AD-768 649

FINITE ELEMENT ANALYSIS OF STRESSES

DEFORMATIONS AND PROGRESSIVE FAILURE OF NON-HOMOGENEOUS FISSURED ROCK. VOLUME II. COMPUTER PROGRAMS USER'S MANUAL

OHIO STATE UNIVERSITY

MARCH 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

OSURF-3177-73-
2F

STRESSES, DEFORMATIONS AND PROGRESSIVE FAILURE OF NON-HOMOGENEOUS FISSURED ROCK

Final Report
Volume 2 – Computer Program User's Manual
March 1973

U.S. BUREAU OF MINES
Contract Number HO210017

Sponsored by
ADVANCED RESEARCH PROJECTS AGENCY
ARPA Order No. 1579, Amend. No. 2
Program Code 1F10

Principal Investigators

R.S. Sandhu
T.H. Wu
J.R. Hooper

THE OHIO STATE UNIVERSITY RESEARCH FOUNDATION
1314 KINNEAR ROAD COLUMBUS, OHIO 43212

UNCLASSIFIED
Security Classification

3200.8 (Att 1 to Encl 1)
Mar 7, 66

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) The Ohio State University Research Foundation	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Finite Element Analysis of Stresses. Deformations and Progressive Failure of Non-Homogeneous Fissured Rock - Volume 2, Computer Programs User's Manual

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report - March 31, 1973

5. AUTHOR(S) (First name, middle initial, last name)

Ranbir S. Sandhu

6. REPORT DATE March 31, 1973	7a. TOTAL NO. OF PAGES 138	7b. NO. OF REFS None
8a. CONTRACT OR GRANT NO. HO 210017 b. PROJECT NO. RF 3177 A1 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) OSURF-3177-73-2F 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES Computer programs available on Magnetic tape, see AD-768 651.	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Washington, D.C. 20301

13. ABSTRACT The objective of this research program was development of finite element procedures for prediction of stresses and deformations in the vicinity of underground excavation.

Two models of rock behavior were selected. In one the rock is treated as isotropic elastic-plastic following a generalized Mohr-Coulomb law and in the other the rock is isotropic elastic-brittle subject to Griffith or modified Griffith failure theory.

For each model, mathematical relationships governing stress-strain behavior and progressive failure were developed. Finite element computer programs incorporating each of the two models were coded. Preliminary to this development, a revised version of Zienkiewicz's no-tension analysis was programmed.

The procedures developed allow for initial stresses in rock, arbitrary shape and size of the opening, any given sequence of construction/excavation, material nonhomogeneity and progressive failure.

This report is in three parts: Volume 1-Main Document; Volume 2-Computer Program User's Manual; Volume 3-Computer Programs.

Volume 2 -Computer Program User's Manual, contains descriptions, instructions for usage and fortran listings of computer programs used to obtain the numerical results presented and discussed in Volume 1 -Main Document.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
computation						
crack propagation						
deformation						
elasticity						
excavation						
failure						
finite element method						
foundation						
fracture						
mining						
plasticity						
progressive failure						
research						
rock mechanics						
stresses						
structural supports						
tunnels						
underground excavation						

FINAL REPORT

ARPA Order Number: 1579, Amend 2

Program Code Number: 1F10

Name of Contractor:
The Ohio State University
Research Foundation

Effective Date of Contract:
February 1, 1971

Contract Expriation Date:
March 31, 1973

Amount of Contract:
$71,613.00

Contract Number: HO210017

Principal Investigators: R. S. Sandhu
T. H. Wu
J. R. Hooper
Telephone Number: (614) 422-7531

Project Scientist or Engineer:
R. S. Sandhu
Telephone Number: (614) 422-7531

Short Title of Work:
Stresses, Deformations and
Progressive Failure of
Nonhomogeneous Fissured Rock

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the United States Bureau of Mines under Contract Number HO210017.

ic

FOREWORD

The final report on work done under contract HO210017 between the Ohio State University and the United States Bureau of Mines is in three parts as follows:

Volume 1:	Main Document
Volume 2:	Computer Program User's Manual
Volume 3:	Computer Programs

Volume 2 of the report contains documentation relating to three computer programs including fortran listings and description of program structure.

It is obvious that these computer programs be used only under the conditions and assumptions for which they were developed. These are described in Volume 1 of this report. Although the programs have been tested by applications to several problems, no warranty is made regarding the accuracy and reliability of the programs and no responsibility is assumed by the authors or by the sponsors of this research project.

The technical report summary is included in all three volumes of the report.

R. S. Sandhu
Principal Investigator

TECHNICAL REPORT SUMMARY

Program Objectives

The objective of this research program was development of finite element procedures to predict stresses, deformations and progressive failure of rock associated with underground excavations. For applicability to arbitrary sequence of excavation operations, it was necessary that the procedures developed allow for arbitrary initial stresses in rock, arbitrary size and shape of the opening and progressive failure. Plane strain conditions and two different types of material behavior were considered. Rock was treated as an isotropic elastic-plastic generalized Mohr-Coulomb material in one model and as an elastic-brittle material following Griffith theory of fracture in the other.

Background

In previous applications of the finite element method to rock mechanics, elastic-plastic behavior of rock has been modeled as nonlinear elastic for computational convenience. Further, it was assumed that the results of a one-dimensional test could be generalized to three-dimensional analysis through the use of an equivalent stress-equivalent strain curve. In some applications, two stress or strain parameters were used. These procedures are unsatisfactory. Assumption of isotropic elasticity assumes that the principal directions of stress and strain coincide. In plasticity this is not true. Also, rock behavior is characterized by a significant part of deformation being irreversible. For this reason, the mechanical behavior in unloading is different from that in loading. For rock with preexisting joints or developing tensile cracks, a 'no tension' procedure is often adopted. In this method, a linear elastic solution is obtained and all tensile stress redistributed simultaneously. Actually, as cracking progresses, the rock on either side of the crack is relieved of stress and a stress concentration develops near the crack tip. Conventional procedures ignore these effects and the progressive nature of crack development, leading to erroneous conclusions regarding stresses around underground openings.

Accomplishments Under the Present Program

The research conducted under this contract has resulted in development of computer programs based on more realistic simulation of material behavior. The incremental theory of plasticity has been used to characterize the stress-strain behavior of elastic-plastic rock. Role of kinematic constraint of plane strain in development of residual stresses in rock has been examined on the basis of Hill's theory. New techniques have been developed for study of initiation and propagation of fracture in rock following Griffith's theory or the modified Griffith theory. Allowing for sequential fracture of various elements in a system, the effect of progressive stress redistribution in the remaining system is correctly incorporated. Arbitrary initial stress states, arbitrary sequence of excavation (or construction), arbitrary size and shape of opening, and nonhomogeneous material properties were allowed for. The actual construction operations can be simulated. The procedures developed were applied to several typical problems in rock mechanics as well as to some theoretical and laboratory studies for the purpose of verification and illustration. These were used to carry out parametric studies to examine the influence of rock properties upon the stresses in steel supports in a tunnel.

DOD Implications

The procedures developed provide useful means for study of stability of underground excavations based on stresses and deformations associated with the mining operations, structural support evaluation, safety analyses of openings, study of blasting effectiveness under certain conditions, evaluation of mining sequences, study of vulnerability and serviceability of underground structures etc.

Organization of the Report

This report is in three parts as follows:

Volume 1 - Main Document
Volume 2 - Computer Program User's Manual
Volume 3 - Computer Programs

Volume 1 contains the main body of the report including the theoretical development, program verification and case studies. Chapter I reviews previous efforts in the general research area and describes the objectives and methods of the present research in the historical context. Chapter II describes the mechanical behavior of rock and the idealizations used in the research under report. The basis and methods of the finite element theory are briefly discussed in Chapter III leading to the formulation of matix equations. Chapter IV gives details of the analysis technique for isotropic elastic-plastic generalized Mohr-Coulomb rock materials and Chapter V gives the numerical analysis procedure for jointed rock and rock subjected to progressive fracture following Griffith or modified Griffith theory. Examples of application are included in Chapters IV and V. Chapter VI presents application of the elastic-plastic analysis computer program to a parametric study to evaluate the influence of rock properties on stresses in steel supports for specified initial stresses and design of the opening.

In the original proposal, model testing to verify some aspects of rock behavior under plane strain conditions was foreseen. The effort under the present contract covered procurement of suitable plane strain test equipment and design of suitable test material. Appendix B includes a report on this effort.

Volume 2 of the report contains description of the three computer programs developed under the contract along with fortran listings and instructions for input preparation. The input definition and the listings are for the IBM 370/165 version.

The programs are the primary content of volume 3. These are available on magnetic tape from DDC-TC, U.S. Department of Commerce, Springfield, Virginia 22151, telephone (703) 321-8517.

ACKNOWLEDGEMENTS

The research was supported by the U.S. Government through the Advanced Research Projects Agency, ARPA, and its agent the U.S. Bureau of Mines, Department of the Interior. James J. Olson, Twin Cities Mining Research Center, was the ARPA program coordinator and Dr. William J. Karwoski, Spokane Mining Research Center, was the Project Officer. In early stages of work. Dr. Syd Peng, Twin Cities Mining Research Center acted as the Project Officer. Constant cooperation and several constructive suggestions from these individuals are gratefully appreciated.

A number of graduate students worked on the project. The contributions of Messrs. Ram Dhan Singh, S. W. Huang and Kamar Jit Singh were specially noteworthy. The Instruction and Research Computer Center of the Ohio State University provided the computational facilities.

R. S. Sandhu
Project Supervisor

TABLE OF CONTENTS

		Page
FOREWORD		i
TECHNICAL REPORT SUMMARY		ii
ACKNOWLEDGEMENTS		vi
TABLE OF CONTENTS		vii
CHAPTER I:	NOTENS--A Computer Program for Two-Dimensional Analysis of Rock as a No Tension Material	1
	1.1. Purpose and Capability	1
	1.2. Program Organization	1
	1.3. Input Data	4
	1.4. Output Information	8
	1.5. Limitations	8
	1.6. Fortran Listing	9
CHAPTER II:	ELPL--A Computer Program for Plane Strain Analysis of Progressive Failure of Nonhomogeneous Rock Following Incremental Theory of Plasticity, Plane Strain Conditions, Generalized Mohr-Coulomb Yiel Criterion	30
	2.1. Purpose and Capability	30
	2.2. Program Organization	30
	2.3. Input Data	36
	2.4. Output Information	45
	2.5. Fort an Listing	46
CHAPTER III:	GRIFTH--A Computer Program for Two-Dimensional Analysis of Progressive Failure of Rock Following Griffith and Modified Griffith Theory	76
	3.1. Purpose and Capability	76
	3.2. Organization	76
	3.3. Input Data	78
	3.4. Output Information	86
	3.5. Fortran Listing	87

CHAPTER I: NOTENS-- A Computer Program for Analysis of Rock as a No-Tension Material

1.1. Purpose and Capability

This computer program is based on a modification of Zienkiewicz's approach (Zienkiewicz et.al. 1968). A linear elastic analysis of a given two-dimensional system is carried out to determine the regions of tensile stress. The tensions, wherever they occur, are replaced by equivalent nodal point loads and the tensile stresses in the corresponding elements are neutralized. For these equivalent loads, the elastic analysis is repeated. This iterative process is continued until all tensile stresses are liquidated. The modification to the original approach is discussed in Part I: Technical Report.

The program allows for an arbitrary initial stress field, gravity loads, concentrated loads, distributed loads applied to the boundary, temperature loads and temperature dependent material properties.

1.2. Organization

The program is in Fortran language. Scratch tapes are 1 and 2. Tapes 5 and 6 are used for input/output respectively. The listing in section 1.6 used double precision for real numbers. The program capacity can be altered by changing the dimension of arrays AA and IA. These correspond to the total locations required in real arrays and integer arrays respectively. NTOT, MTOT at lines MAIN 23, MAIN 24 are set equal to the dimension of AA, IA respectively.

The program consists of the following units:

a. Program MAIN

In this unit, the control information is read in and the location of various dimensioned variables defined in the system arrays AA and IA.

b. Subroutine INPT

This subroutine is called by the MAIN. In this, information regarding material properties; nodal point coordinates, loads, temperatures, boundary conditions; element geometry, material and initial stresses is read in. Missing information is generated. Information regarding boundary pressure is read in. Dimensions of the matrix for solution of stiffness equations are defined and subroutine SOLVE is called to complete the solution proeess.

c. Subroutine SOLVE

Called by INPT after reading in all data, this subroutine sets up the system stiffness matrix in blocks. It ealls subroutines ONED and ELEMEN to obtain stiffness properties for one-dimensional and two-dimensional elements respectively. The stiffness is assembled by the direet stiffness procedure and modified for specified boundary displacements. The matrix is triangularized by ealling SYMBAN (1). Subroutine LOAD is called to recover the system load veetor. Reduetion of the load vector and baek-substitution are aeeomplished by calling SYMBAN (2). The cumulative as well as the ineremental displacements are pointed out. Subroutine STRESS is called for evaluation and print out of stresses. Because the solution is obtained by elimination of tension in an iterative sequenee, the steps from calling LOAD to caleulation of stress are repeated a preset number of times or until convergenee is obtained.

d. Subroutine ONED

Called by SOLVE, this subroutine sets up the stiffness matrix for a one-dimensional element.

e. Subroutine ELEMEN

Called by SOLVE, this subroutine calculates the stiffness matrix for two-dimensional elements. These maybe triangular elements or quadrilaterals made up of four constant strain triangles. If other types of quadrilateral elements are to be used, this subroutine has to be replaced. Linear isotropic elasticity is assumed. Loads due to initial stresses or temperature changes from a reference temperature, and gravity loads are calculated in this subroutine.

f. Subroutine MODIFY

Called by SOLVE, this subroutine modifies the system stiffness matrix to allow for prescribed displacement boundary conditions.

g. Subroutine SYMBAN

This subroutine is called by SOLVE. Called with augment 1, it triangularizes the stiffness matrix and called with augment 2, it backsubstitutes to evaluate the solution for displacements for a given load vector.

h. Subroutine LOAD

This subroutine called by SOLVE, sets up the system load vector including nodal point loads, distributed boundary pressure loads, as well as unbalanced element loads, initial stresses, temperature stresses and gravity loads.

j. Subroutine STRESS

Stresses in x-y coordinate system as well as principal stresses are calculated in the STRESS subroutine called by SOLVE. The stresses are checked for tension, and if necessary, the tensions are replaced by equivalent unbalanced stresses to be transformed into load in the next cycle. A check on convergence of the iterative scheme is applied and the maximum unbalanced stress in any element is printed out.

1.3. Input Data

Input to the program NOTENS consists of the following sequence of cards.

a. First Card. Job Title (18A4).

This card gives the descriptive identification of the job.

b. Second Card. Control Information (4I5, 3F10.2, I5, D15.4)

Information	Columns
Total number of nodal points (NUMNP)	1-5
Total number of elements (NUMEL)	6-10
Number of different materials (NUMMAT)	11-15
Number of pressure boundary cards (NUMPL)	16-20
Acceleration in x-direction (ACELR)	21-30
Acceleration in y-direction (ACELZ)	31-40
Reference (stress-free) temperature (Q)	41-50
Maximum number of approximations or iterations allowed (NP)	51-55
Tolerance for convergence of iterations (TOL)	56-70

c. Material Property Cards

One set of cards must be provided for each material. In each set:

i. First card (2I5, 1F10.3, I5) gives the following information:

Material identification number ($1 \leq$ MTYPE ≤ 10)	1-5
Number of temperature cards (NTC)	6-10

Mass density of the material (RO)	11-20
Material code (MTC): 1 for no-tension material 0 otherwise	21-25

ii. Subsequent cards, <u>one for each temperature</u>, the number of data sets defined in columns 6-10 of the first card for the material, will carry the the following information (4F10.3):

<u>Information</u>	<u>Columns</u>
Temperature	1-10
Elastic Modulus	11-20
Poisson's ratio	21-30
Coefficient of thermal expansion (or cross-sectional area of 1-D element)	31-40

d. Nodal Point Cards (I5, F5.1, 5F10.4)

Nodal point number (N)	1-5
Code of nodal point (CODE)	6-10
X-ordinate (R)	11-20
Y-ordinate (Z)	21-30
UR	31-40
UZ	41-50
Temperature (T)	51-60

If the number in columns 6-10 is

0. UR is the specified X-load and UZ is the specified Y-load

1. UR is the specified X-displacement and UZ is the specified Y-load

2. UR is the specified X-load and UZ is the specified Y-displacement

3. UR is the specified X-displacement and UZ is the specified Y-displacement

All loads are considered to be total forces acting on an element of unit thickness. Nodal point cards must be in numerical sequence. If cards are omitted, the omitted nodal points are generated at equal intervals along a straight line between the defined nodal points. Similarly, the corresponding temperatures are determined by linear interpolation. The codes (CODE) of these generated nodal points, as well as UR and UZ, are set equal to 0.

e. Element cards

i. Material type information of elements (16I5)

Every 5 columns of each data card give the material identification number of each element in sequence. Each card contains 16 material identification numbers (80 columns of 16 elements).

ii. Nodal points and initial stresses (5I5, 5X, 3F10.0). One card for each element.

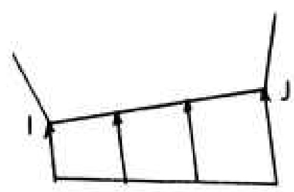
Information	Columns
Number of the element	1-5
Nodal point I	6-10
Nodal point J	11-15
Nodal point K	16-20
Nodal point L	21-25
Initial σ_{xx}	31-40
Initial σ_{yy}	41-50
Initial σ_{xy}	51-60

Nodal points I, J, K, L are corners of each individual element in a counterclockwise order for a right-handed system of co-ordinates. For 1-D elements, nodal points K and L are set same as J and I respectively. In the case of triangular elements, L is set to be same as K. The element cards must be in numerical sequence. Any element cards omitted will be automatically generated in the program by incrementing each of the I, J, K and L nodal points by 1. The corresponding initial stresses of the generated elements are calculated by the program using linear interpolation. Such interpolation is done basing upon element number rather than distance.

f. Pressure Boundary cards (2I5, 2F10.3)

One card for each boundary element, which is subjected to a normal pressure, will carry the following information:

Information	Columns
Nodal point I (IBC)	1-5
Nodal point J (JBC)	6-10
Normal pressure at I (PR)	11-20
Normal pressure at J (PR)	21-30



As shown in the sketch, the boundary element must be on the left as one progresses counter-clockwise from I to J. Surface tensile force is input as a negative pressure.

1.4. Output Data

The following information is developed and printed by the program:

a. Printing of input (and generated)data

b. Incremental and cumulative nodal displacements of each iteration

c. Stresses at center of each element and the unbalanced force of the elements during each iteration

d. Maximum unbalanced force of the elements for the iteration

e. In the case where maximum unbalanced force is less than the set tolerance (TOL), convergence is noted and the number of cycles to reach this is printed.

1.5. Limitations

Limitations of the approach are discussed in detail in Part I-Technical Report. In this approach it is assumed that all the elements having tensile stress crack and are relieved of tension simultaneously. This is unrealistic. Actually the process of cracking will be sequential with cracking of each element influencing the stress distribution and consequently the continuation of the cracking sequence. The program included here eliminates some of the procedural errors of earlier attempts. However, it cannot predict sequential or progressive development of cracks. Also convergence is often very slow and the advantage of using the same system stiffness matrix throughout the iterative computation is lost because of the large number of iterations required.

1.6. Fortran Listing

```
C     ****************************************************************
C     PROGRAM NOTENS
C     NO TENSION ANALYSIS FOR PLANE STRESS AND PLANE STRAIN.
C     LINEAR PRESSURE BOUNDARY
C     PROGRAMMED BY R.S.SANDHU, R.D.SINGH AND H.LIU. THE OHIO STATE
C     UNIVERSITY, COLUMBUS.
C     THE FORMULATION IS DOCUMENTED IN THE FINAL REPORT DATED MARCH 31,
C     1973, ON CONTRACT H0210017 BETWEEN THE OHIO STATE UNIVERSITY AND
C     THE UNITED STATES BUREAU OF MINES SUPPORTED BY THE ADVANCED
C     RESEARCH PROJECTS AGENCY. INSTRUCTIONS FOR USE OF THE PROGRAM
C     ARE CONTAINED IN PART II OF THE REPORT.
C     ****************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON AA(10000),IA(1500)
      COMMON/ONE/ACELR,ACELZ,TEMP,W,TOL,VOL,NUMNP,NUMEL,NUMMAT,NUMPC,
     *MTYPE,LLL,N,MBAND,NUMBLK,KKK,NCHECK,NP,NL,NEQ,HED(18),NPC
      COMMON/TWO/C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),XC,YC,
     *EE(3),LM(4),E(5,4,10),RC(10),NTC(10),MTC(10)
      COMMON/THREE/MTOT,NTOT,N11,M4,STOP
      READ(5,1000) HED,NUMNP,NUMEL,NUMMAT,NUMPC,ACELR,ACELZ,Q,NP,TOL
      WRITE(6,2000)HED,NUMNP,NUMEL,NUMMAT,NUMPC,ACELR,ACELZ,Q,NP,TOL
      NTOT=10000
      MTOT=1500
      NPC=NUMPC
      IF(NUMPC.EQ.0)NPC=1
      N1=1
      N2=N1+NUMNP
      N3=N2+NUMNP
      N4=N3+NUMNP
      N5=N4+NUMNP
      N6=N5+NUMNP
      N7=N6+2*NPC
      N8=N7+6*NUMEL
      N9=N8+NUMNP
      N10=N9+2*NUMNP
      N11=N10+2*NUMNP
      M1=1
      M2=M1+5*NUMEL
      M3=M2+NPC
      M4=M3+NPC
      NEQ=2*NUMNP
      JJ=M4-MTOT
      IF(JJ.LE.0)GO TO 100
      WRITE(6,3000)JJ
      CALL EXIT
  100 CONTINUE
      CALL INPT(AA(N1),AA(N2),AA(N3),AA(N4),AA(N5),AA(N6),AA(N7),
     *AA(N8),AA(N9),AA(N10),IA(M1),IA(M2),IA(M3))
 1000 FORMAT(18A4/4I5,3F10.2,I5,D15.4)
```

```
 2000 FORMAT (1H1 18A4/
     1 30H0 NUMBER OF NODAL POINTS------ I3 /
     2 30H0 NUMBER OF ELEMENTS---------- I3 /
     3 30H0 NUMBER OF DIFF. MATERIALS--- I3 /
     4 30H0 NUMBER OF PRESSURE CARDS---- I3 /
     5 30H0 X-ACCELERATION-------------- E12.4/
     6 30H0 Y-ACCELERATION-------------- E12.4/
     7 30H0 REFERENCE TEMPERATURE------- E12.4/
     8 30H0 NO. OF APPROXIMATIONS------- I5/
     9 30H0 TOLERANCE FOR CONVERGENCE--- E12.4)
 3000 FORMAT (70H PROGRAM EXECUTION TERMINATED. REQUIRED CORE EXCEEDS NT
     *OT BY                                                           I10)
      END
```

```
      SUBROUTINE INPT(R,Z,UR,UZ,T,PR,SIGI,CODE,B,CU,IX,IBC,JBC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON AA(10000),IA(1500)
      COMMON/ONE/ACELR,ACELZ,TEMP,Q,TOL,VOL,NUMNP,NUMEL,NUMMAT,NUMPC,
     *MTYPE,LLL,N,MBAND,NUMBLK,KKK,NCHECK,NP,NL,NEQ,HED(18),NPC
      COMMON/TWO/C(3,3),S(10,10),SIG(6),P(6),ST(3,10),RR(5),ZZ(5),XC,YC,
     *EE(3),LM(4),E(5,4,10),RO(10),NTC(10),MTC(10)
      COMMON/THREE/MTOT,NTOT,N11,M4,SIOP
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),T(NUMNP),PP(NPC,2)
     *,SIGI(NUMEL,6),CODE(NUMNP),IX(NUMEL,5),IBC(NPC),JBC(NPC)
      DIMENSION B(NEQ),CU(NEQ)
      DO 50 M=1,NUMMAT
      READ (5,1001) MTYPE,NTC(MTYPE),RO(MTYPE),MTC(MTYPE)
      WRITE(6,2001) MTYPE,NTC(MTYPE),RO(MTYPE),MTC(MTYPE)
      NUMTC=NTC(MTYPE)
      READ (5,1002) ((E(I,J,MTYPE),J=1,4),I=1,NUMTC)
      WRITE (6,2002) ((E(I,J,MTYPE),J=1,4),I=1,NUMTC)
   50 CONTINUE
      WRITE (6,2003)
      L=0
   60 READ  (5,1003) N,CODE(N),R(N),Z(N),UR(N),UZ(N),T(N)
      NL=L+1
      IF(N.EQ.1) GO TO 70
      ZX=N-L
      DR=(R(N)-R(L))/ZX
      DZ=(Z(N)-Z(L))/ZX
      DT=(T(N)-T(L))/ZX
   70 L=L+1
      IF(N-L) 100,90,80
   80 CODE(L)=0.0
      R(L)=R(L-1)+DR
      Z(L)=Z(L-1)+DZ
      T(L)=T(L-1)+DT
      UR(L)=0.0
      UZ(L)=0.0
      GO TO 70
   90 CONTINUE
      IF(NUMNP-N) 100,110,60
  100 WRITE (6,2005) N
      CALL EXIT
  110 WRITE(6,2004)((K,CODE(K),R(K),Z(K),UR(K),UZ(K),T(K)),K=1,NUMNP)
      READ(5,1007) (IX(N,5),N=1,NUMEL)
      WRITE (6,2006)
      N=0
  130 READ (5,1004) M,(IX(M,I),I=1,4) ,(SIGI(M,I),I=1,3)
      IF(M.EQ.1) GO TO 140
      ZX=M-N
      DO 135 I=1,3
  135 SIG(I) =(SIGI(M,I)-SIGI(N,I)) / ZX
  140 N=N+1
```

```
      IF(M.LE.N)GO TO 170
      IX(N,1)=IX(N-1,1)+1
      IX(N,2)=IX(N-1,2)+1
      IX(N,3)=IX(N-1,3)+1
      IX(N,4)=IX(N-1,4)+1
      DO 160 I=1,3
  160 SIGI(N,I)=SIGI(N-1,I)+SIG(I)
  170 IF(M.GT.N) GO TO 140
      IF(N.LT.NUMEL)GO TO130
      WRITE(6,2007)((N,(IX(N,I),I=1,5),(SIGI(N,I),I=1,3)),N=1,NUMEL)
      IF(NUMPC.EQ.0) GO TO 310
  290 WRITE (6,2008)
      DO 300 L=1,NUMPC
      READ(5,1005) IBC(L),JBC(L),PR(L,1),PR(L,2)
  300 WRITE(6,2009) IBC(L),JBC(L),PR(L,1),PR(L,2)
  310 CONTINUE
      J=0
      DO 340 N=1,NUMEL
      DO 340 I=1,4
      DO 325 L=1,4
      KK=IABS(IX(N,I)-IX(N,L))
      IF(KK.GT.J) J=KK
  325 CONTINUE
  340 CONTINUE
      MBAND=2*J+2
      WRITE(6,3000)MBAND
      NL=(NTOT-N11+1)/MBAND
      NLL=NEQ+3
      IF(NL.GT.NLL) NL=NLL
      NL=NL/4
      NL=4*NL
      NBAND=2*MBAND
      IF(NL.GE.NBAND)GO TO 350
      WRITE (6,4010)NL,MBAND
      CALL EXIT
  350 CONTINUE
      N12=N11+NL*MBAND
      JJ=N12-NTOT
      IF(JJ.LE.0)GO TO 360
      WRITE(6,3050)JJ
      CALL EXIT
  360 CONTINUE
      WRITE(6,4000)N12,M4
          CALL SOLVE(R,Z,UR,UZ,T,PR,SIGI,CODE,B,CU,AA(N11),IX,
     *IBC,JBC)
      RETURN
 1001 FORMAT (2I5,1F10.3,I5)
 1002 FORMAT (4F10.3)
 1003 FORMAT (I5,F5.1,5F10.4)
 1004 FORMAT(5I5,5X,3F10.0)
```

```
 1005 FORMAT(2I5,2F10.3)
 1007 FORMAT(16I5)
 2001 FORMAT (17HOMATERIAL NUMBER= I3, 30H, NUMBER OF TEMPERATURE CARDS=
     1 I3, 15H, MASS DENSITY= F12.4,16H, MATERIAL CODE= I5)
 2002 FORMAT (15HO  TEMPERATURE 10X 5HE      9X 6HNU     10X 5HALPHA/
     1(F15.2,3E15.5))
 2003 FORMAT (108HINODAL POINT         TYPE  X ORDINATE  Y ORDINATE  X LO
     1AD OR DISPLACEMENT  Y LOAD OR DISPLACEMENT TEMPERATURE )
 2004 FORMAT (I12,F12.2,2F12.5,2E24.7,F12.3)
 2005 FORMAT (26HONODAL POINT CARD ERROR N= I5)
 2006 FORMAT(108H1ELEMENT NO.      I     J     K     L    MATERIAL
     1SIGIXX         SIGIYY        SIGIXY                            )
 2007 FORMAT (I13,4I6,I112,3F12.3)
 2008 FORMAT(29HOPRESSURE BOUNDARY CONDITIONS/40H      I      J    PRESSUR
     1E I    PRESSURE J)
 2009 FORMAT(2I6,2F14.3)
 3000 FORMAT(30H BAND WIDTH------ -----------                        I5)
 3050 FORMAT (70H PROGRAM EXECUTION TERMINATED. REQUIRED CORE EXCEEDS NI
     *OT BY                                                            I10)
 4000 FORMAT (47H FOR THIS PROGRAM THE LOCATION USED IN AA IS =     I5,
     *17H AND IN IA IS =          I5)
 4010 FORMAT (25H0 NL IS LESS THAN 2*MBAND                              /
     *5H0 NL=                                                          I5/
     *8H0 MBAND=                                                       I5)
      END
```

```
      SUBROUTINE SOLVE(R,Z,UR,UZ,T,PR,SIGI,CODE,B,CU,A,IX,IBC,JBC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON AA(10000),IA(1500)
      COMMON/ONE/ACELR,ACELZ,TEMP,Q,TOL,VOL,NUMNP,NUMEL,NUMMAT,NUMPC,
     *MTYPE,LLL,N,MBAND,NUMBLK,KKK,NCHECK,NP,NL,NEQ,HED(18),NPC
      COMMON/TWO/C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),XC,YC,
     *EE(3),LM(4),E(5,4,10),RO(10),NTC(10),MTC(10)
      COMMON/THREE/MTOT,NTOT,N11,M4,STOP
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),T(NUMNP),PR(NPC,2)
     *,SIGI(NUMEL,6),CODE(NUMNP),IX(NUMEL,5),IBC(NPC),JBC(NPC)
      DIMENSION CU(NEQ),A(NL,MBAND),B(NEQ)
      DO 10 N=1,NUMEL
      DO 10 I=4,6
   10 SIGI(N,I)=0.
      DO 20 N=1,NUMNP
      NN=2*N
      CU(NN-1)=0.
   20 CU(NN)=0.
      REWIND 2
      ND2=NL
      NO=NL/2
      NB=NO/2
      STOP=0.0
      NUMBLK=0
      DO 50 N=1,ND2
      DO 50 M=1,MBAND
   50 A(N,M)=0.0
   60 NUMBLK=NUMBLK+1
      NH=NB*(NUMBLK+1)
      NM=NH-NB
      NZ=NM-NB+1
      KSHIFT=2*NZ-2
      DO 210 N=1,NUMEL
      NN=IX(N,1)
      DO 65 I=2,4
      IF(IX(N,I).LT.NN) NN=IX(N,I)
   65 CONTINUE
      IF((NN.LT.NZ).OR.(NN.GT.NM)) GO TO 210
      IF(IX(N,3).NE.IX(N,2)) GO TO 95
      CALL ONED(R,Z,UR,UZ,T,PR,SIGI,CODE,CU,A,B,IX,IBC,JBC)
      MM=2
      GO TO 130
   95 CALL ELEMEN (R,Z,UR,UZ,T,PR,SIGI,CODE,CU,A,B,IX,IBC,JBC)
      IF(VOL.GT.0)GO TO 110
  100 WRITE(6,2000) N
      STOP=1.0
  110 MM=4
      IF(IX(N,3).EQ.IX(N,4))MM=3
  130 CONTINUE
      DO 140 I=1,MM
```

```
  140 LM(I)=2*IX(N,I)-2
      DO 200 I=1,MM
      DO 200 K=1,2
      II=LM(I)+K-KSHIFT
      KK=2*I-2+K
      DO 200 J=1,MM
      DO 200 L=1,2
      JJ=LM(J)+L-II+1-KSHIFT
      LL=2*J-2+L
      IF(JJ.LE.0) GO TO 200
      IF(ND.GE.JJ) GO TO 195
  180 WRITE (6,2001) N
      STOP=1.0
      GO TO 210
  195 A(II,JJ)=A(II,JJ)+S(KK,LL)
  200 CONTINUE
  210 CONTINUE
      DO 400 M=NZ,NH
      IF(M.GT.NUMNP) GO TO 410
      N=2*M-1-KSHIFT
      IF(CODE(M)) 390,400,316
  316 IF(CODE(M).EQ.1.)GO TO 370
      IF(CODE(M).EQ.2.)GO TO 390
      GO TO 380
  370 CALL MODIFY(A,ND2,MBAND,N)
      GO TO 400
  380 CALL MODIFY(A,ND2,MBAND,N)
  390 CONTINUE
      N=N+1
      CALL MODIFY(A,ND2,MBAND,N)
  400 CONTINUE
  410 CONTINUE
      WRITE(2) ((A(N,M),M=1,MBAND),N=1,ND)
      DO 420 N=1,ND
      K=N+ND
      DO 420 M=1,MBAND
      A(N,M)=A(K,M)
  420 A(K,M)=0.0
      IF(NM.LT.NUMNP) GO TO 60
      IF(STOP.NE.0.)CALL EXIT
      NCHECK=1
      KKK=1
      CALL SYMBAN(ND2,A,B,NEQ,MBAND,NUMBLK,KKK)
      KKK=2
      DO 550 LLL=1,NP
      CALL LOAD(R,Z,UR,UZ,T,PR,SIGI,CODE,CU,A,B,IX,IBC,JBC)
      CALL SYMBAN(ND2,A,B,NEQ,MBAND,NUMBLK,KKK)
      DO 510 N=1,NUMNP
      NN=2*N
      CU(NN-1)=CU(NN-1)+B(NN-1)
```

```
  510 CU(NN)=CU(NN)+B(NN)
      WRITE (6,2013) LLL
      WRITE(6,2010) (N,B(2*N-1),B(2*N),CU(2*N-1),CU(2*N), N = 1,NUMNP)
      CALL STRESS(R,Z,UR,UZ,T,PR,SIGI,CODE,CU,A,B,IX,IBC,JBC)
      IF(NCHECK.EQ.0) GO TO 600
  550 CONTINUE
      GO TO 990
  600 WRITE (6,2011)LLL
  990 RETURN
 2000 FORMAT (26HONEGATIVE AREA ELEMENT NO. I4)
 2001 FORMAT (29HOBAND WIDTH EXCEEDS ALLOWABLE I4)
 2010 FORMAT(12HON.P.NUMBER 17X 3HDUX 17X 3HDUY I8X 2HUX 18X 2HUY/
     1(1I12,4E20.7))
 2011 FORMAT(35H0 NUMBER OF CYCLES TO CONVERGENCE =    I5)
 2013 FORMAT(30H1 RESULTS OF ITERATION  NO.=      I5///)
      END
```

```
      SUBROUTINE UNED(R,Z,UR,UZ,T,PR,SIGI,CODE,CO,A,B,IX,IBC,JBC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON AA(10000),IA(1500)
      COMMON/ONE/ACELR,ACELZ,TEMP,Q,TOL,VOL,NUMNP,NUMEL,NUMMAT,NUMPC,
     *MTYPE,LLL,N,MBAND,NUMBLK,KKK,NCHECK,NP,NL,NEQ,HED(18),NPC
      COMMON/TWO/C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),XC,YC,
     *EE(3),LM(4),E(5,4,10),KC(10),NTC(10),MTC(10)
      COMMON/THREE/MTOT,NTOT,NI1,M4,STOP
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),T(NUMNP),PR(NPC,2)
     *,SIGI(NUMEL,6),CODE(NUMNP),IX(NUMEL,5),IBC(NPC),JBC(NPC)
C
C
      DO 100 I=1,8
      P(I)=0.0
      DO 100 J=1,8
  100 S(I,J)=0.0
      MTYPE=IX(N,5)
      I=IX(N,1)
      J=IX(N,2)
      DX=R(J)-R(I)
      DY=Z(J)-Z(I)
      XL=DSQRT(DX**2+DY**2)
      COSA=DX/XL
      SINA=DY/XL
      COMM=E(1,2,MTYPE)*E(1,4,MTYPE)/XL
C
      S(1,1)=COSA*COSA*COMM
      S(1,2)=COSA*SINA*COMM
      S(1,3)=-S(1,1)
      S(1,4)=-S(1,2)
      S(2,1)=S(1,2)
      S(2,2)=SINA*SINA*COMM
      S(2,3)=-S(1,2)
      S(2,4)=-S(2,2)
      S(3,1)=S(1,3)
      S(3,2)=S(2,3)
      S(3,3)=S(1,1)
      S(3,4)=S(1,2)
      S(4,1)=S(1,4)
      S(4,2)=S(2,4)
      S(4,3)=S(3,4)
      S(4,4)=S(2,2)
C
C
      RETURN
C
      END
```

```
      SUBROUTINE ELEMEN(R,Z,UR,UZ,T,PR,SIGI,CODE,CU,A,B,IX,IBC,JBC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON AA(10000),IA(1500)
      COMMON/ONE/ACELR,ACELZ,TEMP,Q,TOL,VOL,NUMNP,NUMEL,NUMMAT,NUMPC,
     *MTYPE,LLL,N,MBAND,NUMBLK,KKK,NCHECK,NP,NL,NEQ,HED(18),NPC
      COMMON/TWO/C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),XC,YC,
     *EE(3),LM(4),E(5,4,10),RO(10),NTC(10),MTC(10)
      COMMON/THREE/MTOT,NTOT,N11,M4,STOP
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),T(NUMNP),PR(NPC,2)
     *,SIGI(NUMEL,6),CODE(NUMNP),IX(NUMEL,5),IBC(NPC),JBC(NPC)
      DIMENSION CU(NEQ),A(NL,MBAND),B(NEQ)
      DIMENSION U(3),V(3)
C
      I=IX(N,1)
      J=IX(N,2)
      K=IX(N,3)
      L=IX(N,4)
      MTYPE=IX(N,5)
      VOL=0.
      TEMP=(T(I)+T(J)+T(K)+T(L))/4.0
      RATIO=0.0
      NUMTC=NTC(MTYPE)
      IF (NUMTC.EQ.1) GO TO 100
      DO 50 M=2,NUMTC
      IF (E(M,1,MTYPE)-TEMP) 50,60,60
   50 CONTINUE
   60 DEN=E(M,1,MTYPE)-E(M-1,1,MTYPE)
      IF(DEN.EQ.0.) GO TO 80
      RATIO=(TEMP-E(M-1,1,MTYPE))/DEN
   80 DO 90 KK=1,3
   90 EE(KK)=E(M-1,KK+1,MTYPE)+RATIO*(E(M,KK+1,MTYPE)-E(M-1,KK+1,MTYPE))
      GO TO 110
  100 DO 105 KK=1,3
  105 EE(KK)=E(1,KK+1,MTYPE)
  110 COMM=EE(1)/(1.-EE(2)**2)
      C(1,1)=COMM
      C(1,2)=COMM*EE(2)
      C(1,3)=0.
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=0.
      C(3,1)=0.
      C(3,2)=0.
      C(3,3)=.5*COMM*(1.-EE(2))
      DO 130 J=1,10
      DO 120 I=1,3
  120 ST(I,J)=0.
      DO 130 I=1,10
  130 S(I,J)=0.
      DO 140 I=1,4
```

```
      NPP=IX(N,1)
      RR(1)=R(NPP)
  140 ZZ(1)=Z(NPP)
      IF(IX(N,3).EQ.IX(N,4))GO TO 150
      XC=(RR(1)+RR(2)+RR(3)+RR(4))/4.
      YC=(ZZ(1)+ZZ(2)+ZZ(3)+ZZ(4))/4.
      RR(5)=XC
      ZZ(5)=YC
      K=5
      J=1
      I=4
      LM(3)=9
      NT=4
      GO TO 160
  150 NT=1
      LM(3)=5
      I=1
      K=3
      J=2
      XC=(RR(1)+RR(2)+RR(3))/3.
      YC=(ZZ(1)+ZZ(2)+ZZ(3))/3.
      RR(5)=RR(3)
      ZZ(5)=ZZ(3)
  160 DO 200 NN=1,NT
      LM(1)=2*I-1
      LM(2)=2*J-1
      U(1)=ZZ(J)-ZZ(K)
      U(2)=ZZ(K)-ZZ(I)
      U(3)=ZZ(I)-ZZ(J)
      V(1)=RR(K)-RR(J)
      V(2)=RR(I)-RR(K)
      V(3)=RR(J)-RR(I)
      AREA=(RR(J)*U(2)+RR(I)*U(1)+RR(5)*U(3))/2.
      VOL=VOL+AREA
      COMM=.25/AREA
      XNT=NT
      COM=2./XNT
      COM=COM*COMM
      DO 180 L=1,3
      II=LM(L)
      ST(1,II)=ST(1,II)+U(L)*COM
      ST(2,II+1)=ST(2,II+1)+V(L)*COM
      ST(3,II)=ST(3,II)+V(L)*COM
      ST(3,II+1)=ST(3,II+1)+U(L)*COM
      DO 180 M=1,3
      JJ=LM(M)
      S(II,JJ)=S(II,JJ)+(U(L)*C(1,1)*U(M)+V(L)*C(3,3)*V(M))*COMM
      S(II,JJ+1)=S(II,JJ+1)+(U(L)*C(1,2)*V(M)+V(L)*C(3,3)*U(M))*COMM
      S(II+1,JJ+1)=S(II+1,JJ+1)+(V(L)*C(1,1)*V(M)+U(L)*C(3,3)*U(M))*COMM
      S(JJ+1,II)=S(II,JJ+1)
```

```
  180 CONTINUE
      I=J
      J=J+1
  200 CONTINUE
      IF(IX(N,3).EQ.IX(N,4))GO TO 250
      DO240 I=1,2
      KK=10-I
      DO 240 K=1,KK
      CC=S(KK+1,K)/S(KK+1,KK+1)
      DO 230 J=1,3
  230 ST(J,K)=ST(J,K)-CC*ST(J,KK+1)
      DO 240 J=1,KK
  240 S(J,K)=S(J,K)-CC*S(J,KK+1)
  250 CONTINUE
      IF(LLL.EQ.1) GO TO 400
      SIG(1)=-SIGI(N,4)
      SIG(2)=-SIGI(N,5)
      SIG(3)=-SIGI(N,6)
      GO TO 500
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * *
C     INITIAL STRESSES ADDED  TO THE CALCULATED STRESSES
C     * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * *
  400 DT=TEMP-Q
      DX=FE(3)*DT
      DY=FE(3)*DT
C
C     INITIAL STRESSES CALCULATE4 TOTAL HEIGHT =88 FT. AND VALUE OF
C     COEFF. KO =0.2  AND UNIT WEIGHT = 150 FOR ALL MATERIALS
      SIGI(N,2)= -150.0 * (88.0 - YC)
      SIGI(N,1)= 0.2* SIGI(N,2)
      SIGI(N,3)=0.0
C
      SIG(1)=-C(1,1)*DX-C(1,2)*DY+SIGI(N,1)
      SIG(2)=-C(2,1)*DX-C(2,2)*DY+SIGI(N,2)
      SIG(3)=SIGI(N,3)
C
  500 DO 520 I=1,8
      P(I)=0.0
      DO 510 J=1,3
  510 P(I)=P(I)-ST(J,I)*SIG(J)
  520 P(I)=P(I)*VOL
      IF(LLL.EQ.1) GO TO 540
      DO 530 I=1,3
  530 SIG(I)=0.0
      GO TO 600
  540 MM=4
      IF(IX(N,3).EQ.IX(N,4)) MM=3
      XMM=MM
      DY=VOL*ACELZ*RO(MTYPE)/XMM
      DX=VOL*ACELR*RO(MTYPE)/XMM
```

```
      DO 550 I=1,MM
      P(2*I)=P(2*I)+DY
  550 P(2*I-1)=P(2*I-1)+DX
  600 RETURN
      END
```

```
      SUBROUTINE STRESS(R,Z,UR,UZ,T,PR,SIGI,CODE,CU,A,B,IX,IBC,JBC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON AA(10000),IA(1500)
      COMMON/ONE/ACELR,ACELZ,TEMP,Q,TOL,VOL,NUMNP,NUMEL,NUMMAT,NUMPC,
     *MTYPE,LLL,N,MBAND,NUMBLK,K2K,NCHECK,NP,NL,NEQ,HED(18),NPC
      COMMON/TWO/C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),XC,YC,
     *EE(3),LM(4),E(5,4,10),RO(10),NTC(10),MTC(10)
      COMMON/THREE/MTOT,NTOT,N11,M4,STOP
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),T(NUMNP),PR(NPC,2)
     *,SIGI(NUMEL,6),CODE(NUMNP),IX(NUMEL,5),IBC(NPC),JBC(NPC)
      DIMENSION CU(NEQ),A(NL,MBAND),B(NEQ)
      FOR = 0.0
      MPRINT=0
      DO 600 M=1,NUMEL
      N=M
      MTYPE=IX(N,5)
      SIGI(N,4)=0.
      SIGI(N,5)=0.
      SIGI(N,6)=0.
      IF(IX(N,3).NE.IX(N,2))GO TO 90
      I=IX(N,1)
      J=IX(N,2)
      XC=(R(I)+R(J))/2.0
      YC=(Z(I)+Z(J))/2.0
      DX=R(J)-R(I)
      DY=Z(J)-Z(I)
      XL=DSQRT(DX**2+DY**2)
      DU=B(2*J-1)-B(2*I-1)
      DV=B(2*J)-B(2*I)
      DL=DV*DY/XL+DU*DX/XL
      SIG(1)=E(1,4,MTYPE)*DL*E(I,2,MTYPE)/XL+SIGI(N,1)*E(1,4,MTYPE)
      IF(SIG(1).GT.0.) GO TO 100
      SIGI(N,1)=SIG(1)
      GO TO 500
  100 SIGI(N,4)=E(1,2,MTYPE)*DL/XL+SIGI(N,1)
      SIGI(N,1)=0.
      GO TO 420
   90 CALL ELEMEN(R,Z,UR,UZ,T,PR,SIGI,CODE,CU,A,B,IX,IBC,JBC)
      MM=4
      IF(IX(N,3).NE.IX(N,4))GO TO 170
      MM=3
  170 DO 180 I=1,3
      RR(I)=0.
      DO 180 J=1,MM
      II=2*J
      JJ=2*IX(N,J)
  180 RR(I)=RR(I)+ST(I,II)*B(JJ)+ST(I,II-1)*B(JJ-1)
      DO 190 I=1,3
      DO 185 J=1,3
  185 SIG(I)=SIG(I)+C(I,J)*RR(J)
```

```
  190 CONTINUE
      IF(LLL.EQ.1) GO TO 195
      DO 192 I=1,3
  192 SIG(I)=SIG(I)+SIGI(N,I)
  195 CONTINUE
      CC=(SIG(1)+SIG(2))/2.0
      BB=(SIG(1)-SIG(2))/2.
      CR=DSQRT(BB**2+SIG(3)**2)
      SIG(4)=CC+CR
      SIG(5)=CC-CR
      SIG(6)=0.0
      IF((BB.EQ.0.).AND.(SIG(3).EQ.0.)) GO TO 200
      SIG(6)=28.648*DATAN2(SIG(3),BB)
      DX=0.0
  200 SIGI(N,1)=SIG(1)
      SIGI(N,2)=SIG(2)
      SIGI(N,3)=SIG(3)
      IF((SIG(4).LE.0).OR.(MTC(MTYPE).EQ.0)) GO TO 500
      IF(SIG(5).GE.0.00001) GO TO 370
      EPS=SIG(6)/57.296
      CC=DCOS(EPS)
      SS=DSIN(EPS)
      C2=CC*CC
      S2=SS*SS
      SC=SS*CC
C
      DX=EE(2)*SIG(4)
      SIGI(N,4)= SIG(4)*C2+DX*S2
      SIGI(N,5)= SIG(4)*S2+DX*C2
      SIGI(N,6)=SIG(4)*SC-DX*SC
C
C
      GO TO 400
  370 SIGI(N,4)=SIG(1)
      SIGI(N,5)=SIG(2)
      SIGI(N,6)=SIG(3)
  400 SIGI(N,1)=SIG(1)-SIGI(N,4)
      SIGI(N,2)=SIG(2)-SIGI(N,5)
      SIGI(N,3)=SIG(3)-SIGI(N,6)
  420 DX=SIGI(N,4)**2+SIGI(N,5)**2+SIGI(N,6)**2
      DX=DSQRT(DX)
      IF(DX.LE.FOR) GO TO 450
      IJK=N
      FOR=DX
  450 CONTINUE
  500 IF(MPRINT.NE.0) GO TO 550
      WRITE(6,2000)
      MPRINT=50
  550 MPRINT=MPRINT-1
      WRITE(6,2001)N,XC,YC,(SIG(I),I=1,6),DX
```

```
  600 CONTINUE
      WRITE(6,2002)FOR,IJK
      IF(FOR.LE.TOL) NCHECK = 0
      RETURN
 2000 FORMAT (7H1EL.NO.  7X 1HX  7X 1HY 4X 8HX-STRESS 4X 8HY-STRESS 3X
     1 9HXY-STRESS 2X 10HMAX-STRESS 2X 10HMIN-STRESS 7H  ANGLE 2X 17HUNB
     2ALANCED FORCE    )
 2001 FORMAT (I7,2F8.2,1P5E12.4,0P1F7.2,1PE20.4)
 2002 FORMAT(30H0MAXIMUM UNBALANCED FORCE =    E12.5,16H IN ELEMENT NO.
     1 I5)
      END
```

```
      SUBROUTINE MODIFY(A,NEQ,MBAND,N)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NEQ,MBAND)
      DO 250 M=2,MBAND
      K=N-M+1
      IF(K) 235,235,230
  230 A(K,M) = 0.0
  235 K=N+M-1
      IF(NEQ-K) 250,240,240
  240 A(N,M) = 0.0
  250 CONTINUE
      A(N,1)=1.0
      RETURN
      END
```

```
      SUBROUTINE LOAD(R,Z,UR,UZ,T,PR,SIGI,CODE,CO,A,B,IX,IBC,JBC)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON AA(10000),IA(1500)
      COMMON/ONE/ACELR,ACELZ,TEMP,Q,TOL,VOL,NUMNP,NUMEL,NUMMAT,NUMPC,
     *MTYPE,LLL,N,MBAND,NUMBLK,KKK,NCHECK,NP,NL,NEQ,HED(18),NPC
      COMMON/TWO/C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),XC,YC,
     *EE(3),LM(4),E(5,4,10),RO(10),NTC(10),MTC(10)
      COMMON/THREE/MTOT,NTOT,N11,M4,STOP
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),T(NUMNP),PR(NPC,2)
     *,SIGI(NUMEL,6),CODE(NUMNP),IX(NUMEL,5),IBC(NPC),JBC(NPC)
      DIMENSION CO(NEQ),A(NL,MBAND),B(NEQ)
      DO 50 N=1,NUMNP
      B(2*N-1)=UR(N)
      B(2*N)=UZ(N)
      UR(N)=0.
      UZ(N)=0.
   50 CONTINUE
      IF((NUMPC.EQ.0).OR.(LLL.GT.1)) GO TO 300
      DO 200 L=1,NUMPC
      I=IBC(L)
      J=JBC(L)
      DR=Z(I)-Z(J)
      DZ=R(J)-R(I)
      PP2=(PR(L,2)+PR(L,1))/6.
      PP1=PP2+PR(L,1)/6.
      PP2=PP2+PR(L,2)/6.
      II=2*I
      JJ=2*J
      B(II-1)=B(II-1)+PP1*DR
      B(II)=B(II)+PP1*DZ
      B(JJ-1)=B(JJ-1)+PP2*DR
      B(JJ)=B(JJ)+PP2*DZ
  200 CONTINUE
  300 DO 700 N=1,NUMEL
      I=IX(N,1)
      J=IX(N,2)
      K=IX(N,3)
      L=IX(N,4)
      MTYPE=IX(N,5)
      IF(SIGI(N,4).NE.0.) GO TO 320
      IF(SIGI(N,5).NE.0.) GO TO 320
      IF(SIGI(N,6).NE.0.) GO TO 320
      IF(LLL.EQ.1) GO TO 320
      GO TO 700
  320 CONTINUE
      IF(LLL.EQ.1) GO TO 330
      IF(MTC(MTYPE).EQ.0) GO TO 700
  330 IF(J.EQ.K) GO TO 500
      CALL ELEMEN(R,Z,UR,UZ,T,PR,SIGI,CODE,CO,A,B,IX,IBC,JBC)
      GO TO 600
```

```
  500 CALL DNED(R,Z,UR,UZ,T,PR,SIGI,CODE,CU,A,B,IX,IBC,JBC)
      DX=R(J)-R(I)
      DY=Z(J)-Z(I)
      EP=-SIGI(N,4)/E(1,2,MTYPE)
      DX=DX*EP
      DY=DY*EP
      P(1)=S(1,1)*DX+S(1,2)*DY
      P(2)=S(2,1)*DX+S(2,2)*DY
      P(3)=-P(1)
      P(4)=-P(2)
  600 DO 620 II=1,4
  620 LM(II)=2*IX(N,II)-1
      DO 650 JJ=1,4
      II=LM(JJ)
      B(II)=B(II)+P(2*JJ-1)
  650 B(II+1)=B(II+1)+P(2*JJ)
  700 CONTINUE
      DO 750 N=1,NUMNP
      IF(CODE(N).EQ.0.) GO TO 750
      IF((CODE(N).EQ.1.).OR.(CODE(N).EQ.3.)) B(2*N-1)=0.
      IF((CODE(N).EQ.2.).OR.(CODE(N).EQ.3.)) B(2*N)=0.0
  750 CONTINUE
  800 RETURN
      END
```

```
      SUBROUTINE SYMBAN(NG,A,B,NEQ,MBAND,NUMBLK,KKK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NG,MBAND),B(NEQ)
      NN=NG/2
C
C
      NL=NN+1
      NH=NN+NN
      REWIND 1
      GO TO (1000,2000),KKK
 1000 REWIND 2
      NB=0
      GO TO 150
  100 NB=NB+1
      DO 125 N=1,NN
      NM=NN+N
      DO 125 M=1,MBAND
      A(N,M)=A(NM,M)
  125 A(NM,M)=0.
C
      IF(NUMBLK-NB) 150,200,150
  150 READ (2)((A(N,M),M=1,MBAND),N=NL,NH)
      IF(NB) 200,100,200
  200 DO 300 N=1,NN
      IF(A(N,1).EQ.0.)GO TO 300
      DO 275 L=2,MBAND
      IF(A(N,L).EQ.0.) GO TO 275
      C=A(N,L)/A(N,1)
      I=N+L-1
      J=0
      DO 250 K=L,MBAND
      J=J+1
  250 A(I,J)=A(I,J)-C*A(N,K)
      A(N,L)=C
  275 CONTINUE
  300 CONTINUE
      WRITE(1) ((A(N,M),M=1,MBAND),N=1,NN)
      IF(NUMBLK.EQ.NB)GO TO 900
      GO TO 100
 2000 NQ=0
      NB=0
C
C
      GO TO 450
  400 NB=NB+1
      DO 425 N=1,NN
      NM=NN+N
      DO 425 M=1,MBAND
      A(N,M)=A(NM,M)
  425 A(NM,M)=0.
```

```
      IF(NUMBLK.EQ.NB)GO TO 500
  450 READ (1) ((A(N,M),M=1,MBAND),N=NL,NH)
      IF(NB.EQ.0)GO TO 400
  500 DO 550 N=1,NN
      J=NQ+N
      DO 540 L=2,MBAND
      I=J+L-1
      IF(NEQ-I) 545,540,540
  540 B(I)=B(I)-A(N,L)*B(J)
  545 IF(A(N,1).EQ.0.) A(N,1)=1.
  550 B(J)=B(J)/A(N,1)
      IF(NUMBLK-NB) 600,650,600
  600 NQ=NQ+NN
      GO TO 400
  650 BACKSPACE 1
  700 DO 750 M=1,NN
      N=NN+1-M
      J=NQ+N
      DO 750 L=2,MBAND
      IF(A(N,L).EQ.0.)GO TO 750
      I=J+L-1
      IF(NEQ.LT.I)GO TO 750
      B(J)=B(J)-A(N,L)*B(I)
  750 CONTINUE
      NB=NB-1
      IF(NB.EQ.0) GO TO 900
      BACKSPACE 1
      DO 800 N=1,NN
      NM=NN+N
      DO 800 M=1,MBAND
      A(NM,M)=A(N,M)
  800 A(N,M)=0.
      READ(1)((A(N,M),M=1,MBAND),N=1,NN)
      BACKSPACE 1
      NQ=NQ-NN
      GO TO 700
  900 RETURN
      END
```

CHAPTER II: ELPL - A Computer Program for Plane Strain Analysis of Stresses, Deformations and Progressive Failure in Elastic-Perfectly Plastic Rock

2.1. Purpose and Capability

This program is applicable to plane strain analysis of stresses, deformations and progressive failure in elastic-perfectly plastic material following generalized Mohr-Coulomb yield criterion, and the incremental theory of plasticity. Arbitrary initial stresses, arbitrary sequence of excavation and construction, arbitrary history of load application can be simulated. One-dimensional elements with prescribed prestress can be included in the analysis. The computer program is applicable to study of stability of underground or surface excavations, evaluation of alternative schemes for excavation, comparative study of support stresses and deformation.

Theoretical development incorporated in the program is discussed in Part I-Technical Report, of this report.

2.2. Program Organization

The computer program is in Fortran language. Tapes 1 and 2 are used as scratch files. Tapes 5 and 6 are the input/output files. The listing in section 2.5 uses double precision for real numbers. The program capacity can be altered by changing the dimension of arrays AA and IA. These correspond to the total locations required in double precision real arrays and integer arrays respectively. NTOT, MTOT at lines MAIN 28, MAIN 27 are set equal to the dimension of AA and IA.

The program consists of the following units.

a. MAIN

In this unit, the control information including maximum number of elements, nodal points, different materials, boundary pressure cards, the number of steps of excavation or construction, the maximum number of elements removed from the system or added to it at any step, is read in. This information is used to organize the dimensions of various arrays in the analysis. This done, the analysis moves on to the next unit.

b. Subroutine INPT

This subroutine is called by the unit MAIN. The first step is to read in material property data, for all the different materials that may participate in the system at any stage. Also nodal point coordinates, loads and code on boundary conditions are read in or generated. Element geometry, initial stresses and element thickness is also read in or generated. The initial stresses may be input or computed within the program. The thickness, if not specified, is assumed to be unity. Maximum bandwidth for the system is calculated and dimensions of blocks for generation and storage of the system stiffness matrix are defined. After defining these controls, the incremental structure is analyzed in steps. For each step the number of nodal points, the number of elements, the number of elements removed or added, if any, the number of boundary pressure cards, the material type of the elements added or changed in material properties, the number of nodal points removed or added to the system in the step considered, the thickness of elements added, are read in. Additional information defines whether a step involves excavation or construction. At any step, several changes can be simulataneously introduced into a structure.

For example, addition of several sets of elements of different materials, changes in material properties of other elements, removal of certain elements can all be introduced simultaneously if desired. For each step the boundary pressures are read in. After all the information is assembled, the solution process is transferred to subroutine SOLVE.

c. Subroutine SOLVE

This subroutine called by INPT is concerned with obtaining the stresses and deformations at a given stage of the incremental structure allowing for progressive failure. To trace the progressive failure, the solution process traces a sequence of elements reaching yield under the loads applied. This sequence of yield is associated with a proportion of the load application and is described in the output as successive approximations with increasing 'stress ratio.' The procedure consists of applying the total load and then scaling it according to the minimum rates of load increment needed to ensure an excursion to yield by one element at a time.

The SOLVE subroutine calls ONED and QUAD to obtain stiffness of one-dimensional and two-dimensional elements respectively. This is added to system stiffness which is stored on tape after modification for prescribed displacement boundary conditions. Solution to the stiffness equations is obtained in subroutine BANSOL and subroutine STRESS defines the stresses, the stress ratio, the scaling of stress increments and the control for continuation of progressive failure analysis.

d. Subroutine ONED

This subroutine generates stiffness of one-dimensional elements as well as the forces corresponding to the unbalanced stress defined by the difference in total load application and the load taken by the system in the current approximation in progressive failure analysis.

e. Subroutine QUAD

This subroutine generates the stiffness for two-dimensional elements (quadrilaterals or triangles). For the current load increment, an element is either elastic or has reached yield and is plastic. The subroutine STRSTR is called to obtain the stress-strain relationship. Gravity loads, and loads corresponding to unbalanced stress are generated in this subroutine.

f. Subroutine STRSTR

This subroutine is called by QUAD and also by STRESS. It defines the stress-strain relationship for the elastic or plastic materials as the case may be. For the elastic case the relationship is

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E^*}{1 - \nu^{*2}} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu^*}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix}$$

and $\sigma_z = \nu (\sigma_x + \sigma_y)$

where $E^* = \dfrac{E}{1 - \nu^2}$, $\nu^* = \dfrac{\nu}{1 - \nu}$

and E, ν are respectively the Young's modulus and Poisson's ratio for the isotropic elastic material. If anisotropy is to be considered, the above relationship can be modified to reflect that property.

For the plastic domain the stress-strain relation is

$$\begin{Bmatrix} \dot{\sigma}_x \\ \dot{\sigma}_y \\ \dot{\tau}_{xy} \\ \dot{\sigma}_z \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \\ D_{41} & D_{42} & D_{43} \end{bmatrix} \begin{Bmatrix} \dot{\epsilon}_x \\ \dot{\epsilon}_y \\ \dot{\gamma}_{xy} \end{Bmatrix}$$

where

$$D_{11} = 2G(1 - h_2 - 2h_1 \sigma_x - h_3 \sigma_x^2)$$

$$D_{21} = D_{12} = -2G(+h_2 + h_1(\sigma_x + \sigma_y) + h_3 \sigma_x \sigma_y)$$

$$D_{31} = D_{13} = -2G(h_1 \tau_{xy} + h_3 \sigma_x \tau_{xy})$$

$$D_{22} = 2G(1 - h_2 - 2h_1 \sigma_y - h_3 \sigma_y^2)$$

$$D_{32} = D_{23} = -2G(h_1 \tau_{xy} + h_3 \sigma_y \tau_{xy})$$

$$D_{33} = 2G(.5 - h_3 \tau_{xy}^2)$$

$$D_{41} = -2G(h_2 + h_1(\sigma_x + \sigma_z) + h_3 \sigma_x \sigma_z)$$

$$D_{42} = -2G(h_2 + h_1(\sigma_y + \sigma_z) + h_3 \sigma_y \sigma_z)$$

$$D_{43} = -2G(h_1 \tau_{xy} + h_3 \tau_{xy} \sigma_z)$$

$$2G = \frac{E}{(1 + \nu)}$$

$$h_1 = \frac{.5\, h_4}{h_5\, J_2^{\frac{1}{2}}}$$

$$h_2 = \frac{h_4\, h_6}{h_5} - \frac{\nu}{(1 - 2\nu)} \quad \frac{k}{h_5\, J_2^{\frac{1}{2}}}$$

$$h_3 = \frac{.5}{h_5\, J_2}$$

$$h_4 = 3\alpha \frac{K}{G} - \frac{J_1'}{3 J_2^{\frac{1}{2}}}$$

$$h_5 = 1 + 9\alpha^2 \frac{K}{G}$$

$$h_6 = \frac{J_1'}{J_2^{\frac{1}{2}}}$$

K = bulk modulus $= \dfrac{E}{3(1 - 2\nu)}$

$J_1' = \sigma_x + \sigma_y + \sigma_z$ = first invariant of the stress tensor

J_2 = the second invariant of the stress deviation tensor.

g. Subroutine MODIFY

This subroutine is called by the subroutine SOLVE to modify the stiffness matrix for prescribed displacement boundary conditions. The modified matrix is returned to SOLVE.

h. Subroutine BANSOL

This subroutine called by the subroutine SOLVE solves the stiffness equations by gaussian elimination using auxiliary storage files 1 and 2. Results are stored in B array and returned to SOLVE.

i. Subroutine STRESS

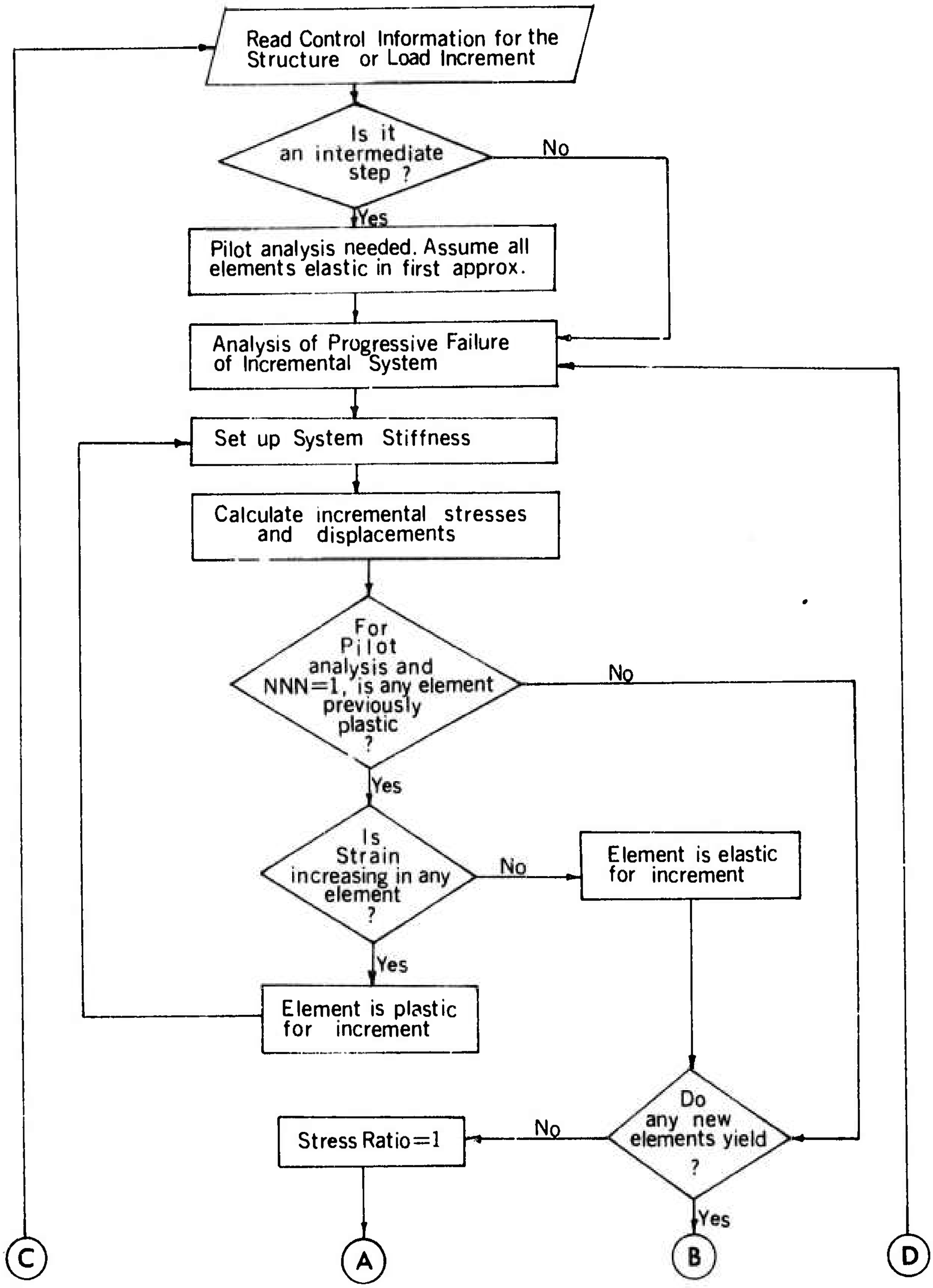
This subroutine is called by the subroutine SOLVE after displacements corresponding to a load increment have been computed. As a first step the entire load is assumed to be applied and the resulting stress state checked for possible excursion beyond yield. If the total load application shows an element passing from the elastic to the plastic stage, the stress ratio is computed as explained in Part I-Technical Report. The minimum stress ratio of all elements corresponds to the least load increment necessary for at least one more element to pass from elastic to the plastic range. The stress ratio also has to be such that elements previously on the yield surface do not depart from this surface by more than the designed tolerance. To avoid unnecessarily lengthy computation, the ratio is kept above a certain minimum. Elements yielding within a small fraction of the load corresponding to the stress ratio are assumed also to have become plastic for the purpose of subsequent computations.

The stiffness for elements in plastic range is stress dependent. Thus over an increment of load the stiffness will change. Noting that plastic yield is, in general, local in character, it is reasonable to assume that the change in stiffness will affect the stress without significantly altering the strains or displacements. An iterative procedure is included to allow for this. Also to determine whether elements initially in plastic range unload elastically in a non-monotonic loading sequence, a pilot analysis is carried out, assuming all elements to be elastic, whenever the structure geometry is changed or a new loading applied. Details of these procedures are discussed in Part I-Technical Report.

After evaluation of stress ratio, the correct stresses in all elements are calculated corresponding to yield of the next group of elements. The difference between the incremental stress corresponding to the total load increment and the stress allowed by the stress ratio is treated as unbalanced stress and applied as psuedo-load in the next iteration.

2.3. Input Data

Input to the program is a sequence of punched cards in the following order and format:



Read Control Information for the Structure or Load Increment
Is it an intermediate step ?
No
Yes
Pilot analysis needed. Assume all elements elastic in first approx.
Analysis of Progressive Failure of Incremental System
Set up System Stiffness
Calculate incremental stresses and displacements
For Pilot analysis and NNN=1, is any element previously plastic ?
No
Yes
Is Strain increasing in any element ?
No
Element is elastic for increment
Yes
Element is plastic for increment
Do any new elements yield ?
No
Stress Ratio=1
Yes
C
A
B
D

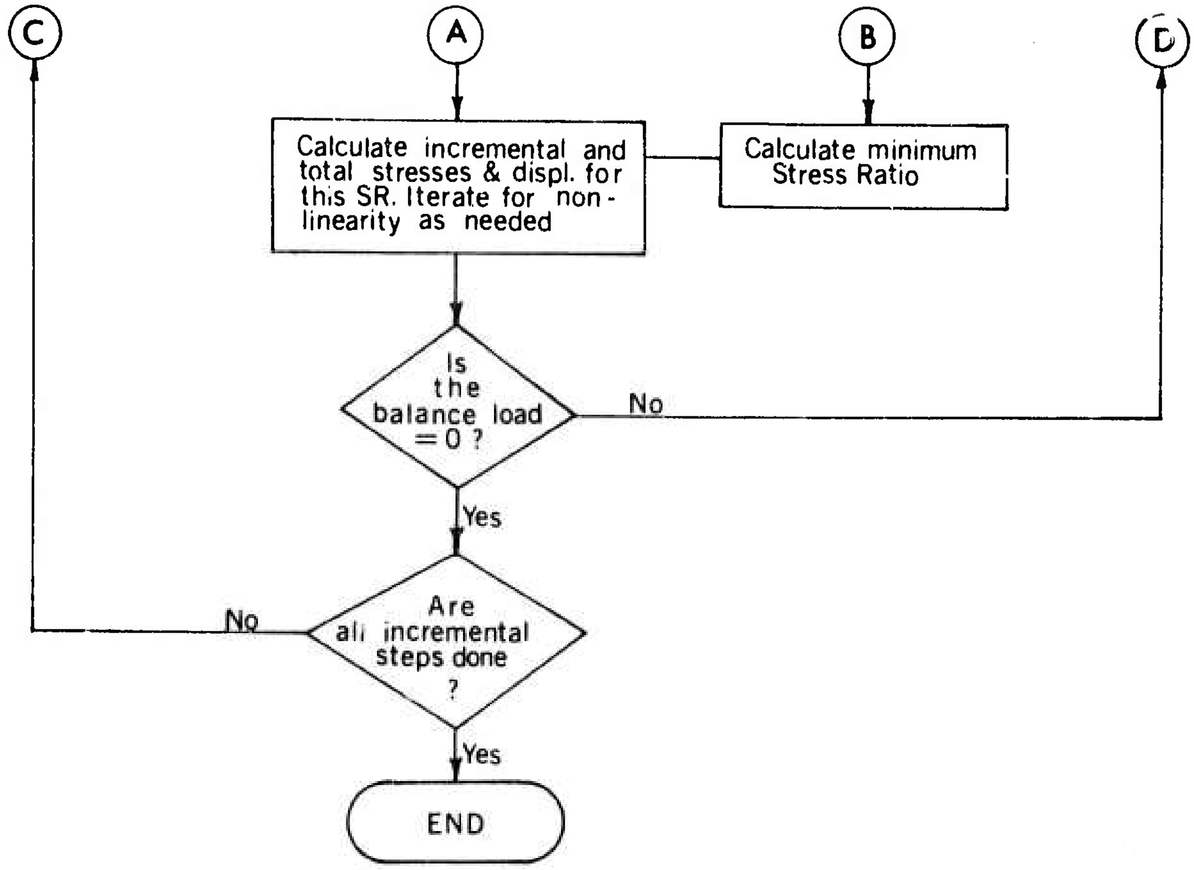


FIG. II-1. Flow Chart for Progressive Failure of Elastic-Plastic Incremental System

a. First Card. Job Title (18A4)

This card carries a descriptive title to identify the job.

b. Second Card. Control Information (4I5, 2F10.2, 3I5)

This card carries the control information for the problem being solved.

Information	Columns
Total number of nodal points (NUMNP)	1-5
Total number of elements (NUMEL)	6-10
Number of different materials (NUMMAT)	11-15
Number of pressure cards (NUMPC)	16-20
Acceleration in X-direction (ACELR)	21-30
Acceleration in Y-direction (ACELZ)	31-45
Total number of excavation and construction steps (NSTEP)	46-50
Code to designate whether initial stress is being fed as data or will be evaluated before the first step (MCASE) MCASE = 0, initial stress will be evaluated = 1, initial stress is fed as data	51-55
Maximum number of elements/nodal points removed or added in any one step (NMR).	56-60

c. Material property cards

One set of cards will be provided for each material. Each set will consist of the following cards:

i. First Card (I5, F10.0)

Information	Columns
Material identification number (MTYPE)	1-5
Mass density of material (RO)	6-15

ii. Second Card (5F10.0) for two-dimensional element

Information	Columns
Elastic modulus	1-10
Poisson's ratio	11-20
Cohesion	21-30
Angle of internal friction in degrees	31-40

or

iii. Second card (5F10.0) for one-dimensional element

Information	Columns
Elastic modulus	1-10
Poisson's ratio	11-20
Code = 1, if element is prestressed = 0, if element is not prestressed	21-30
Allowable compressive strength of the material if the element has prestressing (code = 1)	31-40
Area of one-dimensional element	41-50

d. Nodal Point Data (I5, F5.0, 5 F10.0)

One card for each nodal point with the following information is provided.

Information	Columns
Nodal point number	1-5
Type of nodal point (CODE)	6-10
X-ordinate	11-20

Y-ordinate	21-30
X load or displacement (XR)	31-40
Y load or displacement (XZ)	41-50

The code will be defined as follows:

CODE	Implication
0.0	XR is the specified X-load and XZ is the specified Y-load
1.0	XR is the specified X-displacement and XZ is the specified Y-load
2.0	XR is the specified X-load and XZ is the specified Y-displacement
3.0	XR is the specified X-displacement and XZ is the specified Y-displacement.

Nodal point cards must be in numerical sequence. Nodal points for which no cards are input will be generated by interpolation between specified nodal points. These points will have CODE and loads set equal to zero. The X, Y coordinates will be linearly interpolated.

e. Element Data (6I5, 5F10.0)

One card for each element, in numerical sequence, will show:

Information	Columns
Element number	1-5
Nodal point I	6-10
Nodal point J	11-15
Nodal point K	16-20
Nodal point L	21-25
Material type	26-30

Nodal points are labelled I, J, K, L counter-clockwise.

Initial stresses:

Component in X-direction	31-40
Component in Y-direction	41-50
Shearing stress in X-Y plane	51-60
Component in Z-direction	61-70
Thickness of element (TH)	71-80

If the columns for thickness are left blank, thickness will be taken as 1.0. Elements omitted from the sequence will be generated. The material type and thickness for generated elements is the same as for the preceding element.

f. If the initial stresses are to be evaluated (i.e. if MCASE = 0 in card b), the following card should be included; otherwise proceed to g.

Initial Stress Evaluation

i. First card (18A4)

This card gives the descriptive title of the step.

ii. Second card (4I5)

Information	Columns
Number of nodal points in this step	1-5
Number of elements in this step	6-10
Blank	11-15
Number of pressure boundary cards	16-20

g. Incremental Step Information

One set of cards will be provided for each step of construction or dismantling as follows:

i. First card (18A4)

This card will give a descriptive title of the step.

ii. Second card (10I5)

Information	Columns
Number of nodal points in this step	1-5
Number of elements in this step	6-10
Number of elements removed/added	11-15
Number of pressure boundary cards	16-20
Material type of new elements	21-25
Code = 0 for dismantling = 1 for construction	26-30
Number of nodal points removed/added	31-35

If in addition to adding/removing elements, it is desired to change the material type of some existing elements, use columns 36-45.

Number of elements for which material type is to be changed	36-40
New material type of changed elements	41-45

If it is desired to add/remove two material types in one step, use columns 46-50.

KMORE = 1, another material is being added/removed = 0, no other material is being added/removed.	46-50

iii. Elements removed/added (16I5)

One or more cards will indicate the element numbers removed or added in this step. Total number of elements should be the same as in columns 11-15 in card ii.

iv. Nodal points removed/added (16I5)

One or more cards will indicate the nodal point numbers removed or added in this step. The total number should be the same as in columns 31-35 of card ii.

v. Elements for which material type changed (16I5)

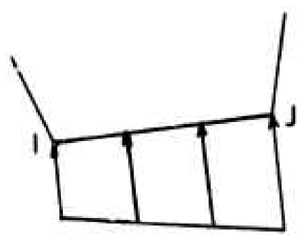
One or more cards will indicate the element numbers for which material type has been changed. Total number should be the same as in columns 36-40 in card ii.

If KMORE = 1, repeat card i to v for the next material; otherwise proceed to card vi.

vi. Pressure boundary cards (2I5, 2F10.0)

The cards representing pressure at the boundary will be as follows:

Information	Columns
Nodal point I	1-5
Nodal point J	6-10
Total normal pressure at I	11-20
Total normal pressure at J	21-30



As shown in the sketch, the boundary of the element must be on the left hand side as one progresses from I to J. Surface tensile force is input as a negative pressure.

2.4. Output information

The following information is developed and printed by the program:

a. Reprint of the input data

b. Cumulative nodal point displacement after each step

c. Stresses at the center of each element after each step

d. Stress ratios for elements which can yield with full application of the balance load

e. The strss ratio applicable for any iteration and the consideration leading to its determination e.g. yielding of next element, minimum stress ratio, maximum stress ratio, ensuring that elements previously in yield surface do not move away from that surface beyond a specified tolerance.

f. The proportion of total load increment applied upto the end of current iteration.

g. The next group of elements reaching yield

h. The failure ratio for each element.

2.5. Fortran Listing

```
C     *    FINITE ELEMENT ANALYSIS OF ELASTIC-PERFECTLY PLASTIC       *
C     *    MOHR-COULOMB SOLIDS UNDER PLANE STRAIN.  INCREMENTAL       *
C     *    CONSTRUCTION  EXCAVATION AND INITIAL STRESSES ARE          *
C     *    CONSIDERED.   A 4-CST QUADRILATERAL ELEMENT IS USED.       *
C     *    THE FORMULATION IS DOCUMENTED IN THE FINAL REPORT          *
C     *    ON CONTRACT H0210017 BETWEEN THE OHIO STATE UNIVERSITY     *
C     *    AND THE UNITED STATES BUREAU OF MINES SUPPORTED BY         *
C     *    THE ADVANCED RESEARCH PROJECTS AGENCY.                     *
C     *    PROGAMMERS:   R.D.SINGH AND K.J.SINGH                      *
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TOTAL/ AA(30000), IA(7000)
      COMMON/ONE/NUMNP,NUMEL,NUMMAT,NUMPC,NPMAX,NELMAX,N,NNN,KKK,KLK,
     .    ISHIFT,MBAND,NBAND,NUMBLK,MTYPE,NCASE,MCASE,NMR,NL,NPP,NEQ,NPC
     .    ,NP,NSTEP,NCODE
      COMMON/TWO/RR(5),ZZ(5),S(10,10),P(10),ST(3,10),C(4,4),SIG(7),FF(4)
     .    ,TITLE(18),SR1,SR2,XC,YC,ACELR,ACELZ,BCELR,BCELZ,VOL
      COMMON/THREE/STOP,MTOT,NTOT,N13,N14,M8
C
C        ACELR,ACELZ ARE BODY FORCE INTENSITIES.   MCASE IS INPUT AS
C        ZERO IF INITIAL STRESSES HAVE TO BE COMPUTED AS PART OF
C        THE ANALYSIS.
C
      READ(5,1000)TITLE ,NUMNP,NUMEL,NUMMAT,NUMPC,ACELR,ACELZ,
     .             NP,NSTEP,MCASE,NMR
      WRITE (6,2000) TITLE,NUMNP,NUMEL,NUMMAT,NUMPC,ACELR,ACELZ,NP,
     .             NSTEP,NMR
      MTOT=7000
      NTOT=30000
      NPC=NUMPC
      IF(NUMPC.EQ.0) NPC=1
      IF(NMR.EQ.0) NMR=1
      N1=1
      N2=N1+NUMNP
      N3=N2+NUMNP
      N4=N3+NUMNP
      N5=N4+NUMNP
      N6=N5+NUMNP
      N7=N6+2*NUMNP
      N8=N7+5*NUMMAT
      N9=N8+NUMMAT
      N10=N9+2*NPC
      N11=N10+8*NUMEL
      N12=N11+NUMEL
      N13=N12+NUMEL
      M1=1
      M2=M1+5*NUMEL
      M3=M2+NUMEL
      M4=M3+NPC
      M5=M4+NPC
      M6=M5+NMR
```

```
      M7=M6+NMR
      M8=M7+NMR
      JJ=M8-MTOT
      IF(JJ.LE.0) GO TO 100
      WRITE(6,3000)JJ
      CALL EXIT
  100 CONTINUE
      NEQ=2*NUMNP
      CALL INPT  (AA(N1),AA(N2),AA(N3),AA(N4),AA(N5),AA(N6),AA(N7),
     *            AA(N8),AA(N9),AA(N10),AA(N11),AA(N12),  IA(M1),IA(M2),
     *            IA(M3),IA(M4),IA(M5),IA(M6),IA(M7)  )
 1000 FORMAT(18A4/4I5,2F10.2,4I5)
 2000 FORMAT(1H1,18A4/
     .30H0  NUMBER OF NODAL POINTS                                    I10/
     .30H0  NUMBER OF ELEMENTS                                        I10/
     .30H0  NUMBER OF DIFF. MATERIALS                                 I10/
     .30H0  NUMBER OF PRESSURE CARDS                                  I10/
     .29H0  X-ACCELERATION                                          F11.1/
     .29H0  Y-ACCELERATION                                          F11.1/
     .30H0  NUMBER OF APPROXIMATIONS                                  I10/
     .30H0  NUMBER OF STEPS                                           I10/
     .69H0  MAX. NO. OF NODAL POINTS OR ELEMENTS ADDED OR REMOVED AT ANY
     . STEP            I5)
 3000 FORMAT(70H  EXECUTION TERMINATED IN MAIN PROGRAM. REQUIRED CORE EX
     .CEEDS MTOT BY                              I10)
      STOP
      END
```

```
      SUBROUTINE INPT (R,Z,UR,UZ,CODE,BO,E,RO,PR,SIGI,TH,ER,IX,MTAG,
     *           IBC,JBC,NUMR,NUMR1,NNP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TOTAL/ AA(30000), IA(7000)
      COMMON/ONE/NUMNP,NUMEL,NUMMAT,NUMPC,NPMAX,NELMAX,N,NNN,KKK,KLK,
     .    ISHIFT,MBAND,NBAND,NUMBLK,MTYPE,NCASE,MCASE,NMR,NL,NPP,NEQ,NPC
     .    ,NP,NSTEP,NCODE
      COMMON/TWO/RR(5),ZZ(5),S(10,10),P(10),ST(3,10),C(4,4),SIG(7),EE(4)
     .    ,TITLE(18),SR1,SR2,XC,YC,ACELR,ACELZ,BCELR,BCELZ,VOL
      COMMON/THREE/STOP,MTOT,NTOT,N13,N14,MB
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),CODE(NUMNP),
     .    BO(NEQ),E(5,NUMMAT),RO(NUMMAT),PR(NPC   ,2),SIGI(NUMEL,8),
     .    TH(NUMEL),ER(NUMEL),IX(NUMEL,5),MTAG(NUMEL),IBC(NPC),
     .    JBC(NPC   ),NUMR(NMR),NUMR1(NMR),NNP(NMR)
C
C         INPUT MATERIAL PROPERTIES
C
      DO 55 M=1,NUMMAT
      READ (5,1001) MTYPE,RO(MTYPE)
      WRITE(6,2001) MTYPE,RO(MTYPE)
      READ (5,1002) (E(J,MTYPE),J=1,5)
      WRITE(6,2002) (E(J,MTYPE),J=1,5)
   55 CONTINUE
C
C         READ AND PRINT NODAL POINT DATA
C
      WRITE (6,2003)
      L=0
   60 READ  (5,1003) N,CODE(N),R(N),Z(N),UR(N),UZ(N)
      NL=L+1
      IF (N.EQ.1) GO TO 70
      ZX=N-L
      DR=(R(N)-R(L))/ZX
      DZ=(Z(N)-Z(L))/ZX
   70 L=L+1
      IF(N-L) 100,90,80
   80 CODE(L)=0.0
      R(L)=R(L-1)+DR
      Z(L)=Z(L-1)+DZ
      UR(L)=0.0
      UZ(L)=0.0
      GO TO 70
   90 IF(NUMNP-N) 100,110,60
  100 WRITE (6,2005) N
      CALL EXIT
  110 WRITE (6,2004)((K,CODE(K),R(K),Z(K),UR(K),UZ(K)),K=1,NUMNP)
C
C         READ AND PRINT ELEMENT DATA
C
      WRITE (6,2006)
```

```
      N=0
  130 READ (5,1004) M,(IX(M,I),I=1,5),(SIGI(M,I),I=1,4),TH(M)
      IF (TH(M).EQ.0.) TH(M)=1.0
      ZX=M-N
      IF (N.EQ.0) GO TO 140
      DO 135 I=1,4
  135 SIG(I)=(SIGI(M,I)-SIGI(N,I))/ZX
  140 N=N+1
      IF(M.LE.N) GO TO 170
      IX(N,1)=IX(N-1,1)+1
      IX(N,2)=IX(N-1,2)+1
      IX(N,3)=IX(N-1,3)+1
      IX(N,4)=IX(N-1,4)+1
      IX(N,5)=IX(N-1,5)
      DO 160 I=1,4
  160 SIGI(N,I)=SIGI(N-1,I)+SIG(I)
      TH(N)=TH(N-1)
  170 IF(M.GT.N) GO TO 140
      IF(NUMEL.GT.N) GO TO 130
      WRITE(6,2007) ((N,(IX(N,I),I=1,5),(SIGI(N,I),I=1,4),TH(N)),N=1,NUM
     .                  EL)
C
C        INITIALIZATION OF UNBALANCED STRESSES AND CUMULATIVE
C        DISPLACEMENTS
C
      DO 195 N=1,NUMEL
      MTAG(N)=0
      SIGI(N,5)=0.
      SIGI(N,6)=0.
      SIGI(N,7)=0.
  195 SIGI(N,8)=0.
      DO 200 N=1,NUMNP
      BO(2*N-1)=0.
  200 BO(2*N)=0.
C
C        CALCULATE MAXIMUM BAND-WIDTH FOR THE INCREMENTAL SYSTEM
C
      J=0
      DO 250 N=1,NUMEL
      DO 250 I=1,4
      DO 250 L=1,4
      KK=IABS(IX(N,I)-IX(N,L))
      IF(KK.GT.J) J=KK
  250 CONTINUE
      NBAND=2*J+2
      WRITE(6,3000) NBAND
C
C        CALCULATE BLOCKSIZE CONSISTENT WITH AVAILABLE CORE
C
      NL=(NTOT-N13+1)/(NBAND+1)
```

```
      NLL=NEQ+3
      IF(NL.GT.NLL) NL=NLL
      NL=NL/4
      NL=4*NL
      N14=N13+NL
      NB=2*NBAND
      NZ=N13+NB+NB*NBAND-1
      WRITE (6,4010) NZ
      IF(NZ.GT.NTOT) CALL EXIT
      IF (NL.LT.NB) CALL EXIT
C     ..............................................................
C     .  READ CONTROL INFORMATION FOR THE NEXT STEP IN INCREMENTAL  .
C     .  ANALYSIS                                                    .
C     ..............................................................
      NCASE=0
      IF(MCASE.NE.0) NCASE=1
      IF(MCASE.NE.0) MCASE=1
  300 READ(5,1008) TITLE
      WRITE(6,2011) TITLE
      READ(5,1009) NPMAX,NELMAX,NUMER,NUMPC,MTYPE,NCODE,NPMIS,
     .             NUMER1,MTYPE1,KMORE,THICK
      IF(THICK.EQ.0.) THICK=1.0
      IF(NUMER.EQ.0) GO TO 310
      READ(5,1007)  (NUMR(N), N= 1,NUMER)
      DO 305  I=1,NUMER
      NUM=NUMR(I)
      TH(NUM)=THICK
  305 CONTINUE
  310 IF(NPMIS.EQ.0) GO TO 320
      READ(5,1007)  (NNP(M), M = 1,NPMIS)
  320 IF (NUMER1.EQ.0) GO TO 330
      READ(5,1007) ( NUMR1(N),N=1,NUMER1)
C
C        DETERMINE BAND-WIDTH FOR THIS STEP
C
  330 J=0
      IF(KMORE.NE.0) GO TO 342
      DO 340 N = 1,NELMAX
      DO 340 I=1,4
      DO 340 L=1,4
      KK=IABS(IX(N,I)-IX(N,L))
      IF(KK.GT.J) J=KK
  340 CONTINUE
      MBAND=2*J+2
  342 IF(NPMIS.EQ.0) GO TO 346
      DO 345 I = 1, NPMIS
      J=NNP(I)
      BO(2*J-1)=0.
      BO(2*J)=0.
  345 CONTINUE
```

```
C
C        PRINT CONTROL INFORMATION FOR THE CURRENT STEP IN INCREMENTAL
C        ANALYSIS
C
  346 IF(NCODE.EQ.1) GO TO 410
      IF(NUMER.EQ.0) GO TO 351
      WRITE(6,2012)NCASE,NPMAX,NELMAX,NUMPC,NPMIS,MTYPE,
     .             (NUMR(N),N=1,NUMER)
      GO TO 352
  351 WRITE(6,2012)NCASE,NPMAX,NELMAX,NUMPC,NPMIS,MTYPE
  352 IF(NPMIS.EQ.0)  GO TO 420
      WRITE(6,2014) (NNP(M) ,M = 1,NPMIS)
      GO TO 420
  410 IF(NUMER.EQ.0) GO TO 411
      WRITE(6,2013)NCASE,NPMAX,NELMAX,NUMPC,NPMIS,MTYPE,
     .             (NUMR(N),N=1,NUMER)
      GO TO 412
  411 WRITE(6,2013)NCASE,NPMAX,NELMAX,NUMPC,NPMIS,MTYPE
  412 IF(NPMIS.EQ.0)  GO TO 420
      WRITE(6,2014) (NNP(M) ,M = 1,NPMIS)
  420 CONTINUE
      IF (NUMER1.EQ.0) GO TO 440
      WRITE (6,2018) (NUMR1(N),N=1,NUMER1)
      WRITE (6,2020) MTYPE1
  440 IF (KMORE.NE.0) GO TO 510
      WRITE (6,2025) MBAND
C
C        READ AND PRINT BOUNDARY PRESSURES, IF ANY
C
      IF(NUMPC.EQ.0) GO TO 510
      WRITE (6,2008)
      DO 500 L=1,NUMPC
      READ(5,1005) IBC(L),JBC(L),PR(L,1),PR(L,2)
  500 WRITE(6,2009)IBC(L),JBC(L),PR(L,1),PR(L,2)
  510 CONTINUE
C
C        FOR ELEMENTS REMOVED OR NEWLY ADDED IN THIS STEP
C        SET INITIAL STRESSES EQUAL TO ZERO
C
      IF (NUMER.EQ.0) GO TO 601
      DO 600 I=1,NUMER
      NUM = NUMR(I)
      IX(NUM,5)=MTYPE
      SIGI(NUM,1)=0.0
      SIGI(NUM,2)=0.0
      SIGI(NUM,3)=0.0
      SIGI(NUM,4)=0.0
      MTAG(NUM)=0
  600 CONTINUE
  601 IF (NUMER1.EQ.0) GO TO 610
```

```
      DO 606 I=1,NUMERI
      NUM=NUMR1(I)
  606 IX(NUM,5)=MTYPE1
  610 IF (KMORE.NE.0) GO TO 300
C
C           EVERY ELEMENT IS ASSUMED ELASTIC AT THE START OF EACH STEP
C
      SR1=1.0
      SR2=0.0
      NPP=NP
      IF (NCASE.LT.1) NPP=1
      CALL SOLVE(P,Z,UP,UZ,CODE,BO,E,RO,PR,SIGI,TH,FR,AA(N13),AA(N14),
     *                  IX,MTAG,IBC,JBC,NUMR,NUMR1,NNP)
      IF(NCASE.LE.NSTEP) GO TO 300
 1001 FORMAT (I15,1F10.0)
 1002 FORMAT (6F10.0)
 1006 FORMAT(A6)
 1003 FORMAT (I5,F5.0,4F10.0)
 1004 FORMAT(6I5,5F10.0)
 1005 FORMAT (2I5,2F10.0)
 1007 FORMAT(16I5)
 1008 FORMAT(18A4)
 1009 FORMAT(10I5,F10.0)
 2001 FORMAT (17HOMATERIAL NUMBER= I3, 15H, MASS DENSITY= F12.4)
 2002 FORMAT(16HOELASTIC MODULUS 14X 2HNU 8X 8HCOHESION 2X 14HFRICTION A
     .NGLE 2X 13HAREA FOR ONED /(2E16.5,3F16.5))
 2003 FORMAT (111H1NODAL POINT          TYPE  X ORDINATE  Y ORDINATE  X LO
     .AD OR DISPLACEMENT  Y LOAD OR DISPLACEMENT  PORE PRESSURE )
 2004 FORMAT (I12,F12.2,2F12.3,2E24.7)
 2005 FORMAT (26HONODAL POINT CARD ERROR N= I5)
 2006 FORMAT(109H1ELEMENT NO.      I      J      K      L    MATERIAL    X-S
     .TRESS     Y-STRESS    XY-STRESS     Z-STRESS    THICKNESS          )
 2007 FORMAT(I12,4I6,1I12,5F12.3)
 2008 FORMAT (29HOPRESSURE BOUNDARY CONDITIONS/ 40H      I       J     PRESS
     .URE I      PRESSURE J                                               )
 2009 FORMAT (2I6,2F12.3)
 2011 FORMAT(1H1 18A4////)
 2012 FORMAT( 51H0 RESULTS AFTER STEP NO.-------------------------   I5/
     . 58H0 TOTAL NUMBER OF NODAL POINTS IN SYSTEM AT THIS STEP----   I5/
     . 58H0 TOTAL NUMBER OF ELEMENTS IN SYSTEM AT THIS STEP--------   I5/
     . 58H0 TOTAL NUMBER OF PRESSURE CARDS AT THIS STEP------------   I5/
     . 58H0 TOTAL NUMBER OF NODAL POINTS MISSING AT THIS STEP------   I5/
     . 58H0 MATERIAL TYPE OF ELEMENTS ADDED/REMOVED IN THIS STEP---   I5/
     . 58H0 ELEMENTS REMOVED IN THIS STEP ARE  -------------------     /
     . (20I5))
 2013 FORMAT( 51H0 RESULTS AFTER STEP NO.-------------------------   I5/
     . 58H0 TOTAL NUMBER OF NODAL POINTS IN SYSTEM AT THIS STEP----   I5/
     . 58H0 TOTAL NUMBER OF ELEMENTS IN SYSTEM AT THIS STEP--------   I5/
     . 58H0 TOTAL NUMBER OF PRESSURE CARDS AT THIS STEP------------   I5/
     . 56H0 TOTAL NUMBER OF NODAL POINTS MISSING AT THIS STEP------   I5/
```

```
     . 58HO MATERIAL TYPE BEING PUT IN THIS STEP ------------------  I5/
     . 58HO ELEMENTS ADDED   IN THIS STEP ARE  --------------------    /
     . (20I5))
2014 FORMAT( 53HO NODAL POINTS REMOVED OR ADDED IN THIS STEP ARE-----
     ./(20I5))
2016 FORMAT(I8,2F8.2,1P4E12.4,0P1F8.5)
2017 FORMAT(I10,2F20.7)
2018 FORMAT (52HO ELEMENTS FOR WHICH MATERIAL HAS BEEN CHANGED ARE--
     ./(20I5))
2020 FORMAT (54HO NEW MATERIAL TYPE OF THE ABOVE CHANGED ELEMENTS IS--
     . I5)
2025 FORMAT(27H BAND WIDTH FOR THIS STEP          I5)
3000 FORMAT(//12H BAND WIDTH                 I5)
4010 FORMAT( 65H THE MINIMUM DIMENSION OF AA AND NTOT NEEDED FOR THIS P
     .ROBLEM IS                 I10)
      RETURN
      END
```

```
      SUBROUTINE SOLVE (R,Z,UR,UZ,CODE,BO,E,RO,PR,SIGI,TH,FR,B,A,IX,
     *                  MTAG,IBC,JBC,NUMR,NUMR1,NNP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TOTAL/ AA(30000), IA(7000)
      COMMON/ONE/NUMNP,NUMEL,NUMMAT,NUMPC,NPMAX,NELMAX,N,NNN,KKK,KLK,
     .    ISHIFT,MBAND,NBAND,NUMBLK,MTYPE,NCASE,MCASE,NMR,NL,NPP,NEQ,NPC
     .    ,NP,NSTEP,NCODE
      COMMON/TWO/RR(5),ZZ(5),S(10,10),P(10),ST(3,10),C(4,4),SIG(7),FF(4)
     .    ,TITLE(18),SR1,SR2,XC,YC,ACELR,ACELZ,BCELR,BCELZ,VOL
      COMMON/THREE/STOP,MTOT,NTOT,N13,N14,MB
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),CODE(NUMNP),
     .    BO(NEQ),E(5,NUMMAT),RO(NUMMAT),PR(NPC   ,2),SIGI(NUMEL,8),
     .    TH(NUMEL),FR(NUMEL),IX(NUMEL,5),MTAG(NUMEL),IBC(NPC),
     .    JBC(NPC   ),NUMR(NMR),NUMR1(NMR),NNP(NMR)
      DIMENSION B(NL),A(NL,NBAND)
      DIMENSION LM(4)
C
      DO 750 NNN=1,NPP
      KKK=0
      REWIND 2
      ND=NL/2
      NB=ND/2
      STOP=0.0
      NUMBLK=0
      DO 50 N=1,NL
      B(N)=0.0
      DO 50 M=1,MBAND
   50 A(N,M)=0.0
      IF(NCASE.GT.MCASE) GO TO  51
      IF (NNN.GT.1) GO TO  51
      BCELR =ACELR
      BCELZ =ACELZ
   51 CONTINUE
      IF (NNN.GT.1) GO TO  52
      ACELR =BCELR
      ACELZ =BCELZ
      GO TO  55
   52 ACELR =0.0
      ACELZ =0.0
      NUMPC =0
   55 CONTINUE
   60 NUMBLK=NUMBLK+1
      NH=NB*(NUMBLK+1)
      NM=NH-NB
      NS=NM-NB+1
      KSHIFT=2*NS-2
      DO 210 N=1,NELMAX
      IF(IX(N,5).LE.0) GO TO 210
      MTYPE=IX(N,5)
      IF (RO(MTYPE).EQ.0.) GO TO 210
```

```
      DO 80 I=1,4
      IF(IX(N,I).LT.NS) GO TO 80
      IF(IX(N,I).LE.NM) GO TO 90
   80 CONTINUE
      GO TO 210
   90 IF(IX(N,3).NE.IX(N,4)) GO TO 95
      CALL ONED (R,Z,UR,UZ,CODE,RO,E,RO,PR,SIGI,TH,FR,IX,MTAG,
     *            IBC,JBC,NUMP,NUMP1,NNP)
      IX(N,5)=-IX(N,5)
      MM=2
      GO TO 130
   95 CALL QUAD (R,Z,UR,UZ,CODE,RO,E,RO,PR,SIGI,TH,FR,IX,MTAG,
     *            IBC,JBC,NUMP,NUMP1,NNP)
      IX(N,5)=-IX(N,5)
      IF (VOL.GT.0.) GO TO 110
      WRITE(6,2000) N
      STOP=1.0
  110 MM=4
      IF (IX(N,3).EQ.IX(N,4)) MM=3
  130 DO 140 I=1,MM
  140 LM(I)=2*IX(N,I)-2
      DO 200 I=1,MM
      DO 200 K=1,2
      II=LM(I)+K-KSHIFT
      KK=2*I-2+K
      B(II)=B(II)+P(KK)
      DO 200 J=1,MM
      DO 200 L=1,2
      JJ=LM(J)+L-II+1-KSHIFT
      LL=2*J-2+L
      IF(JJ.LE.0) GO TO 200
      A(II,JJ)=A(II,JJ)+S(KK,LL)*TH(N)
  200 CONTINUE
  210 CONTINUE
      DO 220 N=NS,NM
      IF (N.GT.NPMAX) GO TO 220
      K=2*N-KSHIFT
      B(K)=B(K)+UZ(N)
      B(K-1)=B(K-1)+UR(N)
  220 CONTINUE
C
      IF (NUMPC.EQ.0) GO TO 310
      DO 300 L=1,NUMPC
      I=IBC(L)
      J=JBC(L)
      DR=Z(I)-Z(J)
      DZ=R(J)-R(I)
      PP2=(PR(L,2)+PR(L,1))/6.
      PP1=PP2+PR(L,1)/6.
      PP2=PP2+PR(L,2)/6.
```

```
      II=2*I-KSHIFT
      JJ=2*J-KSHIFT
      IF((II.LE.0).OR.(II.GT.ND)) GO TO 265
      B(II-1)=B(II-1)+PP1*DR
      B(II)=B(II)+PP1*DZ
  265 IF((JJ.LE.0).OR.(JJ.GT.ND)) GO TO 300
      B(JJ-1)=B(JJ-1)+PP2*DR
      B(JJ)=B(JJ)+PP2*DZ
  300 CONTINUE
  310 DO 400 M=NS,NH
      IF(M.GT.NPMAX) GO TO 400
      U=UR(M)
      N=2*M-1-KSHIFT
      IF (CODE(M)) 390,400,316
  316 IF(CODE(M).EQ.1.) GO TO 370
      IF(CODE(M).EQ.2.) GO TO 390
      IF (CODE(M)-3.) 390,380,390
  370 CALL MODIFY(A,B,NL ,MBAND,NBAND,N,U)
      GO TO 400
  380 CALL MODIFY(A,B,NL ,MBAND,NBAND,N,U)
  390 U=UZ(M)
      N=N+1
      CALL MODIFY(A,B,NL ,MBAND,NBAND,N,U)
  400 CONTINUE
      WRITE (2) (B(N),(A(N,M),M=1,MBAND),N=1,ND)
      DO 420 N=1,ND
      K=N+ND
      B(N)=B(K)
      B(K)=0.0
      DO 420 M=1,MBAND
      A(N,M)=A(K,M)
  420 A(K,M)=0.0
      IF(NM.LT.NPMAX) GO TO 60
      IF (STOP.NE.0.) CALL EXIT
      DO 600 N=1,NUMNP
      UZ(N)=0.
  600 UR(N)=0.
      N15=N14+2*NUMNP
      N16=N15+4*NUMEL
      N17=N16+NUMEL
      CALL BANSOL (NL ,NBAND,MBAND,NUMBLK,B,A)
      CALL STRESS (R,Z,UR,UZ,CODE,BO,E,PO,PR,SIGI,TH,EP,B,AA(N15),
     *             AA(N16),AA(N17),IX,MTAG,IBC,JBC,NUMR,NUMR1,NNP)
      DO 700 N=1,NPMAX
      NN=2*N
      BO(NN-1)=BO(NN-1)+B(NN-1)
  700 BO(NN)=BO(NN)+B(NN)
C
C          PRINT OUT CUMULATIVE DISPLACEMENTS
C
```

```
      WRITE(6,2002)(N,BO(2*N-1),BO(2*N),N=1,NUMNP)
      IF(KLK.EQ.1) GO TO 750
      IF((KKK.EQ.0).OR.(SR2.GT.1.)) GO TO 800
  750 CONTINUE
  800 IF(NUMER.GT.0) GO TO 850
      IF(NCASE.GT.0) GO TO 850
C
C     IF INITIAL STRESSES ARE EVALUATED IN THIS STEP INITIALIZE BO
C
      DO 810 N=1,NEQ
  810 BO(N)=0.
  850 NCASE=NCASE+1
      DO 900 N=1,NELMAX
  900 IF(ER(N).GT.1.) ER(N)=1.
      RETURN
 1000 FORMAT (1H1,40X,'LOAD MATRIX'/(10E13.4))
 2000 FORMAT (26HONEGATIVE AREA ELEMENT NO. I4)
 2002 FORMAT(12H1 N.P NUMBER 18X 2HUX 18X 2HUY /(I12,2F20.7))
      END
```

```
      SUBROUTINE ONED (R,Z,UR,UZ,CODE,BO,E,RO,PR,SIGI,TH,FR,IX,MTAG,
     *                 IBC,JBC,NUMR,NUMR1,NNP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TOTAL/ AA(30000), IA(7000)
      COMMON/ONE/NUMNP,NUMEL,NUMMAT,NUMPC,NPMAX,NELMAX,N,NNN,KKK,KLK,
     .   ISHIFT,MBAND,NBAND,NUMBLK,MTYPE,NCASE,MCASE,NMR,NL,NPP,NEQ,NPC
     .   ,NP,NSTEP,NCODE
      COMMON/TWO/RR(5),ZZ(5),S(10,10),P(10),ST(3,10),C(4,4),SIG(7),EE(4)
     .   ,TITLE(18),SR1,SR2,XC,YC,ACELR,ACELZ,BCELR,BCELZ,VOL
      COMMON/THREE/STOP,MTOT,NTOT,N13,N14,MB
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),CODE(NUMNP),
     .    BO(NEQ),E(5,NUMMAT),RO(NUMMAT),PR(NPC  ,2),SIG1(NUMEL,8),
     .    TH(NUMEL),FR(NUMEL),IX(NUMEL,5),MTAG(NUMEL),IBC(NPC),
     .    JBC(NPC  ),NUMR(NMR),NUMR1(NMR),NNP(NMR)
C
      DO 100 I=1,8
      P(I)=0.0
      DO 100 J=1,8
  100 S(I,J)=0.0
      MTYPE=IX(N,5)
      I=IX(N,1)
      J=IX(N,2)
      DX=R(J)-R(I)
      DY=Z(J)-Z(I)
      XL=DSQRT(DX**2+DY**2)
      COSA=DX/XL
      SINA=DY/XL
      COMM= E(1,MTYPE)*E(5,MTYPE)/XL
      S(1,1)=COSA*COSA*COMM
      S(1,2)=COSA*SINA*COMM
      S(1,3)=-S(1,1)
      S(1,4)=-S(1,2)
      S(2,1)=S(1,2)
      S(2,2)=SINA*SINA*COMM
      S(2,3)=-S(1,2)
      S(2,4)=-S(2,2)
      S(3,1)=S(1,3)
      S(3,2)=S(2,3)
      S(3,3)=S(1,1)
      S(3,4)=S(1,2)
      S(4,1)=S(1,4)
      S(4,2)=S(2,4)
      S(4,3)=S(3,4)
      S(4,4)=S(2,2)
      II=4
      IF(NNN.EQ.1) II=0
      EP=SIGI(N,II+1)/E(1,MTYPE)
      DX=DX*EP
      DY=DY*EP
      P(1)=S(1,1)*DX+S(1,2)*DY
```

```
      P(2)=S(2,1)*DX+S(2,2)*DY
      P(3)=-P(1)
      P(4)=-P(2)
      RETURN
      END
```

```
      SUBROUTINE QUAD (R,Z,UR,UZ,CODE,BD,E,RO,PR,SIGI,TH,FR,IX,MTAG,
     *                IBC,JBC,NUMR,NUMR1,NNP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TOTAL/ AA(30000), IA(7000)
      COMMON/ONE/NUMNP,NUMEL,NUMMAT,NUMPC,NPMAX,NELMAX,N,NNN,KKK,KLK,
     .    ISHIFT,MBAND,NBAND,NUMBLK,MTYPE,NCASE,MCASE,NMR,NL,NPP,NEQ,NPC
     .    ,NP,NSTEP,NCODE
      COMMON/TWO/RR(5),ZZ(5),S(10,10),P(10),ST(3,10),C(4,4),SIG(7),EE(4)
     .    ,TITLE(18),SR1,SR2,XC,YC,ACELR,ACELZ,BCELR,BCELZ,VOL
      COMMON/THREE/STOP,MTOT,NTOT,N13,N14,M8
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),CODE(NUMNP),
     .    BD(NEQ),E(5,NUMMAT),RO(NUMMAT),PR(NPC  ,2),SIGI(NUMEL,8),
     .    TH(NUMEL),FR(NUMEL),IX(NUMEL,5),MTAG(NUMEL),IBC(NPC),
     .    JBC(NPC  ),NUMR(NMR),NUMR1(NMR),NNP(NMR)
      DIMENSION LM(4),U(3),V(3)
C
      CALL STPSTR (R,Z,UR,UZ,CODE,BD,E,RO,PR,SIGI,TH,FR,IX,MTAG,
     *             IBC,JBC,NUMR,NUMR1,NNP)
      DO 130 J=1,10
      P(J)=0.
      DO 120 I=1,3
  120 ST(I,J)=0.
      DO 130 I=1,10
  130 S(I,J)=0.
      DO 140 I=1,4
      NPN=IX(N,I)
      RR(I)=R(NPN)
  140 ZZ(I)=Z(NPN)
      XC=(RR(1)+RR(2)+RR(3)+RR(4))/4.
      YC=(ZZ(1)+ZZ(2)+ZZ(3)+ZZ(4))/4.
      RR(5)=XC
      ZZ(5)=YC
      K=5
      J=1
      I=4
      LM(3)=9
      NT=4
      IF(IX(N,3).NE.IX(N,4)) GO TO 160
      NT=1
      LM(3)=5
      I=1
      K=3
      J=2
      XC=(RR(1)+RR(2)+RR(3))/3.
      YC=(ZZ(1)+ZZ(2)+ZZ(3))/3.
      RR(5)=RR(3)
      ZZ(5)=ZZ(3)
  160 DO 200 NN=1,NT
      LM(1)=2*I-1
      LM(2)=2*J-1
```

```
      U(1)=ZZ(J)-ZZ(K)
      U(2)=ZZ(K)-ZZ(I)
      U(3)=ZZ(I)-ZZ(J)
      V(1)=RR(K)-RR(J)
      V(2)=RR(I)-RR(K)
      V(3)=RR(J)-RR(I)
      AREA=(RR(J)*U(2)+RR(I)*U(1)+RR(K)*U(3))/2.
      VOL=VOL+AREA*TH(N)
      COMM=.25/AREA
      XNT=NT
      COM=2.0/XNT
      COM=COM*COMM
      DX=AREA*TH(N)*RO(MTYPE)/3.
      DY=DX*ACELZ
      DX=DX*ACELR
      DO 180 L=1,3
      II=LM(L)
      ST(1,II)=ST(1,II)+U(L)*COM
      ST(2,II+1)=ST(2,II+1)+V(L)*COM
      ST(3,II)=ST(3,II)+V(L)*COM
      ST(3,II+1)=ST(3,II+1)+U(L)*COM
      P(II)=P(II)+DX
      P(II+1)=P(II+1)+DY
      DO 180 M=1,3
      JJ=LM(M)
      S(II,JJ)=S(II,JJ)+(U(L)*C(1,1)*U(M)+V(L)*C(3,3)*V(M)+V(L)*C(1,3)*U
     .(M)+U(L)*C(1,3)*V(M))*COMM
      S(II,JJ+1)=S(II,JJ+1)+(U(L)*C(1,2)*V(M)+V(L)*C(3,3)*U(M)+V(L)*C(2,
     .3)*V(M)+U(L)*C(1,3)*U(M))*COMM
      S(II+1,JJ+1)=S(II+1,JJ+1)+(V(L)*C(2,2)*V(M)+U(L)*C(3,3)*U(M)+U(L)*
     .C(2,3)*V(M)+V(L)*C(2,3)*U(M))*COMM
      S(JJ+1,II)=S(II,JJ+1)
  180 CONTINUE
      I=J
      J=J+1
  200 CONTINUE
      IF(IX(N,3).EQ.IX(N,4)) GO TO 250
      DO 240 I=1,2
      KK=ID-I
      DO 240 K=1,KK
      CC=S(KK+1,K)/S(KK+1,KK+1)
      P(K)=P(K)-CC*P(KK+1)
      DO 230 J=1,3
  230 ST(J,K)=ST(J,K)-CC*ST(J,KK+1)
      DO 240 J=1,KK
  240 S(J,K)=S(J,K)-CC*S(J,KK+1)
  250 CONTINUE
      II=0
      IF (NNN.GT.1) II=4
      SIG(1)=-SIGI(N,II+1)
```

```
QUAD101          SIG(2)=-SIG1(N,II+2)
QUAD102          SIG(3)=-SIG1(N,II+3)
QUAD103          DO 520 I=1,8
QUAD104          DO 510 J=1,3
QUAD105      510 P(I)=P(I)+ST(J,I)*SIG(J)*VOL
QUAD106      520 CONTINUE
QUAD107          RETURN
QUAD108          END
```

```
      SUBROUTINE STRSTR (R,Z,UR,UZ,CODE,BO,E,RO,PR,SIGI,TH,FR,IX,MTAG,
     *                   IBC,JBC,NUMR,NUMR1,NNP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TOTAL/ AA(30000), IA(7000)
      COMMON/ONE/NUMNP,NUMEL,NUMMAT,NUMPC,NPMAX,NELMAX,N,NNN,KKK,KLK,
     .    ISHIFT,MBAND,NBAND,NUMBLK,MTYPE,NCASE,MCASE,NMR,NL,NPP,NEQ,NPC
     .    ,NP,NSTEP,NCODE
      COMMON/TWO/RR(5),ZZ(5),S(10,10),P(10),ST(3,10),C(4,4),SIG(7),EE(4)
     .    ,TITLE(18),SR1,SR2,XC,YC,ACELR,ACELZ,BCELR,BCELZ,VOL
      COMMON/THREE/STOP,MTOT,NTOT,N13,N14,MB
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),CODE(NUMNP),
     .    BO(NEQ),E(5,NUMMAT),RO(NUMMAT),PR(NPC  ,2),SIGI(NUMEL,8),
     .    TH(NUMEL),FR(NUMEL),IX(NUMEL,5),MTAG(NUMEL),IBC(NPC),
     .    JBC(NPC  ),NUMR(NMR),NUMR1(NMR),NNP(NMR)
C
      MTYPE=IX(N,5)
      VOL=0.
      DO 50 KK=1,4
   50 EE(KK)=E(KK,MTYPE)
      IF((NCASE.GT.1).AND.(NNN.EQ.1)) GO TO 60
      IF(MTAG(N).GT.0) GO TO 70
   60 CC=EE(2)/(1.-EE(2))
      BB=EE(1)/(1.-EE(2)**2)
      COMM=BB/(1.-CC**2)
      C(1,1)=COMM
      C(1,2)=COMM*CC
      C(1,3)=0.
      C(1,4)=0.
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=0.
      C(2,4)=0.
      C(3,1)=0.
      C(3,2)=0.
      C(3,3)=.5*COMM*(1.-CC)
      C(3,4)= 0.
      C(4,1)= C(1,2)
      C(4,2)= C(1,2)
      C(4,3)= 0.
      C(4,4)= 0.
      CC= 2.*DSIN(EE(4)/57.296)
      BB= 1.732*(3.-DSIN(EE(4)/57.296))
      PP=6.*DCOS(EE(4)/57.296)
      EE(4)=CC/BB
      EE(3)= EE(3)*PP/BB
      GO TO 500
   70 CC= 2.*DSIN(EE(4)/57.296)
      BB= 1.732*(3.-DSIN(EE(4)/57.296))
      PP=6.*DCOS(EE(4)/57.296)
      EE(4)=CC/BB
```

```
      FF(3)= FF(3)*PP/BB
      CC=2.*(1.+FF(2))/(3.-6.*FF(2))
      DD=SIGI(N,1)-SIGI(N,2)
      FF=SIGI(N,1)-SIGI(N,4)
      GG=SIGI(N,2)-SIGI(N,4)
      BJ2= (DD**2+FF**2+GG**2)/6. +SIGI(N,3)**2
      BJ2=DSQRT(BJ2)
      BJ1=SIGI(N,1)+SIGI(N,2)+SIGI(N,4)
      DD=BJ1/BJ2
      BB=1.+9.*(FF(4)**2)*CC
      CC=3.*FF(4)*CC-DD/3.
      DD=FF(4)-DD/6.
      H1=.5*CC/(BB*BJ2)
      H2=DD*CC/BB-FF(2)*FF(3)/(BB*BJ2*(1.-2.*FF(2)))
      H3=.5/(BB*BJ2*BJ2)
      BB=FF(1)/(1.+FF(2))
      C(1,1)=BB*(1.-H2-2.*H1*SIGI(N,1)-H3*(SIGI(N,1)**2))
      C(1,2)=-BB*(H2+H1*(SIGI(N,1)+SIGI(N,2))+H3*SIGI(N,1)*SIGI(N,2))
      C(1,3)=-BB*(H1*SIGI(N,3)+H3*SIGI(N,1)*SIGI(N,3))
      C(1,4)=0.
      C(2,1)=C(1,2)
      C(2,2)=BB*(1.-H2-2.*H1*SIGI(N,2)-H3*(SIGI(N,2)**2))
      C(2,3)=-BB*(H1*SIGI(N,3)+H3*SIGI(N,2)*SIGI(N,3))
      C(2,4)=0.
      C(3,1)=C(1,3)
      C(3,2)=C(2,3)
      C(3,3)=BB*(.5-H3*(SIGI(N,3)**2))
      C(3,4)=0.
      C(4,1)=-BB*(H2+H1*SIGI(N,1)+H1*SIGI(N,4)+H3*SIGI(N,1)*SIGI(N,4))
      C(4,2)=-BB*(H2+H1*SIGI(N,2)+H1*SIGI(N,4)+H3*SIGI(N,2)*SIGI(N,4))
      C(4,3)=-BB*(H1*SIGI(N,3)+H3*SIGI(N,3)*SIGI(N,4))
      C(4,4)=0.
  500 RETURN
      END
```

```
      SUBROUTINE MODIFY(A,B,NL ,MBAND,NBAND,N,U)
      IMPLICIT REAL*8(A-H ,O-Z)
      DIMENSION B(NL),A(NL,NBAND)
C
      DO 250 M=2,MBAND
      K=N-M+1
      IF(K.LE.0) GO TO 235
      B(K)=B(K)-A(K,M)*U
      A(K,M)=0.0
  235 K=N+M-1
      IF(NL .LT.K) GO TO 250
      B(K)=B(K)-A(N,M)*U
      A(N,M)=0.0
  250 CONTINUE
      A(N,1)=1.0
      B(N)=U
      RETURN
      END
```

```
      SUBROUTINE BANSOL (ND,NBAND,MM,NUMBLK,B,A)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION B(ND),A(ND,NBAND)
C
      NN=ND/2
      NL=NN+1
      NH=NN+NN
      REWIND 1
      REWIND 2
      NB=0
      GO TO 150
  100 NB=NB+1
      DO 125 N=1,NN
      NM=NN+N
      B(N)=B(NM)
      B(NM)=0.0
      DO 125 M=1,MM
      A(N,M)=A(NM,M)
  125 A(NM,M)=0.0
      IF(NUMBLK.EQ.NB) GO TO 200
  150 READ (2) (B(N),(A(N,M),M=1,MM),N=NL,NH)
      IF(NB.EQ.0) GO TO 100
  200 DO 300 N=1,NN
      IF(A(N,1).EQ.0.) GO TO 300
      B(N)=B(N)/A(N,1)
      DO 275 L=2,MM
      IF (A(N,L).EQ.0.) GO TO 275
      C=A(N,L)/A(N,1)
      I=N+L-1
      J=0
      DO 250 K=L,MM
      J=J+1
  250 A(I,J)=A(I,J)-C*A(N,K)
      B(I)=B(I)-A(N,L)*B(N)
      A(N,L)=C
  275 CONTINUE
  300 CONTINUE
      IF(NUMBLK.EQ.NB) GO TO 400
      WRITE (1) (B(N),(A(N,M),M=2,MM),N=1,NN)
      GO TO 100
  400 DO 450 M=1,NN
      N=NN+1-M
      DO 425 K=2,MM
      L=N+K-1
  425 B(N)=B(N)-A(N,K)*B(L)
      NM=N+NN
      B(NM)=B(N)
  450 A(NM,NB)=B(N)
      NB=NB-1
      IF(NB.EQ.0) GO TO 500
```

```
      BACKSPACE 1
      READ (1) (B(N),(A(N,M),M=2,MM),N=1,NN)
      BACKSPACE 1
      GO TO 400
  500 K=0
      DO 600 NB=1,NUMBLK
      DO 600 N=1,NN
      K=K+1
      NM=N+NN
  600 B(K)=A(NM,NB)
      RETURN
      END
```

```
      SUBROUTINE STRESS (R,Z,UR,UZ,CODE,BO,E,RO,PR,SIGI,TH,FR,B,EP,
     *                   RATIO,NEW,IX,MTAG,IBC,JBC,NUMR,NUMR1,NNP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TOTAL/ AA(30000), IA(7000)
      COMMON/ONE/NUMNP,NUMEL,NUMMAT,NUMPC,NPMAX,NELMAX,N,NNN,KKK,KLK,
     .    ISHIFT,MBAND,NBAND,NUMBLK,MTYPE,NCASE,MCASE,NMR,NL,NPP,NEQ,NPC
     .    ,NP,NSTEP,NCODE
      COMMON/TWO/RR(5),ZZ(5),S(10,10),P(10),ST(3,10),C(4,4),SIG(7),EE(4)
     .    ,TITLE(18),SR1,SR2,XC,YC,ACELR,ACELZ,BCELR,BCELZ,VOL
      COMMON/THREE/STOP,MTOT,NTOT,N13,N14,M8
      DIMENSION R(NUMNP),Z(NUMNP),UR(NUMNP),UZ(NUMNP),CODE(NUMNP),
     .    BO(NEQ),E(5,NUMMAT),RO(NUMMAT),PR(NPC   ,2),SIGI(NUMEL,8),
     .    TH(NUMEL),FR(NUMEL),IX(NUMEL,5),MTAG(NUMEL),IBC(NPC),
     .    JBC(NPC   ),NUMR(NMR),NUMR1(NMR),NNP(NMR)
      DIMENSION EP(NUMEL,4),RATIO(NUMEL),NEW(100),TT(4),TE(4)
      DIMENSION B(NEQ)
C
      TOLL=0.
      SR=1.
      SRATIO=1.0
      NMY=0
      MPRINT=0
      KK=0
      NOPT=0
      TOLLA=10.
      KJK=0
      KLK=0
      IF((NCASE.LE.1).OR.(NNN.GT.1)) GO TO 60
      DO 50 N=1,NELMAX
      IF (MTAG(N).GT.0) KJK=KJK+1
   50 CONTINUE
   60 CONTINUE
      DO 300 N=1,NELMAX
      RATIO(N)=1.
      IX(N,5)=IABS(IX(N,5))
      MTYPE=IX(N,5)
      IF((RO(MTYPE).EQ.0.).OR.(IX(N,3).EQ.IX(N,2))) GO TO 300
      CALL QUAD (R,Z,UR,UZ,CODE,BO,E,RO,PR,SIGI,TH,FR,IX,MTAG,
     *           IBC,JBC,NUMR,NUMR1,NNP)
      MM=4
      IF(IX(N,3).EQ.IX(N,4)) MM=3
      DO 175 I=1,4
      EP(N,I) = 0.
      DO 175 J=1,MM
      II=2*J
      JJ=2*IX(N,J)
  175 EP(N,I)=EP(N,I)+ST(I,II)*B(JJ) + ST(I,II-1)*B(JJ-1)
      DO 190 I=1,4
      SIG(I)=0.0
      DO 190 J=1,3
```

```
  190 SIG(I(=SIG(I(+C(I,J)*FP(N,J)
      DO 195 I=1,4
      II=I+4
  195 SIGI(N,II)=SIGI(N,I(+SIG(I(
      DD=SIGI(N,5(-SIGI(N,6(
      FF=SIGI(N,6)-SIGI(N,8)
      GG=SIGI(N,5(-SIGI(N,8)
      AJ2= (DD**2+GG**2+FF**2(/6.0 +SIGI(N,7)**2
      AJ2=DSQRT(AJ2(
      AJ1=SIGI(N,5(+SIGI(N,6)+SIGI(N,8)
      FAIL=AJ2+FF(4(*AJ1
      IF(MTAG(N(.EQ.0) GO TO 200
      IF (FF(3(.EQ.0.( GO TO 300
      IF ((NCASE.GT.1).AND.(NNN.EQ.1(( GO TO 198
      DD=DABS(FAIL-FF(3((
      CHECK =0.05*FF(3(
      IF(DD.LE.CHECK) GO TO 300
      KKK=1
      CR=CHECK/DD
      IF(CR.GE.SR) GO TO 300
      SR=CR
      NOPT=1
      JJJ=N
      GO TO 300
  198 IF(FAIL.GT.FF(3(( GO TO 300
      MTAG(N(=0
      KJK=KJK-1
      GO TO 300
  200 CONTINUE
      IF(FAIL.LT.FF(3)( GO TO 300
      KK=KK+1
      DD=SIGI(N,1)-SIGI(N,2)
      FF=SIGI(N,1)-SIGI(N,4(
      GG=SIGI(N,2(-SIGI(N,4)
      AJ2= (DD**2+GG**2+FF**2)/6.0 +SIGI(N,3)**2
      AJ1=SIGI(N,1(+SIGI(N,2)+SIGI(N,4)
      DD=SIGI(N,5)-SIGI(N,6)
      FF=SIGI(N,6)-SIGI(N,8(
      GG=SIGI(N,5(-SIGI(N,8)
      BJ2= (DD**2+GG**2+FF**2)/6.0 +SIGI(N,7)**2
      BJ1=SIGI(N,5)+SIGI(N,6)+SIGI(N,8)
      CCC=(SIGI(N,1)-SIGI(N,2))*(SIGI(N,5)-SIGI(N,6))+(SIGI(N,2(-SIGI(N,
     .4))*(SIGI(N,6(-SIGI(N,8()+(SIGI(N,4)-SIGI(N,1()*(SIGI(N,8)-SIGI(N,
     .5((
      CCC=CCC/6.0+ SIGI(N,3)*SIGI(N,7(
      AB=AJ2-(FF(4(*AJ1(**2
      BB=BJ2-(FF(4(*BJ1(**2
      CC=CCC-(FF(4(*FF(4)*AJ1*BJ1(
      DD=FF(4(*FF(3(*AJ1
      EE=FF(4(*FF(3(*BJ1
```

```
      GG=FF(3)*EE(3)
      AAA=AB+BB-2.*CC
      BBB=AB-CC+DD-FF
      CCC=2.*DD-GG+AB
      GGG=BBB*BBB-AAA*CCC
      IF(GGG.LT.0.)   WRITE(6,2008) N
      IF(GGG.LT.0.)  GGG=DABS(GGG)
      GGG=DSQRT(GGG)
      IF(AAA.NE.0.) GO TO 220
      RATIO(N)=.5*CCC/BBB
      GO TO 300
  220 AB=BBB/AAA
      BB=DABS(GGG/AAA)
      RATIO(N)=AB-BB
      IF(RATIO(N).LT.0.) RATIO(N)=AB+BB
      IF(RATIO(N).GE.1.) RATIO(N)=.99999
      IF(RATIO(N).LT.0.) RATIO(N)=0.
      IF (KK.NE.1) GO TO 240
      WRITE (6,3000)
  240 WRITE (6,3005) N,RATIO(N)
  300 CONTINUE
      IF(KK.EQ.0) SR=1.0
      IF(KK.EQ.0) GO TO 410
      DO 350 N=1,NELMAX
      IF(MTAG(N).GT.0) GO TO 350
      MTYPE=IX(N,5)
      IF (RO(MTYPE).EQ.0.) GO TO 350
      DD=RATIO(N)
      IF(DD.GE.SR) GO TO 350
      SR=DD
      NOPT=2
      NMY=N
      KKK=1
  350 CONTINUE
      WRITE (6,3010) SR,SR2
      IF(NNN.EQ.1) GO TO 352
      DD= SR*(1.0-SR2)
      IF(DD.LT.0.03)  NOPT=3
      IF(DD.LT.0.03) DD=0.03
      SR= DD/(1.0-SR2)
      DD= SR*(1.0-SR2)
      IF(DD.GT.0.10)  NOPT=4
      IF(DD.GT.0.10)  DD=0.10
      SR=DD/(1.-SR2)
      IF(SR.GT.1.0) SR=1.00
  352 CONTINUE
      SRATIO=SR
      IF (KJK.GT.0) SR=0.
  410 CONTINUE
      DO 420 N=1,NPMAX
```

```
      II=2*N-1
      B(II)=B(II)*SR
  420 B(II+1)=B(II+1)*SR
      WRITE (6,4010) NMY,SR,SR2
      NOO=0
      ERMAX=0.
      DO 600 N=1,NELMAX
      MTYPE=IX(N,5)
      IF (RO(MTYPE).EQ.0.) GO TO 515
      IF(IX(N,3).NE.IX(N,2)) GO TO 430
C
C        CALCULATE STRESSES IN ONE DIMENSIONAL ELEMENTS
C
      I=IX(N,1)
      J=IX(N,2)
      XC=(R(I)+R(J))/2.0
      YC=(Z(I)+Z(J))/2.0
      DX=R(J)-R(I)
      DY=Z(J)-Z(I)
      XL=DSQRT(DX**2+DY**2)
      DU=B(2*J-1)-B(2*I-1)
      DV=B(2*J)-B(2*I)
      DL=DV*DY/XL+DU*DX/XL
      DO 422 I=1,7
  422 SIG(I)=0.
      DX=E(1,MTYPE)*DL/XL
      SIG(1)=DX*SR
      IF(E(3,MTYPE).EQ.1.) GO TO 425
      SIGI(N,5)=-DX*(1-SR)
      SIGI(N,1)=SIG(1)+SIGI(N,I)
      GO TO 515
  425 SIG(5)=-DX*(1.-SR)
      SIG(1)=SIGI(N,1)+DX*SR
      DY=E(4,MTYPE)/(SIGI(N,1)+DX)-1.
      IF(DABS(DY).LE..05) GO TO 428
      II=1
      IF(NNN.GT.1) II=5
      DZ=SIGI(N,1)+DX
      DX=SIGI(N,II)+DX
      DY=(E(4,MTYPE)-DZ)*SIGI(N,II)/DX
      SIGI(N,1)=SIG(1)+DY
      SIGI(N,5)=SIG(5)+DY
      KLK=1
      SIG(5)=SIGI(N,5)
      SIG(4)=SIGI(N,1)
      GO TO 515
  428 SIGI(N,1)=SIG(1)
      SIGI(N,5)=SIG(5)
      GO TO 515
  430 I=IX(N,1)
```

```
      J=IX(N,2)
      K=IX(N,3)
      L=IX(N,4)
      MTYPE=IX(N,5)
      IF(K.EQ.L) GO TO 440
      XC=(R(I)+R(J)+R(K)+R(L))/4.
      YC=(Z(I)+Z(J)+Z(K)+Z(L))/4.
      GO TO 445
  440 XC=(R(I)+R(J)+R(K))/3.
      YC=(Z(I)+Z(J)+Z(K))/3.
  445 CONTINUE
      XN=0.5
      IF(NNN.EQ.1) XN=1.0
      IF(MTAG(N).LE.0)  XN=1.0
      DO 450 I=1,4
      II=I+4
      TT(I) = SIGI(N,I)
      TF(I)=SIGI(N,II)
      SIG(I)=SIGI(N,II)-SIGI(N,I)
      SIGI(N,II)=SIG(I)*(1.-SR)
      SIGI(N,II)=-SIGI(N,II)
      SIG(I)= XN*SIG(I)*SR+SIGI(N,I)
  450 SIGI(N,I)=SIG(I)
      IF((NMY.EQ.0).OR.(KJK.GT.0))  GO TO 485
      IF (MTAG(N).LE.0) GO TO 481
      ISTOP=0
      DO 470 JJ=1,5
      CALL STRSTR (R,Z,UR,UZ,CODE,BO,E,RO,PR,SIGI,TH,FR,IX,MTAG,
     *              IBC,JBC,NUMR,NUMR1,NNP)
      DO 455 I=1,4
      II=I+4
      SIGI(N,II)=TT(I)
      DO 455 J=1,3
  455 SIGI(N,II)=SIGI(N,II)+C(I,J)*EP(N,J)*SR
      DD=(SIGI(N,5)-SIGI(N,6))**2+(SIGI(N,6)- SIGI(N,8))**2+(SIGI(N,5)-
     .SIGI(N,8))**2
      AJ2=DD/6.0+ SIGI(N,7)**2
      AJ2=DSQRT(AJ2)
      AJ1=SIGI(N,5)+SIGI(N,6)+SIGI(N,8)
      FAIL=AJ2+FE(4)*AJ1
      IF(JJ.EQ.1) GO TO 460
       D=DABS(FAIL-FAIL1)
      TOL=0.005*FAIL
      IF(DD.LE.TOL)  ISTOP=1
  460 FAIL1 =FAIL
      DO 465 I=1,4
      II=I+4
      SIG(I)=SIGI(N,II)-TT(I)
  465 SIGI(N,I)=TT(I)+SIG(I)*XN
      IF(N.NE.1) GO TO 451
```

```
      WRITE(6,1000) (SIGI(N,I),I=1,8)
 1000 FORMAT(10H SIGI(N,I) / (8F12.4))
  451 CONTINUE
      IF(ISTOP.GT.0) GO TO 475
  470 CONTINUE
  475 DO 480 I=1,4
      II=I+4
      SIGI(N,I)=SIGI(N,II)
      SIGI(N,II)=-(TF(I)-SIGI(N,I))
      SIG(I)=SIGI(N,I)
  480 CONTINUE
  481 TR=SRATIO
      TOL=0.05*TR
      IF(TR.GE.0.95) GO TO 485
      IF(TOL.LT.0.005) TOL=0.005
      IF(MTAG(N).GT.0) GO TO 485
      IF(RATIO(N).LE.TR) GO TO 482
      DD=RATIO(N)-TR
      IF(DD.GT.TOL) GO TO 485
  482 NOD=NOD+1
      NEW(NOD)=N
      MTAG(N)=1
  485 CONTINUE
      DO 490 I=1,4
  490 FF(I)=F(I,MTYPE)
      CC= 2.*DSIN(FF(4)/57.296)
      BB= 1.732*(3.-DSIN(FF(4)/57.296))
      PP=6.*DCOS(FF(4)/57.296)
      FF(4)=CC/BB
      FF(3)= FF(3)*PP/BB
      SIG(7)=SIGI(N,4)
      CC=(SIG(1)+SIG(2))/2.
      BB=(SIG(1)-SIG(2))/2.
      CR=DSQRT(BB**2+SIG(3)**2)
      SIG(4)=CC+CR
      SIG(5)=CC-CR
      SIG(6)=0.0
      IF ((BB.EQ.0.0).AND.(SIG(3).EQ.0.0)) GO TO 510
      SIG(6)=28.648*DATAN2(SIG(3),BB)
  510 CONTINUE
      DD=(SIG(1)-SIG(2))**2+(SIG(2)-SIG(7))**2+(SIG(7)-SIG(1))**2
      AJ2=DD/6. +SIG(3)**2
      AJ2=DSQRT(AJ2)
      AJ1=SIG(1)+SIG(2)+SIG(7)
      FAIL=FF(3)-FF(4)*AJ1
      FR(N)=AJ2/FAIL
      IF (MTAG(N).GT.0) GO TO 515
      IF (KJK.GT.0) GO TO 515
      IF (FR(N).GE.0.99) MTAG(N)=1
      IF(MTAG(N).EQ.0) GO TO 515
```

```
      NOO=NOO+1
      NEW(NOO)=N
  515 IF(MPRINT.GT.0) GO TO 550
      WRITE(6,2000) NNN
      MPRINT=50
  550 MPRINT=MPRINT-1
      IF((IX(N,2)-IX(N,3)).EQ.0)  FR(N)=0.0
      IF(TH(N).NE.1.) SIG(7)=0.0
      IF (RO(MTYPE).NE.0.) GO TO 555
      WRITE (6,2010) N
      GO TO 600
C
C        PRINT OUT STRESSES
C
  555 IF(MTAG(N).LT.1) GO TO 556
      WRITE(6,2011) N,XC,YC,(SIG(I),I=1,7),MTAG(N),FR(N),MTYPE
      GO TO 557
  556 WRITE(6,2001) N,XC,YC,(SIG(I),I=1,7),MTAG(N),FR(N),MTYPE
  557 IF(IX(N,2).EQ.IX(N,3)) GO TO 600
      IF((FR(N).GE.1.) OR.(FR(N).LT.FRMAX)) GO TO 570
      FRMAX=FR(N)
      NMAX=N
  570 CONTINUE
      DD=SIGI(N,5)-SIGI(N,6)
      FF=SIGI(N,6)-SIGI(N,8)
      GG=SIGI(N,5)-SIGI(N,8)
      BJ2= (DD**2+GG**2+FF**2)/6.0 +SIGI(N,7)**2
      BJ2=DSQRT(BJ2)
      TOLL=TOLL+BJ2
  600 CONTINUE
      SR2= (SR1*SR) + SR2
      SR1= (1.0-SR) * SR1
      WRITE(6,2002)TOLL,SR,NMY,KK,SR2
      WRITE(6,5000) FRMAX,NMAX
      IF(NOO.EQ.0) GO TO 620
      WRITE(6,2020) (NEW(I),I=1,NOO)
  620 KKK=1
      IF(TOLL.LE.TOLLA) KKK=0
      NKT=NELMAX/2
      JCK=0
      DO 705 N=1,NELMAX
      IF(MTAG(N).GT.0) JCK=JCK+1
  705 CONTINUE
      IF(JCK.LT.NKT) GO TO 710
      WRITE(6,2007)
      CALL EXIT
  710 CONTINUE
      IF(NNN.EQ.1)  GO TO 800
      GO TO (750,760,770,780),NOPT
      GO TO 800
```

```
  750 WRITE(6,2003) JJJ
      GO TO 800
  760 WRITE(6,2004)
      GO TO 800
  770 WRITE(6,2005)
      GO TO 800
  780 WRITE(6,2006)
  800 RETURN
 2000 FORMAT(1H1/
     .36H STRESSES AFTER APPROXIMATION NUMBER I4////
     .7H EL.NO. 7X 1HX 7X 1HY 4X 8HX-STRESS 4X 8HY-STRESS 3X 9HXY-STRESS
     . 2X 10HMAX-STRESS 2X 10HMIN-STRESS 7H  ANGLE 4X 8HZ-STRESS 3X 7HPL
     .ASTIC 3X 4HFAIL 3X 5HMTYPE     )
 2001 FORMAT (I7,2F8.2,1P5E12.4,0P1F7.2,1PE12.4, I6,0P1F11.3 ,I6 )
 2002 FORMAT(39H0THE UNBALANCED LOAD AT THIS STAGE IS  E14.5//
     .47H THE RATIO FOR CORRECTION OF STORED STRESSES IS F10.4//
     .31H THE NEXT ELEMENT YIELDING IS     I4/
     .91H AND THE TOTAL NUMBER OF ELEMENTS THAT CAN YIELD WITH THE LINEA
     .R ADDITION OF TOTAL LOAD IS         I4/
     .50H LOAD UP TO THIS STAGE AS A FRACTION OF TOTAL IS        F20.5   )
 2003 FORMAT(110H0STRESS RATIO GOVERNED BY STRESS STATE BEING MORE THAN
     .5 PERCENT OUTSIDE YIELD SURFACE FOR ELEMENT NUMBER =           I5)
 2004 FORMAT( 53H0 STRESS RATIO GOVERNED BY NEXT ELEMENT YIELDING        )
 2005 FORMAT(100H0STRESS RATIO GOVERNED BY THE MINIMUM VALUE  OF SR FOR
     . ANY STEP                                                          )
 2006 FORMAT(110H0 STRESS RATIO GOVERNED BY THE MAXIMUM VALUE OF STRESS
     . RATIO FOR ANY STEP                                                )
 2007 FORMAT(64H0 JOB TERMINATED AS HALF OF TOTAL ELEMENTS YIELD AT THIS
     . STEP             )
 2008 FORMAT( 110H ARGUMENT NEGATIVE IN THE EQUATION FOR CALCULATING
     . THE VALUE OF STRESS RATIO FOR ELEMENT NUMBER  =                I5)
 2009 FORMAT(45H0 STRESS RATIO SR FOR THIS CYCLE=                  F10.5)
 2010 FORMAT(I7,50H THIS ELEMENT HAS BEEN REMOVED FROM THE ANALYSIS  )
 2011 FORMAT (I7,2F8.2,1P5E12.4,0P1F7.2,1PE12.4, I6,0P1F11.3 ,I6,3X,1H*)
 2020 FORMAT(48H0 THE FOLLOWING NEW ELEMENTS YIELD IN THIS STEP      //
     .20I5)
 3000 FORMAT (35H0 THE FOLLOWING ELEMENTS CAN YEILD                    //
     .34H0 ELEMENT NO.             RATIO(N)                              )
 3005 FORMAT ((I10,F20.5)
 3010 FORMAT( 17H0 SR            SR2                                    /
     . 2F10.5)
 4010 FORMAT(6H0 NMY=  I5,3HSR=  E20.5, 4HSR2=  E20.5     )
 5000 FORMAT(17H0MAXIMUM FAIL IS  F6.3,17H FOR ELEMENT NO.   I5)
      END
```

CHAPTER III: GRIFTH - A Computer Program for Two-Dimensional Analysis of Progressive Failure of Rock Following Griffith and Modified Griffith Theory

3.1. Purpose and Capability

This computer program is applieable to plane stress or plane strain analysis of stresses, deformations and progressive fracture in elastic brittle roek following Griffith and modified Griffith theory. Arbitrary initial stresses, arbitrary sequence of construction or exeavation, arbitrary history of load application ean be simulated. One dimensional elements are ineluded. The program is applieable to study of fraeture initiation and propagation in arbitrary elastic brittle strueture systems composed of several different materials. Non-monotonic loading is considered.

Theoretical development ineorporated in the program is documented in Part I-Teehnieal Report of this report.

3.2. Program Organization

The eomputer program is in Fortran language. Files 1 and 2 are used to store system equations and element properties respectively. Tapes 5 and 6 are the input/output files. The program capaeity can be altered by ehanging the dimensions of arrays AA and IA. These eorrespond to the total loeations required for real and integer arrays respeetively. NTOT, MTOT at lines MAIN 29 , MAIN 30 are set equal to the dimensions of AA and IA.

The program consists of the following units:

a. MAIN

In this unit, the eontrol information including maximum number of elements, nodal points, different materials, boundary pressure cards, the number of steps of excavation or construction, the maximum number of elements removed or added to the system at any stage is read in. This information is used to organize the dimensions of various arrays. This done, further processing of data is done in subroutine INPT.

b. Subroutine INPT

This subroutine is called by the unit MAIN. The first step is to read in material property data for all different materials in the system. Nodal point coordinates, loads and code descriptors for boundary conditions are read in or generated. Element geometry, initial stresses, initial crack openings, if any, are read in or generated. Maximum band width for the system is calculated and dimensions of blocks for generation and storage of system stiffness defined. After defining these controls, the incremental structure is analyzed in steps. For each step the number of nodal points, the number of elements, the number of elements and nodal points removed or added, if any, the number of boundary pressure cards and the material type of the elements added or changes in material properties are read in. After the information is assembled the solution process is transferred to subroutine SOLVE.

c. Subroutine SOLVE

This subroutine called by INPT is concerned with obtaining stresses, deformations and sequence of progressive fracture of elements in a given step of loading/construction/excavation. To trace progressive fracture, the solution process traces a sequence of elements reaching fracture along with the effects of stress redistribution associated with fracture leading to secondary fractures. The process consists of applying the total load and then scaling it according to the minimum ratio of load increment needed to ensure one element reaching fracture. Once an element fractures, the associated stress redistribution will result in secondary fractures at the same total load. This is referred to as system stability iteration in the program.

The SOLVE subroutine calls ONED and QUAD to obtain element stiffness for one or two-dimensional elements respectively. This information is stored on File 2 and is updated in case of fracture or failure. Solution to the stiffness equations is obtained in subroutine BANSOL. Subroutine STRESS defines the stresses corresponding to a load application. This is referred to as the initial state for any load increment. As several elements may crack during a load increment, it is necessary to scale it to pin-point the sequential fracture phenomenon. This is accomplished in subroutine SCALE.

d. Subroutine SCALE

This subroutine calls subroutine GRIFTH to check each element for fracture or closure of cracks. The stress ratio for each element is calculated, if necessary, using interpolation (subroutine INTER). Iterations to define a value of orientation of fracture are referred to as beta-stability iterations. After choosing the minimum stress ratio applicable and the element that next fractures, system stability iteration is accomplished to define all secondary fractures associated with the primary fracture. This process is illustrated in the flow chart as Figure III-1.

3.3. Input Data

a. Job Title (18A4). This card will give the descriptive identification for the job.

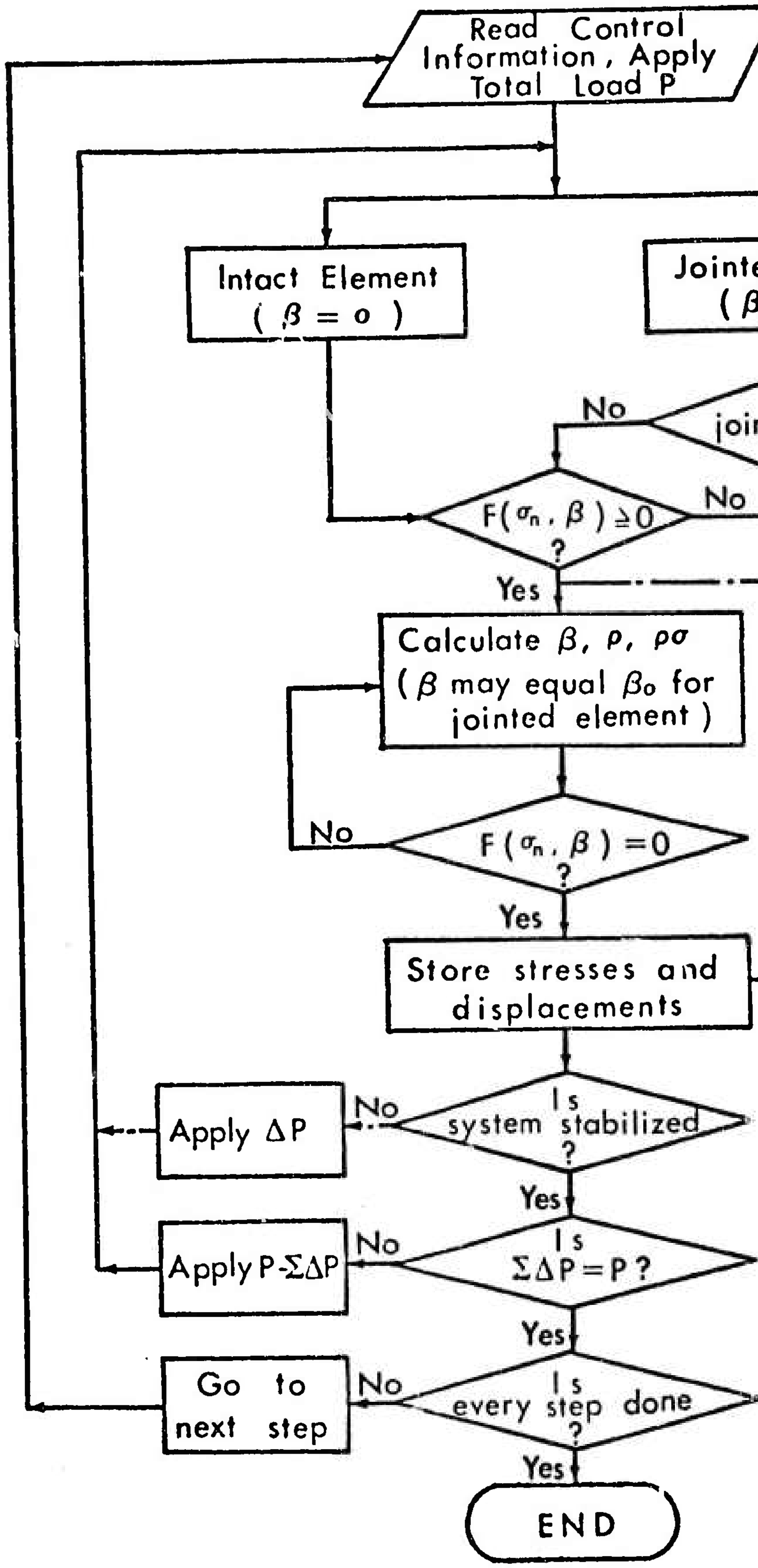


Figure 3.1. Flow Chart

Analysis of Progressive Fracture of Rock

b. Control Information (4I5, 3F10.2, 3I5)

Information	Columns
Maximum number of nodal points	1-5
Maximum number of elements	6-10
Number of different materials	11-15
Maximum number of pressure cards	16-20
Body force in X-direction	21-30
Body force in Y-direction	31-40
Reference (stress-free) temperature	41-50
Number of initially open cracks	51-55
Code to designate plane stress or plane strain NPLANE = 1 for plane stress = 2 for plane strain	56-60
Total number of excavation & construction steps	61-65

c. Material Property Cards. One set of cards must be provided for each material.

In each set:

i. First card (2I5, F10.3, 2I5) will give the following information:

Information	Columns
Material identification number	1-5
Number of temperature cards (8 maximum)	6-10
Mass density of the material	11-20
Material code to designate materials which will follow the fracture criteria	21-25

code = 1 for materials which will follow the fracture criteria
= 0 for materials which will not fracture

Material code to identify the initial anisotropy due to initial cracks 26-30
code = 1 for initial anisotropy
= 0 for isotropy

ii. If columns 21-25 in card i is not zero, the following information must be provided (3F10.3)

Information	Columns
Tensile strength	1-10
Internal frictional coefficient	11-20
Tolerance for crack closure	21-30

iii. Subsequent cards, one for each temperature, the number being defined in columns 6-10 of the first card, will carry the following information (F10.0, E10.0, 2F10.0):

Information	Columns
Temperature	1-10
Elastic modulus	11-20
Poisson's ratio	21-30
Coefficient of thermal expansion	31-40

iv. If columns 26-30 in card i is not zero, the following information must be providied (4F10.4):

Information	Columns
Modulus ratio	1-10
Angle of fault	11-20
Frictional coefficient	21-30
Shear strength	31-40

d. Nodal Point Cards (I5, F5.0, 5F10.0). One card for each nodal point with the following information:

Information	Columns
Nodal point number	1-5
Type of nodal point	6-10
X-ordinate	11-20
Y-ordinate	21-30
XR	31-40
XZ	41-50
Temperature	51-60

If the number in columns 6-10 is:

0 = XR is the specified X-load and XZ is the specified Y-load

1 = XR is the specified X-displacement and XZ is the specified Y-load

2 = XR is the specified X-load and XZ is the specified Y-displacement

3 = XR is the specified X-displacement and XZ is the specified Y-displacement.

All loads are considered to be total forces acting on an element of unit thickness. Nodal point cards must be in numerical sequence. If cards are omitted, the omitted nodal points are generated at equal intervals along a straight line between the defined nodal points. The necessary temperatures are determined by linear interpolation. The type of the nodal point, as well as XR, XZ, are set equal to zero.

e. Element Material Cards (16I5). These cards shall carry the material type of all the elements. Each card shall have material types for 16 elements in sequence. The material type for each element must be read in as no interpolation has been provided for.

f. Elements Cards (5I5, 5X, 3F10.0). One card for each element will provide the following data:

Information	Columns
Number of element	1-5
Nodal point I	6-10
Nodal point J	11-15
Nodal point K	16-20
Nodal point L	21-25
Initial stresses:	
Component in X-direction	31-40
Component in Y-direction	41-50
Shearing stress on X-Y planes	51-60

Nodal points I, J, K, L are corners of each individual element in a counter-clockwise order for a right handed system of coordinates. For triangular elements set nodal point L same as nodal point K. The element cards must be in the numerical sequence. Any cards that are omitted will be automatically generated in the program by incrementing each of the I, J, K and L nodal points by one. The material type will be taken same as for the last element defined.

g. Initial Cracks Cards. If columns 41-45 in card b is not zero, the following information must be provided for each initial crack (2I5, F10.0).

Information	Columns
Element number	1-5

Tag number	6-10
Crack Angle	11-20

The tag number is used to designate the crack mode

Tag = 1 for single crack
= 2 for double crack

The crack angle is defined as the angle between X-axis and the normal to the crack plane, positive counter-clockwise.

h. Incremental Step Information. One set of cards must be provided for each step of construction or dismantling. Construction and dismantling may not be mixed in one step.

i. First card (18A4). This gives the descriptive title for the step for which information follows.

ii. Second card (7I5). Following information is given for the step which is described in Title.

Information	Columns
Number of nodal points in this step	1-5
Number of elements in this step	6-10
Number of elements removed or added with reference to previous step	11-15
Number of pressure boundary cards (total pressure for this step)	16-20
New material type for the elements added or eliminated	21-25
Code to designate addition to structure of dismantling	26-30

Code = 0 for dismantling
= 1 for construction

Number of nodal points included in analysis but not part of structure at this step	31-35

iii. Addition or Dismantling of Structure. One or more cards shall indicate the numbers of elements removed or added in the step under consideration (16I5).

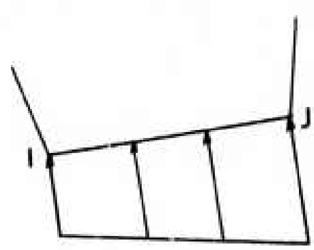
The total number of elements included here must be the same as indicated in columns 11-15 of card h-ii.

iv. Nodal Points Included in Analysis but not Taking Any Load (16I5). One or more cards shall indicate the numbers of the nodal points included in the analysis at this step but not forming part of the load carrying system.

The total number of nodal points listed here must be the same as indicated in columns 31-35 of card h-ii.

v. Pressure Boundary Cards (2I5, 2F10.0). If there are any boundary pressures for this step, then one card for each boundary element which is subjected to normal pressure will carry the following information:

Information	Columns
Nodal Point I	1-5
Nodal Point J	6-10
Total Normal Pressure at I	11-20
Total Normal Pressure at J	21-30



As shown in the sketch, the boundary element must be on the left as one progresses from I to J. Surface tensile force is input as a negative pressure.

j. Last Card (A6). The last card at the end of data deck is "stop" card. It carries the characters STOP in columns 1 through 4.

3.4. Output Information

The following information is developed and printed by the program:

a. Print out of problem data. This includes information in material properties, mesh layout, geometry and boundary conditions, loads and constraints, initial cracks, etc.

b. Initial stresses before any excavation or construction at the center of each element.

c. The incremental as well as cumulative nodal point displacements and the stresses after application of a load increment.

d. The stresses upon application of total load increment.

e. The stresses, crack orientation, type of crack, at the first element cracking under a load increment.

f. The stresses, crack orientation, type of crack for each beta-iteration to define the correct crack orientation.

g. The stresses, crack orientation, type of crack for each system stability iteration to define secondary fractures following the first fracture in a load increment .

h. The stress ratio as a proportion of total load at each stability iteration.

3.5. Fortran Listing

```
C       ****************************************************************
C       * PROGRAM IDENTIFICATION: PROGRAM PFA                          *
C       * PROGRAMMER: S.W. HUANG,I.S.RAI, THE OHIO STATE UNIVERSITY    *
C       * PURPOSE: PROGRESSIVE FRACTURE ANALYSIS                       *
C       * FAILURE CRITERIA: FRACTURE ACCORDING TO GRIFFITH AND MODIFIED *
C       *           GRIFFITH THEORIES                                  *
C       *  THE FORMULATION IS DOCUMENTED IN THE FINAL REPORT DATED MARCH *
C       * 31,1973, ON CONTRACT H0210017 BETWEEN THE OHIO STATE UNIVERSITY*
C       * AND THE UNITED STATES BUREAU OF MINES SUPPORTED BY THE ADVANCED*
C       * RESEARCH PROJECTS AGENCY. INSTRUCTIONS FOR USE OF THE PROGRAM *
C       * ARE CONTAINED IN PART II OF THE REPORT.                      *
C       ****************************************************************
C
        COMMON AA(16000),IA(4500)
        COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .    VOL,ACELR,ACELZ,Q,HED(18),STOP,SR,SR1,TOTAL,TOL,TOL1,XC,YC,
     .    TEMP,SIGN,SIGDI,SIGDJ,
     .    LLL,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMER,NCODE,JA,NEQ
     .    ,NBAND,NCRACK,NSTEP,N15,M7,NTOT
        COMMON/TWO/ C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(51,2215),
     .    LM(4),EE(3),EPS(3)
        COMMON/THREE/  E(8,4,8),EO(4,8),TENS(8),XNU(8),RO(8),EPST(8),
     .    MTC(8),NIC(8),MID(8)
        DEFINE FILE 1(100,1500,U,NBK),2(510,230,U,ID)
        CALL ERRSET(208,256,-1,1)
C
C       READ AND WRITE CONTROL INFORMATION FOR THE PROBLEM
C
        NTOT=16000
        MTOT=4500
        TOL1=0.001
        READ (5,1000) HED,NUMNP,NUMEL,NUMMAT,NUMPC,ACELR,ACELZ,Q,NCRACK,
     .     NPLANE,NSTEP
        IF(NPLANE.EQ.1) WRITE(6,2000)
        IF(NPLANE.EQ.2) WRITE(6,2005)
        NEQ=2*NUMNP
        WRITE (6,2010) HED,NUMNP,NUMEL,NUMMAT,NUMPC,ACELR,ACELZ,Q,NSTEP
        NPC=NUMPC
        IF(NPC.EQ.0) NPC=1
        N1=1
        N2=N1+NUMNP
        N3=N2+NUMNP
        N4=N3+NUMNP
        N5=N4+NEQ
        N6=N5+NEQ
        N7=N6+NEQ
        N8=N7+NEQ
        N9=N8+NUMNP
        N10=N9 +6*NUMEL
        N11=N10+6*NUMEL
```

```
      N12=N11+  NUMEL
      N13=N12+  NUMEL
      N14=N13+  NUMEL
      N15=N14+2*NPC
      M1=1
      M2=M1+5*NUMEL
      M3=M2+NUMEL
      M4=M3+NUMEL
      M5=M4+NUMEL
      M6=M5+NPC
      M7=M6+NPC
      JJ=M7-MTOT
      IF(JJ.LE.0) GO TO 100
      WRITE (6,3000)  JJ
      CALL EXIT
  100 CONTINUE
      CALL INPT  (AA(N1),AA(N2),AA(N3),AA(N4),AA(N5),AA(N6),AA(N7),
     .   AA(N8),AA(N9),AA(N10),AA(N11),AA(N12),AA(N13),AA(N14),IA(M1),
     .   IA(M2),IA(M3),IA(M4),IA(M5),IA(M6))
      STOP
 1000 FORMAT (18A4/4I5,3F10.2,3I5)
 2000 FORMAT(1H1,' PLANE STRESS ANALYSIS')
 2005 FORMAT(1H1,' PLANE STRAIN ANALYSIS')
 2010 FORMAT (1H0 18A4/
     1 30H0 NUMBER OF NODAL POINTS------ I3 /
     2 30H0 NUMBER OF ELEMENTS---------- I3 /
     3 30H0 NUMBER OF DIFF. MATERIALS--- I3 /
     4 30H0 NUMBER OF PRESSURE CARDS--   I3/
     5 30H0 X-ACCELERATION-------------- E12.4/
     6 30H0 Y-ACCELERATION-------------- E12.4/
     7 30H0 REFERENCE TEMPERATURE------- E12.4/
     8 30H0 NUMBER OF STEPS------------- I3 )
 3000 FORMAT (70H PROGRAM EXECUTION TERMINATED. REQUIRED CORE EXCEEDS MT
     .OT BY                                      I10)
      END
```

```
      SUBROUTINE INPT   (R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,IX,MTAG,NTAG,JNT,IBC,JBC)
C
      COMMON AA(16000),IA(4500)
      COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .   VOL,ACELR,ACELZ,Q,HED(18),STOP,SR,SRI,TOTAL,TOL,ICL(,XC,YC,
     .   TEMP,SIGN,SIGDI,SIGDJ,
     .   LLL,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMER,NCODE,JA,NEQ
     .   ,NLAND,NCRACK,NSTEP,N15,M7,NTOT
      COMMON/TWO/ C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),
     .   LM(4),EE(3),EPS(3)
      COMMON/THREE/  E(8,4,8),EO(4,8),TENS(8),XNU(8),RO(8),EPST(8),
     .   MTC(8),NTC(8),MID(8)
      DIMENSION TITLE(18),NNP(200),MAD(200),NUMR(200)
      DIMENSION R(NUMNP),Z(NUMNP),CODE(NUMNP),UU(NUMNP,2),CU(NEQ),
     .BI(NEQ    (NEQ),T(NUMNP),SIGI(NUMEL,6),EPSI(NUMEL,6),SIGNM(NUMEL),
     .BETA(NU    ),RATIO(NUMEL),PR(NPC,2),IX(NUMEL,5),MTAG(NUMEL),
     .NTAG(NUM   ),JNT(NUMEL),IBC(NPC),JBC(NPC)
      DO 50 M=1,NUMMAT
      READ (5,1010) MTYPE,NTC(MTYPE),RO(MTYPE),MTC(MTYPE),MID(MTYPE)
      WRITE(6,2010) MTYPE,NTC(MTYPE),RO(MTYPE),MTC(MTYPE),MID(MTYPE)
      IF(MTC(MTYPE).EQ.0) GO TO 45
      READ(5,1015)    TENS(MTYPE),XNU(MTYPE),EPST(MTYPE)
      WRITE(6,2015)   TENS(MTYPE),XNU(MTYPE),EPST(MTYPE)
   45 CONTINUE
      NUMTC=NTC(MTYPE)
      READ (5,1020) ((E(I,J,MTYPE),J=1,4),I=1,NUMTC)
      WRITE (6,2020) ((E(I,J,MTYPE),J=1,4),I=1,NUMTC)
      IF(MID(MTYPE).EQ.0) GO TO 50
      READ (5,1022) (EO(K,MTYPE),K=1,4)
      WRITE(6,2022) (EO(K,MTYPE),K=1,4)
      EO(2,MTYPE)=EO(2,MTYPE)/57.296
   50 CONTINUE
C
C     READ NODAL POINT DATA,GENERATE INTERMEDIATE POINTS AND WRITE
C
      WRITE (6,2025)
      L=0
   60 READ  (5,1025) N,CODE(N),R(N),Z(N),UU(N,1),UU(N,2),T(N)
      IF(N.EQ.1) GO TO 70
      ZX=N-L
      DR=(R(N)-R(L))/ZX
      DZ=(Z(N)-Z(L))/ZX
      DT=(T(N)-T(L))/ZX
   70 L=L+1
      IF(N-L) 100,90,80
   80 CODE(L)=0.0
      R(L)=R(L-1)+DR
      Z(L)=Z(L-1)+DZ
      T(L)=T(L-1)+DT
```

```
      UU(L,1)=0.
      UU(L,2)=0.
      GO TO 70
   90 IF(NUMNP-N) 100,110,60
  100 WRITE (6,2035) N
      CALL EXIT
  110 WRITE(6,2030) (N,CODE(N),R(N),Z(N),UU(N,1),UU(N,2),T(N),N=1,NUMNP)
C
C     READ ELEMENT DATA , GENERATE INTERMEDIATE ELEMENTS AND WRITE
C
      READ(5,1030) (IX(N,5),N=1,NUMEL)
      WRITE (6,2040)
      N=0
  130 READ(5,1035) M,(IX(M,I),I=1,4)
  140 N=N+1
      IF (M-N) 170,170,150
  150 IX(N,1)=IX(N-1,1)+1
      IX(N,2)=IX(N-1,2)+1
      IX(N,3)=IX(N-1,3)+1
      IX(N,4)=IX(N-1,4)+1
  170 CONTINUE
      SIGI(N,1)=0.
      SIGI(N,2)=0.
      SIGI(N,3)=0.
      MTAG(N)=0
      JNT(N)=0
      BETA(N)=0.
      IF (M-N) 180,180,140
  180 IF (NUMEL-N) 190,190,130
  190 WRITE (6,2045) (N,(IX(N,I),I=1,5),(SIGI(N,I),I=1,3),N=1,NUMEL)
      IF(NCRACK.EQ.0) GO TO 194
      WRITE(6,2050)
      DO 192 M=1,NCRACK
      READ (5,1040) N,MTAG(N),BETA(N)
      WRITE(6,1040) N,MTAG(N),BETA(N)
  192 BETA(N)=BETA(N)/57.296
  194 CONTINUE
C
C     INITIALIZE STRAINS
C
      DO 196 N=1,NUMEL
      NTAG(N)=0
      DO 196 I=1,3
  196 EPSI(N,I)=0.0
C
C     INITIALIZE CUMULATIVE DISPLACEMENTS
C
      DO 198 I=1,NEQ
  198 CU(I)=0.0
      JJ=0
```

```
      DO 199 I=1,NUMEL
      DO 199 J=1,4
      DO 199 K=1,4
      KK=IABS(IX(I,J)-IX(I,K))
      IF(KK.GT.JJ) JJ=KK
  199 CONTINUE
      NBAND=2*JJ+2
      NL=(NTOT-N15+1)/(NBAND+1)
      NLL=NEQ+3
      IF(NL.GT.NLL) NL=NLL
      NL=NL/4
      ND=2*NL
      NL=4*NL
      IF(ND.GE.NBAND) GO TO 209
      WRITE (6,4010) NL,NBAND
      CALL EXIT
  209 N16=N15+NL
      N17=N16+NL*NBAND
      JJ=N17-NTOT
      IF(JJ.LE.0) GO TO 210
      WRITE (6,3000) JJ
      CALL EXIT
  210 WRITE (6,4000) N17,M7,NL,NBAND
C
C     READ CONTROL INFORMATION FOR THE NEXT STEP IN INCREMENTAL ANALYSIS
C
C     NUMBER OF NODAL POINTS MISSING  NPMIS IS INTRODUCED TO TAKE
C     CARE OF EMPTY NODAL POINTS IN THE PREVIOUS STEP AND NEW NODAL
C     POINTS INTRODUCED AS A CONSEQUENCE OF CONSTRUCTION OR EXCAVATION
C     DONE DURING THIS STEP. NPMIS IS TAKEN AS THE SUM OF EMPTY NODAL
C     POINTS DURING PREVIOUS STEP PLUS THE NODAL POINTS VACATED OR
C     ADDED INTHIS STEP UPTO  NPMAX  .
C
      NCASE=0
  200 READ(5,1045) TITLE
      WRITE(6,2055) TITLE
      READ(5,1050) NPMAX, NELMAX, NUMER, NUMPC, MTYPE, NCODE ,NPMIS,MADD
      LLL=1
      III=0
      JJJ=0
      TOTAL=0.0
      SIGOI=0.
      SIGOJ=0.
  202 IF(NUMER.EQ.0) GO TO 204
      READ(5,1030)  (NUMR(N), N= 1,NUMER)
  204 IF(NPMIS.EQ.0) GO TO 206
      READ(5,1030)  (NNP(M), M = 1,NPMIS)
C
C     DETERMINE BANDWITH
C
```

```
  206 NUMEL=NELMAX
      NUMNP=NPMAX
      J=0
      DO 208 N=1,NUMEL
      DO 208 I=1,4
      DO 208 L=1,4
      KK=IABS(IX(N,I)-IX(N,L))
      IF (KK-J) 208,208,207
  207 J=KK
  208 CONTINUE
      MBAND=2*J+2
      IF (NPMIS.EQ.0) GO TO 215
      DO 212 I = 1, NPMIS
      J = NNP(I)
      CU(2*J-1) = 0.
  212 CU(2*J) = 0.
C
C     PRINT OUT CONTROL INFORMATION FOR THE CURRENT STEP IN INCREMENTAL
C     ANALYSIS
C
  215 IF(NCODE.EQ.1) GO TO 240
      IF(NUMER.EQ.0) GO TO 220
      WRITE(6,2060)NCASE,NPMAX,NELMAX,NUMPC,NPMIS,(NUMR(N),N=1,NUMER )
      GO TO 230
  220 WRITE (6,2060) NCASE,NPMAX,NELMAX,NUMPC,NPMIS
  230 IF(NPMIS.EQ.0)  GO TO 270
      WRITE(6,2065) (NNP(M) ,M = 1,NPMIS)
      GO TO 270
  240 IF(NUMER.EQ.0) GO TO 250
      WRITE(6,2070)NCASE,NPMAX,NELMAX,NUMPC,NPMIS,(NUMR(N),N=1,NUMER   )
      GO TO 260
  250 WRITE(6,2070)NCASE,NPMAX,NELMAX,NUMPC,NPMIS
  260 IF(NPMIS.EQ.0)  GO TO 270
      WRITE(6,2065) (NNP(M) ,M = 1,NPMIS)
C
C     IF THERE ARE ANY BOUNDARY PRESSURES FOR THIS STEP READ AND PRINT
C     THIS DATA
C
  270 IF (NUMPC) 290,310,290
  290 WRITE (6,2075)
      DO 300 L=1,NUMPC
      READ(5,1060) IBC(L),JBC(L),PR(L,1),PR(L,2)
  300 WRITE(6,2080) IBC(L),JBC(L),PR(L,1),PR(L,2)
  310 CONTINUE
C
C     CHANGE MATERIAL TYPE FOR ORIGINAL ELEMENTS
C
      IF(MADD.EQ.0) GO TO 336
      WRITE(6,2082)
      READ(5,1030) (MAD(I),IX(MAD(I),5),I=1,MADD)
```

```
      WRITE(6,2083)(MAD(I),IX(MAD(I),5),I=1,MADD)
      DO 335 I=1,MADD
      N=MAD(I)
      DO 335 J=1,3
      SIGI(N,J)=0.
  335 EPSI(N,J)=0.
  336 CONTINUE
C
C     IF INITIAL STRESS CONDITION IS TO BE ANALYSED INSTEAD OF INPUT,
C     USE NUMER = 0 AND DIRECTLY PROCEED TO STIFF.
C
C
C     CORRECT MATERIAL TYPE FOR ELEMENTS REMOVED OR ADDED. SET INITIAL
C     STRESSES EQUAL TO ZERO
C
      IF (NUMER.EQ.0) GO TO 340
      DO 330 I=1,NUMER
      NUM = NUMR(I)
      IX(NUM,5)=MTYPE
      DO 330 J=1,3
      SIGI(NUM,J) = 0.0
  330 EPSI(NUM,J)=0.0
  340 SR=0.
  345 CONTINUE
      DO 350 N=1,NUMEL
  350 RATIO(N)=1.0
C
C     FORM STIFFNESS MATRIX IN BLOCKS FOR THIS STEP
C
      CALL       SOLVE (R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,AA(N15),AA(N16),IX,MTAG,NTAG,JNT,IBC,JBC)
      IF(ISTOP.EQ.1.) CALL EXIT
C
C     SOLVE MATRIX EQUATIONS
C
      CALL BANSOL (AA(N15),AA(N16),MBAND,NBAND,NUMBLK,NL,JA)
      CALL       STRESS(R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,AA(N15),AA(N16),IX,MTAG,NTAG,JNT,IBC,JBC)
C
C
      CALL SCALE (IX,MTAG,NTAG, JNT,RATIO,BETA,SIGI,SIGNM,EPSI,CU,
     .   AA(N15),AA(N16))
C
C
      JJJ=1
      IF(JCHECK.EQ.0) GO TO 400
      III=III+1
      WRITE(6,2084) LLL,III
      NKT=0
      NML=NUMEL/2
```

```
      DO 380 N=1,NUMEL
      IF(MTAG(N).GT.2) NKT=NKT+1
      IF(NKT.LT.NML) GO TO 380
      WRITE(6,2085) NKT,NML
      CALL EXIT
  380 CONTINUE
      GO TO 345
  400 IF(TOTAL.EQ.1.0  ) GO TO 600
      LLL=LLL+1
      III=0
  450 GO TO 345
  600 CONTINUE
      IF(NCASE.GE.1) GO TO 800
C
C     IF THE INITIAL STRESSES ARE EVALUATED IN THIS STEP
C    1INITIALIZE  CU
C
      DO 750 I = 1,NEQ
  750 CU(I) = 0.0
  800 NCASE = NCASE + 1
      IF(NCASE.LE.NSTEP) GO TO 200
 1010 FORMAT (2I5,1F10.3,2I5)
      RETURN
 1015 FORMAT(3F10.3)
 1020 FORMAT(F10.3,E10.0,2F10.4)
 1022 FORMAT(4F10.4)
 1025 FORMAT (I5,F5.1,5F10.4)
 1030 FORMAT(16I5)
 1035 FORMAT(5I5)
 1040 FORMAT(2I5,F10.5)
 1045 FORMAT(18A4)
 1050 FORMAT(8I5)
 1055 FORMAT (16I5)
 1060 FORMAT(2I5,2F10.3)
 2010 FORMAT (17HOMATERIAL NUMBER= I3, 30H, NUMBER OF TEMPERATURE CARDS=
     1 I3, 15H, MASS DENSITY= E12.4,16H, MATERIAL CODE= I3
     2,16H, MATERIAL I.D.= I3)
 2015 FORMAT(18HOTENSILE STRENGTH=E12.4,21H, COEFF. OF FRICTION=E12.4
     *,19H, INITIAL OPENING = E12.4)
 2020 FORMAT (15HO  TEMPERATURE 10X 5HE     9X 6HNU     5X  8HALPHA    /
     *(F15.2,3E15.3))
 2022 FORMAT(15HO MODULUS RATIO,15H ANGLE OF FAULT,15H FRICTION COEF.,
     *15H SHEAR STRENGTH /(4E15.3))
 2025 FORMAT (108H1NODAL POINT         TYPE  X ORDINATE  Y ORDINATE  X LO
     1AD OR DISPLACEMENT  Y LOAD OR DISPLACEMENT TEMPERATURE )
 2030 FORMAT (I12,F12.2,2F12.5,2E24.7,F12.3)
 2035 FORMAT (26HONODAL POINT CARD ERROR N= I5)
 2040 FORMAT(120H1ELEMENT NO.     I      J     K      L     MATERIAL
     1SIGIXX       SIGIYY        SIGIXY                                  )
 2045 FORMAT (I13,4I6,I12,3F12.3)
```

```
 2050 FORMAT(1H0,' INITIALLY CRACKED ELEMENTS'/,'  EL. MTAG       BETA'/)
 2055 FORMAT(1H1 18A4////)
 2060 FORMAT( ' RESULTS AFTER STEP NO.-----------------------------------',I5/
     1' TOTAL NUMBER OF NODAL POINTS IN SYSTEM AT THIS STEP--------',I5/
     2' TOTAL NUMBER OF    ELEMENTS    IN SYSTEM AT THIS STEP--------',I5/
     3' TOTAL NUMBER OF PRESSURE CARDS AT THIS STEP ----------------',I5/
     4' TOTAL NUMBER OF NODAL POINTS MISSING AT THIS STEP----------',I5/
     5' ELEMENTS REMOVED IN THIS STEP ARE ' /(20I5))
 2065 FORMAT(      '  NUMBERS OF NODAL POINTS MISSING IN THIS STEP ARE   '/
     1(20I5))
 2070 FORMAT( ' RESULTS AFTER STEP NO.-----------------------------------',I5/
     1' TOTAL NUMBER OF NODAL POINTS IN SYSTEM AT THIS STEP--------',I5/
     2' TOTAL NUMBER OF    ELEMENTS    IN SYSTEM AT THIS STEP--------',I5/
     3' TOTAL NUMBER OF PRESSURE CARDS AT THIS STEP ----------------',I5/
     4' TOTAL NUMBER OF NODAL POINTS MISSING AT THIS STEP----------',I5/
     5' ELEMENTS ADDED    IN THIS STEP ARE ' /(20I5))
 2075 FORMAT(29H0PRESSURE BOUNDARY CONDITIONS/40H      I      J    PRESSUR
     1E I     PRESSURE J)
 2080 FORMAT(2I6,2F14.3)
 2082 FORMAT(///' MATERIAL TYPES CHANGED AT THIS STEP'//'  EL.  MTYPE')
 2083 FORMAT(2I5/)
 2084 FORMAT(1H1,' START ITERATION TO OBTAIN SYSTEM STABILITY UNDER THE
     *LOAD INCREMENT NO.',I3/
     *' SYSTEM STABILITY ITERATION NO.',I3//)
 2085 FORMAT(' NUMBER OF ELEMENTS CRACKED -',I4,'- REACHES THE LIMIT-',I
     *4,'- PROGRESSIVE FRACTURE IS LEADING THE SYSTEM TO TOTAL FAILURE')
 3000 FORMAT (70H PROGRAM EXECUTION TERMINATED. REQUIRED CORE EXCEEDS MT
     .OT BY                                              I10)
 4000 FORMAT (47H FOR THIS PROGRAM THE LOCATION USED IN AA IS =     I5,
     .    17H AND IN IA IS =          I5/
     .    18H MAX BLOCK SIZE =          I5/
     .    18H MAX BAND WIDTH =          I5/ )
 4010 FORMAT (25H0 NL IS LESS THAN 2*MBAND       /
     .    5H0 NL=             I5/
     .    8H0 MBAND=          I5)
      END
```

```
      SUBROUTINE SOLVE (R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,B,A,IX,MTAG,NTAG,JNT,IBC,JBC)
C
      COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .   VOL,ACELR,ACELZ,Q,HED(18),STOP,SR,SR1,TOTAL,TOL,TOLI,XC,YC,
     .   TEMP,SIGN,SIGDI,SIGDJ,
     .   LLL,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMER,NCODE,JA,NEQ
     .   ,NBAND,NCRACK,NSTEP,N15,M7,NTOT
      COMMON/TWO/ C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),
     .   LM(4),EE(3),EPS(3)
      COMMON/THREE/  E(6,4,8),EO(4,8),TENS(8),XNU(8),RO(8),EPST(8),
     .   MTC(8),NTC(8),MID(8)
      DIMENSION R(NUMNP),Z(NUMNP),CODE(NUMNP),UU(NUMNP,2),CU(NEQ),
     .BI(NEQ),BJ(NEQ),T(NUMNP),SIGI(NUMEL,6),EPSI(NUMEL,6),SIGNM(NUMEL),
     .BETA(NUMEL),RATIO(NUMEL),PR(NPC,2),IX(NUMEL,5),MTAG(NUMEL),
     .NTAG(NUMEL),JNT(NUMEL),IBC(NPC),JBC(NPC)
      DIMENSION B(NL),A(NL,NBAND)
      NB=NL/4
      ND=2*NB
      ND2=2*ND
      STOP=0.0
      NUMBLK=0
      NBK=1
      JA=ND *(MBAND+1)/1500+1
      DO 50 N=1,ND2
      B(N)=0.0
      DO  50 M=1,MBAND
   50 A(N,M)=0.0
   60 NUMBLK=NUMBLK+1
      NH=NB*(NUMBLK+1)
      NM=NH-NB
      NLL=NM-NB+1
      KSHIFT=2*NLL-2
      DO 210 N=1,NUMEL
      MTAGI=MTAG(N)
      ICHECK=0
      LCHECK=1
      ID=N
      MM=4
      IF(IX(N,3).EQ.IX(N,4)) MM=3
      IF(IX(N,3).EQ.IX(N,2)) MM=2
      IF (IX(N,5)) 210,210,65
   65 DO 80 I=1,4
      IF (IX(N,I)-NLL) 80,70,70
   70 IF (IX(N,I)-NM) 90,90,80
   80 CONTINUE
      GO TO 210
   90 IF(NTAG(N).EQ.1) GO TO 92
      IF(MTAG(N).GT.2) GO TO 99
      IF((LLL.GT.1).AND.(MTAG(N).EQ.0)) GO TO 99
```

```
      IF((III.GT.0).AND.(MTAG(N).EQ.0)) GO TO 99
   92 IF(IX(N,3)-IX(N,2)) 95,85,95
   85 CALL       ONED  (R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,B,A,IX,MTAG,NTAG,JNT,IBC,JBC)
      GO TO 98
   95 CALL       ELEMEN(R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,B,A,IX,MTAG,NTAG,JNT,IBC,JBC)
      NTAG(N)=0
   98 ICHECK=1
C     ******************************************************************
C     * WRITE ELEMENT INFORMATION ON FILE 2                            *
C     ******************************************************************
      WRITE(2'10) ((C(IIK,JJK),JJK=1,3),EE(IIK),IIK=1,3)
     1 ,((S(JJI,KKI),KKI=1,8),JJI=1,8),((ST(IKK,JKK),JKK=1,6),IKK=1,3)
     2 ,(RR(JII),ZZ(JII),JII=1,4),XC,YC,TEMP,VOL,MTYPE,N
   99 IX(N,5)=-IX(N,5)
      IF((LLL.GT.1).OR.(III.GT.0)) GO TO 101
      IF(VOL) 100,100,101
  100 WRITE(6,2000) N
      STOP=1.0
C     ******************************************************************
C     *   CALCULATE NODAL POINT FORCES                                 *
C     ******************************************************************
  101 IF((LLL.EQ.1).AND.(III.EQ.0)) GO TO 145
      IF(ICHECK.EQ.1) GO TO 105
      READ(2'10)  ((C(IIK,JJK),JJK=1,3),EL(IIK),IIK=1,3)
     1 ,((S(JJI,KKI),KKI=1,8),JJI=1,8),((ST(IKK,JKK),JKK=1,8),IKK=1,3)
     2 ,(RR(JII),ZZ(JII),JII=1,4),XC,YC,TEMP,VOL,MTYPE,N
      GO TO 166
C
C     EVALUATE STRESS TRANSFORMATION MATRIX AND STRESSES TO BE RELEASED
C        FOR NEWLY FRACTURED ELEMENT ONLY
C
  105 IF(IX(N,2).EQ.IX(N,3)) GO TO 166
      DO 106 I=1,3
      IF(SIGI(N,I).NE.0.) GO TO 110
  106 CONTINUE
      LCHECK=0
      GO TO 154
  110 IF (MTAG(N).NE.0) GO TO 111
      SIG(1)=0.
      SIG(2)=0.
      SIG(3)=0.
      GO TO 154
  111 IF(MTAG(N).NE.4) GO TO 112
      DO 112 I=1,3
      SIG(I)=-SIGI(N,I)
  112 SIGI(N,I)=0.
      GO TO 154
  113 EPSX=BETA(N)
```

```
      CC=COS(EPSX)
      SS=SIN(EPSX)
      S2=SS*SS
      C2=CC*CC
      SC=SS*CC
      SC2=2.D*SC
      CSD=C2-S2
      SIGXX=C2*SIGI(N,1)+S2*SIGI(N,2)+SC2*SIGI(N,3)
      SIGYY=S2*SIGI(N,1)+C2*SIGI(N,2)-SC2*SIGI(N,3)
      SIGXY=-SC*(SIGI(N,1)-SIGI(N,2))+CSD*SIGI(N,3)
      DX=EE(2)*SIGXX
      SIG(1)=C2*SIGXX+S2*DX-SC2*SIGXY
      SIG(2)=S2*SIGXX+C2*DX+SC2*SIGXY
      SIG(3)=SC*(SIGXX-DX)+CSD*SIGXY
      DO 144 I=1,3
      SIGI(N,I)=SIGI(N,I)-SIG(I)
  144 SIG(I)=-SIG(I)
      GO TO 154
C
C     CALCULATE TEMPERATURE STRESSES ,FOR LLL=1 AND III=0 ONLY
C
  145 DT=TEMP-O
      DX=EE(3)*DT
      SIG(1)=-(C(1,1)+C(1,2))*DX+SIGI(N,1)
      SIG(2)=-(C(2,2)+C(1,2))*DX+SIGI(N,2)
      SIG(3)=SIGI(N,3)
  154 IF(IX(N,2).EQ.IX(N,3)) GO TO 166
      DO 160 I=1,8
      P(I)=0.0
      IF(LCHECK.EQ.0) GO TO 160
      DO 155 J=1,3
  155 P(I)=P(I)-ST(J,I)*SIG(J)
  160 P(I)=P(I)*VOL
      IF((LLL.GT.1).OR.(III.GT.0)) GO TO 166
C
C     CALCULATE BODY FORCES, FOR LLL=1 AND III=0 ONLY
C
  163 XMM=MM
      DY=VOL*ACELZ*RO(MTYPE)/XMM
      DX=VOL*ACELR*RO(MTYPE)/XMM
      DO 165 I=1,MM
      P(2*I)=P(2*I)+DY
  165 P(2*I-1)=P(2*I-1)+DX
  166 CONTINUE
  167 DO 168 I=1,MM
  168 LM(I)=2*IX(N,I)-2
      DO 200 I=1,MM
      DO 200 K=1,2
      II=LM(I)+K-KSHIFT
      KK=2*I-2+K
```

```
      IF(ICHECK.EQ.0) GO TO 172
      B(II)=B(II)+P(KK)
  172 DO 200 J=1,MM
      DO 200 L=1,2
      JJ=LM(J)+L-II+1-KSHIFT
      LL=2*J-2+L
      IF(JJ) 200,200,175
  175 IF(ND-JJ) 180,195,195
  180 WRITE (6,2001) N
      STOP=1.0
      GO TO 210
  195 A(II,JJ)=A(II,JJ)+S(KK,LL)
  200 CONTINUE
  210 CONTINUE
      IF((LLL.GT.1).OR.(III.GT.0)) GO TO 301
      DO 220 N=NLL,NM
      IF(N-NUMNP) 215,215,220
  215 K=2*N-KSHIFT
      B(K)=B(K)+UU(N,2)
      B(K-1)=B(K-1)+UU(N,1)
  220 CONTINUE
      IF(NUMPC) 225,301,225
  225 DO 300 L=1,NUMPC
      I=IBC(L)
      J=JBC(L)
      DR=Z(I)-Z(J)
      DZ=R(J)-R(I)
      PP2=(PR(L,2)+PR(L,1))/6.
      PP1=PP2+PR(L,1)/6.
      PP2=PP2+PR(L,2)/6.
      II=2*I-KSHIFT
      JJ=2*J-KSHIFT
      IF(II) 265,265,235
  235 IF(II-ND) 240,240,265
  240 B(II-1)=B(II-1)+PP1*DR
      B(II)=B(II)+PP1*DZ
  265 IF(JJ) 300,300,270
  270 IF(JJ-ND) 275,275,300
  275 B(JJ-1)=B(JJ-1)+PP2*DR
      B(JJ)=B(JJ)+PP2*DZ
  300 CONTINUE
  301 CONTINUE
      II=ND*(NUMBLK-1)+1
      JJ=ND*NUMBLK
      KK=0
      IF(III.NE.0) GO TO 303
      DO 302 N=II,JJ
  302 BJ(N)=0.
  303 IF((LLL.EQ.1).AND.(III.EQ.0)) GO TO 305
      GO TO 313
```

```
  305 DO 306 N=II,JJ
      KK=KK+1
  306 BI(N)=B(KK)
      GO TO 310
  313 FACTOR=SR
      IF(II.EQ.0) FACTOR=1.-TOTAL
      IF(LL.EQ.1) FACTOR=TOTAL
      DO 314 N=II,JJ
      KK=KK+1
      BJ(N)=BJ(N)+B(KK)
  314 B(KK)=FACTOR*BI(N)+BJ(N)
  310 DO 400 M=NLL,NH
      IF (M-NUMNP) 315,315,400
  315 U=UU(M,1)
      N=2*M-1-KSHIFT
      IF (CODE(M)) 390,400,316
  316 IF (CODE(M)-1.) 317,370,317
  317 IF (CODE(M)-2.) 318,390,318
  318 IF (CODE(M)-3.) 390,380,390
  370 CALL MODIFY(A,B,ND2,MBAND,NBAND,N,U)
      GO TO 400
  380 CALL MODIFY(A,B,ND2,MBAND,NBAND,N,U)
  390 U=UU(M,2)
      N=N+1
      CALL MODIFY(A,B,ND2,MBAND,NBAND,N,U)
  400 CONTINUE
C     ******************************************************************
C     *  WRITE BLOCK INFORMATION ON FILE 1                             *
C     ******************************************************************
      WRITE(1'NBK)(B(N),(A(N,M),M=1,MBAND),N=1,ND)
      NBK=NBK+JA
      DO 420 N=1,ND
      K=N+ND
      B(N)=B(K)
      B(K)=0.0
      DO 420 M=1,MBAND
      A(N,M)=A(K,M)
  420 A(K,M)=0.0
      IF (NH-NUMNP) 60,480,480
  480 CONTINUE
      RETURN
 2000 FORMAT (26HONEGATIVE AREA ELEMENT NO. I4)
 2001 FORMAT (29HOBAND WIDTH EXCEEDS ALLOWABLE I4)
      END
```

```
      SUBROUTINE ONED  (R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,B,A,IX,MTAG,NTAG,JNT,IBC,JBC)
      COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .   VOL,ACELR,ACELZ,Q,HED(18),STOP,SR,SR1,TOTAL,TOL,TOL1,XC,YC,
     .   TEMP,SIGN,SIGDI,SIGDJ,
     .   LLL,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMER,NCODE,JA,NEQ
     .   ,NBAND,NCRACK,NSTEP,N15,N7,NTOT
      COMMON/TWO/ C(2,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),
     .   LM(4),EE(3),EPS(3)
      COMMON/THREE/  E(8,4,8),EO(4,8),TENS(8),XNU(8),RO(8),EPST(8),
     .   MTC(8),NTC(8),MOD(8)
      DIMENSION R(NUMNP),Z(NUMNP),CODE(NUMNP),UU(NUMNP,2),CU(NEQ),
     .BI(NEQ),BJ(NEQ),T(NUMNP),SIGI(NUMEL,6),EPSI(NUMEL,6),SIGNM(NUMEL),
     .BETA(NUMEL),RATIO(NUMEL),PR(NPC,2),IX(NUMEL,5),MTAG(NUMEL),
     .NTAG(NUMEL),JNT(NUMEL),IBC(NPC),JBC(NPC)
      DIMENSION B(NL),A(NL,NBAND)
C
      DO 100 I=1,8
      P(I)=0.0
      DO 100 J=1,8
  100 S(I,J)=0.0
      MTYPE=IX(N,5)
      I=IX(N,1)
      J=IX(N,2)
      DX=R(J)-R(I)
      DY=Z(J)-Z(I)
      XL= SQRT(DX**2+DY**2)
      COSA=DX/XL
      SINA=DY/XL
      EE(1)=E(1,2,MTYPE)
      EE(2)=E(1,3,MTYPE)
      COMM=EE(1)*EE(2)/XL
      S(1,1)=COSA*COSA*COMM
      S(1,2)=COSA*SINA*COMM
      S(1,3)=-S(1,1)
      S(1,4)=-S(1,2)
      S(2,1)=S(1,2)
      S(2,2)=SINA*SINA*COMM
      S(2,3)=-S(1,2)
      S(2,4)=-S(2,2)
      S(3,1)=S(1,3)
      S(3,2)=S(2,3)
      S(3,3)=S(1,1)
      S(3,4)=S(1,2)
      S(4,1)=S(1,4)
      S(4,2)=S(2,4)
      S(4,3)=S(3,4)
      S(4,4)=S(2,2)
      EP=E(1,4,MTYPE)*E(1,1,MTYPE)
      DX=DX*EP
```

```
      DY=DY*EP
      P(1)=S(1,1)*DX+S(1,2)*DY
      P(2)=S(2,1)*DX+S(2,2)*DY
  136 P(3)=-P(1)
      P(4)=-P(2)
      VOL=1.
      DO 140 I=1,4
      NPP=IX(N,I)
      RR(I)=R(NPP)
  140 ZZ(I)=Z(NPP)
      XC=0.5*(RR(1)+RR(2))
      YC=0.5*(ZZ(1)+ZZ(2))
      DO 200 I=1,3
      DO 200 J=1,8
  200 ST(I,J)=0.0
      DO 300 I=1,3
      DO 300 J=1,3
  300 C(I,J)=0.0
      DO 400 I=1,3
  400 EE(I)=E(1,I+1,MTYPE)
C
      RETURN
C
      END
```

```
      SUBROUTINE ELEMEN(R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,B,A,IX,MTAG,NTAG,JNT,IBC,JBC)
      COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .   VOL,ACELR,ACELZ,Q,HED(18),STUP,SR,SR1,TOTAL,TOL,TOL1,XC,YC,
     .   TEMP,SIGN,SIGDI,SIGDJ,
     .   LLL,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMER,NCODE,JA,NEQ
     .   ,NBAND,NCHALK,NSTEP,N15,MT,NTOT
      COMMON/TWO/ C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),
     .   LM(4),EE(3),EPS(3)
      COMMON/THREE/  E(6,4,8),FU(4,8),IENS(8),XNU(8),RO(8),EPST(8),
     .   MIC(8),NTC(8),MID(8)
      DIMENSION R(NUMNP),Z(NUMNP),CODE(NUMNP),UU(NUMNP,2),CU(NEQ),
     .BI(NEQ),BJ(NEQ),T(NUMNP),SIGI(NUMEL,6),EPSI(NUMEL,6),SIGNM(NUMEL),
     .BETA(NUMEL),RATIO(NUMEL),PR(NPC,2),IX(NUMEL,5),MTAG(NUMEL),
     .NTAG(NUMEL),JNT(NUMEL),IBC(NPC),JBC(NPC)
      DIMENSION B(NL),A(NL,NBAND)
      DIMENSION U(3),V(3)
C
      EX=0.00001
      I=IX(N,1)
      J=IX(N,2)
      K=IX(N,3)
      L=IX(N,4)
      MTYPE=IX(N,5)
      VOL=0.
      TEMP=(T(I)+T(J)+T(K)+T(L))/4.0
      RAT=0.0
      NUMTC=NTC(MTYPE)
      IF (NUMTC.EQ.1) GO TO 100
      DO 50 M=2,NUMTC
      IF (E(M,1,MTYPE)-TEMP) 50,60,60
   50 CONTINUE
   60 DEN=E(M,1,MTYPE)-E(M-1,1,MTYPE)
      IF(DEN) 70,80,70
   70 RAT  =(TEMP-E(M-1,1,MTYPE))/DEN
   80 DO 90 KK=1,3
   90 EE(KK)=E(M-1,KK+1,MTYPE)+RAT  *(E(M,KK+1,MTYPE)-E(M-1,KK+1,MTYPE))
      GO TO 106
  100 DO 105 KK=1,3
  105 EE(KK)=E(1,KK+1,MTYPE)
  106 CONTINUE
C     ******************************************************************
C     * CALCULATE ORTHOTROPIC ELASTICITY MATRIX FOR MTAG=0,1,2         *
C     ******************************************************************
      KKK=MTAG(N)+1
      GO TO (117,107,108),KKK
  107 CONTINUE
C
C     MTAG=1
C
```

```
      EX=EEX
      EY=1.
      GO TO 109
  108 CONTINUE
C
C     MTAG=2
C
      EX=EEX
      EY=EEX
  109 CONTINUE
      GO TO (110,111),NPLANE
  110 CONTINUE
C
C     PLANE STRESS
C
      EE1=EE(1)*EX
      EE2=EE(1)*EY
      EEN=EE(2)*EY
      COMM=(1.-(EE1/EE2)*(EEN**2))
      C11=EE1/COMM
      C12=EEN*C11
      C21=C12
      C22=EE2/COMM
      GO TO 115
  111 CONTINUE
C
C     PLANE STRAIN
C
      EE1=EE(1)*EX
      EE2=EE(1)*EY
      EN1=EE(2)*EX
      EN2=EE(2)*EY
      EN3=EE(2)
      XA=1.-EN1*EN3
      XB=1.-EN2*EN3
      XC=1.+EN3
      XD=EN2*XC
      XE=EE2/EE1
      COMM=(XE*XA*XB-(EN2**2)*(XC**2))/EE2
      C11=XB/COMM
      C12=XD/COMM
      C21=C12
      C22=XE*XA/COMM
  115 CONTINUE
      EPSX=BETA(N)
      KCK=0
      IF(MID(MTYPE).EQ.0) GO TO 112
      EPSX=EO(2,MTYPE)
      IF(MTAG(N).NE.0) EPSX=EPSX-BETA(N)
  112 CONTINUE
```

```
ELEM101       S1= SIN(EPSX)
ELEM102       S2=S1*S1
ELEM103       C1= COS(EPSX)
ELEM104       C2=C1*C1
ELEM105       C1S1=C1*S1
ELEM106       C2S2=C2*S2
ELEM107       C4=C2**2
ELEM108       S4=S2*S2
ELEM109       DA=C11+C22
ELEM110       DB=C11-C12
ELEM111       DC=C22-C12
ELEM112       DD=2.*C2S2*C1.
ELEM113       C(1,1)=C4*C11+DD+S4*C22
ELEM114       C(1,2)=C2S2*DA+(C4+S4)*C12
ELEM115       C(1,3)=C1S1*(C2*DB-S2*DC)
ELEM116       C(2,2)=S4*C11+DD+C4*C22
ELEM117       C(2,3)=C1S1*(S2*DB-C2*DC)
ELEM118       C(3,3)=C2S2*(DB+DC)
ELEM119       IF((MID(MTYPE).EQ.0).OR.(KCK.EQ.1)) GO TO 116
ELEM120       IF(MTAG(N).NE.0) KCK=1
ELEM121       G=0.5*EE1/(1.+(EX**2)*EE2)
ELEM122       EPSX=2.*EPSX
ELEM123       S1=SIN(EPSX)
ELEM124       C1=COS(EPSX)
ELEM125       SC=S1*C1
ELEM126       S1=S1*S1
ELEM127       C1=C1*C1
ELEM128       C11=G*S1
ELEM129       C12=-C11
ELEM130       C13=-G*SC
ELEM131       C22=C11
ELEM132       C23=-C13
ELEM133       C33=G*C1
ELEM134       C(1,1)=C(1,1)+C11
ELEM135       C(1,2)=C(1,2)+C12
ELEM136       C(1,3)=C(1,3)+C13
ELEM137       C(2,2)=C(2,2)+C22
ELEM138       C(2,3)=C(2,3)+C23
ELEM139       C(3,3)=C(3,3)+C33
ELEM140       IF(KCK.EQ.0) GO TO 116
ELEM141       EX=EEX
ELEM142       EY=1.
ELEM143       IF(MTAG(N).EQ.1) GO TO 113
ELEM144       EY=EEX
ELEM145   113 CONTINUE
ELEM146       C11=EX*C(1,1)
ELEM147       C12=EX*C(1,2)
ELEM148       C22=EY*C(2,2)
ELEM149       C21=C12
ELEM150       EPSX=BETA(N)
```

```
      GO TO 112
  116 CONTINUE
      C(2,1)=C(1,2)
      C(3,2)=C(2,3)
      C(3,1)=C(1,3)
      IF(MTAG(N).EQ.0) GO TO 118
      MTAG(N)=MTAG(N)+2
      GO TO 118
  117 CONTINUE
      IF((MID(MTYPE).EQ.0).OR.(EO(1,MTYPE).EQ.1.)) GO TO 125
      EX=EO(1,MTYPE)
      EY=1.0
      GO TO 109
  125 CONTINUE
C     ****************************************************************
C     * CALCULATE ISOTROPIC ELASTICITY MATRIX                        *
C     ****************************************************************
      EE1=EE(1)
      EE2=EE(2)
      IF(NPLANE.EQ.1) GO TO 119
      EE1=EE1/(1.-EE2**2)
      EE2=EE2/(1.-EE2)
  119 CONTINUE
      COMM=EE1/(1.-EE2**2)
      C(1,1)=COMM
      C(1,2)=COMM*EE2
      C(1,3)=0.
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=0.
      C(3,1)=0.
      C(3,2)=0.
      C(3,3)=.5*COMM*(1.-EE2)
  118 CONTINUE
      DO 130 J=1,10
      DO 120 I=1,3
  120 ST(I,J)=0.
      DO 130 I=1,10
  130 S(I,J)=0.
  135 CONTINUE
      DO 140 I=1,4
      NPP=IX(N,I)
      RR(I)=R(NPP)
  140 ZZ(I)=Z(NPP)
      IF(IX(N,3)-IX(N,4)) 145,150,145
  145 XC=(RR(1)+RR(2)+RR(3)+RR(4))/4.
      YC=(ZZ(1)+ZZ(2)+ZZ(3)+ZZ(4))/4.
      RR(5)=XC
      ZZ(5)=YC
      K=5
```

```
      J=1
      I=4
      LM(3)=9
      NT=4
      GO TO 160
  150 NT=1
      LM(3)=5
      I=1
      J=2
      K=3
      XC=(RR(1)+RR(2)+RR(3))/3.
      YC=(ZZ(1)+ZZ(2)+ZZ(3))/3.
      RR(5)=RR(3)
      ZZ(5)=ZZ(3)
  160 DO 200 NN=1,NT
      LM(1)=2*I-1
      LM(2)=2*J-1
      U(1)=ZZ(J)-ZZ(K)
      U(2)=ZZ(K)-ZZ(I)
      U(3)=ZZ(I)-ZZ(J)
      V(1)=RR(K)-RR(J)
      V(2)=RR(I)-RR(K)
      V(3)=RR(J)-RR(I)
      AREA=(RR(J)*U(2)+RR(I)*U(1)+RR(5)*U(3))/2.
      VOL=VOL+AREA
      COMM=.25/AREA
      XNT=NT
      COM=2./XNT
      COM=COM*COMM
      DO 180 L=1,3
      II=LM(L)
      ST(1,II)=ST(1,II)+U(L)*COM
      ST(2,II+1)=ST(2,II+1)+V(L)*COM
      ST(3,II)=ST(3,II)+V(L)*COM
      ST(3,II+1)=ST(3,II+1)+U(L)*COM
      DO 180 M=1,3
      JJ=LM(M)
      S(II,JJ)=S(II,JJ)+(C(1,1)*U(L)*U(M)+C(1,3)*(V(L)*U(M)+U(L)*V(M))
     * +C(3,3)*V(L)*V(M))*COMM
      S(II,JJ+1)=S(II,JJ+1)+(U(L)*(C(1,2)*V(M)+C(1,3)*U(M))+V(L)*(C(2,3)
     *    *V(M)+C(3,3)*U(M)))*COMM
      S(II+1,JJ+1)=S(II+1,JJ+1)+(C(2,2)*V(L)*V(M)+C(2,3)*(U(L)*V(M)+V(L)
     * *U(M))+C(3,3)*U(L)*U(M))*COMM
      S(JJ+1,II)=S(II,JJ+1)
  180 CONTINUE
      I=J
      J=J+1
  200 CONTINUE
      IF(IX(N,3)-IX(N,4)) 220,250,220
  220 DO 240 I=1,2
```

```
      KK=10-I
      DO 240 K=1,KK
      IF(S(KK+1,KK+1).EQ.0.) S(KK+1,KK+1)=.000001
      CC=S(KK+1,K)/S(KK+1,KK+1)
      DO 230 J=1,3
  230 ST(J,K)=ST(J,K)-CC*ST(J,KK+1)
      DO 240 J=1,KK
  240 S(J,K)=S(J,K)-CC*S(J,KK+1)
  250 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE STRESS(R,Z,CODE,UU,CU,BI,BJ,T,SIGI,EPSI,SIGNM,BETA,
     .   RATIO,PR,B,A,IX,MTAG,NTAG,JNT,IBC,JBC)
      COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .   VOL,ACELR,ACELZ,Q,HED(18),STOP,SR,SRI,TOTAL,TOL,TOLI,XC,YC,
     .   TEMP,SIGN,SIGDI,SIGDJ,
     .   LLE,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMER,NCODE,JA,NEQ
     .   ,NBAND,NCRACK,NSTEP,N15,M7,NTOT
      COMMON/TWO/ C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),
     .   LM(4),EE(3),EPS(3)
      COMMON/THREE/  E(8,4,8),EO(4,8),TENS(8),XNU(8),RO(8),EPST(8),
     .   MTC(8),NDC(8),MID(8)
      DIMENSION R(NUMNP),Z(NUMNP),CODE(NUMNP),UU(NUMNP,2),CU(NEQ),
     .BI(NEQ),BJ(NEQ),T(NUMNP),SIGI(NUMEL,6),EPSI(NUMEL,6),SIGNM(NUMEL),
     .BETA(NUMEL),RATIO(NUMEL),PR(NPC,2),IX(NUMEL,5),MTAG(NUMEL),
     .NTAG(NUMEL),JNT(NUMEL),IBC(NPC),JBC(NPC)
      DIMENSION B(NL),A(NL,NBAND)
      DO 600 M=1,NUMEL
      ID=M
      FIND(2'ID)
      IX(M,5)=IABS(IX(M,5))
      DO 50 I=1,6
   50 SIG(I)=0.0
C     ******************************************************************
C     * READ ELEMENT INFORMATION FROM FILE 2                           *
C     ******************************************************************
      READ(2'ID)  ((C(IIK,JJK),JJK=1,3),EE(IIK),IIK=1,3)
     1 ,((S(JJI,KKI),KKI=1,8),JJI=1,8),((ST(IKK,JKK),JKK=1,8),IKK=1,3)
     2 ,(RR(JII),ZZ(JII),JII=1,4),XC,YC,TEMP,VOL,MTYPE,N
      IF(IX(N,3)-IX(N,2)) 90,60,90
C
C     ******************************************************************
C     *    ONE-DIMENSIONAL ELEMENT                                     *
C     ******************************************************************
   60 IF(MTAG(N).EQ.4) GO TO 70
      I=IX(N,1)
      J=IX(N,2)
      DX=RR(2)-RR(1)
      DY=ZZ(2)-ZZ(1)
      XL= SQRT(DX**2+DY**2)
      DU=B(2*J-1)-B(2*I-1)
      DV=B(2*J)-B(2*I)
      DL=DV*DY/XL+DU*DX/XL
      SIG(1)=DL*EE(1)/XL
      IF(EE(2).EQ.0.) SIG(1)=0.
   70 SIGI(N,4)=SIG(1)
      SIGI(N,5)=0.
      SIGI(N,6)=0.
      GO TO 600
C
C     ******************************************************************
```

```
C      *     TWO-DIMENSIONAL ELEMENT                                          *
C      ****************************************************************
   90 MM=4
      IF(IX(N,3).EQ.IX(N,4)) MM=3
  170 DO 180 I=1,3
      RR(I)=0.
      DO 180 J=1,MM
      II=2*J
      JJ=2*IX(N,J)
  180 RR(I)=RR(I)+ST(I,II)*B(JJ)+ST(I,II-1)*B(JJ-1)
      DO 190 I=1,3
      DO 190 J=1,3
      SIG(I)=SIG(I)+C(I,J)*RR(J)
      IF(MTAG(N).EQ.4) SIG(I)=0.
  190 CONTINUE
      IF(MTAG(N).NE.3) GO TO 191
      EPSX=BETA(N)
      CC=COS(EPSX)
      SS=SIN(EPSX)
      S2=SS*SS
      C2=CC*CC
      SC=SS*CC
      SC2=2.*SC
      CSD=C2-S2
      SIGXX=C2*SIG(1)+S2*SIG(2)+SC2*SIG(3)
      SIGYY=S2*SIG(1)+C2*SIG(2)-SC2*SIG(3)
      SIGXY=-SC*(SIG(1)-SIG(2))+CSD*SIG(3)
      DX=FF(2)*SIGXX
      SIG(1)=SIG(1)-(C2*SIGXX+S2*DX-SC2*SIGXY)
      SIG(2)=SIG(2)-(S2*SIGXX+C2*DX+SC2*SIGXY)
      SIG(3)=SIG(3)-(SC*(SIGXX-DX    )+CSD*SIGXY)
  191 CONTINUE
      DO 192 I=1,3
      SIGI(N,I+3)=SIG(I)
      EPSI(N,I+3)=RR(I)
  192 CONTINUE
  600 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRIFTH (MTAG,JNT,RATIO,BETA,SIGI,SIGNM)
C
      COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .   VOL,ACELR,ACELZ,Q,HED(18),STOP,SR,SR1,TOTAL,TOL,TOLI,XC,YC,
     .   TEMP,SIGN,SIGDI,SIGDJ,
     .   LLL,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMBR,NCODE,JA,NEQ
     .   ,NBAND,NCRACK,NSTEP,N15,M7,NTOT
      COMMON/TWO/ C(3,3),S(10,10),SIG(6),P(8),S1(3,10),RR(5),ZZ(5),
     .   LM(4),EE(3),EPS(3)
      COMMON/THREE/  E(8,4,8),EO(4,8),TENS(8),XNU(8),RO(8),EPST(8),
     .   MTC(8),NIC(8),MID(8)
      DIMENSION MTAG(NUMEL),JNT(NUMEL),RATIO(NUMEL),BETA(NUMEL),
     .   SIGI(NUMEL,6),SIGNM(NUMEL)
C    ***********************************************************
C     * XNU,XN= COEF. OF FRICTION FOR THE MODIFIED GRIFFITH CRITERION  *
C     * TENS,SIGT= TENSILE STRENGTH OF THE MATERIAL                    *
C     * SIGN= STRESS COMPONENT NORMAL TO THE FRACTURE PLANE            *
C     * EPSX= ANGLE BETWEEN P AND SIGN                                 *
C     * BETA= ANGLE BETWEEN X-AXIS AND SIGN                            *
C     * ANGLES ARE POSITIVE COUNTERCLOCKWISE                           *
C     ******************************************************************
      MTAGI=MTAG(N)
      JNTI=JNT(N)
      SIGYY=0.
      SIGN=0.
      IF(MTAG(N).LT.2) GO TO 600
      MTAG(N)=0
      JNT(N)=0
      BETA(N)=0.
      PP=SIG(4)
      Q=SIG(5)
      R=SIG(6)
      SIGT=TENS(MTYPE)
      XN=XNU(MTYPE)
      SUM=3.0*PP+Q
      TOL=0.005*SIGT
      TOLL=TOL
      IF(JCK.EQ.0) TOLL=0.
      MD=MID(MTYPE)
      IF((MTAGI.EQ.2).AND.(JNTI.LT.4)) GO TO 30
C     CHECK INITIAL WEAK PLANE
      IF(MD.EQ.0) GO TO 30
      WX=2.*(EO(2,MTYPE)-R/57.296)
      W=0.5*WX
      X=COS(WX)
      PQADD=PP+Q
      PQSUB=PP-Q
      SIGYY=0.5*(PQADD+PQSUB*X)
      SIGT=EO(4,MTYPE)
      XN=EO(3,MTYPE)
```

```
      IF(SIGYY.LE.O.) GO TO 200
      SIGN=SIGYY
      GO TO 40
   30 CONTINUE
      IF(SUM.LT.O.) GO TO 100
      SIGN=PP
   40 CONTINUE
      DSIG=SIGN-SIGI
      IF((JCK.EQ.1).AND.(ABS(DSIG).LT.TOLL)) GO TO 50
      IF(DSIG.LT.TOLL) GO TO 600
C     ******************************************************************
C     * FRACTURE PLANE COINCIDES WITH SIG(5) IF CRITICAL ANGLE IS ZERO *
C     * APPLY ORIGINAL GRIFFITH CRITERION                              *
C     ******************************************************************
   50 CONTINUE
      IF(MD.EQ.0) GO TO 70
      IF((MTAGI.EQ.2).AND.(JNTI.LT.4)) GO TO 70
      MTAG(N)=1
      JNT(N)=4
      BETA(N)=ED(2,MTYPE)
      GO TO 320
   70 JNT(N)=1
      SIGN=PP
      SIGYY=SIGN
      EPSX=0.0
      GO TO 300
  100 CONTINUE
C     ******************************************************************
C     * CRITICAL ANGLE IS NOT ZERO                                     *
C     * APPLY THE ORIGINAL GRIFFITH IF THE INITIAL FLAW IS OPEN        *
C     ******************************************************************
      PQADD=PP+Q
      PQSUB=PP-Q
      X=-0.5*PQSUB/PQADD
      SIGYY=0.5*(PQADD+PQSUB*X)
      IF(SIGYY.LT.O.) GO TO 200
      SIGN=-PQSUB**2/(8.*PQADD)
      DSIG=SIGN-SIGI
      IF((JCK.EQ.1).AND.(ABS(DSIG).LT.TOLL)) GO TO 150
      IF(DSIG.LT.TOLL) GO TO 600
  150 CONTINUE
      WX= ARCOS(X)
      W=0.5*WX
      JNT(N)=2
      Z=Q/PP
      S1= SIN(W)
      S2=S1**2
      SS= SIN(WX)
      ZZZ=Z*Z-1.0
      Z=Z-1.0
```

```
      ARG=(Z*S2+1.0- SQRT(ZZZ*S2+1.0))/(0.5*Z*SS)
      EPSX= ATAN(ARG)-W
      GO TO 300
  200 CONTINUE
C     ****************************************************************
C     * APPLY THE MODIFIED GRIFFITH IF THE INITIAL FLAW IS CLOSED    *
C     ****************************************************************
      IF(MO.EQ.0) GO TO 240
C     CHECK SHEAR STRENGTH
      SIGN=0.5*(PQSUB*SQRT(1.+XN**2)+PQADD*XN)
      DSIG=SIGN-SIGT
      IF((JCK.EQ.1).AND.(ABS(DSIG).LT.TOLL)) GO TO 220
      IF(DSIG.LT.TOLL) GO TO 600
  220 SIGM=.5*PQADD
      TAU=.5*PQSUB
      IF(TAU.EQ.0.) GO TO 600
      WX=ATAN(XN)
      X=((SIGM+SIGT/XN)/TAU)*SIN(WX)
      IF(X.GT.1.) X=1.
      AX=ARSIN(X)
      B1=-.5*(AX+WX)
      B2=.5*(3.1416+AX-WX)
      BW=.5*ATAN2(1.,XN)
C     CHECK BOUNDS OF ANGLE
      IF((BW.GE.B1).AND.(BW.LE.B2)) GO TO 230
      GO TO 600
  230 XN=XNU(MTYPE)
      SIGM=.25*(PQSUB*SQRT(1.+XN**2)+PQADD*XN)
      IF(SIGN.GE.SIGM) GO TO 240
      JNT(N)=5
      GO TO 270
  240 CONTINUE
      X=1.0/XN
      WX= ATAN(X)
      SIGYY=0.5*(PQADD+PQSUB* COS(WX))
      IF(SIGYY.GE.0.1) GO TO 600
      W=0.5*WX
      SIGN=0.25*(PQSUB* SQRT(1.+XN**2)+PQADD*XN)
      DSIG=SIGN-SIGT
      IF((JCK.EQ.1).AND.(ABS(DSIG).LT.TOLL)) GO TO 250
      IF(DSIG.LT.TOLL) GO TO 600
  250 CONTINUE
      JNT(N)=3
  270 CONTINUE
      EPSX=0.7854-W
  300 CONTINUE
      MTAG(N)=1
      BETA(N)=R/57.296-EPSX
      ANG=57.296*BETA(N)
      IF(ANG.GT.90.0) BETA(N)=BETA(N)-3.1416
```

```
      IF(ANG.LT.-90.) BETA(N)=BETA(N)+3.1416
  320 CONTINUE
      IF(KCHECK.EQ.0) GO TO 600
C     ****************************************************************
C     *   EVALUATE STRESS RATIO FOR EACH ELEMENT                     *
C     ****************************************************************
      A0=SIGI(N,1)+SIGI(N,2)
      B0=SIGI(N,1)-SIGI(N,2)
      A1=SIG(1)+SIG(2)
      B1=SIG(1)-SIG(2)
      C0=B0**2+4.0*SIGI(N,3)**2
      C1=B0*B1+4.0*SIGI(N,3)*SIG(3)
      C2=B1**2+4.0*SIG(3)**2
      IJNT=JNT(N)+1
      GO TO (600,350,400,500,340,500),IJNT
  340 C4=X**4
      C1=C1*C4
      C2=C2*C4
  350 SIGT2=2.*SIGI
      DSIGT=SIGT2-A0
      AAA=A1**2-C2
      BB=-A1*DSIGT-C1
      CC=DSIGT**2-C0
      GO TO 550
  400 CONTINUE
      AAA=C2
      BB=C1+4.0*SIGT*A1
      CC=8.0*SIGT*A0+C0
      GO TO 550
  500 CONTINUE
      XN2=XN**2
      RXN2=1.0+XN2
      AAA=RXN2*C2-XN2*A1**2
      IF(JNT(N).EQ.3) GO TO 520
      BB=RXN2*C1-XN2*A1*A0+2.0*SIGT*A1*XN
      CC=RXN2*C0-XN2*A0*A0+4.0*SIGT*A0*XN- 4.0*SIGT**2
      GO TO 550
  520 BB=RXN2*C1-XN2*A1*A0+4.0*SIGT*A1*XN
      CC=RXN2*C0-XN2*A0*A0+8.0*SIGT*A0*XN-16.0*SIGT**2
  550 CONTINUE
      DD=BB**2-AAA*CC
      IF(DD.GE.0.0) GO TO 560
      GO TO 600
  560 ROOT= SQRT(DD)
      RI=(-BB+ROOT)/AAA
      RJ=(-BB-ROOT)/AAA
      RATIO(N)=RI
      IF((RATIO(N).LT.TOLI ).AND.(RJ.GE.TOLI )) RATIO(N)=RJ
      IF((RATIO(N).GT.1.).AND.(RJ.LT.1.)) RATIO(N)=RJ
      IF( ABS(1.-RATIO(N)).LE.TOLI ) RATIO(N)=1.0
```

```
      IF(RATIO(N).LT.TOLI) RATIO(N)=TOLI
      IF (RATIO(N).GT.1.) RATIO(N)=1.
  600 CONTINUE
      SIGNM(N)=SIGN
      IF(MTAG(N).NE.MTAGI) JCHECK=1
      IF(MTAGI.EQ.2) MTAG(N)=2
      IF(MTAG(N).NE.1) RATIO(N)=1.0
      RETURN
C
      END
```

```
      SUBROUTINE SCALE(IX,MTAG,NTAG, JNT,RATIO,BETA,SIGI,SIGNM,EPSI,CU,
     . B,A)
C
      COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .   VOL,ACELR,ACELZ,Q,HED(18),STOP,SR,SR1,TOTAL,TOL,TOLI,XC,YC,
     .   TEMP,SIGN,SIGDI,SIGDJ,
     .   LLL,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMCR,NCODE,JA,NEQ
     .   ,NBAND,NCRACK,NSTEP,N15,M7,NTOT
      COMMON/TWO/ C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),
     .   LM(4),EE(3),EPS(3)
      COMMON/THREE/  E(8,4,8),EO(4,8),TENS(8),XNU(8),RO(8),EPST(8),
     .   MTC(8),NTC(8),MID(8)
      DIMENSION IX(NUMEL,5),MTAG(NUMEL),NTAG(NUMEL),JNT(NUMEL),RATIO(NUM
     .EL),BETA(NUMEL),SIGI(NUMEL,6),SIGNM(NUMEL),EPSI(NUMEL,6),CU(NEQ)
      DIMENSION B(NL),A(NL,NBAND)
      DIMENSION UX(8),UY(8),F(4)
      KIK=0
      JCK=1
      ICHK=1
      ICHECK=0
      IF(III.GT.0) GO TO 350
      JCK=0
      KCHECK=1
      KKK=0
      SRR=1.
      SR1=1.0
      WRITE(6,2005) LLL
      GO TO 600
  100 CONTINUE
      NKT=0
      KKK=KKK+1
      SR=1.0
      DO 200 M=1,NUMEL
  200 IF(RATIO(M).LT.SR) SR=RATIO(M)
      SR3=SR
      TOLL=0.
      DO 210 M=1,NUMEL
      IF(MTAG(M).NE.1) GO TO 210
      MTYPE=IX(M,5)
      TOLJ=0.005
      TOLK=ABS(RATIO(M)-SR)
      IF(TOLK.GT.TOLJ) GO TO 210
      IF(TOLK.GT.TOLL) TOLL=TOLK
  210 CONTINUE
      IF(SR3.LT.1.) SR3=SR+TOLL
      IF(SR3.GT.1.) SR3=1.0
      DO 220 M=1,NUMEL
  220 IF( ABS(RATIO(M)-SR).LE.TOLJ ) RATIO(M)=SR3
      SR=SR3
      IF(SR3.NE.1.0) GO TO 240
```

```
      KCHECK=0
      SR=SR*SR1
      GO TO 250
C
C     INTERPOLATE OR EXTRAPOLATE SCALING FACTOR
C
  240 CALL       INTER (IX,MTAG,JNT,RATIO,BETA,SIGI,SIGNM,EPSI,CU,B,A)
  250 SR1=SR
      DO 280 M=1,NUMEL
      IF(MTAG(M).GT.1) GO TO 280
      IF(MTAG(M).EQ.1) NKT=NKT+1
      IF(RATIO(M).EQ.SR3) GO TO 280
      IF(MTAG(M).EQ.1) NKT=NKT-1
      RATIO(M)=1.0
      BETA(M)=0.
      JNT(M)=0
      MTAG(M)=0
  280 CONTINUE
      IF(KCHECK.NE.0) GO TO 345
      KKK=KKK-1
      SR=SR*(1.-TOTAL)
      KIK=0
      IF((KKK.EQ.0).AND.(SR.EQ.1.)) KIK=1
      WRITE(6,2006)
      WRITE(6,2002) LLL,SR,KKK
      GO TO 350
  345 WRITE(6,2003) LLL,KKK,SR
      IF(SR.GE.0.0001) GO TO 350
      WRITE(6,2004) SR
      SR=0.
      RETURN
  350 SRR=SR1
      IF(III.GT.0) SRR=1.0
      GO TO 600
  352 IF((ICHK.EQ.0).AND.(KCHECK.EQ.1)) GO TO 100
C     *****************************************************************
C     * SCALE DOWN DISPLACEMENTS                                      *
C     *****************************************************************
      IF((III.EQ.0).AND.(KCHECK.EQ.0)) GO TO 357
      GO TO 358
  357 TOTAL=TOTAL+SR
      IF((TOTAL.GE..999).OR.(SR.EQ.0.0)) TOTAL=1.0
  358 CONTINUE
      IF((III.GT.0).AND.(ICHECK.EQ.0)) GO TO 550
      IF(KIK.EQ.1) GO TO 359
      WRITE(6,2006)
      WRITE(6,2000)
  359 DO 500 M=1,NUMNP
      N1=2*M-1
      N2=N1+1
```

```
      BX=SRR*B(N1)
      BY=SRR*B(N2)
      IF(ICHECK.EQ.1) GO TO 360
      BXT=BX+CU(N1)
      BYT=BY+CU(N2)
      GO TO 370
  360 CU(N1)=CU(N1)+BX
      CU(N2)=CU(N2)+BY
      BXT=CU(N1)
      BYT=CU(N2)
  370 IF(KIK.EQ.1) GO TO 500
      WRITE(6,1000) M,BX,BY,BXT,BYT
  500 CONTINUE
      KIK=0
  550 CONTINUE
      ICHK=0
      IF(III.NE.0) GO TO 900
      IF(KCHECK.EQ.0) GO TO 900
      GO TO 100
C     ****************************************************************
C     * SCALE DOWN STRESSES                                          *
C     ****************************************************************
  600 CONTINUE
      SM=0.
      JCHECK=0
      KCK=0
      IF((KKK.GT.0).AND.(KCHECK.EQ.1)) KCK=1
      IF((KIK.EQ.1).OR.(KCK.EQ.1)) GO TO 605
      WRITE(6,2001)
  605 CONTINUE
      DO 800 M=1,NUMEL
      ID=M
C     ****************************************************************
C     * READ ELEMENT INFORMATION FROM FILE 2                         *
C     ****************************************************************
      READ(2'ID)  ((C(IIK,JJK),JJK=1,3),EE(IIK),IIK=1,3)
     1 ,((S(JJI,KKI),KKI=1,8),JJI=1,8),((ST(IKK,JKK),JKK=1,8),IKK=1,3)
     2 ,(RR(JII),ZZ(JII),JII=1,4),XC,YC,TEMP,VOL,MTYPE,N
      DO 640 I=1,3
  640 SIG(I)=SRR*SIGI(M,I+3)+SIGI(M,I)
      IF(IX(M,3).NE.IX(M,2)) GO TO 650
      DX=RR(2)-RR(1)
      DY=ZZ(2)-ZZ(1)
      SIG(6)=90.
      IF(DX.NE.0.) SIG(6)=57.296*ATAN2(DY,DX)
      SIG(4)=SIG(1)
      SIG(5)=0.
      GO TO 765
  650 CC=(SIG(1)+SIG(2))/2.0
      BB=(SIG(1)-SIG(2))/2.0
```

```
      CR= SQRT(BB**2+SIG(3)**2)
      SIG(4)=CC+CR
      SIG(5)=CC-CR
      SIG(6)=0.0
      IF((BB.EQ.0.).AND.(SIG(3).EQ.0.)) GO TO 660
      SIG(6)=28.648* ATAN2(SIG(3),BB)
  660 MTYPE=IX(M,5)
      IF(MTC(MTYPE).EQ.0) GO TO 765
      IF(III.EQ.0) GO TO 755
      IF((MTAG(M).EQ.3).AND.(SIG(4).GT.    TENS(MTYPE)))  MTAG(M)=2
      IF(MTAG(M).EQ.2) JCHECK=1
      IF(MTAG(M).GT.2) GO TO 765
      CALL       GRIFTH (MTAG,JNT,RATIO,BETA,SIG1,SIGNM)
      GO TO 765
  755 CONTINUE
      IF(KCHECK.EQ.1) GO TO 757
      IF((MTAG(M).EQ.3).AND.(SIG(4).GT.    TENS(MTYPE)))  MTAG(M)=2
      IF((MTAG(M).EQ.1).OR.(MTAG(M).EQ.2)) JCHECK=1
      IF(MTAG(M).GT.2) GO TO 765
  757 CALL       GRIFTH (MTAG,JNT,RATIO,BETA,SIG1,SIGNM)
  765 SN=0.
      IF(KCHECK.EQ.1) GO TO 770
      IF((MTAG(M).EQ.0).OR.(MTAG(M).GT.2)) GO TO 770
      SN=SIGNM(M)/TENS(MTYPE)
      IF(SIGNM(M).GT.SM) SM=SIGNM(M)
  770 IF(KCK.EQ.1) GO TO 800
      IF((SN.GE.1.).OR.(RATIO(M).GT.1.)) RATIO(M)=1.
      IF(KIK.EQ.1) GO TO 800
      ANG=57.296*BETA(M)
      WRITE(6,1001) M,XC,YC,(SIG(I),I=1,6),MTAG(M),JNT(M),ANG,RATIO(M)
     *,SN
  800 CONTINUE
      IF(III.NE.0) GO TO 830
      IF(KCHECK.NE.0) GO TO 828
      IF(JJJ.NE.0) GO TO 828
C
C     ******************************************************************
C     * CHECK CRACK CLOSURE WHEN III=0,KCHECK=0 & JJJ=0  .             *
C     ******************************************************************
C
      WRITE(6,2007)
      DO 827 N=1,NUMEL
      IF(MTAG(N).LT.3) GO TO 827
      IF(IX(N,2).EQ.IX(N,3)) GO TO 827
      ID=N
      MM=4
      IF(IX(N,1).EQ.IX(N,4)) MM=3
      FIND(2'ID)
      KTAG=0
      DO 802 I=1,MM
```

```
      UX(I)=0.
  802 UY(I)=0.
      READ(2'ID)  ((C(IIK,JJK),JJK=1,3),EE(IIK),IIK=1,3)
     1 ,((S(JJI,KKI),KKI=1,8),JJI=1,8),((SI(IKK,JKK),JKK=1,8),IKK=1,3)
     2 ,(RR(JII),ZZ(JII),JII=1,4),XC,YC,TEMP,VOL,MTYPE,N
      ICK=0
      EPT=EPST(MTYPE)
C
C     DECOMPOSE JOINT OPENING INTO X AND Y COMPONENTS
C
      EPSX=1.5708+BETA(N)
      IF(EPSX.GE.1.5708) EPSX=1.56
  804 SLOPE=TAN(EPSX)
      DO 806 I=1,MM
  806 F(I)=SLOPE*(RR(I)-XC)-(ZZ(I)-YC)
      EPSX=BETA(N)
      IF(BETA(N).LT.0.) EPSX=3.1416+EPSX
      IF(KTAG.EQ.1) EPSX=EPSX-1.5708
      DUX=EPT*COS(EPSX)
      DUY=EPT*SIN(EPSX)
      DO 808 I=1,MM
      IF(F(I).LT.0.) GO TO 808
      UX(I)=UX(I)+DUX
      UY(I)=UY(I)+DUY
  808 CONTINUE
      IF(KTAG.EQ.1) GO TO 810
      IF(MTAG(N).EQ.3) GO TO 812
      KTAG=1
      EPSX=BETA(N)
      GO TO 804
  810 EPSX=EPSX+1.5708
  812 CONTINUE
C
C     EVALUATE EQUIVALENT STRAINS
C
      DO 816 I=1,3
      RR(I)=0.0
      DO 816 J=1,MM
      K=2*J
  816 RR(I)=RR(I)+ST(I,K)*UY(J)+ST(I,K-1)*UX(J)
  817 CONTINUE
      C1=COS(EPSX)
      S1=SIN(EPSX)
      SC=C1*S1
      C1=C1*C1
      S1=S1*S1
      IF(ICK.EQ.1) GO TO 818
      DP=SC*RR(3)
      EPT1=ABS(C1*RR(1)+S1*RR(2)+DP)
      EPT2=ABS(S1*RR(1)+C1*RR(2)-DP)
```

```
      IF(BETA(N).GT.0.) GO TO 818
      EPSX=BETA(N)
      ICK=1
      GO TO 817
C
C     EVALUATE ELEMENT PRINCIPAL STRAINS, CHECK CLOSURE POSSIBILITY
C
  818 EPS(1)=SRR*EPSI(N,4)+EPSI(N,1)
      EPS(2)=SRR*EPSI(N,5)+EPSI(N,2)
      EPS(3)=SRR*EPSI(N,6)+EPSI(N,3)
      DP=SC*EPS(3)
      EPS1=C1*EPS(1)+S1*EPS(2)+DP
      EPS2=S1*EPS(1)+C1*EPS(2)-DP
      C1=EPT1+EPS1
      S1=EPT2+EPS2
      IF(C1.GT.0.) GO TO 824
      IF(S1.GT.0.) GO TO 823
      MTAG(N)=0
      GO TO 826
  823 MTAG(N)=MTAG(N)-3
      IF(MTAG(N).EQ.0) GO TO 826
      BETA(N)=BETA(N)+1.5708
      IF(BETA(N).GT.1.5708) BETA(N)=BETA(N)-3.1416
      GO TO 826
  824 IF(S1.GT.0.) GO TO 827
      IF(MTAG(N).EQ.3) GO TO 827
      MTAG(N)=1
  826 JCHECK=1
      NTAG(N)=1
      ANG=57.296*BETA(N)
      WRITE(6,2015)
      WRITE(6,2020) N,MTAG(N),ANG,EPT,EPT1,EPT2,EPS1,EPS2,C1,S1
  827 CONTINUE
  828 IF((DTAL..G.1.) ICHECK=1
      IF((KCHECK.EQ.0).AND.(NKT.LE.0)) ICHECK=1
      GO TO 840
C
C     SEARCH FOR MAX. SIGNM
C
  830 DO 835 N=1,NUMEL
      IF( MTAG(N).GT.2) GO TO 835
      DN=ABS(SIGNM(N)-SM)
      IF(DN.LE.TOL) GO TO 835
      IF(MTAG(N).NE.2) GO TO 834
      MTAG(N)=3
      DA=1.5708
      IF(BETA(N).GT.0.) DA=-DA
      BETA(N)=BETA(N)+DA
      GO TO 835
  834 MTAG(N)=0
```

```
      JNT(N)=0
      BETA(N)=0.
      RATIO(N)=1.
  835 ICHECK=1
  840 IF(JCHECK.EQ.1) ICHECK=0
      IF(ICHECK.EQ.1) GO TO 850
      GO TO 352
  850 DO 860 M=1,NUMEL
      DO 860 I=1,3
      SIGI(M,I)=SRR*SIGI(M,I+3)+SIGI(M,I)
      EPSI(M,I)=SRR*EPSI(M,I+3)+EPSI(M,I)
  860 CONTINUE
      GO TO 352
  900 CONTINUE
      WRITE(6,2010) TOTAL
      RETURN
 1000 FORMAT(I12,4E20.7)
 1001 FORMAT(I7,2F8.2,1P5E12.4,0P1F7.2,2I5,F7.2,1X,F7.4,F7.2)
 2000 FORMAT(12HON.P.NUMBER ,17X,3HDUX,17X,3HDUY,18X,2HUX,18X,2HUY)
 2001 FORMAT (7H0EL.NO.  7X 1HX  7X 1HY 4X 8HX-STRESS 4X 8HY-STRESS 3X
     1 9HXY-STRESS,2X,10HMAX-STRESS,2X,10HMIN-STRESS,7H  ANGLE
     2,1X,4HMTAG,2X,3HJNT,3X,4HBETA,2X,5HRATIO,2X,5HSIGNM)
 2002 FORMAT(/ ' LOAD INCREMENT NO.',I5,/ ' LOAD APPLIED AS A FRACTION O
     *F TOTAL LOAD =',F9.5,/' NUMBER OF BETA STABILITY ITERATIONS=',I5/)
 2003 FORMAT(1H0,' LOAD INCREMENT NO.',I5,'   STABILITY ITERATION NO.',
     * I3,'   SCALING FACTOR =',G10.4)
 2004 FORMAT(//' SR=',G10.4,'  CALL EXIT')
 2005 FORMAT(1H1,' INITIAL STATE FOR LOAD INCREMENT',I5)
 2006 FORMAT(1H1)
 2007 FORMAT(//  ' CHECK CRACK OPENING STRAINS ')
 2010 FORMAT(1H0,' LOAD ACCUMULATED AS A FRACTION OF THE TOTAL IS',G15.5
     *//)
 2015 FORMAT(1H0,' N  TAG',8X,'BETA',8X,' EPT',8X,'EPT1',8X,'EPT2',8X,
     *'EPS1',8X,'EPS2',7X,'DEPS1',7X,'DEPS2')
 2020 FORMAT(I3,I5,F12.2,6E12.4)
 2025 FORMAT(I8,2F10.4,3E12.5)
      END
```

```
      SUBROUTINE INTER (IX,MTAG,JNT,RATIO,BETA,SIGI,SIGNM)
C
      COMMON/ONE/ NUMNP,NUMEL,NUMMAT,NUMPC,NPC,MBAND,NUMBLK,NL,MTYPE,N,
     .   VOL,ACELR,ACELZ,Q,HED(18),STOP,SR,SR1,TOTAL,TOL,TOLI,XC,YC,
     .   TEMP,SIGN,SIGDI,SIGDJ,
     .   LLL,III,JJJ,JCK,KCHECK,JCHECK,NPLANE,NUMFR,NCODE,JA,NEQ
     .   ,NBAND,NCRACK,NSTEP,N15,M7,NTOT
      COMMON/TWO/ C(3,3),S(10,10),SIG(6),P(8),ST(3,10),RR(5),ZZ(5),
     .   LM(4),LE(3),EPS(3)
      COMMON/THREE/  E(6,4,8),EO(4,8),TENS(8),XNU(8),RO(8),EPST(8),
     .   MTC(8),NIC(8),MID(8)
      DIMENSION IX(NUMEL,5),MTAG(NUMEL),JNT(NUMEL),RATIO(NUMEL),
     .   BETA(NUMEL),SIGI(NUMEL,6),SIGNM(NUMEL)
      JCK=1
      SRR=SR1
      IJJ=0
      DO 200 N=1,NUMEL
      IF(RATIO(N).NE.SR) GO TO 200
      SR3=RATIO(N)
      BETAI=BETA(N)
      MTAGI=MTAG(N)
      JNTI=JNT(N)
      SR=SR*SR1
      MTYPE=IX(N,5)
      SIGT=TENS(MTYPE)
      IF(IX(N,2).EQ.IX(N,3)) GO TO 200
   50 CONTINUE
      DO 100 I=1,3
  100 SIG(I)=SRR*SIGI(N,I+3)+SIGI(N,I)
      CC=(SIG(1)+SIG(2))/2.0
      BB=(SIG(1)-SIG(2))/2.0
      CR= SQRT(BB**2+SIG(3)**2)
      SIG(4)=CC+CR
      SIG(5)=CC-CR
      SIG(6)=0.0
      IF((BB.EQ.0.).AND.(SIG(3).EQ.0.)) GO TO 150
      SIG(6)=28.648* ATAN2(SIG(3),BB)
  150 CONTINUE
      CALL       GRIFTH (MTAG,JNT,RATIO,BETA,SIGI,SIGNM)
      IF(IJJ.GT.0) GO TO 300
      SIGDI=SIGN-SIGT
      SRR=SR
      IF(SRR.LT.TOLI) SRR=TOLI
      IJJ=1
      GO TO 50
  200 CONTINUE
  300 CONTINUE
      IF (IX(N,2).EQ.IX(N,3)) GO TO 400
      RATIO(N)=SR3
      BETA(N)=BETAI
```

```
      MTAG(N)=MTAGI
      JNI(N)=JNTI
      SIGDJ=SIGN-SIGT
      IF(( ABS(SIGDI).LE.TOL  ).OR.( ABS(SIGDJ).LE.TOL  )) GO TO 350
      SR=(SR1*SIGDJ-SR*SIGDI)/(SIGDJ-SIGDI)
      GO TO 400
  350 CONTINUE
      IF( ABS(SIGDI).GT.TOL  ) GO TO 370
      SR=SR1
      GO TO 390
  370 IF( ABS(SIGDJ).GT.TOL  ) GO TO 400
      SR=SRR
  390 CONTINUE
  400 CONTINUE
      IF(SR.GE.0.99 ) SR=1.0
      IF(SR.LT.0.001) SR=0.001
      IF(SR.EQ.1.0) KCHECK=0
      TOLJ=0.01
      DA=SR-SRR
      DB=SR-SR1
      IF((ABS(DA).LE.TOLJ).OR.(ABS(DB).LE.TOLJ)) KCHECK=0
      WRITE(6,1000) SR1,SRR,SR
      RETURN
 1000 FORMAT(//' OLD SR=',G14.4,' NEW SR=',G14.4, 10X,' INTERPOLATED SR=
     *',G14.4//)
C
      END
```

```
      SUBROUTINE MODIFY(A,B,NEQ,MBAND,NBAND,N,U)
C
C
      DIMENSION B(NEQ),A(NEQ,NBAND)
      DO 250 M=2,MBAND
      K=N-M+1
      IF(K) 235,235,230
  230 B(K)=B(K)-A(K,M)*U
      A(K,M)=0.0
  235 K=N+M-1
      IF(NEQ-K) 250,240,240
  240 B(K)=B(K)-A(N,M)*U
      A(N,M)=0.0
  250 CONTINUE
      A(N,1)=1.0
      B(N)=U
      RETURN
      END
```

```
      SUBROUTINE BANSOL (B,A,MM,NBAND,NUMBLK,NLL,JA)
C
C
      DIMENSION B(NLL),A(NLL,NBAND)
      NN=NLL/2
      NL=NN+1
      NH=NN+NN
      NB=0
      NBK=1
      FIND(1'1)
      GO TO 150
  100 NB=NB+1
      DO 125 N=1,NN
      NM=NN+N
      B(N)=B(NM)
      B(NM)=0.0
      DO 125 M=1,MM
      A(N,M)=A(NM,M)
  125 A(NM,M)=0.0
      IF (NUMBLK-NB) 150,200,150
  150 READ (1'NBK) (B(N),(A(N,M),M=1,MM),N=NL,NH)
      NNK=NBK
      NBK=NBK+JA
      IF (NB) 200,100,200
  200 DO 300 N=1,NN
      IF (A(N,1)) 225,300,225
  225 B(N)=B(N)/A(N,1)
      DO 275 L=2,MM
      IF (A(N,L)) 230,275,230
  230 C=A(N,L)/A(N,1)
      I=N+L-1
      J=0
      DO 250 K=L,MM
      J=J+1
  250 A(I,J)=A(I,J)-C*A(N,K)
      B(I)=B(I)-A(N,L)*B(N)
      A(N,L)=C
  275 CONTINUE
  300 CONTINUE
      NBK=NNK-JA
      IF(NUMBLK.EQ.NB) GO TO 410
      WRITE (1'NBK) (B(N),(A(N,M),M=2,MM),N=1,NN)
      NBK=NNK+JA
      GO TO 100
  410 DO 450 M=1,NN
      N=NN+1-M
      DO 425 K=2,MM
      L=N+K-1
  425 B(N)=B(N)-A(N,K)*B(L)
      NM=N+NN
```

```
      B(NM)=B(N)
  450 A(NM,NB)=B(N)
      NB=NB-1
      IF(NB.EQ.0) GO TO 500
      READ  (1'NBK)  (B(N),(A(N,M),M=2,MM),N=1,NN)
      NBK=NBK-JA
      GO TO 410
  500 K=0
      DO 600 NB=1,NUMBLK
      DO 600 N=1,NN
      NM=N+NN
      K=K+1
  600 B(K)=A(NM,NB)
      RETURN
C
      END
```