROUP |
| 4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Annual Technical Report | | |
| 5. AUTHOR(S) (First name, middle initial, last name) Jesus E. Basas, Robert W. Heins | | |
| 6. REPORT DATE April 1973 | 7a. TOTAL NO. OF PAGES 99 | 7b. NO. OF REFS 4 |
| 8a. CONTRACT OR GRANT NO. H0220041 | 9a. ORIGINATOR'S REPORT NUMBER(S) 144-C745 | |
| b. PROJECT NO. ARPA Order No. 1579 Amendment 3 | 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) | |
| c. Program Code No. 2F10 | d. | |
| 10. DISTRIBUTION STATEMENT Distribution of this document is unlimited. | | |
| 11. SUPPLEMENTARY NOTES | | 12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Washington, D.C. 20301 |
| 13. ABSTRACT A finite element program based on both 2- and 3-dimensional element models is developed to simulate the Brazilian test. Rock isotropy, non-homogeneity, and anisotropy are taken into account in the analysis. The program has two failure criteria options: the old criterion described in the semi-annual report and a new criterion in which the elastic modulus across tension cracks is reduced to zero. Tests runs have shown that the new failure criterion yields far more realistic load-displacement and load-strain curves. A new equation solving procedure, the sparse matrix technique, is used in the program making possible the solution of certain problems which could not be handled by the banded matrix technique because of computer storage limitations. To demonstrate the capabilities of the program, several test problems were run. The results are shown in Chapter 2 of this report. | | |

DD FORM 1473
1 NOV 65

Security Classification

1a

| | GROUP A | | GROUP B | | GROUP C | |
|-------------------------------|---------|----|---------|----|---------|----|
| | ROLL | WT | ROLL | WT | ROLL | WT |
| Brazilian test | | | | | | |
| finite element method | | | | | | |
| 2- and 3-dimensional elements | | | | | | |
| isotropy | | | | | | |
| anisotropy | | | | | | |
| non-homogeneity | | | | | | |
| sparse matrix | | | | | | |
| isoparametric elements | | | | | | |

Security Classification

PREFACE

This report covers the second year accomplishments in the research program entitled, "Mechanical Behavior of Rock Under Static Loading," R.W. Heins, Co-Principal Investigator. The program is Part A of the project entitled, "Aspects of Mechanical Behavior of Rock Under Static and Cyclic Loading" (Contract No. H0220041). Part B of the project is published in a separate volume.

ACKNOWLEDGEMENTS

The reported research was supported by ARPA and monitored by Dr. Syd S. Peng of the Twin Cities Mining Research Center, U.S. Bureau of Mines. His cooperation and that of Mr. Egons S. Podnieks are gratefully acknowledged.

The work reported here has been the principal responsibility of Dr. Jesus E. Basas who was a postdoctoral fellow on this project. The contribution of Mr. A. Hayatdavoudi on a laboratory investigation of size effect in the Brazilian test which was previously reported in a semiannual report should be noted.

SUMMARY

ASPECTS OF MECHANICAL BEHAVIOUR OF ROCK UNDER STATIC LOADING

PART A

Summary of Year's Work

Brazilian tests were carried out on dacite, Valder's limestone, and St. Cloud gray granodiorite to determine size effect on tensile strength. Curves showing variation of tensile strength with specimen dimensions and other findings constitute Chapter 1 of the semi-annual technical progress report⁽¹⁾.

In connection with the theoretical phase of the investigation, a 2-dimensional computer program which simulates the Brazilian test is developed. The program is based upon isoparametric finite elements which may be either isotropic or non-homogeneous and a failure criterion in which a fixed percentage of the stiffnesses of the failed elements is subtracted from the total stiffness during each loading cycle. The complete program listing and the results of several computer runs are part of the semi-annual report⁽¹⁾.

A second program is presented in this report. The program employs 2- and 3-dimensional isoparametric elements which exhibit anisotropy as well as isotropy and non-homogeneity. Two failure criteria are used: the old criterion described in the preceding paragraph and a new criterion in which the elastic modulus across tension cracks is reduced to zero. Based on few test runs made, the new failure criterion yields considerably superior and more realistic load-displacement and load-strain curves.

To demonstrate the capabilities of the program, several problems were run and the results are shown in Chapter 2. The use of a new equation solver, the sparse matrix technique, permitted the solution of certain problems which could not be handled by the banded matrix technique because of the bandwidth limitation.

TABLE OF CONTENTS

| | |
|--|-----|
| PREFACE | i |
| ACKNOWLEDGEMENTS | ii |
| SUMMARY | iii |
| TABLE OF CONTENTS | iv |
| LIST OF FIGURES | v |
| LIST OF TABLES | vi |
| CHAPTER 1 - PROGRAMMING ROUTINES | 1 |
| 1.1 Introduction | 1 |
| 1.2 Nodal Reordering | 1 |
| 1.3 Gaussian Elimination | 11 |
| 1.4 Determination of Principal Stresses in Three Dimensions | 17 |
| References | 21 |
| CHAPTER 2 - NUMERICAL EXAMPLES | 22 |
| 2.1 Introduction | 22 |
| 2.2 Two-dimensional Anisotropic Problem | 24 |
| 2.3 Two-dimensional Isotropic Problem (Old Failure Criterion) | 24 |
| 2.4 Two-dimensional Isotropic Problem (New Failure Criterion) | 31 |
| 2.5 Three-dimensional Isotropic Problem (Old Failure Criterion) | 31 |
| 2.6 Three-dimensional Isotropic Problem (New Failure Criterion) | 31 |
| 2.7 Three-dimensional Anisotropic Problem | 41 |
| 2.8 Three-dimensional Non-homogeneous Problem | 48 |
| 2.9 Discussion of Results and Conclusion | 52 |
| APPENDIX A - PROGRAM LISTING | 55 |
| APPENDIX B - DATA CARDS PREPARATION GUIDE | 95 |

LIST OF FIGURES

| | | |
|-------------|--|-------|
| FIGURE 1.1 | Standard Scheme for Numbering Nodes and Elements of Two-dimensional Disc | 3 |
| FIGURE 1.2 | Numbering Scheme for Nodes in Three-dimensional Mesh | 4 |
| FIGURE 1.3 | Flow Chart of Nodal Reordering Routine | 5-7 |
| FIGURE 1.4 | Unreordered Stiffness Matrix | 9 |
| FIGURE 1.5 | Reordered Stiffness Matrix | 10 |
| FIGURE 1.6 | Flow Chart of Gaussian Elimination Routine | 14-16 |
| FIGURE 1.7 | Graphical Representation of Cubic Polynomial (Eq. 1.1) | 20 |
| FIGURE 2.1 | Progression of Failure in Problem 2.2 | 26 |
| FIGURE 2.2 | Load-Displacement-Strain Curves (Problem 2.2) | 27 |
| FIGURE 2.3 | Progression of Failure in Problem 2.3 | 29 |
| FIGURE 2.4 | Load-Displacement-Strain Curves (Problem 2.3) | 30 |
| FIGURE 2.5 | Progression of Failure in Problem 2.4 | 33 |
| FIGURE 2.6 | Load-Displacement-Strain Curve (Problem 2.4) | 34 |
| FIGURE 2.7 | Progression of Failure in Problem 2.5 | 36 |
| FIGURE 2.8 | Load-Displacement-Strain Curves (Problem 2.5) | 37 |
| FIGURE 2.9 | Progression of Failure in Problem 2.6 | 39 |
| FIGURE 2.10 | Load-Displacement-Strain Curves (Problem 2.6) | 40 |
| FIGURE 2.11 | Progression of Failure in Problem 2.7 | 46 |
| FIGURE 2.12 | Load-Displacement-Strain Curves (Problem 2.7) | 47 |
| FIGURE 2.13 | Progression of Failure in Problem 2.8 | 50 |
| FIGURE 2.14 | Load-Displacement-Strain Curves (Problem 2.8) | 51 |

LIST OF TABLES

| | | |
|-----------|--|-------|
| TABLE 1.1 | Locator Arrays for Non-zero Elements of Figure 1.5c | 11 |
| TABLE 2.1 | Nodal Reordering for Problem 2.7 | 43-45 |
| TABLE 2.2 | Test Problems Comparison | 54 |
| TABLE 2.3 | Program Limits | 54 |
| TABLE A | Input Cards Arrangement | 96 |

PART A

ASPECTS OF MECHANICAL BEHAVIOR OF ROCK
UNDER STATIC AND CYCLIC LOADING

A FINITE ELEMENT MODEL OF ROCK FAILURE
FROM THE BRAZILIAN TEST

Jesus E. Basas
Robert W. Heins

April, 1973

Department of Metallurgical & Mineral Engineering
The University of Wisconsin
Madison, Wisconsin 53706

CHAPTER 1

PROGRAMMING ROUTINES

1.1 Introduction

A finite element program for the prediction of rock fracture characteristics in Brazilian tests is presented herein. The program has all the capabilities of the previous 2-dimensional program (1) plus the following:

1. Availability of both 2- and 3-dimensional elements with isotropic, anisotropic, or non-homogeneous material property.
2. A failure criterion in which the elastic modulus across tension cracks is reduced to zero.
3. An equation solver in which only the non-zero elements of the stiffness matrix are involved in the elimination and back-substitution processes.
4. A nodal reordering scheme.

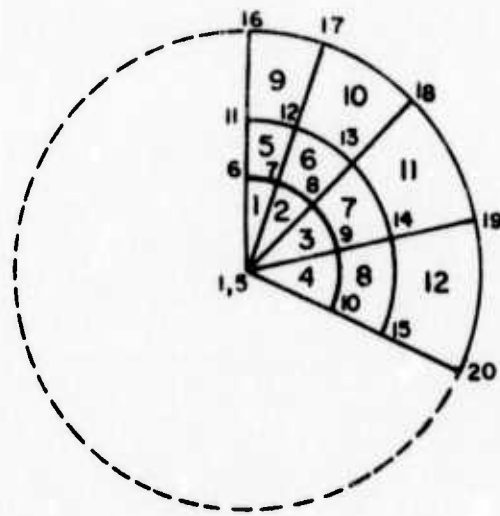
The theoretical considerations upon which the program is based have already been discussed in the semi-annual report (1) and are not repeated here. The program's major routines are described in the following sections.

1.2 Nodal Reordering

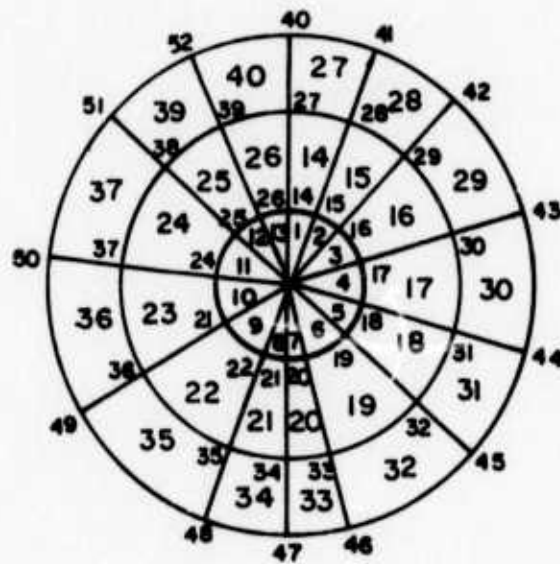
Among the different mathematical undertakings encountered in a finite element program, the solution of the governing linear equilibrium equations determines to a large extent the effectiveness of the program. When the first finite element programs were developed only small-size systems could be treated because of limited memory space for the storage of the global

stiffness matrix. With the introduction of external storage disks, it became possible to store only a small part of the stiffness matrix in computer memory at a time, the balance being in the disk, permitting the solution of considerably larger systems. The most widely used disk-aided equation solving procedure is the banded matrix technique developed in 1965 by E.L. Wilson of the University of California. Wilson's method takes advantage of the banded property of the stiffness matrix resulting in savings in execution time and storage requirement. Its only limitation is in the width of the band. Unfortunately, the maximum bandwidth that most digital computers allow is not large enough in many practical situations. In the context of the present research, such situations arise when the whole circular face of the Brazilian test specimen is considered in the analysis and/or when 3-dimensional problems are treated. In both of these cases, because of the closed geometry of the finite element mesh and the high nodal connectivity of the 3-dimensional elements, a low-numbered node becomes connected to a high-numbered node resulting in prohibitively large bandwidths. Nodal reordering schemes have been developed to minimize the bandwidth. The effectiveness of such schemes, however, is very limited.

An equation solver which stores and processes only the non-zero elements of the stiffness matrix is employed in the present program. To minimize the number of non-zero elements created during the elimination process, a reordering of the nodes is undertaken. The reordering concept used is based upon one described by Jensen and Parks (2). The complete reordering flow chart is shown in Figure 1.3. To start with, the nodes are assigned numbers in the manner shown in Figures 1.1 and 1.2. This allows for the automatic generation, within the program, of the nodes and their connections. The reordering of the nodes then starts. At each stage of the reordering process, the branch table of the last node (JLØC) entered in the



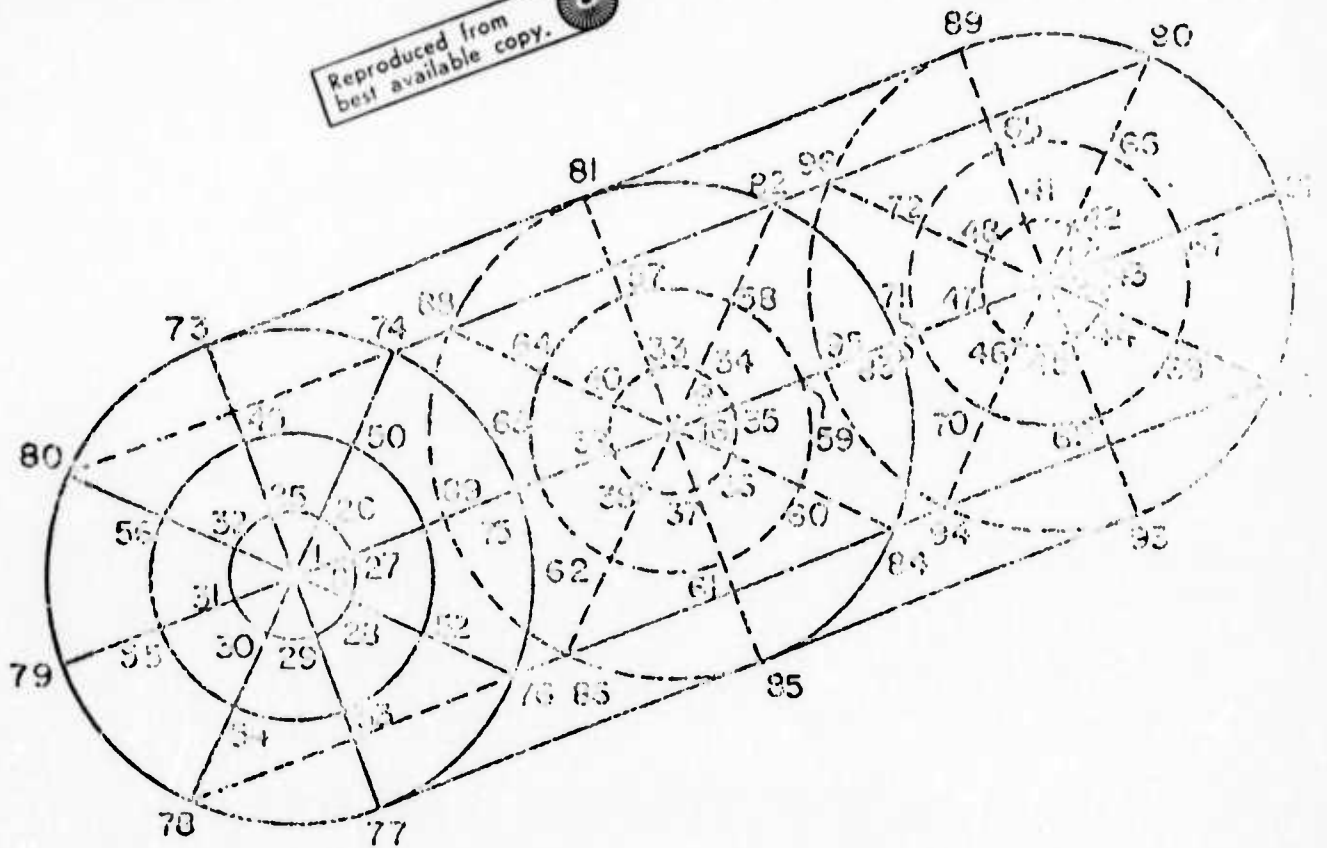
(a) Partial Disc



(b) Full Disc

Figure 1.1. Standard Scheme for Numbering Nodes and Elements of Two-dimensional Disc.

Reproduced from
best available copy.



Note: Numbering of elements follows the same direction.

Figure 1.2 Numbering Scheme for Nodes in Three-dimensional Mesh

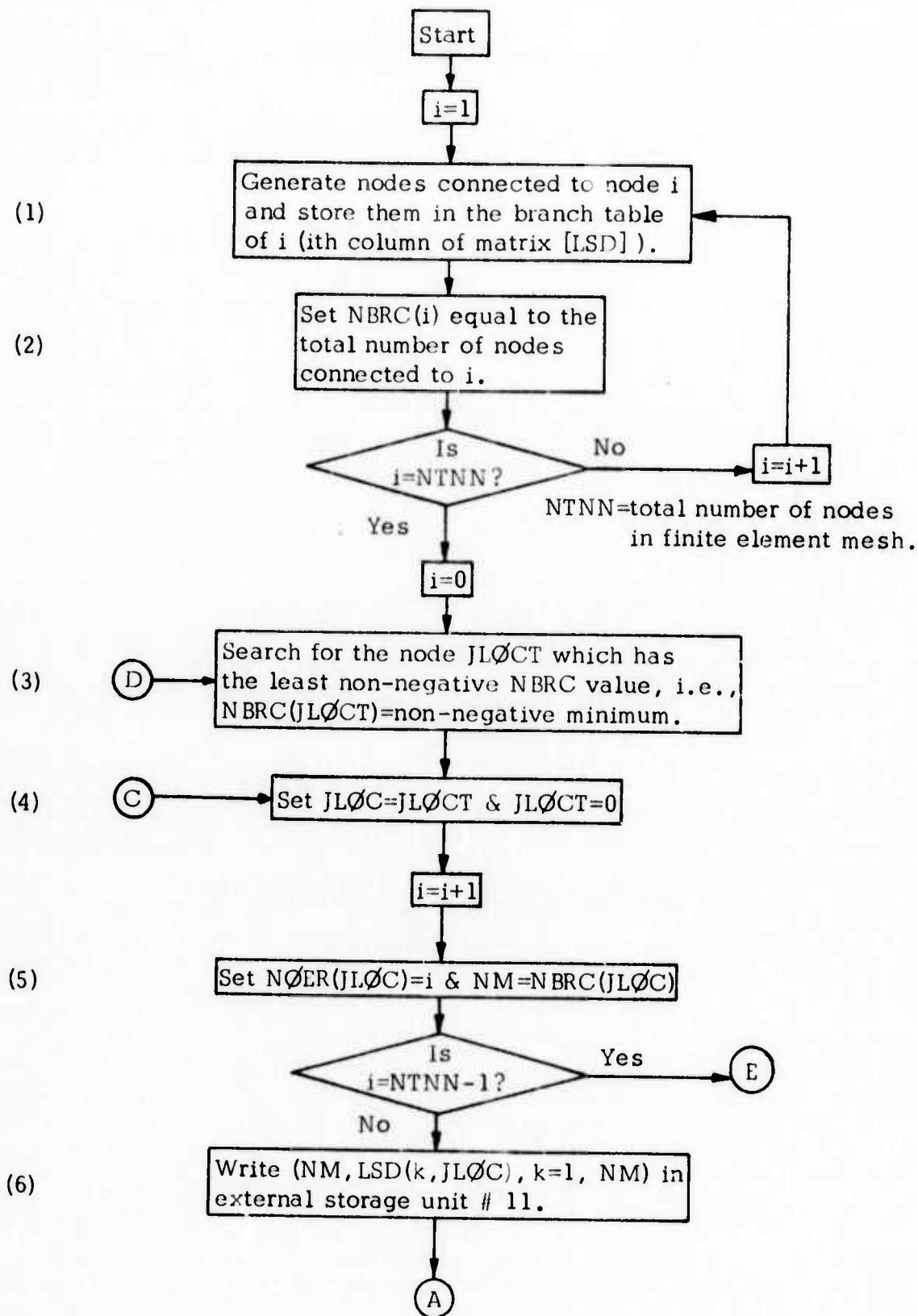


Figure 1.3. Flow Chart of Nodal Reordering Routine.

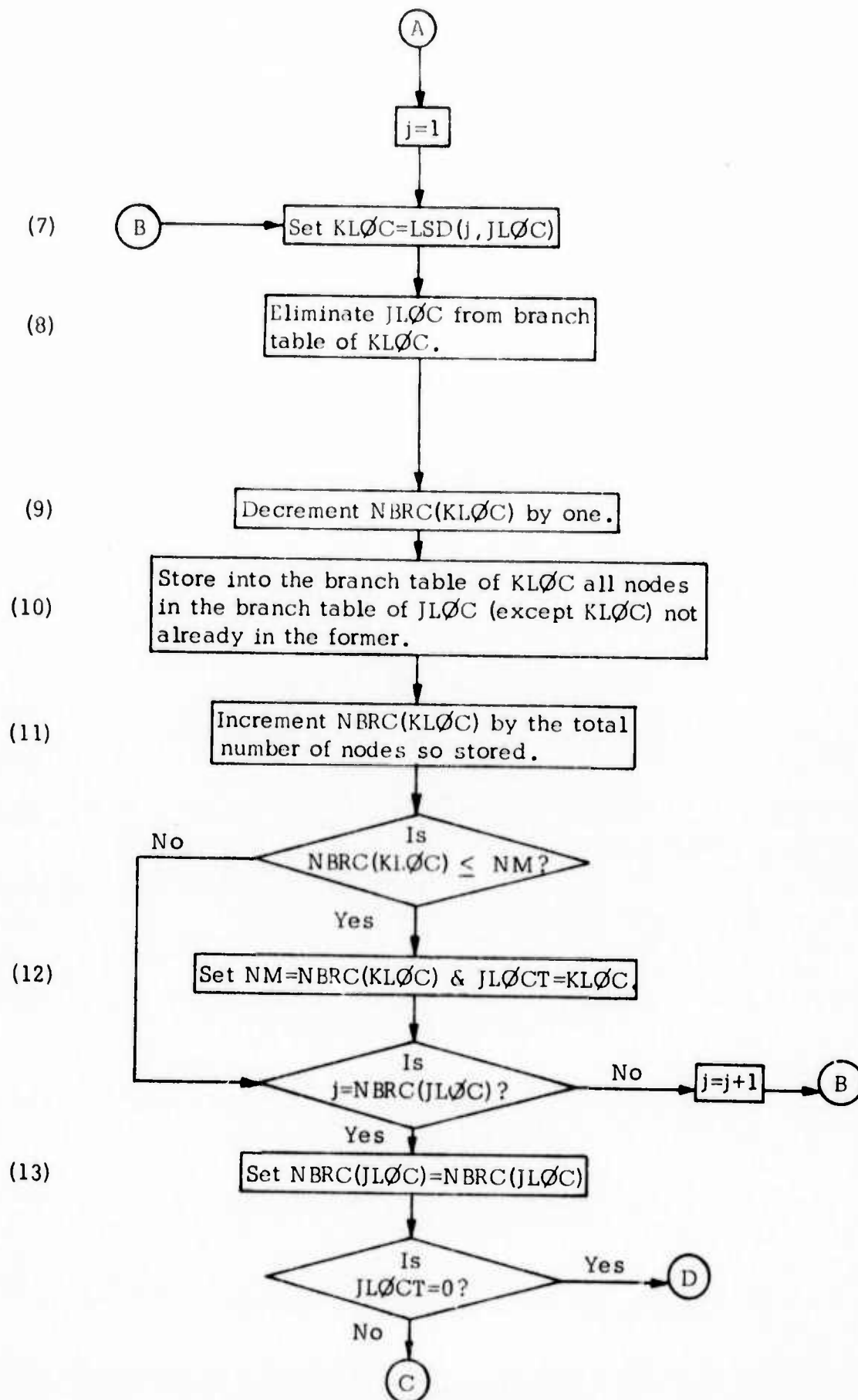


Figure 1.3 Continued.

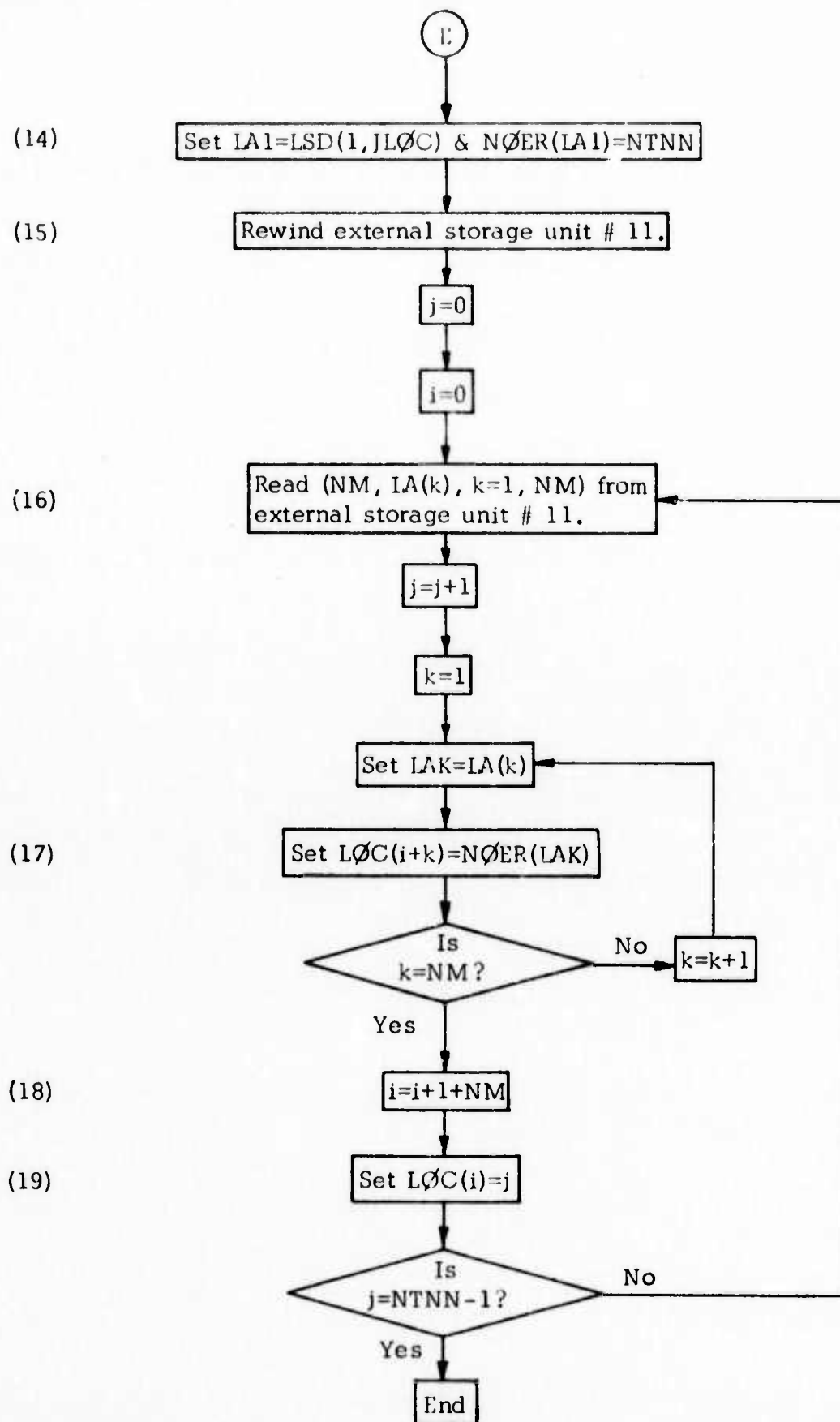


Figure 1.3 Continued.

reordering array* is scanned for the node with the fewest number of connections. If such node has equal or fewer connections than $JL\emptyset C$, then it is placed into the next available space of the reordering array. Otherwise, the node with the fewest number of connections from among those which remain to be renumbered is placed into the next available space of the reordering array. The process is repeated until all nodes have been renumbered. Whenever more than one node satisfy the aforementioned conditions, the node with the lowest number is selected. An example of the effect of reordering is illustrated in Figures 1.4 and 1.5. In Figure 1.4c no reordering is undertaken and 18 non-zero elements (represented by the letter c) are created during the elimination process. In the reordered version, Figure 1.5c, only 11 such elements are formed. The effect of reordering becomes even more significant in large systems where a high percentage of the non-zero elements are created during elimination.

The removal of $JL\emptyset C$ from the branch tables of the nodes connected to it (Step 8 of flow chart) is equivalent to the elimination of non-zero elements located directly below the diagonal element in the row of $JL\emptyset C$. Nodes added to the same branch tables (Step 10) on the other hand, represent the non-zero elements created as a result of the activity in Step 8. Steps 8 and 10 make up the pseudo-elimination process which is an essential part of the reordering routine. The final contents of the branch tables represent the off-diagonal non-zero elements of the upper triangular form of the stiffness matrix. The location of these elements has to be defined before the actual elimination can be carried out.

Two arrays are needed to locate a non-zero element. These are $NR\emptyset W$, generated for convenience during the actual elimination process (Figure 1.6),

* $NRE\emptyset$ in which $NRE\emptyset(i)$ gives the original number of the node assigned the new number i .

and $L\emptyset C$, generated in the reordering routine. $NR\emptyset W(j)$ gives the number assigned to the first off-diagonal non-zero element in row j . $L\emptyset C(i)$ gives the column location of the i th non-zero element. Thus the k th non-zero element would be located in column $L\emptyset C(k)$ and in row m where $NR\emptyset W(m) \leq k < NR\emptyset W(m+1)$. Sequential numbers, starting with one, are used to identify the non-zero elements. The numbering proceeds in the row-wise direction with the off-diagonal elements in a row numbered ahead of the diagonal element. The following tables, used to locate the non-zero elements in Figure 1.5c, serve to clarify the idea presented in this paragraph. The arrows point to the diagonal elements.

| | | | | | | | | | | | | | | | | |
|--------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| $L\emptyset C(j)$ | 5 | 9 | 10 | 1 | 6 | 12 | 15 | 2 | 7 | 11 | 13 | 3 | 8 | 14 | 16 | 4 |
| j | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | |
| $L\emptyset C(j)$ | 6 | 9 | 10 | 15 | 5 | 9 | 10 | 12 | 15 | 6 | 9 | 10 | 11 | 13 | 7 | |
| j | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | | | | | |
| $L\emptyset C(j)$ | 12 | 14 | 15 | 16 | 8 | 10 | 11 | 12 | 13 | 15 | 9 | | | | | |
| j | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | | | | |
| $L\emptyset C(j)$ | 11 | 12 | 13 | 15 | 16 | 10 | 12 | 13 | 14 | 15 | 16 | 11 | | | | |
| j | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | |
| $L\emptyset C(j)$ | 13 | 14 | 15 | 16 | 12 | 14 | 15 | 16 | 13 | 15 | 16 | 14 | 16 | 15 | 16 | |
| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
| $NR\emptyset W(i)$ | 1 | 5 | 9 | 13 | 17 | 22 | 27 | 32 | 37 | 43 | 49 | 55 | 60 | 64 | 67 | |

Table 1.1 Locator Arrays for Non-zero Elements of Figure 1.5c

1.3 Gaussian Elimination

In the discussion in the preceding section, it should be noted that a "non-zero element" is actually a submatrix whose order is equal to the

number of degrees of freedom specified at a node and a "row" are actually rows occupied by the submatrices. Hereafter, these expressions shall convey the same meanings. Thus, a "non-zero element" is either a 2x2 submatrix (2-dimensional cases) or a 3x3 submatrix (3-dimensional cases). Similarly, the "nth row" of the stiffness matrix would be taken to mean the nth 2 rows or the nth 3 rows of the matrix.

The flow chart of the Gaussian elimination routine is shown in Figure 1.6. In Step 3, only the non-zero elements on and to the right of the diagonal need be stored in $[S]$ because of symmetry. The elements are stored in the same order that they would appear in the stiffness matrix with enough spaces reserved for the non-zero elements which will be created later on during the elimination process. Using Figure 1.5c again as an example, $[S]$ would appear as follows during the storage of the different rows of the stiffness matrix:

Row 1:

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} & s_{27} & s_{28} \end{bmatrix}$$

Row 6:

$$\begin{bmatrix} s_{11} & s_{12} & 0 & 0 & s_{15} & s_{16} & s_{17} & s_{18} & s_{19} & s_{1,10} \\ s_{21} & s_{22} & 0 & 0 & s_{25} & s_{26} & s_{27} & s_{28} & s_{29} & s_{2,10} \end{bmatrix}$$

Row 9:

$$\begin{bmatrix} s_{11} & s_{12} & 0 & 0 & 0 & 0 & s_{17} & s_{18} & 0 & 0 \\ s_{21} & s_{22} & 0 & 0 & 0 & 0 & s_{27} & s_{28} & 0 & 0 \end{bmatrix}$$

Row 16:

$$\begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$$

in which the spaces marked s_{ij} are occupied by the x elements while those marked 0 (zero) are reserved for the c elements.

After the formation of any row, say i , the process of eliminating all non-zero elements in the row located to the left of the diagonal immediately follows. Had the entire upper triangular half of the stiffness matrix been considered in the elimination procedure, these elements would be simply the "mirror image" of the elements lying on the i th column of the preceding rows. Since this is not the case, a method for locating the aforementioned elements has to be devised. This is accomplished by introducing the spotter array NSPT. The spotter array keeps track, for each row, of the element in that row whose "mirror image" is next in line for elimination. Because of the row-wise direction of the elimination procedure, the order in which the elements of any particular row are used in the elimination process proceeds from left to right; that is, the "mirror image" of the left-most element is eliminated first and that of the right-most element is eliminated last. Thus, at the start of the elimination process $NSPT(j) = NR\emptyset W(j)$ for any row j and after each elimination the NSPT value of the affected rows is incremented by one. $NR\emptyset W(j)$ is defined on page 11. The procedure discussed in this paragraph corresponds to Steps 4 through 6 of the flow chart shown in Figure 1.6.

It should be noted that after elimination of a row, the elements in that row are stored in the next available space in tape or drum #10. Normalization of an element (pre-multiplication of the element by the inverse of the diagonal

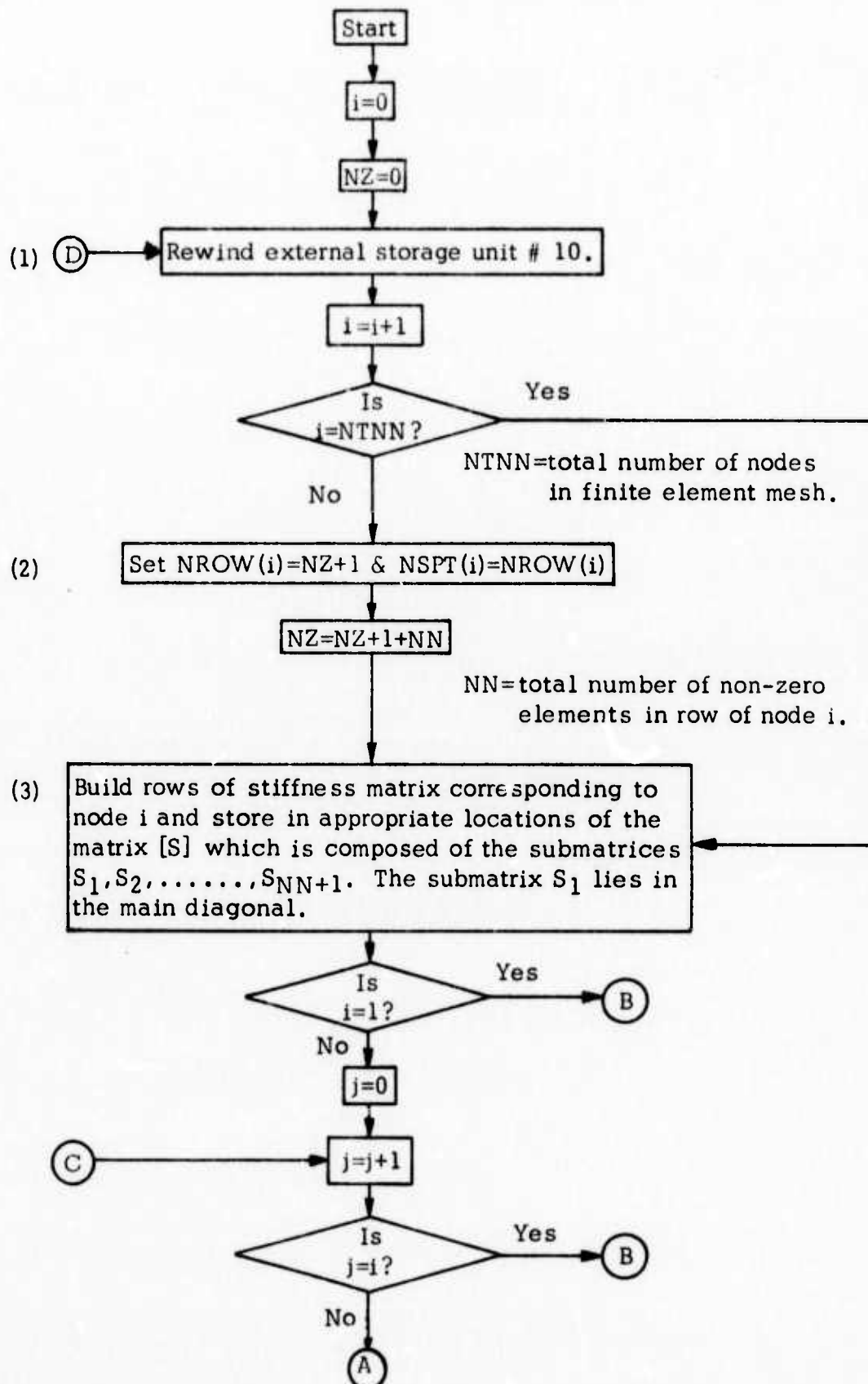


Figure 1.6. Flow Chart of Gaussian Elimination Routine

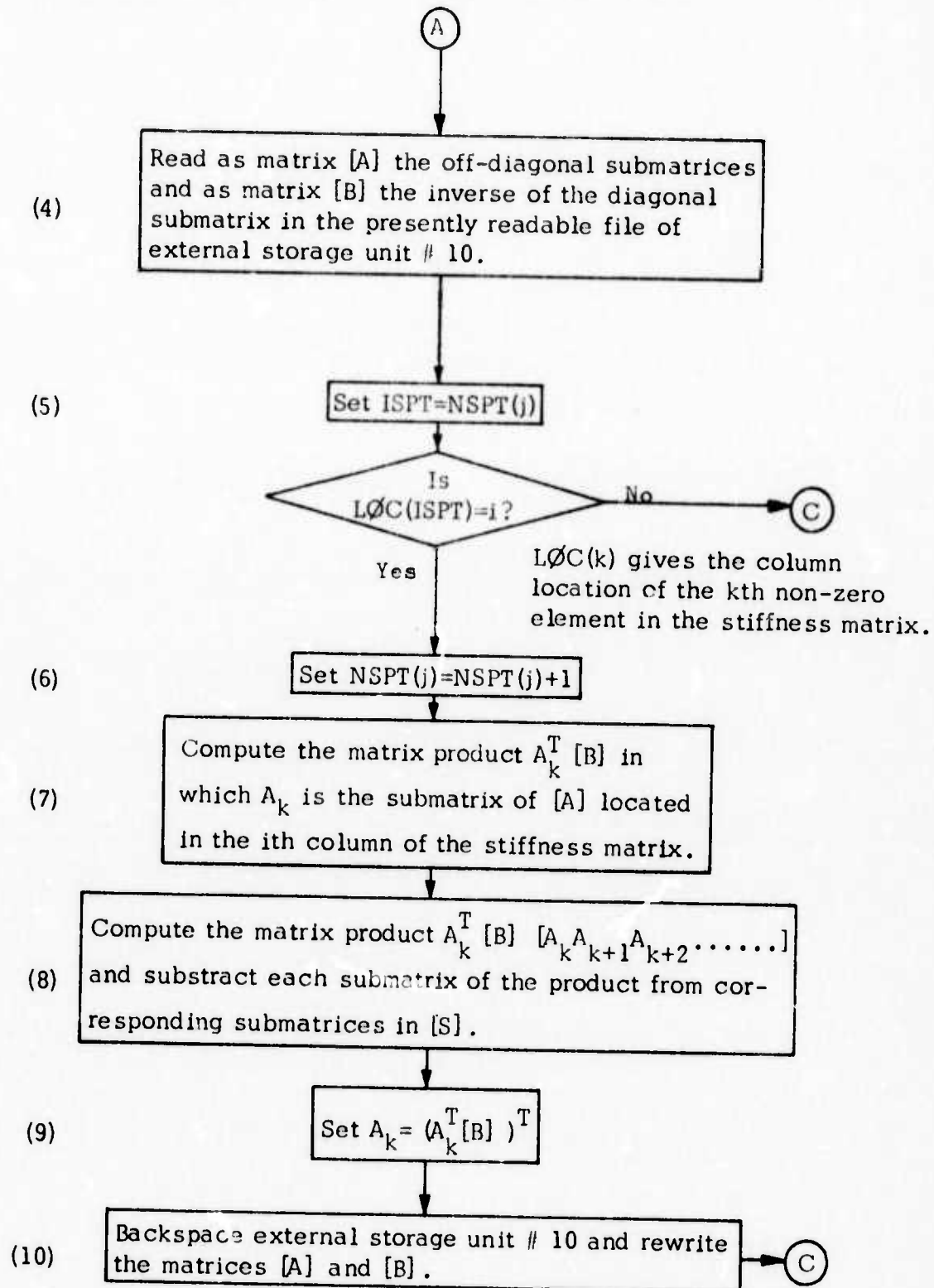


Figure 1.6 Continued.

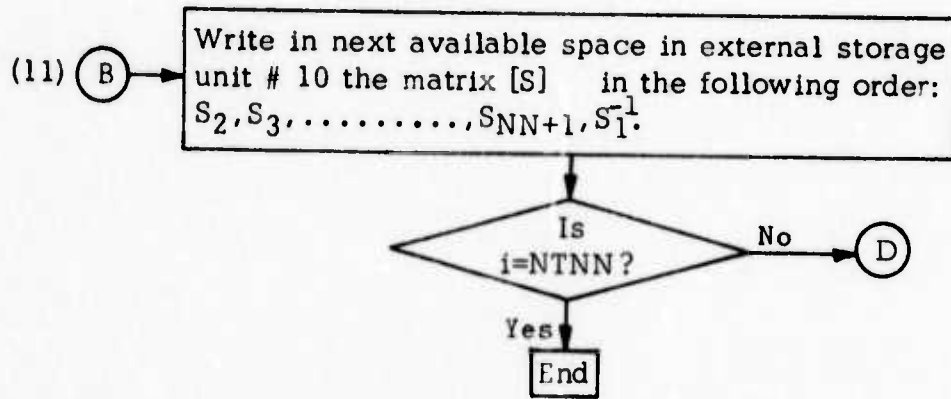


Figure 1.6 Continued.

element) takes place only when the element is no longer needed in the elimination process (Step 9). The actual row operations carried out during the elimination process are represented by Step 8.

2.4 Determination of Principal Stresses in Three Dimensions

It has been determined in Reference 3 that the principal stresses in a 3-dimensional solid are the roots of the cubic equation

$$F(S) = S^3 - a_1 S^2 + a_2 S - a_3 = 0 \quad (1.1)$$

in which

$$a_1 = \sigma_r + \sigma_\theta + \sigma_z$$

$$a_2 = \sigma_r \sigma_\theta + \sigma_\theta \sigma_z + \sigma_r \sigma_z - \tau_{\theta z}^2 - \tau_{r\theta}^2 - \tau_{rz}^2$$

$$a_3 = \sigma_r \sigma_\theta \sigma_z + 2\tau_{\theta z} \tau_{r\theta} \tau_{rz} - \sigma_r \tau_{\theta z}^2 - \sigma_\theta \tau_{rz}^2 - \sigma_z \tau_{r\theta}^2$$

The graph of $F(S)$, with the maxima, minima, and inflection points indicated, is shown in Figure 1.7.

Applying the Newton-Raphson method, the roots of equation (1.1) can be evaluated by means of the iteration formula

$$S_{n+1} = S_n - \frac{F(s_n)}{F'(s_n)} \quad (1.2)$$

in which

$$F'(S) = \frac{dF}{dS} = 3S^2 - 2a_1 S + a_2$$

$$n = 0, 1, 2, \dots$$

$$S_n = \text{approximation of root value at } n\text{th iteration}$$

Setting $S_0 = \frac{a_1}{3}$ (that is, starting the iteration at the inflection point I), equation (1.2) will yield the value of the middle root R_2 . With R_2 known, equation (1.1) may be rewritten as

$$F(S) = (S-R_2)Q(S) = 0 \quad (1.3)$$

in which $Q(S)$ is a quadratic polynomial whose roots are the two remaining roots of equation (1.1). $Q(S)$ may be conveniently written as

$$Q(S) = S^2 + b_1S + b_2 \quad (1.4)$$

in which (4)

$$b_1 = R_2 - a_1$$

$$b_2 = R_2b_1 + a_2$$

Thus the two remaining roots are

$$R_1 = \frac{-b_1 - \sqrt{b_1^2 - 4b_2}}{2}$$

$$R_3 = \frac{-b_1 + \sqrt{b_1^2 - 4b_2}}{2} \quad (1.5)$$

The direction cosines l_i, m_i, n_i of the principal stresses are obtained from the equations

$$(R_i - \sigma_r)l_i - \tau_{r\theta}m_i - \tau_{rz}n_i = 0$$

$$-\tau_{r\theta}l_i + (R_i - \sigma_\theta)m_i - \tau_{\theta z}n_i = 0$$

$$-\tau_{rz} l_i - \tau_{\theta z} m_i + (R_i - \sigma_z) n_i = 0 \quad (1.6)$$

taking into account the relationship

$$l_i^2 + m_i^2 + n_i^2 = 1 \quad (1.7)$$

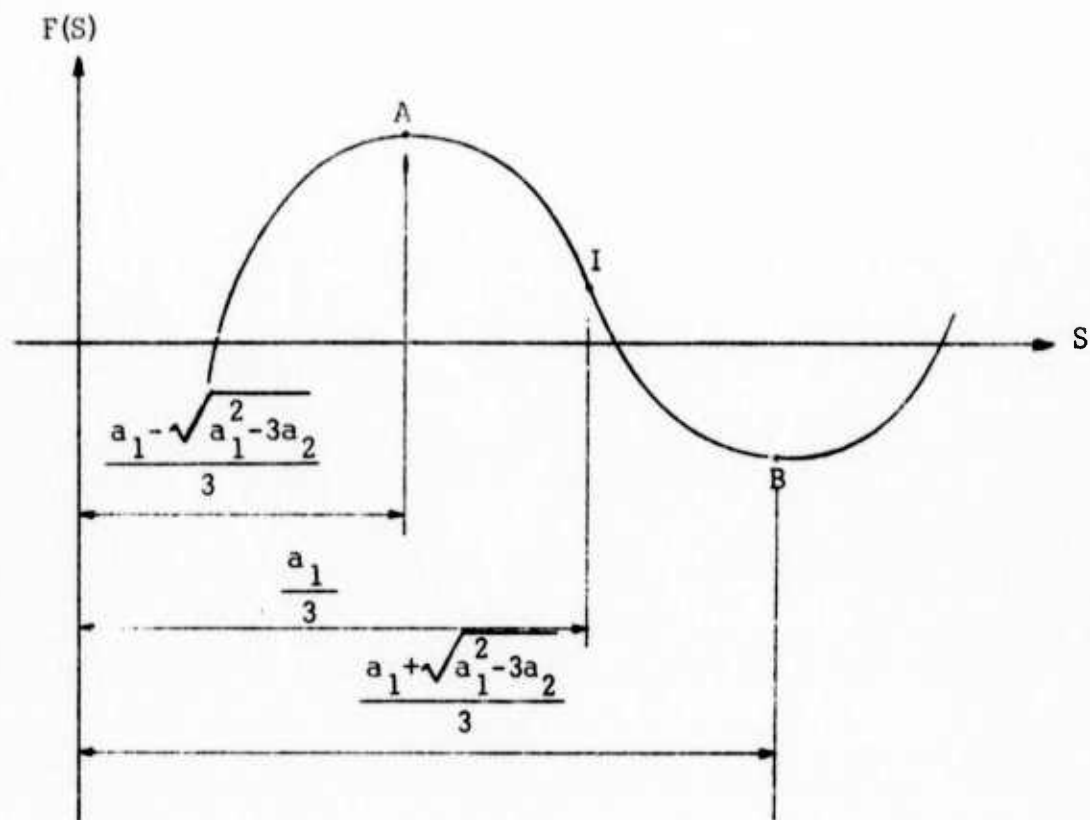


Figure 1.7. Graphical Representation of Cubic Polynomial (Eq. 1.1).

REFERENCES

1. "Aspects of Mechanical Behaviour of Rock Under Static and Cyclic Loading (Part A: Mechanical Behaviour of Rock Under Static Loading)", Semi-Annual Technical Progress Report, Engineering Experiment Station, University of Wisconsin, Madison, Wisconsin, September 1972.
2. Jensen, H.G. and G.A. Parks, "Efficient Solutions for Linear Matrix Equations", Journal of the Structural Division, Proc. ASCE, Vol. 96, No. ST1, Jan. 1970, pp. 49-64.
3. Timoshenko, S. and J.N. Goodier, Theory of Elasticity, McGraw-Hill Book Company, Inc., New York, 1951, p. 217.
4. Henrici, P., Elements of Numerical Analysis, John Wiley & Sons, New York, 1964, p. 86.

CHAPTER 2

NUMERICAL EXAMPLES

2.1 Introduction

To demonstrate the capabilities of the program, several cylinders exhibiting all three rock properties considered in the program are analyzed. Unfortunately, it is not possible to compare the results, particularly with respect to the three-dimensional, anisotropic, and non-homogeneous cases, with any existing alternative solution, so relatively coarse meshes are used to save on computer expense which quite prohibitive in most cases. Isotropic problems are run mainly to compare the load-displacement and load-strain curves obtained by the two failure criteria used in the program. In the new failure criterion, the stiffness matrix of the elements that failed is revised by reducing to zero the elastic modulus across tension cracks when failure is in tension or by removing a fixed percentage (specified by program user) of said matrix when the failure is in compression or shear. In the old failure criterion (used in semi-annual report), the stiffness matrix of the failed elements is reduced by a fixed percentage regardless of whether tensile, compressive or shearing failure occurs. One shortcoming of the old criterion is the unrealistic shape of the resulting load-strain curves and, to a certain extent, load-displacement curves.

In anisotropic discs and cylinders the allowable stresses at a point vary with direction making the determination of the load factor (critical ratio of actual stress to allowable stress) of an element quite involved, one requiring time-consuming trial-and-error procedure. Because of this, a simplifying assumption is made in the program in which the critical direction is considered to coincide with one of the principal stresses as in the case of isotropic discs and cylinders. This assumption makes it necessary only to

determine the allowable stress in the direction of the critical principal stress to obtain the element load factor. In the program the allowable stresses at a point are assumed to vary as an ellipsoidal surface with the principal axes coinciding with the directions of anisotropy.

Considerable effort was expended to check each individual subroutine of the program by long hand calculation. As part of the debugging process, it was also decided to rerun the isotropic problem presented as Problem 2 in the semi-annual report. The results were in close agreement with those obtained by the previous program.

All the cylinders treated herein possess the following dimensions:

Diameter = 3 inches

Length = 1 inch

"Vertical displacement" refers to the average displacement of the loaded nodes in the direction of the load while "horizontal strain" refers to the strain at the central point of the disc or cylinder measured normal to the loaded plane.

2.2 Two-dimensional Anisotropic Problem

Elastic Moduli:

$$E_1 = 3,000,000 \text{ psi}$$

$$E_2 = 5,700,000 \text{ psi}$$

Allowable Compressive Stresses:

$$C_1 = 15,000 \text{ psi}$$

$$C_2 = 27,000 \text{ psi}$$

Allow. Tension = .05 of Allow. Compression

Allow. Shear = .10 of Allow. Compression

Poisson's Ratio = .25

Shear Modulus = 1,600,000. psi

Finite Element Mesh (Whole Disc):

5 radial divisions

16 circumferential divisions

Directions of anisotropy are indicated in Figure 2.1.

2.3 Two-dimension Isotropic Problem (Old Failure Criterion)

Elastic Modulus = 5,700,000. psi

Allow. Compressive Stress = 27,000. psi

Allow. Tension = .05 of Allow. Compression

Allow. Shear = .10 of Allow. Compression

Poisson's Ratio = .25

Finite Element Mesh (Disc Quadrant):

6 radial divisions

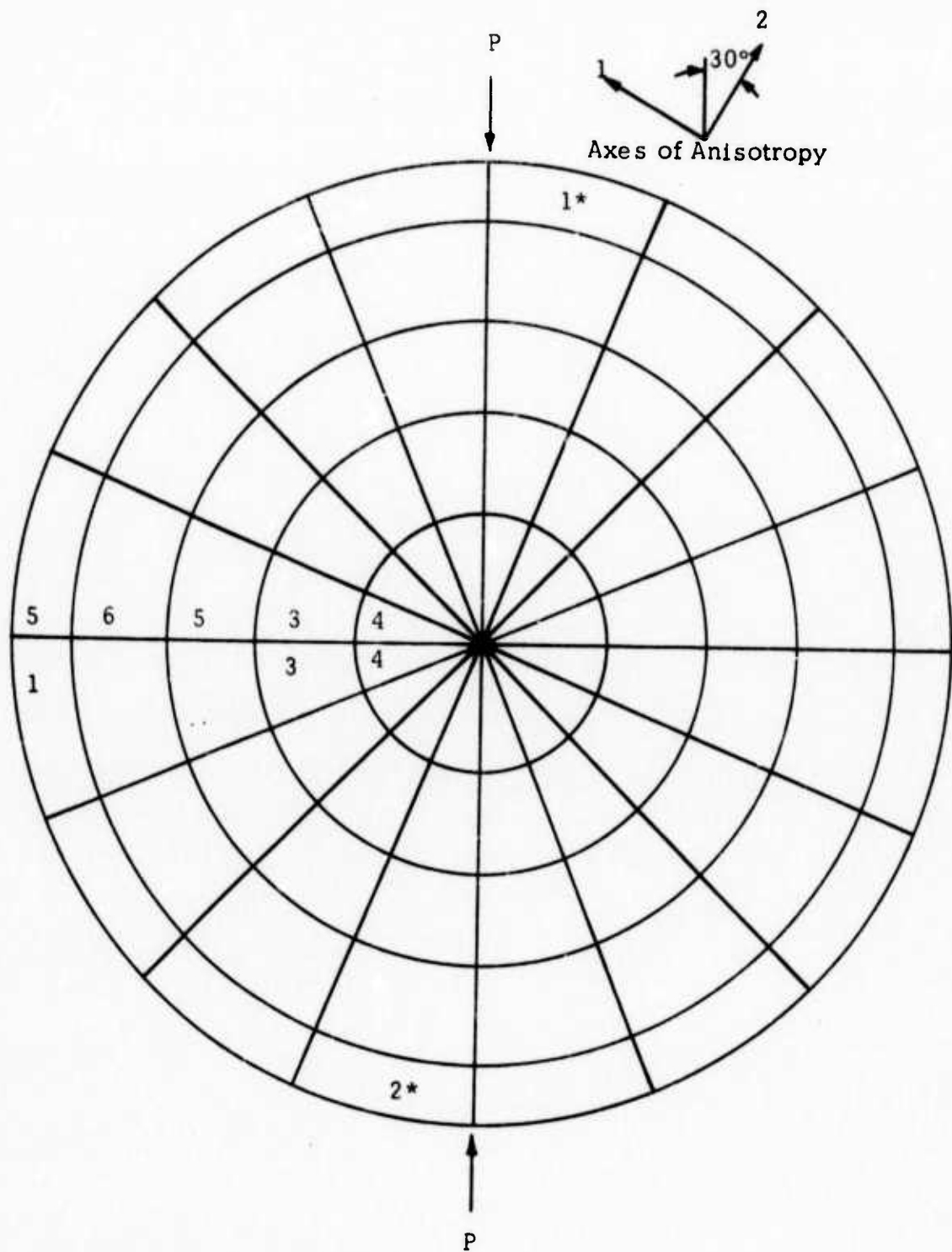
9 circumferential divisions

Stiffness matrices of failed elements are reduced 90%

INPUT DATA LISTING FOR PROBLEM 2.2

Column

| 1 | 5 | 6 | 10 | 11 | 15 | 16 | 20 | 21 | 25 | 26 | 30 | 31 | 35 | 36 | 40 | 41 | 45 | 46 | 50 | 51 | 55 | 56 | 60 | 61 | 65 | 66 | 70 | 71 | Card No. |
|---------|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|---------|----|----|----|----|----------|
| 1 | 5 | 6 | 10 | 11 | 15 | 16 | 20 | 21 | 25 | 26 | 30 | 31 | 35 | 36 | 40 | 41 | 45 | 46 | 50 | 51 | 55 | 56 | 60 | 61 | 65 | 66 | 70 | 71 | 1 |
| .05 | | | | | | | | | | | 2 | | | | 1 | | | | | | | | | | | | | | 2 |
| .07 | | | | | | | | | | | 1 | | | | | | 1 | | | | | | | | | | | | 3 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 4 |
| | | | | | | | | | | | | | | | | | | | | | 1.5 | | | | | | | | 5a |
| | | | | | | | | | | | | | | | | | | | | | 112.5 | | | 135. | | | | | 6a |
| 180. | | | | | | | | | | | | | | | | | | | | | 292.5 | | | 315. | | | | | 6b |
| 30. | | | | | | | | | | | | | | | | | | | | | 2700. | | | 300000. | | | | | 9 |
| 570000. | | | | | | | | | | | | | | | | | | | | | 280000. | | | 280000. | | | | | 10 |



*Failed in shear.
The rest failed in tension.

Figure 2.1 Progression of Failure in Problem 2.2

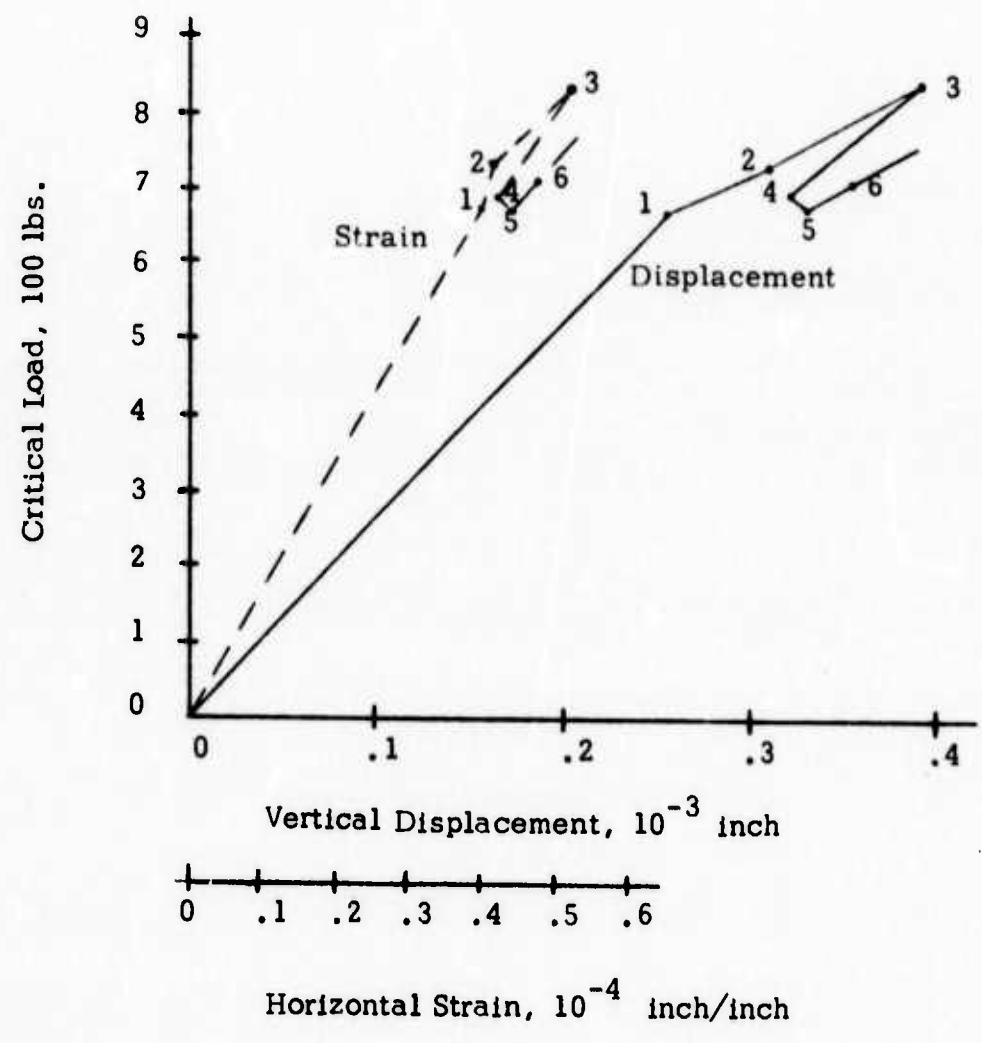
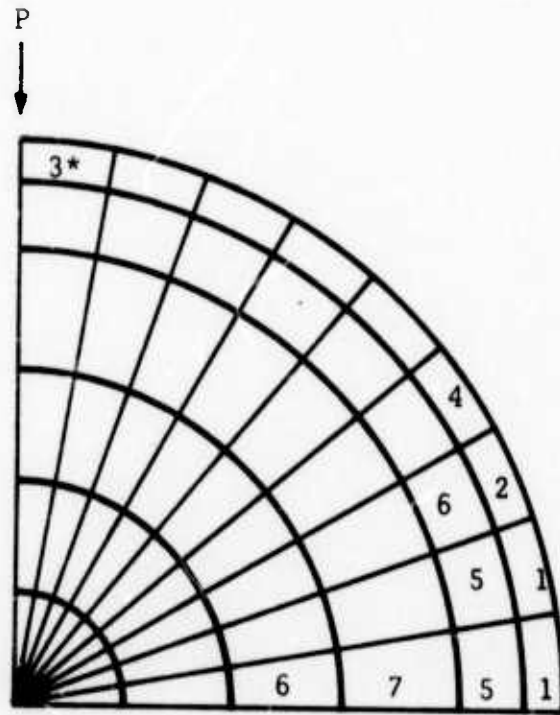


Figure 2.2 Load-Displacement-Strain Curves (Problem 2.2)

INPUT DATA LISTING FOR PROBLEM 2.3

Column

| 1 | 5 | 6 | 10 | 11 | 15 | 16 | 20 | 21 | 25 | 26 | 30 | 31 | 35 | 36 | 40 | 41 | 45 | 46 | 50 | 51 | 60 | 61 | 70 | 71 | Card No. |
|-----|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|
| -1 | | | | | | | | | | | | | | | | | | | | | | | | | 1 |
| .05 | | | | | | | | | | | | | | | | | | | | | | | | | 2 |
| .90 | | | | | | | | | | | | | | | | | | | | | | | | | 3 |
| 10 | | | | | | | | | | | | | | | | | | | | | | | | | 4 |
| 0. | | | | | | | | | | | | | | | | | | | | | | | | | 5a |
| 0. | | | | | | | | | | | | | | | | | | | | | | | | | 6a |
| 80. | | | | | | | | | | | | | | | | | | | | | | | | | 6b |



*Failed in shear. Rest failed in tension.

Figure 2.3. Progression of Failure in Problem 2.3.

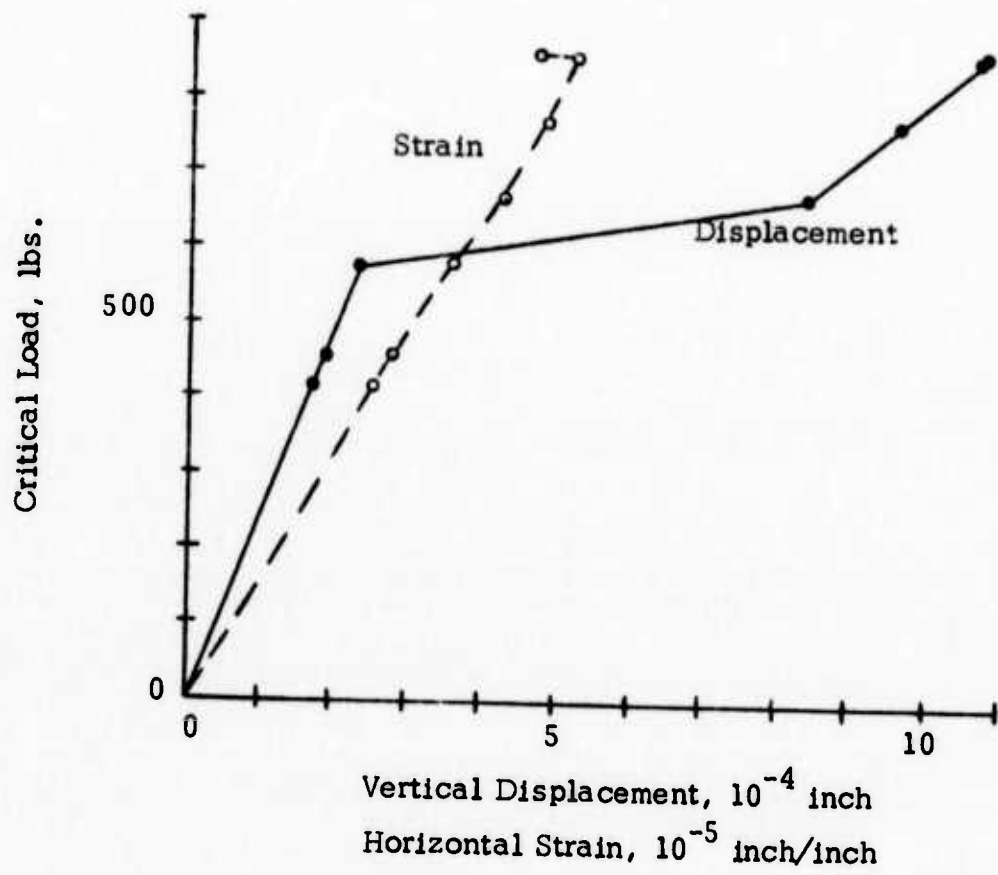


Figure 2.4. Load-Displacement-Strain Curves (Problem 2.3)

2.4 Two-dimensional Isotropic Problem (New Failure Criterion)

Data are the same as Problem 2.3. Stiffness matrices of elements failing in compression or shear are reduced 90%. Elastic moduli across tension cracks are reduced to zero.

2.5 Three-dimensional Isotropic Problem (Old Failure Criterion)

Elastic Modulus = 5,700,000. psi

Allow. Compressive Stress = 27,000. psi

Allow. Tension = .05 of Allow. Compression

Allow. Shear = 9.00 of Allow. Compression

Poisson's Ratio = .25

Finite Element Mesh (Cylinder Quadrant):

5 radial divisions

6 circumferential divisions

2 axial (Z-axis) divisions

Because of symmetry only half the length of cylinder is considered in analysis. Stiffness matrices of failed elements are reduced 90%.

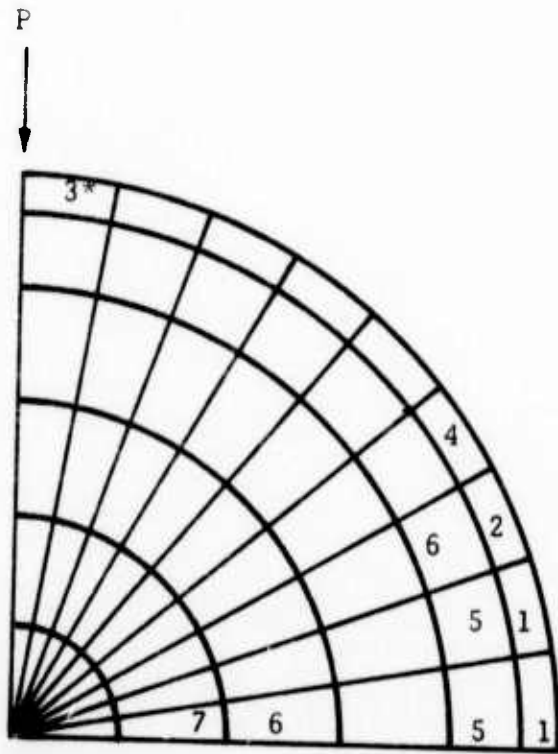
2.6 Three-dimensional Isotropic Problem (New Failure Criterion)

Data are the same as Problem 2.5. Stiffness matrices of elements failing in compression or shear are reduced 90%. Elastic moduli across tension cracks are reduced to zero.

INPUT DATA LISTING FOR PROBLEM 2.4

Column

| | 5 | 6 | 10 | 11 | 15 | 16 | 20 | 21 | 25 | 26 | 30 | 31 | 35 | 36 | 40 | 41 | 45 | 46 | 50 | 51 | 60 | 61 | 70 | 71 | Card No. |
|---|-----|-----|----|----|----|----|----|-----|----------|----|-----|----|----|----|----|----|----|----|----|--------|----|-----|----|-----|----------|
| 1 | 5 | 6 | 10 | 11 | 15 | 16 | 20 | 21 | 25 | 26 | 30 | 31 | 35 | 36 | 40 | 41 | 45 | 46 | 50 | 51 | 60 | 61 | 70 | 71 | |
| | -1 | 6 | | | 0 | | | | | | 2 | 7 | 9 | 0 | | | | | | | | | | | 1 |
| | .05 | .10 | | | | | | | 2700000. | | | | | | | | | | | 27000. | | .25 | | .25 | 2 |
| | .90 | .05 | | | | | | 3. | | | 1. | | | | | | | | | | | | | | 3 |
| | 10 | 20 | 10 | | | | 20 | | | | | | | | | | | | | | | | | | 4 |
| | 0. | .3 | | | | | | .6 | | | .9 | | | | | | | | | 1.4 | | 1.5 | | | 5a |
| | 0. | 10. | | | | | | 20. | | | 30. | | | | | | | | | 50. | | 60. | | 70. | 6a |
| | 80. | 90. | | | | | | | | | | | | | | | | | | | | | | | 6b |



*Failed in shear. Rest failed in tension.

Figure 2.5. Progression of Failure in Problem 2.4.

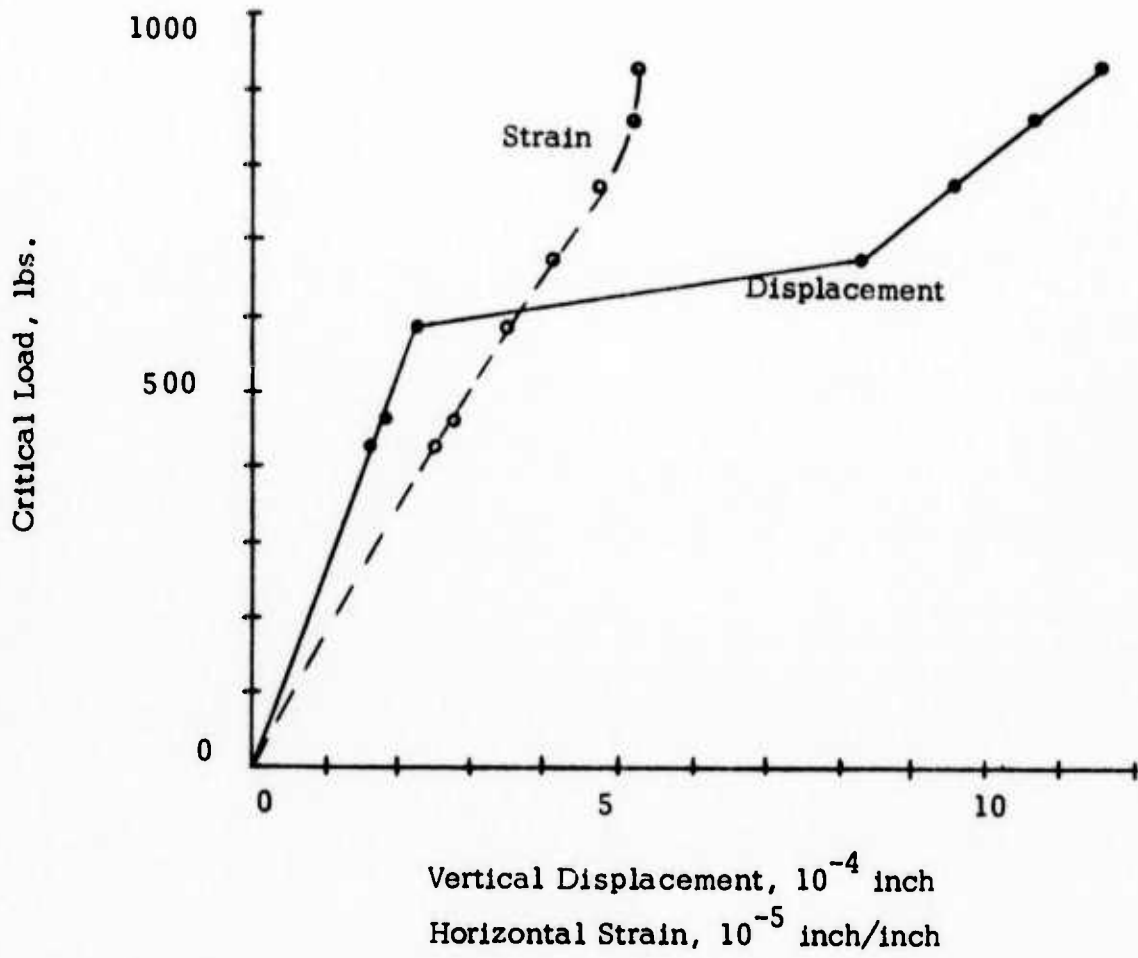
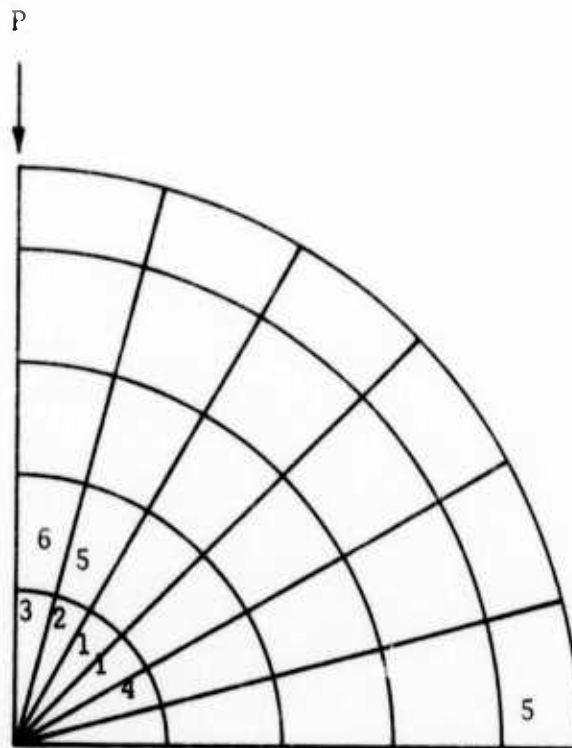


Figure 2.6. Load-Displacement-Strain Curves (Problem 2.4)



Note: All elements failed in tension.

Figure 2.7. Progression of Failure in Problem 2.5.

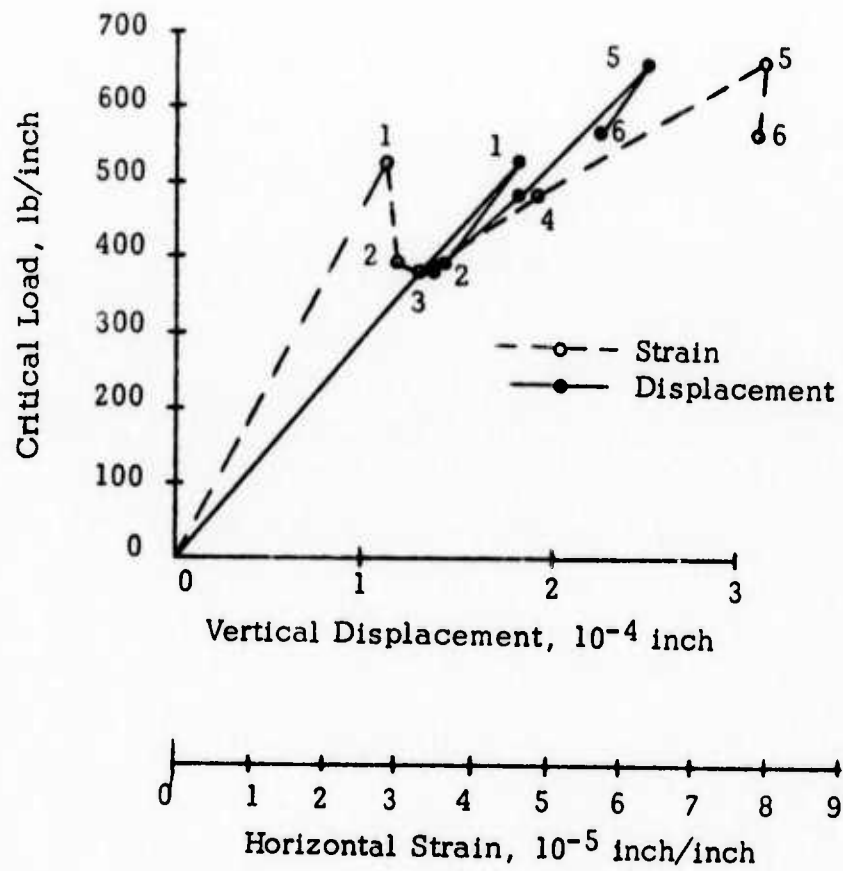
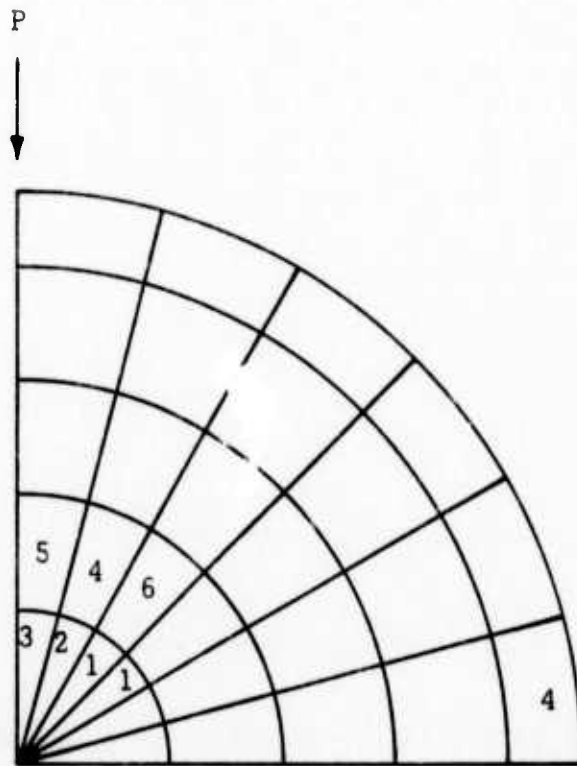


Figure 2.8. Load-Displacement-Strain Curves (Problem 2.5).



Note: All elements failed in tension

Figure 2.9. Progression of Failure in Problem 2.6.

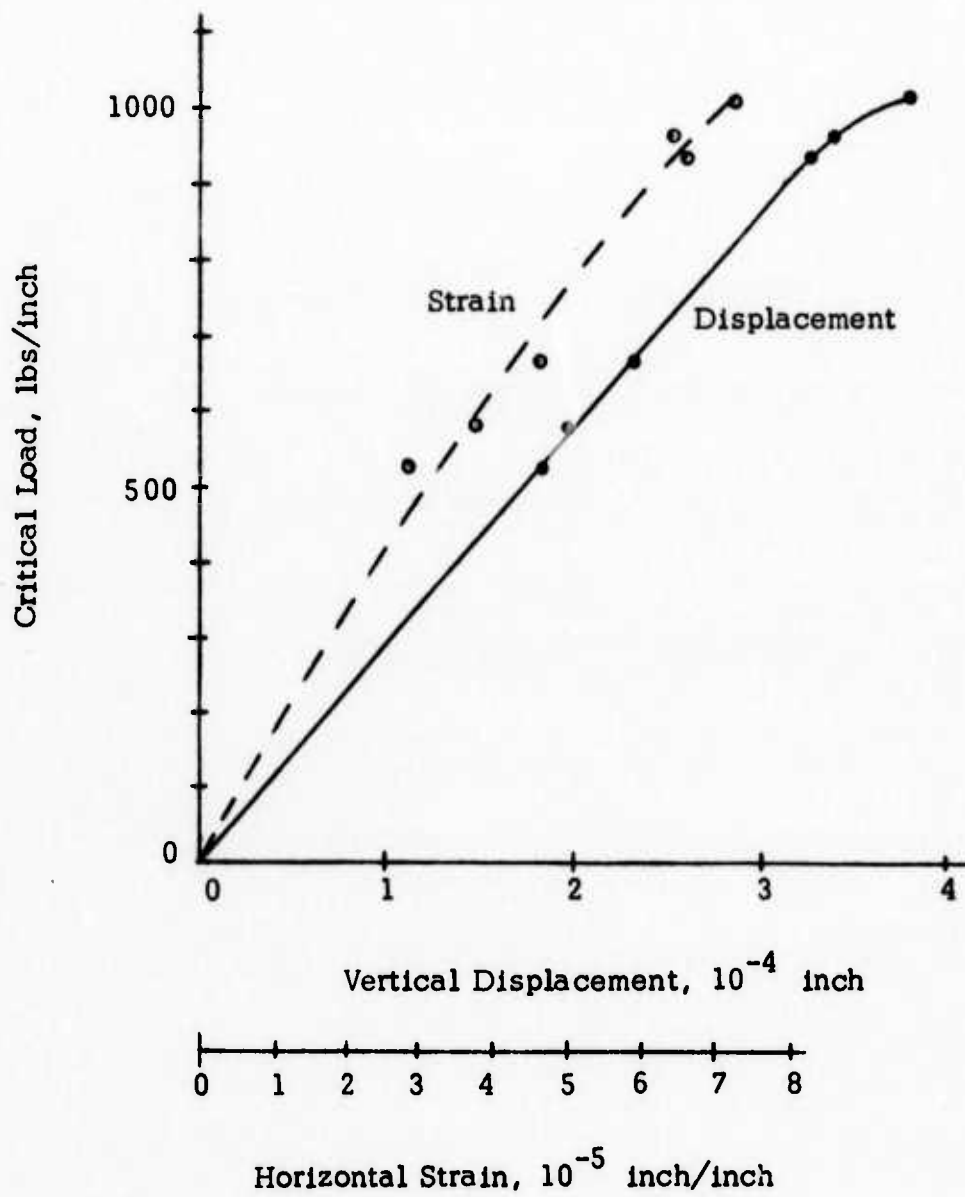


Figure 2.10. Load-Displacement-Strain Curves (Problem 2.6).

2.7 Three-dimensional Anisotropic Problem

Elastic Moduli:

$$E_1 = 3,000,000. \text{ psi}$$

$$E_2 = 5,700,000. \text{ psi}$$

$$E_3 = 5,700,000. \text{ psi}$$

Allow. Compressive Stresses:

$$C_1 = 15,000. \text{ psi}$$

$$C_2 = 27,000. \text{ psi}$$

$$C_3 = 27,000. \text{ psi}$$

Allow. Tension = .05 of Allow. Compression

Allow. Shear = .10 of Allow. Compression

Poisson's Ratios:

$$\mu_1 = .25$$

$$\mu_2 = .23$$

$$\mu_3 = .23$$

Shear Moduli:

$$G_1 = 1,600,000 \text{ psi}$$

$$G_2 = 2,800,000 \text{ psi}$$

$$G_3 = 2,800,000 \text{ psi}$$

Finite Element Mesh (Whole Cylinder):

3 radial divisions

8 circumferential divisions

2 axial (Z-axis) divisions

Directions of anisotropy are indicated in Figure 2.11.

Table 2.1. Nodal Reordering for Problem 2.7.

RESULTS OF NODAL REORDERING

NEW NO. ORIG. NO. NODAL CONNECTIONS (ORIG. NOS.)

| | | | | | | | | | | | |
|----|-----|-----|-----|----|----|----|------|----|----|----|----|
| 1 | 9 | 25 | 33 | 2 | 26 | 10 | 34 | 16 | 40 | 8 | 32 |
| 2 | 11 | 27 | 35 | 2 | 26 | 10 | 34 | 4 | 36 | 12 | 28 |
| 3 | 13 | 29 | 37 | 4 | 36 | 12 | 38 | 30 | 6 | 14 | 38 |
| 4 | 15 | 31 | 39 | 16 | 40 | 8 | 32 | 30 | 6 | 14 | 38 |
| 5 | 9 | 33 | 41 | 18 | 10 | 42 | 34 | 24 | 48 | 16 | 40 |
| 6 | 11 | 35 | 43 | 18 | 10 | 42 | 34 | 44 | 20 | 36 | 12 |
| 7 | 13 | 37 | 45 | 44 | 20 | 36 | 12 | 22 | 14 | 46 | 38 |
| 8 | 15 | 39 | 47 | 24 | 48 | 16 | 40 | 22 | 14 | 46 | 38 |
| 9 | 81 | 49 | 57 | 74 | 82 | 50 | 58 | 80 | 88 | 64 | 56 |
| 10 | 83 | 51 | 59 | 74 | 82 | 50 | 58 | 76 | 84 | 60 | 52 |
| 11 | 85 | 53 | 61 | 76 | 84 | 60 | 52 | 78 | 86 | 62 | 54 |
| 12 | 87 | 55 | 63 | 80 | 88 | 64 | 56 | 78 | 86 | 62 | 54 |
| 13 | 81 | 65 | 57 | 82 | 90 | 66 | 58 | 88 | 96 | 72 | 64 |
| 14 | 83 | 67 | 59 | 82 | 90 | 66 | 58 | 84 | 92 | 68 | 60 |
| 15 | 85 | 69 | 61 | 84 | 92 | 68 | 60 | 86 | 94 | 70 | 62 |
| 16 | 87 | 71 | 63 | 88 | 96 | 72 | 64 | 86 | 94 | 70 | 62 |
| 17 | 25 | 33 | 41 | 2 | 18 | 26 | 10 | 42 | 34 | 24 | 48 |
| 18 | 27 | 35 | 43 | 2 | 18 | 26 | 10 | 42 | 34 | 4 | 44 |
| 19 | 29 | 37 | 45 | 4 | 44 | 20 | 36 | 12 | 28 | 22 | 30 |
| 20 | 31 | 39 | 47 | 24 | 48 | 16 | 40 | 8 | 32 | 22 | 30 |
| 21 | 65 | 49 | 57 | 74 | 82 | 90 | 66 | 50 | 58 | 80 | 88 |
| 22 | 67 | 51 | 59 | 74 | 82 | 90 | 66 | 50 | 58 | 76 | 84 |
| 23 | 69 | 53 | 61 | 76 | 84 | 92 | 68 | 60 | 52 | 78 | 86 |
| 24 | 71 | 55 | 63 | 80 | 88 | 96 | 72 | 64 | 56 | 78 | 86 |
| 25 | 33 | 41 | 49 | 57 | 2 | 18 | 26 | 10 | 50 | 42 | 34 |
| 26 | 35 | 43 | 51 | 59 | 2 | 18 | 26 | 10 | 50 | 42 | 34 |
| 27 | 37 | 45 | 53 | 61 | 4 | 44 | 20 | 36 | 12 | 28 | 30 |
| 28 | 39 | 47 | 55 | 63 | 64 | 56 | 24 | 48 | 16 | 40 | 8 |
| 29 | 33 | 41 | 49 | 57 | 74 | 82 | 90 | 66 | 50 | 42 | 34 |
| 30 | 35 | 43 | 51 | 59 | 74 | 82 | 90 | 66 | 50 | 42 | 34 |
| 31 | 37 | 45 | 53 | 61 | 76 | 84 | 92 | 68 | 60 | 52 | 44 |
| 32 | 39 | 47 | 55 | 63 | 80 | 88 | 96 | 72 | 64 | 56 | 48 |
| 33 | 41 | 49 | 57 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 |
| 34 | 49 | 57 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 | 50 |
| 35 | 16 | 40 | 8 | 32 | 2 | 18 | 26 | 10 | 50 | 42 | 34 |
| 36 | 57 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 | 50 | 42 |
| 37 | 43 | 51 | 59 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 |
| 38 | 51 | 59 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 | 50 |
| 39 | 51 | 59 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 | 50 |
| 40 | 59 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 | 50 | 42 |
| 41 | 45 | 53 | 61 | 4 | 76 | 84 | 92 | 68 | 60 | 52 | 44 |
| 42 | 53 | 61 | 4 | 76 | 84 | 92 | 68 | 60 | 52 | 44 | 20 |
| 43 | 61 | 4 | 76 | 84 | 92 | 68 | 60 | 52 | 44 | 20 | 36 |
| | 41 | 4 | 76 | 84 | 92 | 68 | 60 | 52 | 44 | 20 | 36 |
| | 45 | 53 | 61 | 4 | 76 | 84 | 92 | 68 | 60 | 52 | 44 |
| | 49 | 57 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 | 50 |
| | 53 | 61 | 4 | 76 | 84 | 92 | 68 | 60 | 52 | 44 | 20 |
| | 57 | 2 | 74 | 18 | 82 | 26 | 90 | 10 | 66 | 50 | 42 |
| | 61 | 4 | 76 | 84 | 92 | 68 | 60 | 52 | 44 | 20 | 36 |
| | 65 | 49 | 57 | 74 | 82 | 90 | 66 | 50 | 58 | 80 | 88 |
| | 69 | 53 | 61 | 76 | 84 | 92 | 68 | 60 | 52 | 78 | 86 |
| | 73 | 57 | 65 | 80 | 88 | 96 | 72 | 64 | 56 | 78 | 86 |
| | 77 | 61 | 69 | 84 | 92 | 68 | 60 | 52 | 78 | 86 | 62 |
| | 81 | 65 | 73 | 88 | 96 | 72 | 64 | 56 | 78 | 86 | 62 |
| | 85 | 69 | 77 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 89 | 73 | 81 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 93 | 77 | 85 | 88 | 96 | 72 | 64 | 56 | 78 | 86 | 62 |
| | 97 | 81 | 89 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 101 | 85 | 93 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 105 | 89 | 97 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 109 | 93 | 101 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 113 | 97 | 105 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 117 | 101 | 109 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 121 | 105 | 113 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 125 | 109 | 117 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 129 | 113 | 121 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 133 | 117 | 125 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 137 | 121 | 129 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 141 | 125 | 133 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 145 | 129 | 137 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 149 | 133 | 141 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 153 | 137 | 145 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 157 | 141 | 149 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 161 | 145 | 153 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 165 | 149 | 157 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 169 | 153 | 161 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 173 | 157 | 165 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 177 | 161 | 169 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 181 | 165 | 173 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 185 | 169 | 177 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 189 | 173 | 181 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 193 | 177 | 185 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 197 | 181 | 189 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 201 | 185 | 193 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 205 | 189 | 197 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 209 | 193 | 201 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 213 | 197 | 205 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 217 | 201 | 209 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 221 | 205 | 213 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 225 | 209 | 217 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 229 | 213 | 221 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 233 | 217 | 225 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 237 | 221 | 229 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 241 | 225 | 233 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 245 | 229 | 237 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 249 | 233 | 241 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 253 | 237 | 245 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 257 | 241 | 249 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 261 | 245 | 253 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 265 | 249 | 257 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 269 | 253 | 261 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 273 | 257 | 265 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 277 | 261 | 269 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 281 | 265 | 273 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 285 | 269 | 277 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 289 | 273 | 281 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 293 | 277 | 285 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 297 | 281 | 289 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 301 | 285 | 293 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 305 | 289 | 297 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 309 | 293 | 301 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 313 | 297 | 305 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 317 | 301 | 309 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 321 | 305 | 313 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 325 | 309 | 317 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 329 | 313 | 321 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 333 | 317 | 325 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 337 | 321 | 329 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 341 | 325 | 333 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 345 | 329 | 337 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 349 | 333 | 341 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 353 | 337 | 345 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 357 | 341 | 349 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 361 | 345 | 353 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 365 | 349 | 357 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 369 | 353 | 361 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 373 | 357 | 365 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 377 | 361 | 369 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 381 | 365 | 373 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 385 | 369 | 377 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 389 | 373 | 381 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 393 | 377 | 385 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 397 | 381 | 389 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 401 | 385 | 393 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 405 | 389 | 397 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 409 | 393 | 401 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 413 | 397 | 405 | 92 | 68 | 60 | 52 | 78 | 86 | 62 | 54 |
| | 417 | 401 | 409 | 92 | 68 | 60 | 52</ | | | | |

Table 2.1 Continued.

| | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 44 | 61 | 4 | 76 | 84 | 92 | 68 | 60 | 52 | 44 | 20 | 36 | 12 | 28 | 78 | 22 | 86 | 30 | 94 | 6 | 70 | 14 | 62 | 46 |
| 45 | 39 | 54 | 38 | 63 | 80 | 88 | 96 | 72 | 64 | 56 | 24 | 48 | 16 | 40 | 8 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 |
| 46 | 47 | 14 | 62 | 46 | 54 | 38 | 96 | 72 | 64 | 24 | 48 | 16 | 40 | 8 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 | 14 |
| 47 | 55 | 63 | 80 | 88 | 96 | 72 | 64 | 56 | 24 | 48 | 16 | 40 | 8 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 | 14 | 62 |
| 48 | 63 | 46 | 54 | 38 | 96 | 72 | 64 | 56 | 24 | 48 | 16 | 40 | 8 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 | 14 | 62 |
| 49 | 2 | 54 | 38 | 82 | 72 | 64 | 56 | 24 | 48 | 16 | 40 | 8 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 | 14 | 62 | 46 |
| 50 | 74 | 54 | 18 | 74 | 24 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 |
| 51 | 18 | 52 | 24 | 44 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 |
| 52 | 82 | 24 | 44 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 |
| 53 | 26 | 90 | 10 | 64 | 50 | 42 | 34 | 58 | 4 | 80 | 76 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 |
| 54 | 90 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 |
| 55 | 10 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 |
| 56 | 66 | 50 | 42 | 34 | 58 | 4 | 80 | 76 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 |
| 57 | 50 | 36 | 40 | 12 | 8 | 28 | 32 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 | 16 |
| 58 | 42 | 40 | 12 | 8 | 28 | 32 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 | 16 | 36 |
| 59 | 34 | 58 | 4 | 80 | 76 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 | 16 | 36 | 40 |
| 60 | 58 | 8 | 28 | 32 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 | 16 | 36 | 40 | 12 |
| 61 | 4 | 28 | 32 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 | 16 | 36 | 40 | 12 | 8 |
| 62 | 80 | 32 | 78 | 88 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 | 16 | 36 | 40 | 12 | 8 |
| 63 | 76 | 78 | 22 | 86 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 | 16 | 36 | 40 | 12 | 8 |
| 64 | 88 | 22 | 86 | 84 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 |
| 65 | 84 | 86 | 30 | 94 | 6 | 70 | 14 | 62 | 46 | 54 | 38 | 96 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 48 | 22 |
| 66 | 96 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 78 | 22 | 86 | 30 |
| 67 | 92 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 78 | 22 | 86 | 30 | 94 |
| 68 | 72 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 78 | 22 | 86 | 30 | 94 | 6 |
| 69 | 68 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 |
| 70 | 64 | 60 | 56 | 52 | 24 | 44 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 | 14 |
| 71 | 60 | 56 | 52 | 24 | 44 | 48 | 20 | 16 | 36 | 40 | 12 | 8 | 28 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 | 14 | 62 |
| 72 | 56 | 46 | 54 | 38 | 96 | 72 | 64 | 56 | 24 | 48 | 16 | 40 | 8 | 32 | 78 | 22 | 86 | 30 | 94 | 6 | 70 | 14 | 62 |

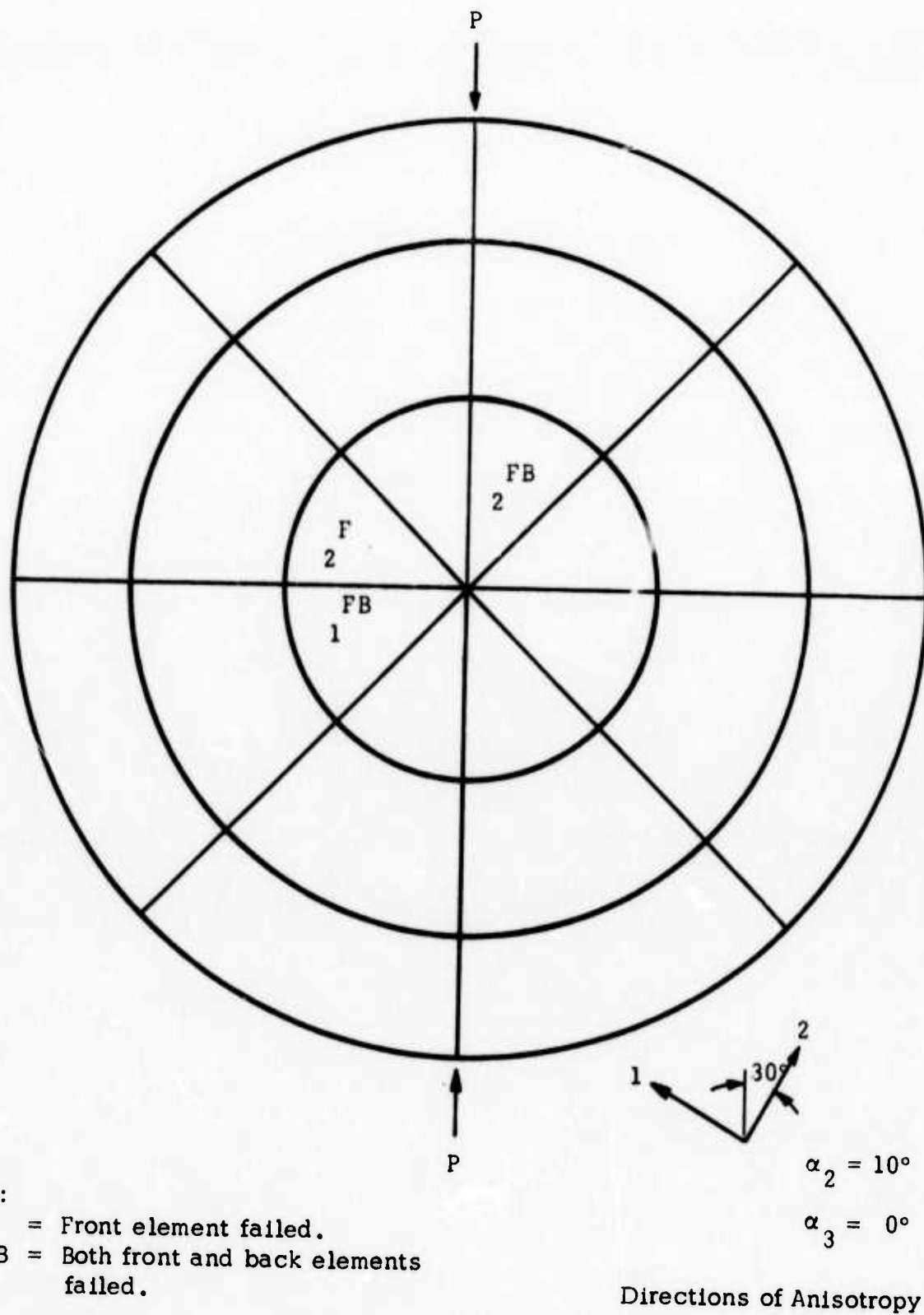


Figure 2.11. Progression of Failure in Problem 2.7.

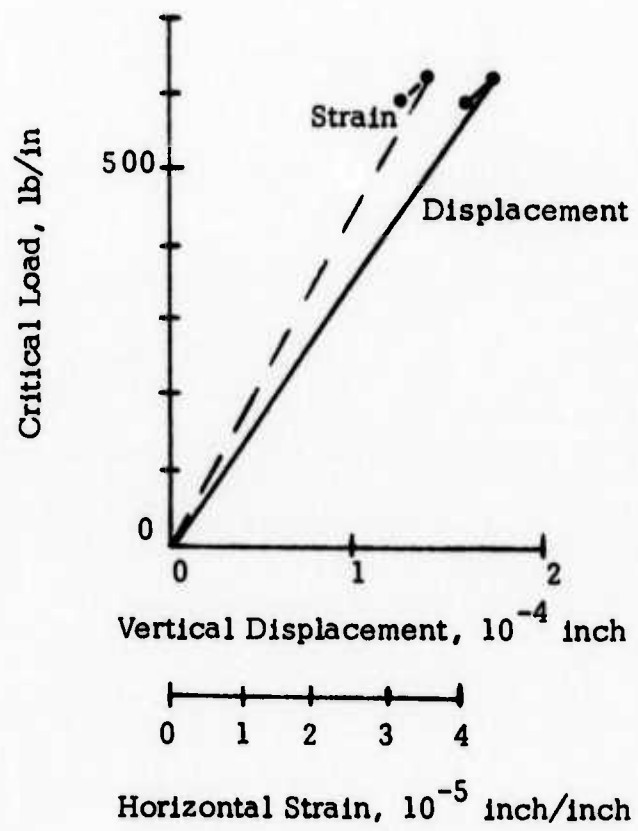


Figure 2.12. Load-Displacement-Strain Curves (Problem 2.7)

2.8 Three-dimensional Non-homogeneous Problem

Elastic Modulus = 5,500,000 to 6,500,000 psi

Allow. Compressive Stress = 22,000 to 32,000 psi

Allow. Tension = .05 of Allow. Compression

Allow. Shear = .10 of Allow. Compression

Poisson's Ratio = .23 to .25

Finite Element Mesh (Whole Cylinder):

3 radial divisions

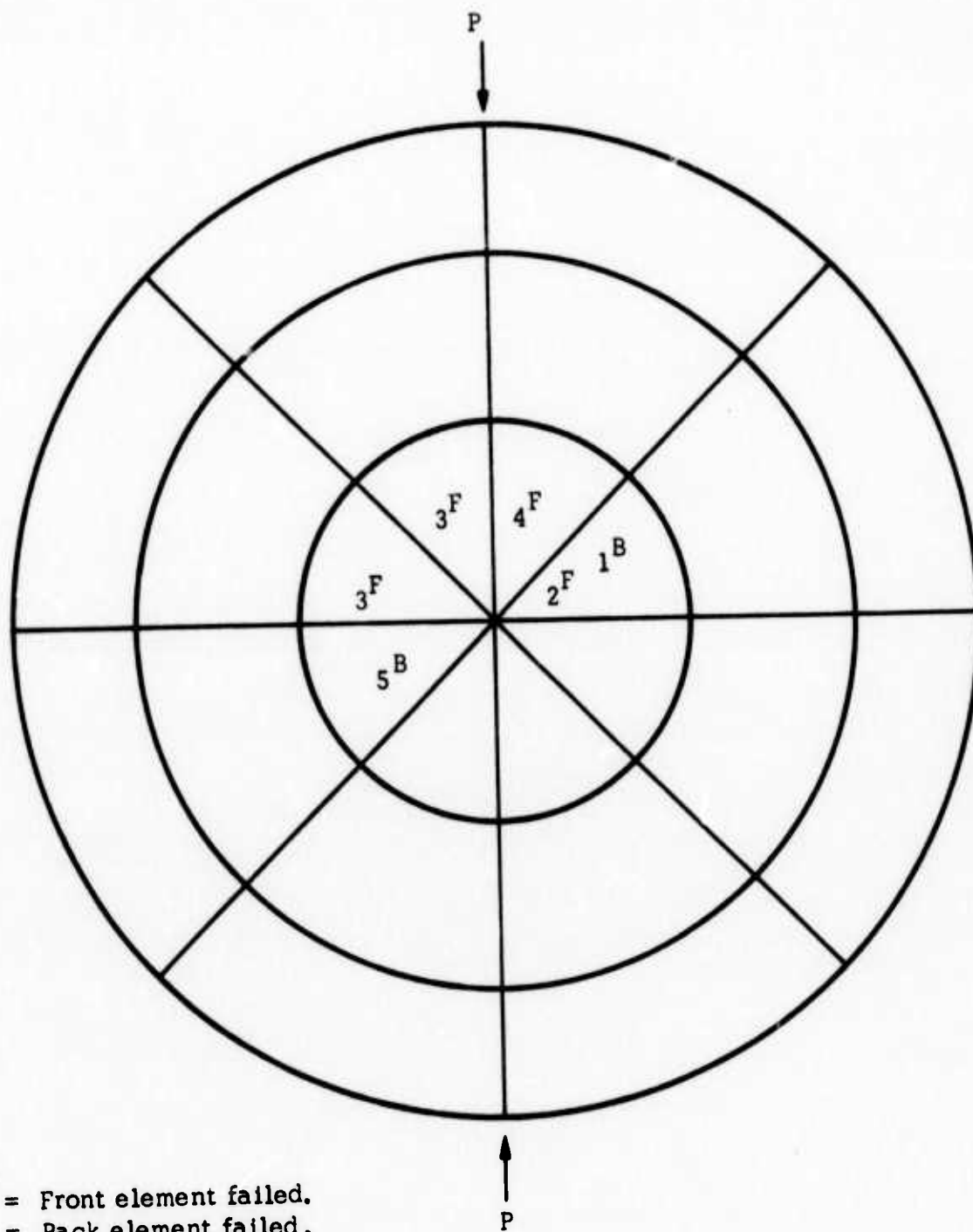
8 circumferential divisions

2 axial divisions

INPUT DATA LISTING FOR PROBLEM 2.8

Column

| 1 | 5 | 6 | 10 | 11 | 15 | 16 | 20 | 21 | 25 | 26 | 30 | 31 | 35 | 36 | 40 | 41 | 45 | 46 | 50 | 51 | 60 | 61 | 70 | 71 | Card No. |
|--------|------|---|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|
| | | | | | | | | | | | | | | | | | | | | | | | | | 1 |
| .05 | | | .10 | | | | | | | | | | | | | | | | | | | | | | 2 |
| .90 | | | .05 | | | | | | | | | | | | | | | | | | | | | | 3 |
| | | | | | | | | | | | | | | | | | | | | | | | | | 4 |
| 0. | | | .6 | | | | | | | | | | | | | | | | | | | | | | 5a |
| 0. | | | .45 | | | | | | | | | | | | | | | | | | | | | | 6a |
| 0. | | | .5 | | | | | | | | | | | | | | | | | | | | | | 7a |
| 395543 | 1329 | | | | | | | | | | | | | | | | | | | | | | | | 8 |



Note:

F = Front element failed.
B = Back element failed.

Figure 2.13. Progression of Failure in Problem 2.8.

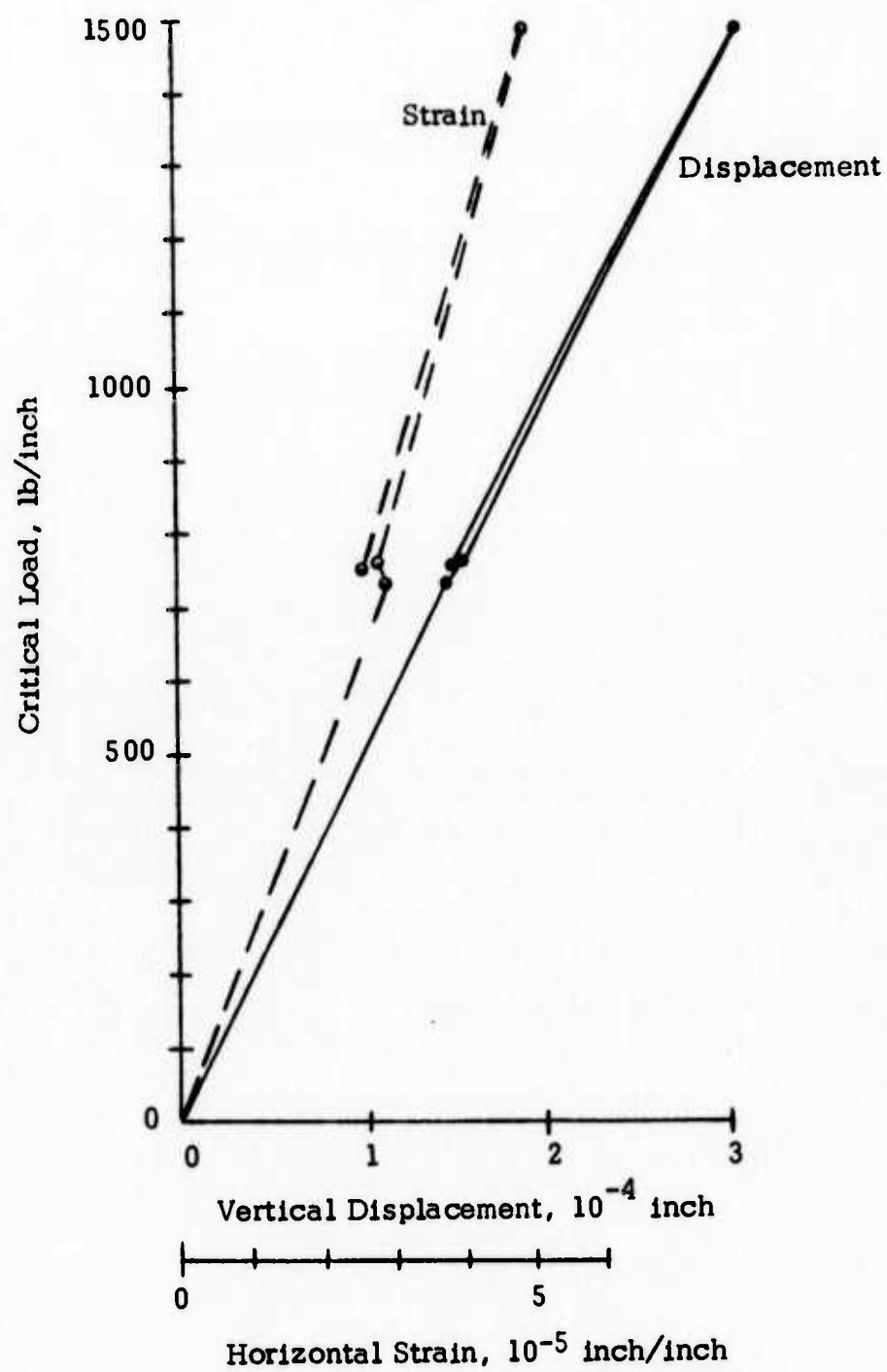


Figure 2.14. Load-Displacement-Strain Curves (Problem 2.8)

2.9 Discussion of Results and Conclusion

It is difficult to verify explicitly the theoretical correctness of the present solution due to lack of an alternative solution to compare the results with. To compensate for this situation, extensive long-hand checks were undertaken on the individual routines that make up the program. The results, on the whole, are fairly reasonable and seem to indicate the correctness of the whole program and the theory upon which it is based.

The unrealistic failure shown in Fig. 2.1 could be attributed to the following:

- 1) Coarseness of the finite element mesh.
- 2) The simplifying assumption used in anisotropic cases (second paragraph of section 2.1) that the critical direction is necessarily along one of the principal stresses.

Item 1) above could also be the cause of the unrealistic failure patterns shown in Figs. 2.3 and 2.5. It should be noted that although the meshes shown in these figures appear to be relatively fine, they are actually approximately six times coarser than the mesh of Problems two and three of the semi-annual report (see page 49 of same report). Unrealistic failure patterns were also observed during the debugging of the previous two-dimensional program when coarse meshes were used.

Problems 2.3 through 2.6 were run to determine the relative merits of the two failure criteria used in the program. There is no significant difference between the two as far as the failure patterns are concerned. The displacement and strain curves, however, differ considerably particularly in the coarser three-dimensional mesh (see Figures 2.8 and 2.10). The new failure criterion seems to yield consistently definable curves while the old criterion does not. Thus although the failure patterns are unrealistic, the plots show a much better relationship between load and strain and load and displacement.

Table 2.1 shows the results of the nodal reordering carried out in connection with Problem 2.7. It took no more than 30 seconds to complete the whole reordering process.

Table 2.2 gives some information concerning the sizes of the test problems described in this chapter. The maximum number of nodes and non-zero elements of the stiffness matrix that the program can handle are indicated in Table 2.3. These limits may still be increased depending on the computer used.

All mathematical operations in the program are carried out in single precision. It may be necessary to resort to double precision to reduce rounding-off errors.

Table 2.2 Test Problems Comparison

| Problem | Number of Nodes | Dimension | Execution Time (min.) | Number of Load Cycles | % of Non-zero Elements |
|---------|-----------------|-----------|-----------------------|-----------------------|------------------------|
| 2.2 | 96 | 2 | 3.10 | 6 | 12.27 |
| 2.3 | 70 | 2 | 1.55 | 7 | 11.20 |
| 2.4 | 70 | 2 | 1.55 | 7 | 11.20 |
| 2.5 | 84 | 3 | 5.60 | 6 | 15.56 |
| 2.6 | 84 | 3 | 5.91 | 6 | 15.56 |
| 2.7 | 96 | 3 | 9.00 | 2 | 21.09 |
| 2.8 | 96 | 3 | 15.00 | 5 | 21.09 |

Table 2.3 Program Limits

| | 2-Dim | 3-Dim |
|--|--------|--------|
| No. of Nodes | 500 | 250 |
| No. of Finite Elements | 480 | 480 |
| No. of Non-zero Elements in Stiffness Matrix | 25,000 | 25,000 |

APPENDIX A
PROGRAM LISTING

The program listed in the following pages is written in Fortran V language for use specifically in the UNIVAC 1108 Computer located at the University of Wisconsin - Madison campus. It requires six external storage units (either tape or drum) which are assigned the numbers 10, 11, 12, 13, 14, and 15. The contents of the files written in units 10 and 12 continually undergo changes during a program run making it necessary for these particular units to possess random access capabilities.

In addition to the subroutines listed, two other subroutines, RANUN(R) and RANUNS(N), are called in the program during execution of non-homogeneous problems. RANUN(R) generates pseudo-random numbers assuming a statistically uniform distribution while RANUNS(N) sets to N the starting point of the random number generator; N is supplied by the user. Both subroutines were developed by the Madison Academic Computing Center (MACC) and are part of the MACC Random Number Routine package.

Except for the replacement of RANUN(R) and RANUNS(N), there should be no difficulty in making this program adaptable to other computer systems.

```

1.      C      MAIN PROGRAM
2.      C      .....
3.      COMMON/FUNKY/NBRC(500),NREQ(500),NOER(500)
4.      COMMON/RUTH/LOC(25000),NTNN,NB,NX,MINUM
5.      COMMON/JNFER/NR,NS,NZ,NWOP,NDIM
6.      COMMON/TANA1/RO,RS,TO,TS,ZS,NY,HNB
7.      COMMON/LEYTE/RC(30),TC(50),ZC(30)
8.      COMMON/BGAND/C(8,8),B(6,24),H
9.      COMMON/SANNA/PD(480,3),PMAT(3,8)
10.     COMMON/ABALN/D(2,600)
11.     COMMON/NANAY/CR(3,3),NW
12.     COMMON/JANET/NSPD,NWSD(100),SPD(100,3)
13.     COMMON/BORAP/NLOD(100),CNLD(100),NE,NER
14.     COMMON/TATAY/ISO,ALF1,ALF2,ALF3,E1,E2,E3,P1,P2,P3,
15.     1      G1,G2,G3
16.     DIMENSION BRC(500),CRT(3,3),CSN(3,3),LA(20),
17.     1LAC(25000),NM(3),NOD(8),P(3),PT(3),S(3,300),
18.     2SL(24,24),SLD(24,24),STRS(6),VV(3),WW(3)
19.     EQUIVALENCE (D(1,301),S),(LOC,LAC)
20.     DATA VV/'COMPRES', 'TENSION', 'SHEAR' /
21.     DATA WW/'SSION', 'N', 'NG' /
22.     DATA (C(1,1),I=1,8)/1.,1.,1.,1.,1.,1.,1.,1./
23.     DATA (C(2,1),I=1,8)/=1.,=1.,=1.,=1.,=1.,=1.,=1.,=1./
24.     DATA (C(3,1),I=1,8)/=1.,=1.,=1.,=1.,=1.,=1.,=1.,=1./
25.     DATA (C(4,1),I=1,8)/1.,=1.,=1.,=1.,=1.,=1.,=1.,=1./
26.     DATA (C(5,1),I=1,8)/=1.,=1.,=1.,=1.,=1.,=1.,=1.,=1./
27.     DATA (C(6,1),I=1,8)/1.,1.,=1.,=1.,=1.,=1.,=1.,=1./
28.     DATA (C(7,1),I=1,8)/1.,=1.,=1.,=1.,=1.,=1.,=1.,=1./
29.     DATA (C(8,1),I=1,8)/=1.,=1.,=1.,=1.,=1.,=1.,=1.,=1./
30.     C      READ IN DATA
31.     C      ISO = -1 , HOMOGENEOUS AND ISOTROPIC
32.     C      ISO = 0 , NONHOMOGENEOUS
33.     C      ISO = 1 , HOMOGENEOUS AND ANISOTROPIC
34.     C      NWOP = 0 , PORTION OF DISC IS ANALYZED
35.     C      NWOP = 1 , WHOLE DISC IS ANALYZED
36.     C      NZ=0 FOR 2-DIMENSIONAL PROBLEMS
37.     READ 100,ISO,NR,NS,NZ,NWOP,NDIM,NCYC,NER,NSS
38.     READ 102,TCRT,SCRT,EH1,EH2,CS1,CS2,PR1,PR2
39.     READ 104,POSH,TRN,DIA,H,NTEP,NGOT
40.     READ 100,NST1,NST2,HST1,HST2
41.     LRM=ISO
42.     NIDA=NDIM-1
43.     INGA=(NDIM-1)*50
44.     INGO=500/(NDIM-1)
45.     NR1=NR+1
46.     NT=NS-NWOP
47.     NS1=NT+1
48.     NZ1=NZ+1
49.     READ 102,(RC(I),I=1,NR1)
50.     READ 102,(TC(I),I=1,NS1)
51.     IF (NDIM.EQ.2)GO TO 3
52.     READ 102,(ZC(I),I=1,NZ1)
53.     3 PRINT 300
54.     NTNN=(NT+1)*(NZ+1)*(NR+1)
55.     NELEM=NR*NS

```

```

56.      IF (NDIM .EQ. 3) NELEM=NELEM*NZ
57.      IF (ISO) 4,6,8
58.      4 DO 5 I=1,NELEM
59.        PD(I,1)=EM1
60.        PD(I,2)=PR1
61.      5 PD(I,3)=CS1
62.        PRINT 301,EM1
63.        PRINT 302,PR1
64.        PRINT 303,CS1
65.        GO TO 9
66.      6 READ 106,N
67.        CALL RANUNS(N)
68.        DO 7 I=1,NELEM
69.          CALL UNIPRM(EM1,EM2,EMS)
70.          CALL UNIPRM(PR1,PR2,PRS)
71.          CALL UNIPRM(CS1,CS2,CSS)
72.          PD(I,1)=EMS
73.          PD(I,2)=PRS
74.      7 PD(I,3)=CSS
75.        PRINT 306,EM1,EM2
76.        PRINT 307,PR1,PR2
77.        PRINT 308,CS1,CS2
78.        GO TO 9
79.      8 READ 102,ALF1,ALF2,ALF3,C1,C2,C3,E1,E2
80.        READ 102,E3,P1,P2,P3,G1,G2,G3
81.        PRINT 326,ALF1,ALF2,ALF3
82.        PALO=C1*C2
83.        PALO=PALO
84.        ALF1=ALF1*.0174532925
85.        ALF2=ALF2*.0174532925
86.        ALF3=ALF3*.0174532925
87.        CS2=COS(ALF2)
88.        SN2=SIN(ALF2)
89.        CS3=COS(ALF3)
90.        SN3=SIN(ALF3)
91.        IF (NDIM .EQ. 2) GO TO 2
92.        PALO=PALO*C3
93.        CRT(3,1)=SN2
94.        CRT(3,2)=CS2*SN3
95.        CRT(3,3)=CS2*CS3
96.      2 PRINT 309,E1,E2,E3
97.        PRINT 310,P1,P2,P3
98.        PRINT 311,G1,G2,G3
99.        PRINT 312,C1,C2,C3
100.     9 NB=4*(NDIM-1)
101.       NW=0
102.       HNB=NB
103.       NX=NB*NDIM
104.       NY=(NDIM-1)*3
105.       DO 10 I=1,NY
106.         DO 10 J=1,NX
107.       10 B(I,J)=0.
108.         DO 14 I=1,3
109.         DO 14 J=1,8
110.       14 PHAT(I,J)=0.
111.       PRINT 304,TCRT
112.       PRINT 305,SCRT

```

```

113.      PRINT 313
114.      PRINT 314,DIA
115.      PRINT 315,H
116.      PRINT 319,NELEM
117.      PRINT 320,NTNN
118.      PRINT 316,(RC(I),I=1,NR1)
119.      PRINT 317,(TC(I),I=1,NS1)
120.      DO 15 I=2,NS1
121.      15 TC(I)=TC(I)*.0174532925
122.      IF (NDIM .EQ. 2)GO TO 16
123.      PRINT 318,(ZC(I),I=1,NZ1)
124.      16 CALL NDTND(LAC,INGA,INGO)
125.      CALL REORD(LAC,INGA,INGO)
126.      IF (MINUM .GT. INGA)GO TO 433
127.      C   GENERATE EXTERNALLY-LOADED NODES (NLOD) AND
128.      C   MAGNITUDES OF LOADS (CNLD)
129.      IF (NSS .EQ. 0)NSS=NS/2
130.      NE=NZ+1
131.      ND=(NT+1)*NE*NR
132.      CLLD=-500.
133.      IF (NWOP .EQ. 1)CLLD=-1000.
134.      CRES=0.
135.      NF=-NT
136.      IF (NZ .EQ. 0)GO TO 19
137.      DO 18 I=1,NZ
138.      NF=NF+NT+1
139.      NLOD(I)=NF*ND
140.      CDUM=.5*(ZC(I+1)-ZC(I))*CLLD
141.      CNLD(I)=CDUM+CRES
142.      CRES=CDUM
143.      18 CONTINUE
144.      CNLD(NE)=CRES
145.      19 IF (NZ .EQ. 0)CNLD(1)=CLLD
146.      NLOD(NE)=ND*NF+NT+1
147.      IF (NWOP .NE. 1)GO TO 22
148.      DO 20 I=1,NE
149.      NLOD(I+NE)=NLOD(I)+NSS
150.      20 CNLD(I+NE)=CNLD(I)
151.      NE=NE+2
152.      22 NLOD(NE+1)=0
153.      NE1=NE-1
154.      IF (NE1 .EQ. 0)GO TO 25
155.      DO 24 I=1,NE1
156.      NLI=NLOD(I)
157.      I1=I+1
158.      DO 24 J=I1,NE
159.      NLJ=NLOD(J)
160.      IF (NOER(NLJ) .GT. NOER(NLI))GO TO 24
161.      T1=NLOD(I)
162.      T2=CNLD(I)
163.      NLOD(I)=NLOD(J)
164.      CNLD(I)=CNLD(J)
165.      NLOD(J)=T1
166.      CNLD(J)=T2
167.      NLI=NLOD(I)
168.      24 CONTINUE
169.      25 PRINT 200,(NLOD(I),I=1,NE)

```

```

170.      NAY=(NT+1)*(NZ+1)
171.      C      GENERATE BOUNDARY NODES (NWS) AND MAGNITUDES
172.      C      OF SPECIFIED DISPLACEMENTS (SPD)
173.      IF (NWOP .EQ. 1) GO TO 30
174.      MD=(NT+1)*(NZ+1)
175.      DO 26 I=1,MD
176.      NWS(I)=1
177.      26 SPD(I,1)=0.
178.      MR=NR*(NZ+1)
179.      MF=MD+MR
180.      NDR=MD-NT
181.      DO 27 I=1,MR
182.      NWS(MD+I)=NDR+NT+1
183.      NDR=NWS(MD+I)
184.      NWS(MF+I)=NDR+NS
185.      SPD(MD+I,1)=200.
186.      27 SPD(MF+I,1)=200.
187.      NSPD=MF+MR
188.      DO 28 I=1,NSPD
189.      SPD(I,2)=0.
190.      28 SPD(I,3)=200.
191.      IF (NGOT .EQ. 1 .OR. NDIM .EQ. 2) GO TO 85
192.      DO 81 J=1,NSPD
193.      I=NWS(J)
194.      J1=(I-1)/NAY
195.      J2=NAY+J1+1
196.      J3=(I-J2)/(NT+1)
197.      IF (J3 .EQ. 0) SPD(J,3)=0.
198.      81 CONTINUE
199.      DO 83 I=1,NTNN
200.      J1=(I-1)/NAY
201.      J2=NAY+J1+1
202.      J3=(I-J2)/(NT+1)
203.      J4=I-J2-J3*(NT+1)
204.      IF (J3 .NE. 0) GO TO 83
205.      IF (J4 .EQ. 0 .OR. J4 .EQ. NT) GO TO 83
206.      IF (J1 .EQ. 0) GO TO 83
207.      NSPD=NSPD+1
208.      NWS(NSPD)=I
209.      SPD(NSPD,1)=200.
210.      SPD(NSPD,2)=200.
211.      SPD(NSPD,3)=0.
212.      83 CONTINUE
213.      GO TO 85
214.      30 NSPD=NE
215.      DO 32 J=1,NSPD
216.      NWS(J)=NL0D(J)
217.      SPD(J,1)=200.
218.      SPD(J,2)=0.
219.      32 SPD(J,3)=200.
220.      IF (NGOT .EQ. 1 .OR. NDIM .EQ. 2) GO TO 34
221.      DO 82 J=1,NSPD
222.      I=NWS(J)
223.      J1=(I-1)/NAY
224.      J2=NAY+J1+1
225.      J3=(I-J2)/(NT+1)
226.      IF (J3 .EQ. 0) SPD(J,3)=0.

```

```

227.      82 CONTINUE
228.      DO 84 I=1,NTNN
229.      J1=(1-I)/NAY
230.      J2=NAY*J1+1
231.      J3=(1-J2)/(NT+1)
232.      J4=1-J2-J3*(NT+1)
233.      IF (J3 .NE. 0)GO TO 84
234.      IF (J1 .EQ. NR .AND. J4 .EQ. 0)GO TO 84
235.      IF (J1 .EQ. NR .AND. J4 .EQ. NSS)GO TO 84
236.      NSPD=NSPD+1
237.      NWSD(NSPD)=1
238.      SPD(NSPD,1)=200.
239.      SPD(NSPD,2)=200.
240.      SPD(NSPD,3)=0.
241.      84 CONTINUE
242.      85 NSI=NSPD-1
243.      DO 29 I=1,NSI
244.      NWI=NWSD(I)
245.      I1=I+1
246.      DO 29 J=I1,NSPD
247.      NWJ=NWSD(J)
248.      IF (NOER(NWJ) .GT. NOER(NWI))GO TO 29
249.      T1=NWSD(I)
250.      T2=SPD(I,1)
251.      T3=SPD(I,2)
252.      T4=SPD(I,3)
253.      NWSD(I)=NWSD(J)
254.      SPD(I,1)=SPD(J,1)
255.      SPD(I,2)=SPD(J,2)
256.      SPD(I,3)=SPD(J,3)
257.      NWSD(J)=T1
258.      SPD(J,1)=T2
259.      SPD(J,2)=T3
260.      SPD(J,3)=T4
261.      NWI=NWSD(I)
262.      29 CONTINUE
263.      34 PRINT 201,(NWSD(I),I=1,NSPD)
264.      C FORM STIFFNESS MATRICES OF TYPICAL ELEMENTS AND
265.      C STORE IN AUXILIARY UNITS 13,14,15
266.      IF (NER .EQ. 1)GO TO 40
267.      MINUM=NELEM/NER/3
268.      IF (NELEM .GT. MINUM*NER*3)MINUM=MINUM*1
269.      MINAM=MINUM
270.      JILL=0
271.      DO 38 I=1,NELEM,NER
272.      JILL=JILL+1
273.      IDRUM=(JILL-1)/MINUM+13
274.      CALL NQDES(NDIM,I,NOD)
275.      NDI=NOD(I)
276.      NDF=NOD(NB=1)
277.      CALL INTEG(NDIM,I,NDI,NDF,SL)
278.      WRITE (IDRUM)((SL(L1,L2),L1=1,NX),L2=1,NX)
279.      PRINT 400,1,((SL(L1,L2),L1=1,8),L2=1,8)
280.      38 CONTINUE
281.      400 FORMAT (25HSTIFF. MATRIX OF ELEMENT,14/(8E11.5))
282.      C BUILD UP GLOBAL STIFFNESS MATRIX AND SOLVE FOR
283.      C NODAL DISPLACEMENTS

```



```

284.      40 CALL FACTOR
285.      DO 41 I=1,NELEM
286.      41 BRC(I)=1.
287.      DO 80 M=1,NCYC
288.      REWIND 12
289.      MX=0
290.      PRINT 202,M
291.      CRLF=0.
292.      MINUM=NTNN
293.      DO 43 I=1,NELEM
294.      IF (BRC(I) .LE. 0.)GO TO 43
295.      CALL NODES(NDIM,1,NOD)
296.      NOI=NOD(I)
297.      NDF=NOD(NB-1)
298.      CALL COORD3(NDI,RDI,TTI,ZZI)
299.      CALL COORD3(NDF,RDF,TTY,ZZF)
300.      RD=.5*(RDF+RDI)
301.      RS=.5*(RDF-RDI)
302.      ZS=.5*(ZZF+ZZI)
303.      IF (TTY .LT. TTI)TTY=6.283185307
304.      TO=.5*(TTY+TTI)
305.      TS=.5*(TTY-TTI)
306.      CALL STRESS(1,NOD,STRS)
307.      CALL PSTRES(NDIM,LRM,STRS,P)
308.      IF (LRM .NE. 1)GO TO 31
309.      A1=ALF1*TO
310.      CRT(1,1)=CS2*SIN(A1)
311.      CRT(2,1)=-CS2*COS(A1)
312.      CRT(1,2)=CS3*COS(A1)+SN3*SN2*SIN(A1)
313.      CRT(2,2)=CS3*SIN(A1)-SN3*SN2*COS(A1)
314.      IF (NDIM .EQ. 2)GO TO 11
315.      CRT(1,3)=CS3*SN2*SIN(A1)+SN3*COS(A1)
316.      CRT(2,3)=-CS3*SN2*COS(A1)+SN3*SIN(A1)
317.      11 DO 23 IA=1,NDIM,NIDA
318.      DO 23 IB=1,NDIM
319.      DUM=0.
320.      DO 21 K=1,NDIM
321.      21 DUM=DUM+CR(K,IA)*CRT(K,IB)
322.      CSN(IA,IB)=DUM*DUM
323.      23 CONTINUE
324.      POL1=CSN(NDIM,1)*C2*C2+CSN(NDIM,2)*C1*C1
325.      POL2=CSN(1,1)*C2*C2+CSN(1,2)*C1*C1
326.      IF (NDIM .EQ. 2)GO TO 12
327.      POL1=POL1+C3*C3+CSN(NDIM,3)*PAL*PAL
328.      POL2=POL2+C3*C3+CSN(1,3)*PAL*PAL
329.      12 PD(1,3)=PALO/SQRT(POL2)
330.      PD(1,1)=PALO/SQRT(POL1)
331.      CCCT=PD(1,1)*TCRT
332.      CCCS=SQRT*(PD(1,3)+PD(1,1))
333.      GO TO 33
334.      31 CCCT=PD(1,3)*TCRT
335.      CCCS=2.*PD(1,3)*SQRT
336.      33 RT1=P(1)/PD(1,3)
337.      RT2=P(NDIM)/CCCT
338.      RT3=(P(NDIM)+P(1))/CCCS
339.      BRC(I)=AMAX1(RT1,RT2,RT3)
340.      IF (BRC(I) .GT. CRLF)CRLF=BRC(I)

```



```

341.      43 CONTINUE
342. C      DETERMINE ELEMENTS THAT FAILED, LA(I)
343.      I1=0
344.      DO 44 I=1,NELEM
345.      IF (BRC(I) .LE. 0.)GO TO 44
346.      IF (BRC(I)/CRLF .LT. (1.-TRN))GO TO 44
347.      I1=I1+1
348.      LA(I1)=I
349.      BRC(I)=-1.
350.      44 CONTINUE
351. C      CRITICAL LOAD
352.      CRL=1000./CRLF
353.      PRINT 199,CRL
354.      N1=NTNN-NOER(NST1)+1
355.      N2=NTNN-NOER(NST2)+1
356.      M1=NTNN-NOER(MST1)+1
357.      M2=NTNN-NOER(MST2)+1
358.      TEMP=(RC(2)-RC(1))*CRLF
359.      STRN=(O(1,N2)-O(1,N1))/TEMP
360.      STRM=(O(1,M2)-O(1,M1))/TEMP
361.      AVE=.5*(STRN+STRM)
362.      PRINT 203
363.      PRINT 204,STRN
364.      PRINT 205,STRM
365.      PRINT 206,AVE
366.      REWIND 13
367.      REWIND 14
368.      REWIND 15
369.      NM(1)=G
370.      NM(2)=0
371.      NM(3)=0
372.      PRINT 207
373.      NLF=(NE+1)/2
374.      DO 42 I=1,NLF
375.      NLO=NLOD(I)
376.      NUN=NTNN-NOER(NLO)+1
377.      CNL1=CNLD(I)*CRL/CLLO
378.      OZ1=D(1,NUN)*CRL/1000.
379.      IF (NE .EQ. 1)GO TO 45
380.      J=I+NLF
381.      IF (J .GT. NE)GO TO 45
382.      NLA=NLOD(J)
383.      NAN=NTNN-NOER(NLA)+1
384.      CNL2=CNLD(J)*CRL/CLLD
385.      DZ2=D(1,NAN)*CRL/1000.
386.      PRINT 209,NLO,CNL1,DZ1,NLA,CNL2,DZ2
387.      GO TO 42
388.      45 PRINT 209,NLO,CNL1,DZ1
389.      42 CONTINUE
390.      PRINT 211
391.      IF (NDIM .EQ. 2)PRINT 214
392.      IF (NDIM .EQ. 3)PRINT 212
393. C      REVISE AND THEN DEDUCT STIFFNESS OF FAILED ELEMENTS
394.      DO 79 I=1,I1
395. C      SHEAR FAILURE
396.      ICODE=3
397.      LDA=LA(I)

```

```

398.      CALL NODES (NDIM,LDA,NOD)
399.      NDI=NOD(1)
400.      NDF=NOD(NB+1)
401.      CALL COORD3(NDI,RDI,TTI,ZZI)
402.      CALL COORD3(NDF,RDF,TF,ZZF)
403.      RO=.5*(RDF+RDI)
404.      RS=.5*(RDF-RDI)
405.      ZS=.5*(ZZF-ZZI)
406.      IF (TF .LT. TTI) TTF=6.283185307
407.      TO=.5*(TTF+TTI)
408.      TS=.5*(TTF-TTI)
409.      CALL STRESS(LDA,NOD,STRS)
410.      CALL PSTRES(NDIM,1,STRS,P)
411.      IF (LRM .NE. 1) GO TO 35
412.      PT(2)=PD(LDA,1)*TCRT
413.      PT(3)=.5*(PD(LDA,3)+PD(LDA,1))*SCRT
414.      GO TO 37
415. 35 PT(2)=PD(LDA,3)*TCRT
416.      PT(3)=PD(LDA,3)*SCRT
417. 37 PT(1)=PD(LDA,3)
418.      RT1=-P(1)/PD(LDA,3)
419.      RT2=P(NDIM)/PT(2)
420.      RT3=.5*(P(NDIM)-P(1))/PT(3)
421.      IF (RT1 .GT. RT2) GO TO 46
422.      TENSILE FAILURE
423.      IF (RT2 .GT. RT3) ICODE=2
424.      GO TO 47
425.  C
426.      COMPRESSIVE FAILURE
427. 46 IF (RT1 .GT. RT3) ICODE=1
428. 47 PRINT 213,LDA,VV(ICODE),WW(ICODE),(STRS(L1),L1=1,NY),
429.      I(P(L1),L1=1,NDIM),PT(ICODE)
430.      IF (MI .EQ. NCYC) GO TO 80
431.      IF (NER .EQ. 1) GO TO 80
432.      NN=(LDA-1)/NER+1
433.      NMN=(NN-1)/MINAM+1
434.      IDRUM=NMN+12
435.      NAM=NN-(NMN-1)*MINAM-NM(NMN)
436.      IF (NAM .EQ. 0) GO TO 52
437.      DO 48 IR=1,NAM
438.      READ (IDRUM)((SL(L1,L2),L1=1,NX),L2=1,NX)
439. 48 CONTINUE
440.      NM(NMN)=NN-(NMN-1)*MINAM
441.      GO TO 52
442. 50 CALL INTEG(NDIM,LDA,NDI,NDF,SL)
443. 52 IF (ICODE .NE. 2) GO TO 60
444.      IF (NTEP .EQ. 1) GO TO 60
445.      NW=1
446.      IF (ISO .EQ. 1) GO TO 56
447.      ISO=1
448.      E1=PD(LDA,1)
449.      E2=E1
450.      IF (NDIM .EQ. 2) E2=0.
451.      E3=0.
452.      P1=PD(LDA,2)
453.      P2=P1
454.      P3=P1
455.      G1=E1/2./(1.+P1)

```

```

455.      G2=G1
456.      G3=G1
457.      GO TO 58
458.      C   FOR ORIGINALLY ANISOTROPIC DISCS
459.      C   56 CONTINUE
460.      C   *****
461.      58 CALL INTEG(NDIM,LDA,NDI,NDF,SLD)
462.      DO 59 L1=1,NX
463.      DO 59 L2=1,NX
464.      59 SLD(L1,L2)=SLD(L1,L2)-SL(L1,L2)
465.      GO TO 62
466.      60 DO 61 L1=1,NX
467.      DO 61 L2=1,NX
468.      61 SLD(L1,L2)=POSM*SL(L1,L2)
469.      62 DO 77 L1=1,NB
470.      LIL=NOD(L1)
471.      LX=NOER(LIL)-MX
472.      MX=NOER(LIL)
473.      IF (MX .LT. MINUM)MINUM=MX
474.      IF (LX .LE. 0)GO TO 65
475.      IF (LX .EQ. 1)GO TO 68
476.      LX=LX-1
477.      DO 64 IA=1,LX
478.      READ (12)NN3,((S(K,J),K=1,NDIM),J=1,NN3)
479.      64 CONTINUE
480.      GO TO 68
481.      65 NXX=1-LX
482.      DO 66 IA=1,NXX
483.      BACKSPACE 12
484.      66 CONTINUE
485.      68 READ (12)NN3,((S(K,J),K=1,NDIM),J=1,NN3)
486.      BACKSPACE 12
487.      IDUM=(L1-1)*NDIM
488.      DO 69 IA=1,NDIM
489.      IIA=IA+IDUM
490.      DO 69 IB=1,NDIM
491.      IIB=IB+IDUM
492.      69 S(IA,IB)=S(IA,IB)+SLD(IIA,IIB)
493.      IF (MX .EQ. NTNN)GO TO 76
494.      NN=NN3/NDIM-1
495.      DO 75 L2=1,NB
496.      LAL=NOD(L2)
497.      JX=NOER(LAL)
498.      IF (JX .LE. MX)GO TO 75
499.      NBC=NBRC(MX)-1
500.      DO 71 L3=1,NN
501.      LL3=L3
502.      IF (LOC(NBC+L3) .EQ. JX)GO TO 72
503.      71 CONTINUE
504.      72 IDEM=LL3*NDIM
505.      IDAM=(L2-1)*NDIM
506.      DO 73 IA=1,NDIM
507.      IIA=IA+IDUM
508.      DO 73 IB=1,NDIM
509.      IJA=IB+IDEM
510.      IIB=IB+IDAM
511.      73 S(IA,IJA)=S(IA,IJA)+SLD(IIA,IIB)

```

```

512.      75 CONTINUE
513.      76 WRITE (12)NN3,((S(K,J),K=1,NDIM),J=1,NN3)
514.      BACKSPACE 12
515.      77 READ (12)NN3,((S(K,J),K=1,NDIM),J=1,NN3)
516.      77 CONTINUE
517.      ISO=LRM
518.      79 CONTINUE
519.      CALL PCTOR
520.      80 CONTINUE
521.      100 FORMAT (9I5)
522.      102 FORMAT (8F10.0)
523.      104 FORMAT (4F10.0,2I5)
524.      106 FORMAT (1I0)
525.      199 FORMAT (17HO CRITICAL LOAD = ,E12.6)
526.      200 FORMAT (27HO EXTERNALLY-LOADED NODES --,
527.      18(14,1H,)/(27X,8(14,1H,)))
528.      201 FORMAT (18HO BOUNDARY NODES --,10(14,1H, )
529.      2/(18X,10(14,1H,)))
530.      202 FORMAT (10HO CYCLE NO.,13,5H ****)
531.      203 FORMAT (36HO HORIZONTAL STRAINS AT CENTRAL POINT)
532.      204 FORMAT (17H FRONT END = ,E10.5)
533.      205 FORMAT (17H BACK END = ,E10.5)
534.      206 FORMAT (17H AVERAGE = ,E10.5)
535.      207 FORMAT (1HO,11X,'RADIAL DISPLACEMENTS AT LOADED NODES'/
536.      12(' NODE',5X,' LOAD ',5X,' DISPLACEMENT',10X))
537.      209 FORMAT (2(14,4X,E10.5,5X,E10.5,10X))
538.      211 FORMAT (1HO,9X,19H STRESSES IN FAILED ,
539.      13) ELEMENTS (DUE TO 1000-LB. LOAD))
540.      213 FORMAT (16,4X,A6,A5,10(1X,E10.5))
541.      212 FORMAT (' ELEMENT FAIL. MODE RADIAL CIRCUM. ',
542.      1' AXIAL SHRRZ SHRRZ SHRCZ ',
543.      2' PRNCP1 PRNCP2 PRNCP3 ALLOW. ')
544.      214 FORMAT (' ELEMENT FAIL. MODE RADIAL CIRCUM. ',
545.      1' SHRRZ PRNCP1 PRNCP2 ALLOW. ')
546.      300 FORMAT (20HO MATERIAL PROPERTIES)
547.      301 FORMAT (23H ELASTIC MODULUS = ,1PE12.6)
548.      302 FORMAT (22H POISSONS RATIO = ,F4.2)
549.      303 FORMAT (28H ALLOW. COMPR. STRESS = ,2PE10.3)
550.      306 FORMAT (23H ELASTIC MODULUS = ,1PE12.6,
551.      14H TO ,1PE12.6)
552.      307 FORMAT (22H POISSONS RATIO = ,F4.2,
553.      14H TO ,F4.2)
554.      308 FORMAT (28H ALLOW. COMPR. STRESS = ,2PE10.3,
555.      14H TO ,2PE10.3)
556.      309 FORMAT (19H ELASTIC MODULI/8X,5HE1 = ,1PE12.6/
557.      18X,5HE2 = ,1PE12.6/8X,5HE3 = ,1PE12.6)
558.      310 FORMAT (20H POISSONS RATIOS/8X,5HP1 = ,F4.2/
559.      18X,5HP2 = ,F4.2/8X,5HP3 = ,F4.2)
560.      311 FORMAT (17H SHEAR MODULI/8X,5HG1 = ,1PE12.6/
561.      18X,5HG2 = ,1PE12.6/8X,5HG3 = ,1PE12.6)
562.      312 FORMAT (27H ALLOW. COMPR. STRESSES/8X,5HC1 = ,
563.      12PE10.3/8X,5HC2 = ,2PE10.3/8X,5HC3 = ,2PE10.3)
564.      304 FORMAT (28H ALLOW. TENSL. STRESS = ,F4.2,
565.      117H OF ALLOW. COMPR.)
566.      305 FORMAT (28H ALLOW. SHEAR. STRESS = ,F4.2,
567.      117H OF ALLOW. COMPR.)
568.      313 FORMAT (28HO DIMENSIONS OF DISC SPECIMEN)

```

```
569.      314 FORMAT (16H      DIAMETER = ,F5.2)
570.      315 FORMAT (17H      THICKNESS = ,F5.2)
571.      316 FORMAT (19HORADIAL COORDINATES/(4X,10(F6.2,1H,)))
572.      317 FORMAT (38HOCIRCUMFERENTIAL COORDINATES (DEGREES)
573.          1/(4X,10(F6.2,1H,)))
574.      318 FORMAT (14HOZ-COORDINATES/(4X,10(F6.2,1H,)))
575.      319 FORMAT (24HOTOTAL NO. OF ELEMENTS = ,I4)
576.      320 FORMAT (21HOTOTAL NO. OF NODES = ,I4)
577.      326 FORMAT ('      ANISOTROPIC'/8X,'ALPHA1 = ',2PE10.3,
578.          1' DEGREES'/8X,'ALPHA2 = ',2PE10.3/8X,'ALPHA3 = ',2PE10.3)
579.      433 STOP
580.          END
```

END OF COMPILATION:

NO DIAGNOSTICS.

```

1.      SUBROUTINE NDTND(LSD,INGA,INGO)
2.      C      .....
3.      C      GENERATES NODES CONNECTED TO NODE I AND WRITES THEM
4.      C      IN AUXILIARY STORAGE
5.      C      FORMS BRANCH COUNTER ARRAY, NRRC(1)
6.      C      .....
7.      C
8.      COMMON/RUTH/LDC(25000),NTNN,NB,NP,MISTY
9.      COMMON/FUNKY/IBRC(500),NREQ(500),NOER(500)
10.     COMMON/DIFFER/UR,NS,NZ,NWOP,IDIM
11.     DIMENSION NCON(27),LSD(INGA,INGO)
12.     NT=NS-NWOP
13.     ND=(NT+1)*(NZ+1)
14.     M=-1
15.     DO 16 I=1,NTNN
16.     M=M+1
17.     J1=(I-1)/ND
18.     J2=ND*J1+1
19.     J3=(I-J2)/(NT+1)
20.     NCON (2)=I-ND
21.     NCON (1)=NCON(2)-1
22.     IF (NWOP .EQ. 1 .AND. I .EQ. (J2+J3*NS))NCON(1)=NCON(2)+NT
23.     NCON (3)=NCON(2)+1
24.     IF (NWOP .EQ. 1 .AND. I .EQ. 1/NS*NS)NCON(3)=NCON(2)-NT
25.     NCON (4)=NCON(3)+ND
26.     NCON (5)=NCON(4)+ND
27.     NCON (6)=1+ND
28.     NCON (8)=NCON(1)+ND
29.     NCON (7)=NCON(8)+ND
30.     IF (IDIM .EQ. 2)GO TO 3
31.     NCON (9)=NCON(1)+NT-1
32.     NCON(10)=NCON(2)+NT-1
33.     NCON(11)=NCON(3)+NT-1
34.     NCON(12)=NCON(11)+ND
35.     NCON(13)=NCON(12)+ND
36.     NCON(14)=NCON(6)+NT-1
37.     NCON(16)=NCON(9)+ND
38.     NCON(15)=NCON(16)+ND
39.     NCON(17)=1+NT-1
40.     NCON(18)=NCON(11)+NT+1
41.     NCON(19)=NCON(2)+NT+1
42.     NCON(20)=NCON(3)+NT+1
43.     NCON(21)=NCON(20)+ND
44.     NCON(22)=NCON(21)+ND
45.     NCON(23)=NCON(6)+NT+1
46.     NCON(25)=NCON(18)+ND
47.     NCON(24)=NCON(25)+ND
48.     NCON(26)=1+NT+1
49.     IF (J3 .EQ. 0 .OR. J3 .EQ. NZ) GO TO 1
50.     GO TO 3
51.     1 NI=(J3/NZ+1)*9
52.     NF=NI+8
53.     DO 2 J=NI,NF
54.     2 NCON(J)=0
55.     3 IF (J1 .EQ. 0 .OR. J1 .EQ. NR) GO TO 4

```

```

54.          GO TO 6
57.          4 NI=1+J1/NR*4
58.          NF=NI+2
59.          DO 5 J=NI,NF
60.          NCON(J)=0
61.          NCON(J+8)=0
62.          5 NCON(J+17)=0
63.          6 IF (NWOP .EQ. 1)GO TO 11
64.          IF (M .EQ. 0)GO TO 7
65.          IF (M .EQ. NT) GO TO 8
66.          GO TO 11
67.          7 NCON (1)=0
68.          NCON (8)=0
69.          NCON (7)=0
70.          NCON (9)=0
71.          NCON(16)=0
72.          NCON(15)=0
73.          NCON(18)=0
74.          NCON(25)=0
75.          NCON(24)=0
76.          GO TO 11
77.          8 DO 9 J=3,5
78.          NCON(J)=0
79.          NCON(J+9)=0
80.          9 NCON(J+17)=0
81.          M=-1
82.          11 NBRC(1)=26
83.          IF (IDIM .EQ. 2)NBRC(1)=8
84.          MM=NBRC(1)
85.          NCON(MM+1)=0
86.          MM1=MM-1
87.          DO 14 J=1,MM1
88.          IF (NCON(J) .NE. 0) GO TO 14
89.          J1=J+1
90.          DO 12 K=J1,MM
91.          KJ=K
92.          IF (NCON(K) .NE. 0) GO TO 13
93.          12 CONTINUE
94.          NBRC(1)=J-1
95.          GO TO 15
96.          13 NCON(J)=NCON(KJ)
97.          NCON(KJ)=0
98.          14 CONTINUE
99.          15 NI=NBRC(1)
100.         DO 10 J=1,NI
101.         10 LSD(J,1)=NCON(J)
102.         16 CONTINUE
103.         RETURN
104.         END

```

END OF COMPILATION: NO DIAGNOSTICS.

```

1.      SUBROUTINE REORD(LSD,INGA,INGO)
2.      C      .....
3.      C      HEAR-OPTIMAL REORDERING OF NODES
4.      C      .....
5.      C
6.      COMMON/RUTH/LDC(25000),NTNN,NB,NP,MISTY
7.      COMMON/FUNKY/NBRC(500),NREO(500),NOFR(500)
8.      COMMON/JNFER/NR,NS,NZ,NWOP,IDIM
9.      DIMENSION LA(100),LSD(INGA,INGO)
10.     REWIND 11
11.     PRINT 100
12.     J=0
13.     NZET=0
14.     A NDUM=1000
15.     DO 7 J=1,NTNN
16.     IF (NBRC(J) .LT. 0) GO TO 7
17.     IF (NBRC(J) .GE. NDUM) GO TO 7
18.     NDUM=NBRC(J)
19.     JLOCT=1
20.     7 CONTINUE
21.     A JLDC=JLOCT
22.     JLDC=0
23.     J=J+1
24.     NREO(J)=JLDC
25.     NOFR(JLDC)=J
26.     NN=NBRC(JLDC)
27.     WRITE (11,200)NN,(LSD(L,JLDC),L=1,NN)
28.     NZET=NZET+NN
29.     NM=NN
30.     IF (J .EQ. NTNN-1)GO TO 24
31.     DO 23 J=1,NN
32.     KLOC=LSD(1,JLDC)
33.     MM=NBRC(KLOC)
34.     IF (MM .GT. INGA)GO TO 30
35.     DO 14 JJ=1,MM
36.     JM=JJ
37.     IF (LSD(JJ,KLOC) .EQ. JLDC)GO TO 15
38.     14 CONTINUE
39.     15 M1=MM-1
40.     DO 16 LL=JM,M1,1
41.     16 LSD(LL,KLOC)=LSD(LL+1,KLOC)
42.     NBRC(KLOC)=NBRC(KLOC)-1
43.     MM=NBRC(KLOC)
44.     DO 19 JJ=1,MM
45.     IF (LSD(JJ,JLDC) .EQ. KLOC)GO TO 19
46.     DO 18 LL=1,MM
47.     IF (LSD(JJ,JLDC) .EQ. LSD(LL,KLOC))GO TO 19
48.     18 CONTINUE
49.     NBRC(KLOC)=NBRC(KLOC)+1
50.     M=NBRC(KLOC)
51.     LSD(M,KLOC)=LSD(JJ,JLDC)
52.     19 CONTINUE
53.     IF (NBRC(KLOC) .GT. MM) GO TO 23
54.     NM=NBRC(KLOC)
55.     JLOCT=KLOC

```



```

56.      23 CONTINUE
57.      NRRC(JLOC)=-NRRC(JLOC)
58.      IF (JLOC.EQ. 0) GO TO 6
59.      GO TO 8
60.      24 NREQ(HTNN)=LSD(1,JLOC)
61.      LA=LSD(1,JLOC)
62.      NOER(LA)=NTNN
63.      TNN=NTNN
64.      ZET=NZET+HTNN
65.      PRCT=ZET/TNN*100./TNN
66.      REWIND 11
67.      NSJ=0
68.      NTA=HTNN-1
69.      DO 29 I=1,NTA
70.      READ (11,200)NN,(LA(L),L=1,NN)
71.      NRRC(I)=-NN
72.      NNA=NN-1
73.      DO 26 J=1,NNA
74.      LAJ=LA(J)
75.      J=J+1
76.      DO 26 K=J1,NN
77.      LAK=LA(K)
78.      IF (NOER(LAK).GT. NOER(LAJ))GO TO 26
79.      TEMP=LA(J)
80.      LA(J)=LA(K)
81.      LA(K)=TEMP
82.      LAJ=LA(J)
83.      26 CONTINUE
84.      PRINT 102,I,NREQ(I),(LA(L),L=1,NN)
85.      DO 27 K=1,NN
86.      LAK=LA(K)
87.      NSJ=NSJ+1
88.      27 LOC(NSJ)=NOER(LAK)
89.      NSJ=NSJ+1
90.      LOC(NSJ)=I
91.      29 CONTINUE
92.      PRINT 102,HTNN,NREQ(HTNN)
93.      PRINT 104,PRCT
94.      100 FORMAT ('1      RESULTS OF NODAL REORDERING'/
95.      1'ONEW NO.  ORIG. NO.  NODAL CONNECTIONS (ORIG. NOS.)')
96.      101 FORMAT ('0***** ARRAY LENGTH IS EXCEEDED *****')
97.      200 FORMAT (32I4)
98.      102 FORMAT (16,5X,16,4X,22I5/(21X,22I5))
99.      104 FORMAT (32HPERCENTAGE OF NON-ZERO TERMS = ,F5.2)
100.      GO TO 31
101.      30 MISTY=MM
102.      PRINT 101
103.      31 RETURN
104.      END

```

END OF COMPILATION: NO DIAGNOSTICS.

```

1.      SUBROUTINE ELTOND (IDIM,I,LCON)
2.      C
3.      C
4.      C
5.      C
6.      C
7.      COMMON/JNFER/NR,NS,NZ,NWOP,NDIM
8.      DIMENSION LCON(8)
9.      NT=NS-NWOP
10.     ND=(NT+1)*(NZ+1)
11.     NG=NS*NZ
12.     IF (IDIM .EQ. 2)NG=NS
13.     J1=(1-I)/ND
14.     J2=ND*J1+1
15.     J3=(1-J2)/(NT+1)
16.     K3=J3
17.     IF (NWOP .EQ. 1) K3=0
18.     LCON(1)=1-(ND-NG)*J1-K3
19.     LCON(2)=LCON(1)-1
20.     IF (NWOP .EQ. 1 .AND. I .EQ. (J2+J3*NS))LCON(2)=LCON(1)+NT
21.     LCON(3)=LCON(2)-NG
22.     LCON(4)=LCON(1)-NG
23.     IF (IDIM .EQ. 2)GO TO 3
24.     LCON(5)=LCON(1)-NS
25.     LCON(6)=LCON(2)-NS
26.     LCON(7)=LCON(6)-NG
27.     LCON(8)=LCON(5)-NG
28.     IF (J3 .EQ. 0 .OR. J3 .EQ. NZ) GO TO 1
29.     GO TO 3
30.     1 NI=5-J3/NZ*4
31.     NF=NI+3
32.     DO 2 J=NI,NF
33.     2 LCON(J)=0
34.     3 IF (J1 .EQ. 0 .OR. J1 .EQ. NR) GO TO 4
35.     GO TO 6
36.     4 NI=3-J1/NR*2
37.     NF=NI+1
38.     DO 5 J=NI,NF
39.     LCON(J)=0
40.     5 LCON(J+4)=0
41.     6 IF (NWOP .EQ. 1)GO TO 9
42.     J4=1-J2-J3*(NT+1)
43.     IF (J4 .EQ. 0) GO TO 7
44.     IF (J4 .EQ. NT) GO TO 8
45.     GO TO 9
46.     7 LCON(2)=0
47.     LCON(3)=0
48.     LCON(6)=0
49.     LCON(7)=0
50.     GO TO 9
51.     8 LCON(1)=0
52.     LCON(4)=0
53.     LCON(5)=0
54.     LCON(8)=0
55.     9 IF (IDIM .EQ. 2)GO TO 10

```

```
56.          J1=LCON(4)
57.          LCON(4)=LCON(5)
58.          LCON(5)=J1
59.          J1=LCON(3)
60.          LCON(3)=LCON(6)
61.          LCON(6)=J1
62.          10 RETURN
63.          END
```

```
END OF COMPILATION:          NO  DIAGNOSTICS.
```

```

1.      SUBROUTINE NODES (IDIM,I,NOD)
2.      C .....
3.      C GENERATES NODAL NUMBERS OF      ELEMENT I
4.      C .....
5.      COMMON/INFER/NR,NS,NZ,NWOP,NDIM
6.      DIMENSION NOD(8)
7.      NT=NS
8.      IF (NWOP .EQ. 1) NT=NS-1
9.      ND=(NT+1)*(NZ+1)
10.     NG=NS*NZ
11.     IF (IDIM .EQ. 2) NG=NS
12.     IJ=(I-1)/NG
13.     IZ=NG*IJ+1
14.     IS=(I-IZ)/NS
15.     KJ=IJ
16.     IF (NWOP .EQ. 1) KJ=0
17.     NOD(1)=I+(ND-NG)*IJ+KJ
18.     NOD(2)=NOD(1)+1
19.     IF (NWOP .EQ. 1 .AND. I .EQ. 1/NS*NS) NOD(2)=NOD(1)-NT
20.     NOD(3)=NOD(2)+ND
21.     NOD(4)=NOD(1)+ND
22.     IF (IDIM .EQ. 2) GO TO 10
23.     NOD(5)=NOD(1)+NT+1
24.     NOD(6)=NOD(2)+NT+1
25.     NOD(7)=NOD(6)+ND
26.     NOD(8)=NOD(5)+ND
27.     TO RETURN
28.     END

```

END OF COMPILATION: NO DIAGNOSTICS.

```

1.      SUBROUTINE COORD3(I,RDC,TTC,ZZC)
2.      C .....
3.      C ASSIGNS RADIAL,CIRCUMFERENTIAL, AND Z ORDINATES TO NODE I
4.      C .....
5.      COMMON/JNFER/NR,NS,NZ,NWOP,NDIM
6.      COMMON/LEYTE/RC(30),TC(50),ZC(30)
7.      ZZC=0.
8.      NT=NS-NWOP
9.      ND=(NT+1)*(NZ+1)
10.     J1=(I-1)/ND
11.     RDC=RC(J1+1)
12.     J2=ND*J1+1
13.     J3=(1-J2)/(NT+1)
14.     IF (NDIM .EQ. 2)GO TO 1
15.     ZZC=ZC(J3+1)
16.     1 J4=1-J2-J3*(NT+1)
17.     TTC=TC(J4+1)
18.     RETURN
19.     END

```

END OF COMPILATION: NO DIAGNOSTICS.

```
1.      SUBROUTINE UNIFRM(A,B,C)
2.      D=RANUN(R)
3.      C=A+(B-A)*D
4.      RETURN
5.      END
```

```
END OF COMPILATION:      NO DIAGNOSTICS.
```

```

1.      SUBROUTINE MTRXB(IDIM,FK,K,TZ,Z,OB)
2.      C .....
3.      C FURMS THE MATRIX B AND THE MATRIX PRODUCT D*B
4.      C .....
5.      COMMON/SANNA/PD(480,3),FNCTR(8),FNCTT(8),FNCTZ(8)
6.      COMMON/BGAND/C(8,8),B(6,24),M
7.      COMMON/TANA1/RAD,RS,TO,TS,ZS,NA,MNB
8.      COMMON/TATAY/ISO,A1,A2,A3,E1,E2,E3,P1,P2,P3,
9.      G1,G2,G3
10.     COMMON/NANAY/TT(3,3),NW
11.     DIMENSION BB(6,24),FNCT(8),DP(6,0),D(6,6),T(6,6)
12.     NX1=NX-1
13.     NY=4*(IDIM-1)*IDIM
14.     IF (IDIM.EQ.2)GO TO 3
15.     FNCT(1)=1./RAD
16.     FNCT(2)=R/RAD
17.     FNCT(3)=TZ/RAD
18.     FNCT(4)=FNCT(3)*R
19.     FNCT(5)=Z/RAD
20.     FNCT(6)=FNCT(5)*R
21.     FNCT(7)=FNCT(5)*TZ
22.     FNCT(8)=FNCT(7)*R
23.     FNCTR(2)=1./RS
24.     FNCTR(4)=TZ/RS
25.     FNCTR(6)=Z/RS
26.     FNCTR(8)=FNCTR(6)*TZ
27.     FNCTT(3)=1./RAD/TS
28.     FNCTT(4)=R/RAD/TS
29.     FNCTT(7)=Z/RAD/TS
30.     FNCTT(8)=FNCTT(7)*R
31.     FNCTZ(5)=1./ZS
32.     FNCTZ(6)=R/ZS
33.     FNCTZ(7)=TZ/ZS
34.     FNCTZ(8)=FNCTZ(7)*R
35.     DO 1 I=1,8
36.     B(1,I)=FNCTR(1)
37.     B(2,I)=FNCT(1)
38.     B(2,I+8)=FNCTT(1)
39.     B(3,I+16)=FNCTZ(1)
40.     B(4,I)=FNCTT(1)
41.     B(4,I+8)=FNCTR(1)-FNCT(1)
42.     B(5,I)=FNCTZ(1)
43.     B(5,I+16)=FNCTR(1)
44.     B(6,I+8)=FNCTZ(1)
45.     I B(6,I+16)=FNCTT(1)
46.     IF (ISO)13,13,6
47.     I3 PR=PD(KK,2)
48.     D3=.5*PD(KK,1)/(1.+PR)
49.     DM=D3*2./(1.-2.*PR)
50.     D1=DM*(1.-PR)
51.     D2=DM*PR
52.     DO 2 I=1,24
53.     UB(1,I)=D1*B(1,I)+D2*(B(2,I)+B(3,I))
54.     BB(2,I)=D1*B(2,I)+D2*(B(1,I)+B(3,I))
55.     BB(3,I)=D1*B(3,I)+D2*(B(1,I)+B(2,I))

```

```

56.      UB(4,1)=03*U(4,1)
57.      UB(5,1)=03*U(5,1)
58.      2 UB(6,1)=03*U(6,1)
59.      GO TO 5
60.      0 IF (NW .EQ. 1) GO TO 7
61.      A=A1-TU-TS+TZ
62.      TT(1,1)=COS(A2)*SIN(A)
63.      TT(2,1)=-COS(A2)*COS(A)
64.      TT(3,1)=-SIN(A2)
65.      TT(1,2)=COS(A3)*COS(A)+SIN(A3)*SIN(A2)*SIN(A)
66.      TT(2,2)=COS(A3)*SIN(A)-SIN(A3)*SIN(A2)*COS(A)
67.      TT(3,2)=SIN(A3)*COS(A2)
68.      TT(1,3)=COS(A3)*SIN(A2)*SIN(A)-SIN(A3)*COS(A)
69.      TT(2,3)=-COS(A3)*SIN(A2)*COS(A)-SIN(A3)*SIN(A)
70.      TT(3,3)=COS(A3)*COS(A2)
71.      7 TT(1,1)=TT(1,1)*TT(1,1)
72.      TT(1,2)=TT(1,2)*TT(1,2)
73.      TT(1,3)=TT(1,3)*TT(1,3)
74.      TT(1,4)=2.*TT(1,1)*TT(1,2)
75.      TT(1,5)=2.*TT(1,1)*TT(1,3)
76.      TT(1,6)=2.*TT(1,2)*TT(1,3)
77.      TT(2,1)=TT(2,1)*TT(2,1)
78.      TT(2,2)=TT(2,2)*TT(2,2)
79.      TT(2,3)=TT(2,3)*TT(2,3)
80.      TT(2,4)=2.*TT(2,1)*TT(2,2)
81.      TT(2,5)=2.*TT(2,1)*TT(2,3)
82.      TT(2,6)=2.*TT(2,2)*TT(2,3)
83.      TT(3,1)=TT(3,1)*TT(3,1)
84.      TT(3,2)=TT(3,2)*TT(3,2)
85.      TT(3,3)=TT(3,3)*TT(3,3)
86.      TT(3,4)=2.*TT(3,1)*TT(3,2)
87.      TT(3,5)=2.*TT(3,1)*TT(3,3)
88.      TT(3,6)=2.*TT(3,2)*TT(3,3)
89.      TT(4,1)=TT(1,1)*TT(2,1)
90.      TT(4,2)=TT(1,2)*TT(2,2)
91.      TT(4,3)=TT(1,3)*TT(2,3)
92.      TT(4,4)=TT(1,2)*TT(2,1)+TT(1,1)*TT(2,2)
93.      TT(4,5)=TT(1,3)*TT(2,1)+TT(1,1)*TT(2,3)
94.      TT(4,6)=TT(1,3)*TT(2,2)+TT(1,2)*TT(2,3)
95.      TT(5,1)=TT(1,1)*TT(3,1)
96.      TT(5,2)=TT(1,2)*TT(3,2)
97.      TT(5,3)=TT(1,3)*TT(3,3)
98.      TT(5,4)=TT(1,2)*TT(3,1)+TT(1,1)*TT(3,2)
99.      TT(5,5)=TT(1,3)*TT(3,1)+TT(1,1)*TT(3,3)
100.     TT(5,6)=TT(1,3)*TT(3,2)+TT(1,2)*TT(3,3)
101.     TT(6,1)=TT(2,1)*TT(3,1)
102.     TT(6,2)=TT(2,2)*TT(3,2)
103.     TT(6,3)=TT(2,3)*TT(3,3)
104.     TT(6,4)=TT(2,2)*TT(3,1)+TT(2,1)*TT(3,2)
105.     TT(6,5)=TT(2,3)*TT(3,1)+TT(2,1)*TT(3,3)
106.     TT(6,6)=TT(2,3)*TT(3,2)+TT(2,2)*TT(3,3)
107.     DUM=E2*E3-E1*E3*P1*P1-E2*E2*P3*P3
108.     I=E1*E2*(2.*P1*P2*P3+P2*P2)
109.     U(1,1)=E1*E2*(E3-P3*P3*E2)/DUM
110.     U(1,2)=E1*E2*(P2*P3*E2+P1*E3)/DUM
111.     U(1,3)=E1*E2*E3*(P1*P3+P2)/DUM
112.     U(2,2)=E2*E2*(E3-P2*P2*E1)/DUM

```



```

113.      U(2,3)=E2*E3*(P1*P2*E1+P3*E2)/DUM
114.      U(3,3)=E3*E3*(E2-P1*P1*E1)/DUM
115.      U(2,1)=D(1,2)
116.      U(3,1)=D(1,3)
117.      U(3,2)=D(2,3)
118.      U(4,4)=G1
119.      U(5,5)=G2
120.      U(6,6)=G3
121.      GO TO 11
122.      3 B(1,2)=1./RS
123.      B(1,4)=T2/RS
124.      B(2,1)=1./RAD
125.      B(2,2)=R/RAD
126.      B(2,3)=T2/RAD
127.      B(2,4)=B(2,3)*R
128.      B(2,7)=1./(RAD*TS)
129.      B(2,8)=B(2,7)*R
130.      U(3,3)=B(2,7)
131.      B(3,4)=B(2,6)
132.      B(3,5)=-B(2,1)
133.      B(3,6)=1./RS-R/RAD
134.      B(3,7)=-B(2,3)
135.      B(3,8)=B(3,6)*T2
136.      IF (150)B,8,9
137.      8 PR=PD(KK,2)
138.      DUM=PD(KK,1)*H/(1.-PR*PR)
139.      DO 4 I=1,8
140.      BB(1,1)=DUM*(B(1,1)+PR*B(2,1))
141.      BB(2,1)=DUM*(B(2,1)+PR*B(1,1))
142.      4 BB(3,1)=DUM*.5*(1.-PR)*B(3,1)
143.      GO TO 5
144.      9 IF (NW .EQ. 1)GO TO 10
145.      A=A1-TU-TS*T2
146.      TT(1,1)=SIN(A)
147.      TT(1,2)=COS(A)
148.      TT(2,1)=-COS(A)
149.      TT(2,2)=SIN(A)
150.      10 T(1,1)=TT(1,1)*T(1,1)
151.      T(1,2)=TT(1,2)*T(1,2)
152.      T(1,3)=2.*TT(1,1)*T(1,2)
153.      T(2,1)=TT(2,1)*T(2,1)
154.      T(2,2)=TT(2,2)*T(2,2)
155.      T(2,3)=2.*TT(2,1)*T(2,2)
156.      T(3,1)=T(1,1)*T(2,1)
157.      T(3,2)=T(1,2)*T(2,2)
158.      T(3,3)=TT(1,2)*TT(2,1)+TT(1,1)*T(2,2)
159.      DUM=L2-P1*P1*E1
160.      U(1,1)=E1*E2/DUM
161.      U(1,2)=P)*D(1,1)
162.      U(2,1)=D(1,2)
163.      U(2,2)=E2*E2/DUM
164.      U(3,3)=G1
165.      11 DO 16 I=1,10IM
166.      DO 16 J=1,4X
167.      UP(I,J)=0.
168.      DO 17 K=1,10IM
169.      17 UP(I,J)=UP(I,J)+D(I,K)*T(J,K)

```

```
170.      18 CONTINUE
171.          10=10/M+1
172.          DO 20 I=10,NX
173.              DO 20 J=1,NX
174.      20  DP(I,J)=D(I,1)*T(J,1)
175.          DO 22 I=1,NX
176.          DO 22 J=1,NX
177.              U(I,J)=0.
178.          DO 21 K=1,NX
179.      21  U(I,J)=U(I,J)+T(I,K)*DP(K,J)
180.      22 CONTINUE
181.          DO 23 I=1,NX1
182.              I1=I+1
183.          DO 23 J=I1,NX
184.      23  U(J,1)=D(I,J)
185.          DO 25 I=1,NX
186.          DO 25 J=1,NY
187.              BB(I,J)=0.
188.          DO 24 K=1,NX
189.      24  BB(I,J)=BB(I,J)+U(I,K)*B(K,J)
190.      25 CONTINUE
191.          5 RETURN
192.          END
```

END OF COMPILATION: NO DIAGNOSTICS.

```

1.      SURROUTINE INTEG(IDIM, KK, NDI, NDF, SL)
2.      C .....
3.      C FORMS STIFFNESS MATRIX OF ELEMENTS
4.      C .....
5.      C
6.      DIMENSION SL(24,24), RR(6,24), AA(2), DX(24,24)
7.      COMMON/TANA1/RAD, RS, TO, TS, ZS, NX, HNB
8.      COMMON/RUTH/LOC(25000), NTNN, NB, NP, MINUM
9.      COMMON/RGAND/C(8,8), R(6,24), H
10.     DATA AA/.577350269189626, -.577350269189626/
11.     CALL COORD3(NDI, R1, T1, Z1)
12.     CALL COORD3(NDF, R2, T2, Z2)
13.     RS=.5*(R2-R1)
14.     RD=.5*(R2+R1)
15.     IF (T2 .LT. T1) T2=4.2831853072
16.     TS=.5*(T2-T1)
17.     TD=.5*(T2+T1)
18.     ZS=.5*(Z2-Z1)
19.     IF (IDIM .EQ. 2) ZS=1.
20.     NCX=IDIM-1
21.     NPI=NP-1
22.     DO 2 I=1, NP
23.     DO 2 J=1, NP
24.     2 SL(I, J)=0.
25.     DO 7 I=1, 2
26.     R=AA(I)
27.     RAD=RD+RS*R
28.     DUM1=RAD*RS*TS*ZS
29.     DO 6 J=1, 2
30.     T=AA(J)
31.     DO 5 K=1, NCX
32.     Z=AA(K)
33.     CALL MTRXR(IDIM, KK, R, T, Z, RR)
34.     DO 4 NR=1, NP
35.     DO 4 NC=NR, NP
36.     DUM2=0.
37.     DO 3 N=1, NX
38.     3 DUM2=DUM2+R(N, NR)*R(N, NC)
39.     4 SL(NR, NC)=SL(NR, NC)+DUM1*DUM2
40.     5 CONTINUE
41.     6 CONTINUE
42.     7 CONTINUE
43.     DO 8 I=1, NPI
44.     J=I+1
45.     DO 8 K=J, NP
46.     8 SL(K, I)=SL(I, K)
47.     II=-1
48.     DO 11 I=1, NP
49.     II=II+1
50.     I2=(I-1)/IDIM+1
51.     DO 10 J=1, NP
52.     DX(I, J)=0.
53.     DO 9 K=1, NB
54.     KI=K+II*NB
55.     9 DX(I, J)=DX(I, J)+C(K, I2)*SL(KI, J)

```

```
56.          DX(I,J)=DX(I,J)/HNB
57.    10 CONTINUE
58.          IF (II .EQ. NCX) II=-1
59.    11 CONTINUE
60.          II=-1
61.          DO 14 J=1,MP
62.            II=II+1
63.            I2=(I-II)/IDIM+1
64.            DO 13 J=1,MP
65.              SL(J,II)=0.
66.              DO 12 K=1,NB
67.                KJ=K+II*NB
68.            12 SL(J,II)=SL(J,I)+DX(J,KJ)*C(K,I2)
69.              SL(J,II)=SL(J,II)/HNB
70.            13 CONTINUE
71.            IF (II .EQ. NCX) II=-1
72.          14 CONTINUE
73.          DO 15 I=1,MP1
74.            J=I+1
75.            DO 15 K=J,MP
76.          15 SL(I,K)=SL(K,II)
77.          RETURN
78.          END
```

END OF COMPILATION: NO DIAGNOSTICS.

```

1.      SUBROUTINE STRESS(KK,NOD,STRS)
2.      COMMON/RUTH/LOC(25000),NTNN,NB,NX,M,NUM
3.      COMMON/TANA1/RO,RS,TO,TS,ZS,NY,HNB
4.      COMMON/JNFER/NR,NS,NZ,NWOP,IDIM
5.      COMMON/ADALN/D(3,600)
6.      COMMON/FUNKY/NBRC(500),NREQ(500),NOER(500)
7.      COMMON/BGAND/C(8,8),B(6,24),H
8.      DIMENSION NOD(8),DSP(24),A(24),BB(6,24),STRS(6)
9.      C      EVALUATES COMPONENTS OF STRESS ALONG CYLINDRICAL AXES
10.     I1=0
11.     DO 11 J=1,NB
12.     NDD=NOD(J)
13.     MTT=NTNN-NOER(NDD)+1
14.     DO 10 K=1,IDIM
15.     I1=I1+1
16.     10 DSP(I1)=0(K,MTT)
17.     11 CONTINUE
18.     I1=-1
19.     DO 15 J=1,NX
20.     A(J)=0.
21.     JJ=(J-1)/IDIM+1
22.     I1=I1+1
23.     DO 13 K=1,NB
24.     13 A(J)=A(J)+C(JJ,K)*DSP(I1*NB+K)
25.     IF (I1 .EQ. (IDIM-1)) I1=-1
26.     15 CONTINUE
27.     CALL MTRXB(IDIM,KK,0.,0.,0.,BB)
28.     DO 19 J=1,NY
29.     STRS(J)=0.
30.     DO 18 K=1,NX
31.     18 STRS(J)=STRS(J)+BB(J,K)*A(K)
32.     19 STRS(J)=STRS(J)/HNB
33.     RETURN
34.     END

```

END OF COMPILATION: NO DIAGNOSTICS.

```

1.      SUBROUTINE PSTRES(IDIM,MMA,B,P)
2.      C      .....
3.      C      COMPUTES MAGNITUDES AND DIRECTION COSINES
4.      C      OF PRINCIPAL STRESSES
5.      C      .....
6.      COMMON/NANAY/CR(3,3),NW
7.      DIMENSION B(6),P(3)
8.      IF (IDIM .EQ. 2)GO TO 13
9.      A1=B(1)+B(2)+B(3)
10.     A2=B(1)*B(2)+B(2)*B(3)+B(1)*B(3)-
11.     1B(4)*B(4)-B(5)*B(5)-B(6)*B(6)
12.     A3=B(1)*B(2)*B(3)+2.*B(4)*B(5)*B(6)-B(1)*B(6)*B(6)
13.     1-B(2)*B(5)*B(5)-B(3)*B(4)*B(4)
14.     P(3)=A1/3.
15.     DM=P(3)
16.     DO 5 I=1,4
17.     FX=P(3)*(A2+P(3)*(P(3)-A1))-A3
18.     FPX=A2+P(3)*(3.*P(3)-2.*A1)
19.     P(3)=P(3)-FX/FPX
20.     5 CONTINUE
21.     PRINT 6,DM,P(3),FX
22.     6 FORMAT (' FIRST P(3) = ',E10.5,' .. FINAL P(3) = ',
23.     1E10.5,' .. ITER. ERROR = ',E10.5)
24.     IF (ABS(FX) .LT. 150.)GO TO 8
25.     DM=P(3)
26.     DO 7 I=1,4
27.     FX=P(3)*(A2+P(3)*(P(3)-A1))-A3
28.     FPX=A2+P(3)*(3.*P(3)-2.*A1)
29.     P(3)=P(3)-FX/FPX
30.     7 CONTINUE
31.     PRINT 6,DM,P(3),FX
32.     8 B1=P(3)-A1
33.     B2=P(3)*B1+A2
34.     DM=SQRT(B1*B1-4.*B2)
35.     P(1)=.5*(-B1-DM)
36.     P(2)=.5*(-B1+DM)
37.     IF (P(3) .GT. P(2))GO TO 10
38.     B2=P(2)
39.     P(2)=P(3)
40.     P(3)=B2
41.     IF (P(2) .GT. P(1))GO TO 10
42.     B1=P(1)
43.     P(1)=P(2)
44.     P(2)=B1
45.     10 IF (MMA .NE. 1)GO TO 20
46.     C      CR = DIRECTION COSINES
47.     DO 12 I=1,3
48.     CC=(P(1)-B(2))*(P(1)-B(3))-B(6)*B(6)
49.     BB=(P(1)-B(1))*(P(1)-B(2))-B(4)*B(4)
50.     IF (ABS(BB) .GT. ABS(CC))GO TO 11
51.     AA=(P(1)-B(3))*B(4)+B(5)*B(6)
52.     BB=(P(1)-B(2))*B(5)+B(4)*B(6)
53.     GO TO 9
54.     11 CC=(P(1)-B(2))*B(5)+B(4)*B(6)
55.     AA=(P(1)-B(1))*B(6)+B(4)*B(5)

```

```
56.          9 DM=SQRT(CC*CC+BB*BB+AA*AA)
57.          CR(1,1)=CC/DM
58.          CR(2,1)=AA/DM
59.          CR(3,1)=BB/DM
60.          12 CONTINUE
61.          GO TO 20
62.          13 DM=SQRT(.25*(B(1)-B(2))**2+B(3)*B(3))
63.          P(1)=.5*(B(1)+B(2))/DM
64.          P(2)=.5*(B(1)+B(2))/DM
65.          IF (MMA .NE. 1)GO TO 20
66.          DO 14 I=1,2
67.          AA=P(1)-B(I)
68.          BB=P(1)-B(2)
69.          IF (ABS(BB) .GT. ABS(AA))GO TO 15
70.          DM=SQRT(AA*AA+B(3)*B(3))
71.          CR(2,1)=AA/DM
72.          CR(1,1)=B(3)/DM
73.          GO TO 14
74.          15 DM=SQRT(BB*BB+B(3)*B(3))
75.          CR(1,1)=BB/DM
76.          CR(2,1)=B(3)/DM
77.          14 CONTINUE
78.          20 RETURN
79.          END
```

END OF COMPILATION: NO DIAGNOSTICS.

```

1.      SUBROUTINE FACTOR
2.      C      .....
3.      C      MATRIX FACTORIZATION
4.      C      .....
5.      COMMON/FUNKY/NROW(500),NREO(500),NOER(500)
6.      COMMON/JNFER/NR,NS,NZX,NWOP,1DIM
7.      COMMON/ABALN/D(3,600)
8.      COMMON/JANET/NSPD,NWSD(100),SPD(100,3)
9.      COMMON/BOPAP/NLOD(100),CNLD(100),NE,NER
10.     COMMON/RUTH/LOC(25000),NTNN,NB,NP,MISTY
11.     DIMENSION A(3,300),B(3,3),C(3,3),JIT(4),LCON(8),
12.     I      NOD(8),NSPT(500),S(3,300),SL(24,24)
13.     EQUIVALENCE (D(1,301),S)
14.     REWIND 15
15.     REWIND 14
16.     REWIND 13
17.     REWIND 12
18.     REWIND 11
19.     REWIND 10
20.     JIT(1)=0
21.     JIT(2)=0
22.     JIT(3)=0
23.     MIM1=0
24.     KD=0
25.     NEE=1
26.     MEE=1
27.     NZ=0
28.     DO 70 I=1,NTNN
29.     NN=0
30.     IF (I .EQ. NTNN)GO TO 6
31.     NN=-NROW(I)
32.     NROW(I)=NZ+1
33.     NSPT(I)=NZ+1
34.     NZ=NZ+NN+1
35.     6 NN3=(NN+1)*1DIM+1
36.     DO 7 J=1,1DIM
37.     DO 7 K=1,NN3
38.     7 S(J,K)=0.
39.     J=NREO(I)
40.     PRINT 100,1,J
41.     100 FORMAT (6HONODE ,I4,11H (ORIG. NO.,I4,1H))
42.     CALL ELTOND (1DIM,J,LCON)
43.     PRINT 101,(LCON(I),I=1,NB)
44.     101 FORMAT (34H      ELEMENTS CONNECTED TO NODE -- ,8(I4,1H,))
45.     DO 15 K=1,NB
46.     IF (LCON(K) .EQ. 0)GO TO 15
47.     KK=LCON(K)
48.     CALL NODES (1DIM,KK,NOD)
49.     PRINT 102,KK,(NOD(I),I=1,NB)
50.     102 FORMAT (21H      NODES OF ELEMENT ,I4,4H -- ,8(I4,1H,))
51.     NDI=NOD(1)
52.     NDF=NOD(NB-1)
53.     IF (NER .EQ. 1)GO TO 56
54.     KS=(KK-1)/NER+1
55.     IF (KS .EQ. KD)GO TO 57

```



```

56.      NUM=(KS-1)/MISTY+1
57.      IDRUM=NUM+12
58.      JNT=KS-(NUM-1)*MISTY-JIT(NUM)
59.      399 FORMAT ('0IDRUM=',13,'...JNT=',13)
60.      PRINT 399,IDRUM,JNT
61.      JIT(NUM)=KS-(NUM-1)*MISTY
62.      IF (JNT .LE. 0)GO TO 51
63.      IF (JNT .EQ. 1)GO TO 54
64.      JNT=JNT-1
65.      DO 50 IA=1,JNT
66.      READ (IDRUM)((SL(L1,L2),L1=1,NP),L2=1,NP)
67.      50 CONTINUE
68.      GO TO 54
69.      51 JMT=1-JNT
70.      DO 52 IA=1,JMT
71.      BACKSPACE IDRUM
72.      52 CONTINUE
73.      54 READ (IDRUM)((SL(L1,L2),L1=1,NP),L2=1,NP)
74.      KD=KS
75.      GO TO 57
76.      56 CALL INTEG(IDIM,KK,NDI,NDF,SL)
77.      57 DO 8 M=1,NB
78.      IF (J .EQ. NOD(M))GO TO 9
79.      8 CONTINUE
80.      9 NDIA=M
81.      DO 10 KM=1,IDIM
82.      KN=(NDIA-1)*IDIM+KM
83.      DO 10 LM=1,IDIM
84.      LN=(NDIA-1)*IDIM+LM
85.      10 S(KM,LM)=S(KM,LM)+SL(KN,LN)
86.      IF (NN .EQ. 0)GO TO 15
87.      DO 14 M=1,NB
88.      NODM=NOD(M)
89.      IF (1 .GE. NOER(NODM))GO TO 14
90.      NRW=NROW(1)-1
91.      DO 13 L=1,NN
92.      NRW=NRW+1
93.      IF (LOC(NRW) .NE. NOER(NODM))GO TO 13
94.      DO 12 KM=1,IDIM
95.      KN=(NDIA-1)*IDIM+KM
96.      DO 12 MM=1,IDIM
97.      LN=(M-1)*IDIM+MM
98.      LM=L*IDIM+MM
99.      12 S(KM,LM)=S(KM,LM)+SL(KN,LN)
100.     GO TO 14
101.     13 CONTINUE
102.     14 CONTINUE
103.     15 CONTINUE
104.     IF (J .NE. NLOD(NEE))GO TO 16
105.     S(1,NN3)=CNLD(NEE)
106.     NEE=NEE+1
107.     16 WRITE (12)NN3,((S(L1,L2),L1=1,IDIM),L2=1,NN3)
108.     IF (MEE .GT. NSPD)GO TO 38
109.     IF (J .NE. NWSO(MEE))GO TO 18
110.     DO 17 K=1,IDIM
111.     IF (SPD(MEF,K) .GE. 100.)GO TO 17
112.     DO 26 L=1,IDIM

```

```

113.          S(L,NN3)=S(L,NN3)-S(L,K)*SPD(MEE,K)
114.          S(K,L)=0.
115. 26 S(L,K)=0.
116.          S(K,K)=1.
117.          S(K,NN3)=SPD(MEE,K)
118. 17 CONTINUE
119.          MEE=MEE+1
120.          IF (MEE .GT. NSPD)GO TO 38
121. 18 MU=MEE
122.          NU=1
123.          NRW=NROW(1)-1
124.          DO 28 K=MU,NSPD
125.             NWS=NWS(K)
126.             DO 30 L=NU,NN
127.                NRW=NRW+L
128.                IF (LOC(NRW) .GT. NOER(NWS))GO TO 28
129.                IF (LOC(NRW) .NE. NOER(NWS))GO TO 30
130.                MM=L
131.                M3=(NRW-NROW(1)+1)*IDIM
132.                DO 39 IA=1,IDIM
133.                   IF (SPD(K,IA) .GE. 100.)GO TO 39
134.                   DO 36 IB=1,IDIM
135.                      S(IB,NN3)=S(IB,NN3)-S(IB,M3+IA)*SPD(K,IA)
136.                      IF (S(IB,IB)-1. .GT. .001)S(IB,M3+IA)=0.
137. 36 CONTINUE
138. 39 CONTINUE
139.          GO TO 37
140. 30 CONTINUE
141.          GO TO 38
142. 37 NU=MM+1
143. 28 CONTINUE
144. 38 IF (MIM1 .EQ. 0)GO TO 62
145.          JILL=1
146.          I1=1-1
147.          DO 60 K=1,I1
148.             READ (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
149.             1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
150.             ISPT=NSPT(K)
151.             IF (LOC(ISPT) .NE. 1)GO TO 60
152.             ITER=(NSPT(K)-NROW(K))*IDIM
153.             NSPT(K)=NSPT(K)+1
154.             IF (JILL .GT. NSPD)GO TO 42
155.             DO 34 LI=JILL,NSPD
156.                NWD=NWS(LI)
157.                KEE=LI
158.                IF (NOER(NWD) .EQ. K)GO TO 35
159.                IF (NOER(NWD) .GT. K)GO TO 42
160. 34 CONTINUE
161. 35 JILL=KEE+1
162.          DO 41 IA=1,IDIM
163.             IF (SPD(KEE,IA) .GE. 100.)GO TO 41
164.             DO 40 IB=1,IDIM
165.                S(IB,NN3)=S(IB,NN3)-SPD(KEE,IA)*A(IA,ITER+IB)
166. 40 A(IA,ITER+IB)=0.
167. 41 CONTINUE
168. C GAUSSIAN ELIMINATION
169. 42 DO 21 IA=1,IDIM

```

```

170.          ID=ITER+IA
171.          DO 21 IB=1, IDIM
172.          C(IA,IB)=0.
173.          DO 19 IC=1, IDIM
174.          19 C(IA,IB)=C(IA,IB)+A(IC,ID)*B(IC,IB)
175.          21 CONTINUE
176.          NNO=NN1-ITER
177.          DO 24 IA=1, IDIM
178.          DO 23 IB=1, NNO
179.          D(IA,IB)=0.
180.          ID=ITER+IB
181.          DO 22 IC=1, IDIM
182.          22 D(IA,IB)=D(IA,IB)+C(IA,IC)*A(IC,ID)
183.          D(IA,IB)=D(IA,IB)/DET
184.          23 CONTINUE
185.          24 CONTINUE
186.          LYN=NNO/IDIM-1
187.          IF (LYN .EQ. 0) GO TO 32
188.          NRW=NROW(1)-1
189.          JS=1
190.          DO 31 L=1, LYN
191.          MSPT=ISPT+L
192.          L3=L*IDIM
193.          DO 27 M=JS, NN
194.          NRW=NRW+1
195.          IF (LOC(MSPT) .NE. LOC(NRW)) GO TO 27
196.          MM=M
197.          M3=(NRW-NROW(1)+1)*IDIM
198.          DO 25 IA=1, IDIM
199.          DO 25 IB=1, IDIM
200.          IL=L3+IB
201.          IM=M3+IB
202.          25 S(IA,IM)=S(IA,IM)-D(IA,IL)
203.          GO TO 29
204.          27 CONTINUE
205.          GO TO 32
206.          29 JS=MM+1
207.          31 CONTINUE
208.          32 DO 33 IA=1, IDIM
209.          S(IA,NN3)=S(IA,NN3)-D(IA,NNO)
210.          DO 33 IB=1, IDIM
211.          ID=ITER+IB
212.          A(IA,ID)=C(IB,IA)/DET
213.          33 S(IA,IB)=S(IA,IB)-D(IA,IB)
214.          BACKSPACE 10
215.          WRITE (10)NN1,((A(L1,L2),L1=1, IDIM),L2=1,NN1),
216.          1((B(L1,L2),L1=1, IDIM),L2=1, IDIM),DET
217.          IF (K .EQ. 1) GO TO 60
218.          BACKSPACE 10
219.          READ (10)NN1,((A(L1,L2),L1=1, IDIM),L2=1,NN1),
220.          1((B(L1,L2),L1=1, IDIM),L2=1, IDIM),DET
221.          60 CONTINUE
222.          67 DET=S(1,1)*S(2,2)-S(1,2)*S(1,2)
223.          IF (IDIM .EQ. 3) GO TO 64
224.          B(1,1)=S(2,2)
225.          B(2,2)=S(1,1)
226.          B(1,2)=-S(1,2)

```

```

227.      GO TO 65
228.      64 R(1,1)=S(2,2)*S(3,3)-S(2,3)*S(2,3)
229.      B(1,2)=S(1,3)*S(2,3)-S(1,2)*S(3,3)
230.      B(1,3)=S(1,2)*S(2,3)-S(1,3)*S(2,2)
231.      R(2,2)=S(1,1)*S(3,3)-S(1,3)*S(1,3)
232.      R(2,3)=S(1,2)*S(1,3)-S(1,1)*S(2,3)
233.      B(3,3)=DET
234.      DET=S(1,3)*R(1,3)+S(2,3)*R(2,3)+S(3,3)*B(3,3)
235.      P(3,2)=R(2,3)
236.      B(3,1)=R(1,3)
237.      65 R(2,1)=R(1,2)
238.      IF (I .EQ. NTNN)GO TO 70
239.      NNI=NN*IDIM+I
240.      N3=IDIM+1
241.      WRITE (11)NNI,((S(L1,L2),L1=1, IDIM),L2=N3,NN3)
242.      WRITE (10)NNI,((S(L1,L2),L1=1, IDIM),L2=N3,NN3),
243.      I((B(L1,L2),L1=1, IDIM),L2=1, IDIM),DET
244.      IF (MIM1 .EQ. 1)GO TO 69
245.      DO 66 K=1,I
246.      ISPT=NSPT(K)
247.      IF (LOC(ISPT) .EQ. 1+I)GO TO 68
248.      66 CONTINUE
249.      GO TO 70
250.      68 MIM1=1
251.      69 REWIND 10
252.      70 CONTINUE
253.      C .....
254.      C BACK-SUBSTITUTION
255.      C .....
256.      DO 76 I=1, IDIM
257.      D(I,1)=0.
258.      DO 71 J=1, IDIM
259.      71 D(I,1)=D(I,1)+B(I,J)*S(J,NN3)
260.      76 D(I,1)=D(I,1)/DET
261.      DO 75 I=2,NTNN
262.      NODE=NTNN-I+1
263.      BACKSPACE 10
264.      READ (10)NNI,((A(L1,L2),L1=1, IDIM),L2=1,NNI),
265.      I((B(L1,L2),L1=1, IDIM),L2=1, IDIM),DET
266.      BACKSPACE 10
267.      DO 77 IA=1, IDIM
268.      D(IA,1)=0.
269.      DO 72 IB=1, IDIM
270.      72 D(IA,1)=D(IA,1)+B(IA,IB)*A(IB,NN1)
271.      77 D(IA,1)=D(IA,1)/DET
272.      NRW1=NROW(NODE)-1
273.      NN=NNI/IDIM
274.      DO 74 J=1,NN
275.      JJ=(J-1)*IDIM
276.      NRW=NRW1+J
277.      MRW=NTNN-LOC(NRW)+1
278.      DO 73 IA=1, IDIM
279.      DO 73 IB=1, IDIM
280.      73 D(IA,1)=D(IA,1)-A(IA,JJ+IB)*D(IB,MRW)
281.      74 CONTINUE
282.      75 CONTINUE
283.      PRINT 300

```

```
284.      300 FORMAT ('NODAL DISPLACEMENTS')
285.          DO 85 J=1,NDIM
286.          PRINT 350,(D(J,K),K=NTNN,1,-1)
287.          85 CONTINUE
288.      350 FORMAT (12E10.5/(10X,11E10.5))
289.          RETURN
290.          END
```

END OF COMPILATION: NO DIAGNOSTICS.

```

1.      SUBROUTINE FCTOR
2.      COMMON/FUNKY/NROW(500),NREQ(500),NOER(500)
3.      COMMON/JNFER/NR,NS,NZ,NWOP, IDIM
4.      COMMON/RUTH/LOC(25000),NTNN,NB,NP,MISTY
5.      COMMON/JANET/NSPD,NWSD(100),SPD(100,3)
6.      COMMON/ABALN/D(3,600)
7.      DIMENSION A(3,300),B(3,3),C(3,3),NSPT(500),S(3,300)
8.      EQUIVALENCE (D(1,301),S)
9.      REWIND 10
10.     REWIND 11
11.     REWIND 12
12.     MEE=1
13.     PRINT 100,MISTY
14.     I00 FORMAT ('STARTING POINT OF GAUSSIAN ELIMINATION IS ROW',I4)
15.     IF (MISTY .EQ. 1)GO TO 8
16.     MRX=MISTY-1
17.     DO 7 I=1,MRX
18.     IF (NREQ(I) .EQ. NWSD(MEE))MEE=MEE+1
19.     READ  (12)NN3,((S(L1,L2),L1=1, IDIM),L2=1,NN3)
20.     READ  (10)NN1,((S(L1,L2),L1=1, IDIM),L2=1,NN1),
21.     I((B(L1,L2),L1=1, IDIM),L2=1, IDIM),DET
22.     READ  (11)NN1,((A(L1,L2),L1=1, IDIM),L2=1,NN1)
23.     NN=NN1/101M
24.     NSPT(I)=NROW(I)+NN
25.     NRW=NROW(I)-1
26.     DO 5 J=1,NN
27.     NRW=NRW+1
28.     IF (LOC(NRW) .LT. MISTY)GO TO 5
29.     NSPT(I)=NRW
30.     NNO=(NRW-NROW(I))*IDIM+1
31.     DO 4 L1=1, IDIM
32.     DO 4 L2=NNO,NN1
33.     4 S(L1,L2)=A(L1,L2)
34.     BACKSPACE 10
35.     WRITE (10)NN1,((S(L1,L2),L1=1, IDIM),L2=1,NN1),
36.     I((B(L1,L2),L1=1, IDIM),L2=1, IDIM),DET
37.     IF (I .EQ. MRX)GO TO 6
38.     BACKSPACE 10
39.     READ  (10)NN1,((S(L1,L2),L1=1, IDIM),L2=1,NN1),
40.     I((B(L1,L2),L1=1, IDIM),L2=1, IDIM),DET
41.     GO TO 6
42.     5 CONTINUE
43.     6 PRINT 101,I,NROW(I),NSPT(I)
44.     101 FORMAT (' I = ',I3,' ** NROW(I) = ',I5,' ** NSPT(I) = ',I5)
45.     7 CONTINUE
46.     REWIND 10
47.     8 DO 39 I=MISTY,NTNN
48.     NSPT(I)=NROW(I)
49.     READ  (12)NN3,((S(L1,L2),L1=1, IDIM),L2=1,NN3)
50.     NN=NN3/IDIM-1
51.     IF (MEE .GT. NSPD)GO TO 17
52.     IF (NREQ(I) .NE. NWSD(MEE))GO TO 11
53.     DO 10 K=1, IDIM
54.     IF (SPD(MEE,K) .GE. 100.)GO TO 10
55.     DO 9 L=1, IDIM

```

```

56.          S(L,NN3)=S(L,NN3)-S(L,K)*SPD(MEE,K)
57.          S(K,L)=0.
58.          9 S(L,K)=0.
59.          S(K,K)=1.
60.          S(K,NN3)=SPD(MEE,K)
61.          10 CONTINUE
62.          MEE=MEE+1
63.          IF (MEE .GT. NSPD)GO TO 17
64.          11 NU=1
65.          NRW=NROW(1)-1
66.          DO 16 K=MEE,NSPD
67.          NWS=NWS(K)
68.          DO 14 L=NU,NN
69.          NRW=NRW+L
70.          IF (LOC(NRW) .GT. NOER(NWS))GO TO 16
71.          IF (LOC(NRW) .NE. NOER(NWS))GO TO 14
72.          MM=L
73.          M3=L*IDIM
74.          DO 13 IA=1,IDIM
75.          IF (SPD(K,IA) .GE. 100.)GO TO 13
76.          DO 12 IB=1,IDIM
77.          S(IB,NN3)=S(IB,NN3)-S(IB,M3+IA)*SPD(K,IA)
78.          IF (S(IB,IB)-1. .GT. .001)S(IB,M3+IA)=0.
79.          12 CONTINUE
80.          13 CONTINUE
81.          GO TO 15
82.          14 CONTINUE
83.          GO TO 17
84.          15 NU=MM+1
85.          16 CONTINUE
86.          17 IF (I .EQ. 1)GO TO 36
87.          JILL=1
88.          I1=1-1
89.          DO 34 K=1,I1
90.          READ (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
91.          I((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
92.          ISPT=NSPT(K)
93.          IF (LOC(ISPT) .NE. 1)GO TO 34
94.          ITER=(NSPT(K)-NROW(K))*IDIM
95.          NSPT(K)=NSPT(K)+1
96.          IF (JILL .GT. NSPD)GO TO 22
97.          DO 18 LI=JILL,NSPD
98.          NWD=NWS(LI)
99.          KEE=LI
100.         IF (NOER(NWD) .EQ. K)GO TO 21
101.         IF (NOER(NWD) .GT. K)GO TO 22
102.         18 CONTINUE
103.         21 JILL=KEE+1
104.         DO 20 IA=1,IDIM
105.         IF (SPD(KEE,IA) .GE. 100.)GO TO 20
106.         DO 19 IB=1,IDIM
107.         S(IB,NN3)=S(IB,NN3)-SPD(KEE,IA)*A(IA,ITER+IB)
108.         19 A(IA,ITER+IB)=0.
109.         20 CONTINUE
110.         C GAUSSIAN ELIMINATION
111.         22 DO 25 IA=1,IDIM
112.         ID=ITER+IA

```

```

113.      00 25 IB=1, IDIM
114.      C(IA,IB)=0.
115.      DO 23 IC=1, IDIM
116.      23 C(IA,IB)=C(IA,IB)+A(IC, ID)*B(IC, IB)
117.      25 CONTINUE
118.      NNO=NN1-ITER
119.      DO 27 IA=1, IDIM
120.      DO 27 IB=1, NNO
121.      D(IA,IB)=0.
122.      ID=ITER+IB
123.      DO 26 IC=1, IDIM
124.      26 D(IA,IB)=D(IA,IB)+C(IA, IC)*A(IC, ID)
125.      D(IA,IB)=D(IA,IB)/DET
126.      27 CONTINUE
127.      LYN=NNO/IDIM-1
128.      IF (LYN .EQ. 0) GO TO 32
129.      NRW=NROW(1)-1
130.      JS=1
131.      DO 31 L=1, LYN
132.      MSPT=1SPT+L
133.      L3=L*IDIM
134.      DO 29 M=JS, NN
135.      NRW=NRW+1
136.      IF (LOC(MSPT) .NE. LOC(NRW)) GO TO 29
137.      MM=M
138.      M3=(NRW-NROW(1)+1)*IDIM
139.      DO 28 IA=1, IDIM
140.      DO 28 IB=1, IDIM
141.      IL=L3+IB
142.      IM=M3+IB
143.      28 S(IA, IM)=S(IA, IM)-D(IA, IL)
144.      GO TO 30
145.      29 CONTINUE
146.      GO TO 32
147.      30 JS=MM+1
148.      31 CONTINUE
149.      32 DO 33 IA=1, IDIM
150.      S(IA, NN3)=S(IA, NN3)-D(IA, NNO)
151.      DO 33 IB=1, IDIM
152.      ID=ITER+IB
153.      A(IA, ID)=C(IB, IA)/DET
154.      33 S(IA, IB)=S(IA, IB)-D(IA, IB)
155.      BACKSPACE 10
156.      WRITE (10) NN1, ((A(L1, L2), L1=1, IDIM), L2=1, NN1),
157.      1((B(L1, L2), L1=1, IDIM), L2=1, IDIM), DET
158.      IF (K .EQ. 1) GO TO 34
159.      BACKSPACE 10
160.      READ (10) NN1, ((A(L1, L2), L1=1, IDIM), L2=1, NN1),
161.      1((B(L1, L2), L1=1, IDIM), L2=1, IDIM), DET
162.      34 CONTINUE
163.      36 DET=S(1, 1)*S(2, 2)-S(1, 2)*S(1, 2)
164.      IF (IDIM .EQ. 3) GO TO 64
165.      B(1, 1)=S(2, 2)
166.      B(2, 2)=S(1, 1)
167.      B(1, 2)=-S(1, 2)
168.      GO TO 65
169.      64 B(1, 1)=S(2, 2)*S(3, 3)-S(2, 3)*S(2, 3)

```



```

170.      B(1,2)=S(1,3)*S(2,3)-S(1,2)*S(3,3)
171.      B(1,3)=S(1,2)*S(2,3)-S(1,3)*S(2,2)
172.      B(2,2)=S(1,1)*S(3,3)-S(1,3)*S(1,3)
173.      B(2,3)=S(1,2)*S(1,3)-S(1,1)*S(2,3)
174.      B(3,3)=DET
175.      DET=S(1,3)*B(1,3)+S(2,3)*B(2,3)+S(3,3)*B(3,3)
176.      B(3,2)=B(2,3)
177.      B(3,1)=B(1,3)
178.      65 B(2,1)=B(1,2)
179.      IF (I .EQ. NTNN)GO TO 39
180.      NN1=NN*IDIM+1
181.      N3=IDIM+1
182.      WRITE (11)NN1,((S(L1,L2),L1=1,IDIM),L2=N3,NN3)
183.      WRITE (10)NN1,((S(L1,L2),L1=1,IDIM),L2=N3,NN3),
184.      1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
185.      REWIND 10
186.      39 CONTINUE
187.      C BACK-SUBSTITUTION
188.      DO 76 I=1,IDIM
189.      D(I,1)=0.
190.      DO 71 J=1,IDIM
191.      71 D(I,1)=D(I,1)+B(I,J)*S(J,NN3)
192.      76 D(I,1)=D(I,1)/DET
193.      DO 75 I=2,NTNN
194.      NODE=NTNN-I+1
195.      BACKSPACE 10
196.      READ (10)NN1,((A(L1,L2),L1=1,IDIM),L2=1,NN1),
197.      1((B(L1,L2),L1=1,IDIM),L2=1,IDIM),DET
198.      BACKSPACE 10
199.      DO 77 IA=1,IDIM
200.      D(IA,1)=0.
201.      DO 72 IB=1,IDIM
202.      72 D(IA,1)=D(IA,1)+B(IA,IB)*A(IB,NN1)
203.      77 D(IA,1)=D(IA,1)/DET
204.      NRW=NROW(NODE)-1
205.      NN=NN1/IDIM
206.      DO 74 J=1,NN
207.      JJ=(J-1)*IDIM
208.      NRW=NRW+J
209.      MRW=NTNN-LOC(NRW)+1
210.      DO 73 IA=1,IDIM
211.      DO 73 IB=1,IDIM
212.      73 D(IA,1)=D(IA,1)-A(IA,JJ+IB)*D(IB,MRW)
213.      74 CONTINUE
214.      75 CONTINUE
215.      RETURN
216.      END

```

END OF COMPILATION: NO DIAGNOSTICS.

APPENDIX B
DATA CARDS PREPARATION GUIDE

The sequential arrangement of the data cards supplied during each program run as well as the input items punched on each card are shown in Table A. Regarding the input parameters, the following explanatory notes are made:

Card # 1

- | | | |
|------|---|---|
| ISØ | - | 1 in anisotropic cases; 0 in nonhomogeneous cases; -1 in isotropic cases. |
| NR | - | Total number of annular surfaces dividing the Brazilian test cylinder. |
| NS | - | Total number of sectors the circular face of the cylinder or its quadrant is divided into. |
| NZ | - | Total number of divisions in the Z-direction; 0 in 2-dimensional problems. |
| NWØP | - | 1 if whole circular face of cylinder is involved in analysis; 0 if only a quadrant is involved in analysis. |
| NDIM | - | 2 in 2-dimensional cases; 3 in 3-dimensional cases. |
| NCYC | - | Total number of load cycles. |
| NER | - | Least total number of similar and consecutively numbered elements in finite element mesh; 1 in nonhomogeneous and anisotropic cases. This parameter is inserted to avoid having to completely treat each element all over again even if all are the same. |
| NSS | - | Difference between the numbers of a loaded top node and the diametrically opposite bottom node; 0 if NWØP is 0 or if finite element mesh is symmetric about plane of diametral loads. |

Table A Input Cards Arrangement

| CARD NUMBER | INPUT DATA | FORMAT |
|---|---|-------------|
| 1 | ISØ, NR, NS, NZ, NWØP, NDIM, NCYC, NER, NSS | 9I5 |
| 2* | TCRT, SCRT, EM1, EM2, CS1, CS2, PR1, PR2 | 8F10.0 |
| 3 | PØSM, TRN, DIA, H, NTEP, NGØT | 4F10.0, 2I5 |
| 4 | NST1, NST2, MST1, MST2 | 4I5 |
| 5a 5b if heeded 5c if heeded . | RC(I), I=1, NR1 . . . | 8F10.0 |
| 6a 6b if heeded 6c if heeded . | TC(I), J=1, NS1 . . . | 8F10.0 |
| 7a** 7b if heeded 7c if heeded . | ZC(I), I=1, NZ1 . . . | 8F10.0 |
| 8*** | N | I10 |
| 9**** | ALF1, ALF2, ALF3, C1, C2, C3, E1, E2 | 8F10.0 |
| 10**** | E3, P1, P2, P3, G1, G2, G3 | 7F10.0 |

* Spaces for EM1, EM2, CS1, CS2, PR1, PR2 are left blank in anisotropic cases.

** Supplied in 3-dimensional cases only.

*** Supplied in nonhomogeneous cases only.

**** Supplied in anisotropic cases only.

Card # 2

- TCRT - Ratio of allowable tension to allowable compression.
- SCRT - Ratio of allowable shear to allowable compression.
- EM1, EM2 - Range of values of elastic moduli.
- CS1, CS2 - Range of values of allowable compressive stresses.
- PR1, PR2 - Range of values of Poisson's ratios.

Card # 3

- PØSM - Factor to be multiplied to the stiffness matrices of failed elements to obtain that portion of the stiffness matrices to be subtracted from the global stiffness matrix.
- TRN - Maximum deviation which an element load factor can have from the critical load factor for the element to be considered failed.
- DIA - Diameter of cylinder.
- H - Length or thickness of cylinder.
- NTEP - 0 if failure criterion in which the elastic modulus across tension cracks is reduced to zero is to be applied; 1 if the old failure criterion (described in first annual report) is to be applied; always 1 in anisotropic cases.
- NGØT - 0 if both finite element mesh and material property are symmetric about a nodal plane perpendicular to the Z-axis; 1 if no symmetry exists. With this parameter, only one-half of the cylinder length or thickness need be analyzed if symmetry exists.

Card # 4

- NST1, NST2
MST1, MST2
- Pairs of points between which magnitudes of strain, usually the horizontal strain at the center point of each end of the cylinder, are desired.

Card # 5a, etc.

- RC(1)
- 0 for solid cylinders; inner radius of hollow cylinders.
- RC(2) → RC(NR1)
- Radial coordinates of annular surfaces.

Card # 6a, etc.

- TC(I), I=1, NS1
- Circumferential coordinates, in degrees, of nodal radial planes measured clockwise from plane of loads. TC(1) = 0.

Card # 7a, etc.

- ZC(I), I=1, NZ1
- Z-coordinates of nodal circles using an end circle as datum. ZC(1) = 0.

Card # 8

- N
- Any 10-digit integer. This is the starting point of random number generator.

Card # 9

- ALF1
- Angle, in degrees, which the line A'B' (see Figure 2.4 of semi-annual report) makes with the plane of the loads.
- ALF2, ALF3
- The angles α_2 and α_3 , in degrees, defined in Figure 2.4 of semi-annual report.
 - These three angles define the directions of anisotropy.

- C1, C2, C3 - Allowable compressive stresses in the direction of axes of anisotropy.
- E1, E2 - Elastic moduli in the direction of axes of anisotropy, 1, 2.

Card # 10

- E3 - Elastic modulus in the direction of axis of anisotropy 3.
- P1, P2, P3 - Poisson's ratios along axes of anisotropy.
- G1, G2, G3 - Shear moduli associated with directions of anisotropy.

The entries below the heading `FORMAT` in Table A are called format specifications. These serve as instructions to the user on how the input data are to be punched in the data cards. In the present program, each item in the input list has the format specification of the form `Fw.d` or `Iw` in which `w` is the number of column spaces on the card reserved for each item. Given the `w`'s of the entire input list, it is an easy matter to determine the exact columns on the card where each input item is to be punched. For example, in card # 3 one punches `PØSM` in columns 1-10, `TRN` in columns 11-20, `DIA` in columns 21-30, `H` in columns 31-40, `NTEP` in columns 41-45, and `NGØT` in columns 46-50. Integers (items with `I` specifications) are punched right-justified within the spaces allocated to them. Floating-point numbers (items with `F` specifications) may be punched with or without the decimal point anywhere within their allocated spaces. If punched without a decimal point, the computer will read the floating-point number as if there was a decimal point `d` places to the left of the right-most digit. The number before a `F` or an `I` is a repetition factor; for instance, `(415) = (15, 15, 15, 15)`.

Input data listings are shown in Chapter 2.