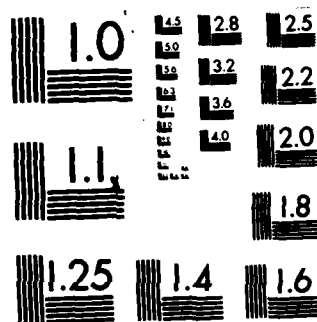




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A139530

ADF 300 393

CONTRACT REPORT ARBRL-CR-00524

FINITE ELEMENT MODEL FOR NONAXISYMMETRIC
STRUCTURE WITH RATE DEPENDENT
YIELD CONDITIONS

Prepared by

University of Illinois
Aeronautical and Astronautical Engineering Dept.
Urbana, Illinois 61801

February 1984



US ARMY ARMAMENT RESEARCH AND DEVELOPMENT CENTER
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.

DTIC FILE COPY

DTIC
ELECTE
MAR 16 1984
E

84 03 15 006

Destroy this report when it is no longer needed.
Do not return it to the originator.

Additional copies of this report may be obtained
from the National Technical Information Service,
U. S. Department of Commerce, Springfield, Virginia
22161.

The findings in this report are not to be construed as
an official Department of the Army position, unless
so designated by other authorized documents.

*The use of trade names or manufacturers' names in this report
does not constitute endorsement of any commercial product.*

UNCLASSIFIED

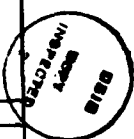
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARBRL-CR-00524	2. GOVT ACCESSION NO. AD-A139530	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FINITE ELEMENT MODEL FOR NONAXISYMMETRIC STRUCTURE WITH RATE DEPENDENT YIELD CONDITIONS		5. TYPE OF REPORT & PERIOD COVERED Final Report
7. AUTHOR(s) ADAM R. ZAK		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Illinois Aeronautical and Astronautical Engineering Dept. Urbana, Illinois		8. CONTRACT OR GRANT NUMBER(s) DAAK11-82-C-0007
11. CONTROLLING OFFICE NAME AND ADDRESS US Army AMCCOM, ARDC Ballistic Research Laboratory, ATTN: DRSMC-BLA-S(A) Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE February 1984
		13. NUMBER OF PAGES 77
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Stress Analysis Finite Element Plasticity Rate-Dependent Yield Stress Anisotropic Yield Conditions		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) jmk → A previously derived model of rate-dependent plasticity is modified for use in a finite element stress analysis code. The code allows nearly axisymmetric bodies to be analyzed by a perturbation of the symmetric solution. It contains anisotropic yield conditions. The formulation of rate effect is added to the code and a test problem of a uniaxial tensile experiment is solved. Convergence to the exact solution is shown. ↵		

TABLE OF CONTENTS

	Page
I. INTRODUCTION.	5
II. RATE DEPENDENT MATERIAL MODEL	6
III. NUMERICAL CALCULATIONS.	12
REFERENCES.	16
APPENDIX A	17
DISTRIBUTION LIST	67

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	



I. INTRODUCTION

The objective of the present investigation is to develop a finite element model and computer program for the purpose of handling elastic-plastic material with rate-dependent yield conditions.¹ The effort is in two parts. The first part involves the analytical development in which the appropriate incremental, stress-strain relations are developed. The second part of the effort involves developing a finite element computer program which incorporates the analytical development. This computer program will be based on previously developed, approximate three-dimensional elastic-plastic computer code, SANX.^{2,3} The code SANX is designed to perform structural analysis on cylindrical configurations which are approximately axisymmetric and which have definite nonaxisymmetric features. The code SANX was developed both for elastic and elastic-plastic materials with no time dependent properties.^{2,3}

The starting point for the development of a three-dimensional finite element code is the one-dimensional analysis¹ which formulates the viscoplastic response in terms of the effective plastic strain. This approach is extended to the development of the three-dimensional model in the present investigation.

-
1. W.H. Drysdale, "A Theory of Rate Dependent Plasticity," Ballistic Research Laboratory Report, APG, MD. (Forthcoming)
 2. A.R. Zak, J.N. Craddock and W.H. Drysdale, "An Elastic-Plastic Analysis of Non-Axisymmetric Structures," International Journal of Computers and Structures, vol. 10, pp. 841-846, 1979.
 3. J.N. Craddock and A.R. Zak, "An Approximate Finite Element Method of Stress Analysis of Non-Axisymmetric Bodies with Elastic-Plastic Materials," Technical Rept. UILU-ENG 79 0501, Aeronautical & Astronautical Engineering Dept., University of Illinois, Urbana, March 1979.

II. RATE DEPENDENT MATERIAL MODEL

The rate dependent plasticity model was introduced for isotropic material and used in the analysis of uniaxial stress case.¹ The same model will be used in the present investigation with a slight modification to allow for use with orthotropic materials^{2,3} when needed. The yield condition for orthotropic materials will be represented by Hill's criterion² and this reduces to the octahedral shear stress criterion in the limit for isotropic materials.¹

Using the rate dependent model for the yield criterion¹ the yield function is taken in the form:

$$f(\sigma_{ij} - \alpha_{ij}) = K(\dot{\epsilon}_{ij}^p) \quad (1)$$

where α_{ij} represents the strain hardening parameters. The rate dependence is defined by the function K which is represented by:

$$K(\dot{\epsilon}_{ij}^p) = [1 + b \ln(1 + \dot{\epsilon}_{ij}^p / \dot{\epsilon}_0)]^2 \quad (2)$$

Equation (2) is an empirical formula which contains material constants b and $\dot{\epsilon}_0$. The dependence of K on the rate is through the variable $\dot{\epsilon}^p$ which will be defined later as the effective plastic strain rate. It may be noted that for isotropic material¹ the function K in equation (2) is multiplied by the square of the uniaxial yield stress at zero rate. In the case of the orthotropic material there are, in general, six yield stresses and it is not possible to separate one stress from the yield condition. The six yield stresses are included in the function f on the left hand side of equation (2).² Using equation (2) to represent rate dependent yield condition for orthotropic materials implies an assumption that each yield stress is dependent on the rate by the same relation to the effective plastic strain rate.

The next step in the development is to obtain an incremental stress-strain relation. From the plastic flow rule the plastic strain changes are related to derivative of yield function²:

$$d\epsilon_{ij}^p = d\lambda \frac{\partial f}{\partial \sigma_{ij}} \quad (3)$$

Using the definition of kinematic strain hardening:

$$d\alpha_{ij} = C d\epsilon_{ij}^p = C d\lambda \frac{\partial f}{\partial \sigma_{ij}} \quad (4)$$

where C is the strain hardening parameter.² During the incremental plastic deformation the stress and strain changes must remain on the yield surface and, therefore, from equation (2):

$$\frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} + \frac{\partial f}{\partial \alpha_{ij}} d\alpha_{ij} - \frac{\partial K}{\partial \dot{\epsilon}_{ij}^p} d\dot{\epsilon}_{ij}^p = 0 \quad (5)$$

It can be shown² that:

$$\frac{\partial f}{\partial \sigma_{ij}} = - \frac{\partial f}{\partial \alpha_{ij}} \quad (6)$$

and, therefore, combining equations (4) and (6) with (5) gives:

$$\frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} - Cd\lambda \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}} - \frac{\partial K}{\partial \epsilon_{ij}^p} d\epsilon_{ij}^p = 0 \quad (7)$$

Solving for the parameter $d\lambda$ from equation (7):

$$d\lambda = \frac{1}{C \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}}} \left[\frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} - \frac{\partial K}{\partial \epsilon_{ij}^p} d\epsilon_{ij}^p \right] \quad (8)$$

Combining equations (3) and (8) and changing the repeated indices in equation (8), for clarity, gives the incremental change in the plastic strain:

$$d\epsilon_{ij}^p = \frac{1}{C \frac{\partial f}{\partial \sigma_{kl}} \frac{\partial f}{\partial \sigma_{kl}}} \left[\frac{\partial f}{\partial \sigma_{mn}} d\sigma_{mn} - \frac{\partial K}{\partial \epsilon_{mn}^p} d\epsilon_{mn}^p \right] \frac{\partial f}{\partial \sigma_{ij}} \quad (9)$$

Consider now the elastic stress-strain relation for incremental changes:

$$d\sigma_{ij} = E_{ijkl}(d\epsilon_{kl} - d\epsilon_{kl}^p) \quad (10)$$

where E_{ijkl} represents elastic material properties² matrix and $d\epsilon_{kl}$ the total strain changes.

Returning to equation (7) and substituting for $d\sigma_{ij}$ from equation (10) and rearranging gives:

$$d\lambda = D \left[\frac{\partial f}{\partial \sigma_{ij}} E_{ijkl} d\epsilon_{kl} - \frac{\partial K}{\partial \epsilon_{ij}^p} d\epsilon_{ij}^p \right] \quad (11)$$

where D is by definition:

$$D = \left[C \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}} + \frac{\partial f}{\partial \sigma_{ij}} E_{ijkl} \frac{\partial f}{\partial \sigma_{kl}} \right]^{-1} \quad (12)$$

Substituting equation (11) into equation (10) gives:

$$d\sigma_{ij} = [E_{ijkl} - DE_{ijrs} E_{mnkl} \frac{\partial f}{\partial \sigma_{mn}} \frac{\partial f}{\partial \sigma_{rs}}] d\epsilon_{kl} + DE_{ijrs} \frac{\partial f}{\partial \sigma_{rs}} \frac{\partial K}{\partial \epsilon_{kl}^p} d\epsilon_{kl}^p \quad (13)$$

Using shorthand notation previously introduced for elastic-plastic materials² permits writing equation (13) in a short form:

$$d\sigma_{ij} = A_{ijkl} d\epsilon_{kl} + DE_{ijrs} \frac{\partial f}{\partial \sigma_{rs}} \frac{\partial K}{\partial \epsilon_{kl}^p} d\epsilon_{kl}^p \quad (14)$$

It may be noted that if the strain rate term is neglected on the right hand side of equation (14), then the remaining terms represent the relation between total incremental stress and strain changes for elastic-plastic material used in SANX model. The next step in the development of the rate dependent model is to introduce the effective plastic strain increment defined by:

$$d\epsilon^p = \sqrt{2/3} d\epsilon_{ij}^p d\epsilon_{ij}^p \quad (15)$$

The objective of this will be to use this concept to replace the strain rate term on the right hand side of equation (14). Using flow rule, equation (3), in equation (15) results in:

$$d\epsilon^p = d\lambda \sqrt{2/3} \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}} \quad (16)$$

Returning to the yield function and the function K, an incremental change in this parameter can now be written as:

$$dK = \frac{\partial K}{\partial \epsilon_{ij}^p} d\epsilon_{ij}^p = \frac{\partial K}{\partial \epsilon^p} \frac{\partial \epsilon^p}{\partial \epsilon_{ij}^p} d\epsilon_{ij}^p \quad (17)$$

But:

$$\frac{\partial \epsilon^p}{\partial \epsilon_{ij}^p} d\epsilon_{ij}^p = d\epsilon^p \quad (18)$$

Therefore:

$$\frac{\partial K}{\partial \epsilon_{ij}^p} d\epsilon_{ij}^p = \frac{\partial K}{\partial \epsilon^p} d\epsilon^p \quad (19)$$

Returning now to equation (7) and substituting for $d\lambda$ from equation (16) and using equation (19) results in:

$$\frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} - C \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}} \cdot \frac{d\epsilon^p}{\sqrt{2/3} \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}}} - \frac{\partial K}{\partial \epsilon^p} d\epsilon^p = 0 \quad (20)$$

The second term in equation (20) can be simplified and equation rearranged as follows:

$$\frac{\partial K}{\partial \epsilon^p} d\epsilon^p + 3/2 C \sqrt{2/3} \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}} d\epsilon^p = \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} \quad (21)$$

Equation (21) can be compared directly to equation (III.5a) of Reference 1, for isotropic materials, with the following substitutions:

$$\frac{\partial K}{\partial \epsilon^p} = 2/3 \sigma_y \frac{\partial \sigma_y}{\partial \epsilon^p} \quad (22)$$

and

$$\sqrt{2/3} \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}} = 2/3 \sigma_y \quad (23)$$

Before proceeding further with equation (21) it is useful to obtain the expression for rate of change of K . Using equation (2) and differentiating:

$$\frac{\partial K}{\partial \epsilon^p} = 2[1 + b \ln(1 + \frac{\epsilon^p}{\epsilon_0})] \frac{b}{\epsilon_0 + \epsilon^p} \quad (24)$$

Consider now equation (21) and apply it to an interval of loading over which the rate of change of stress is represented by a constant.

General equation (21) has variable coefficients. However, if it is applied to a small interval of loading the coefficients can be assumed to be constant over this interval. Over such interval the stress variation can be approximated by a linear variation with time:

$$d\sigma_{ij} = \dot{\sigma}_{ij} dt \quad (25)$$

where $\dot{\sigma}_{ij}$ are constant rates and time t is measured from the beginning of the interval. Equation (21) can now be reduced to an ordinary differential equation over a small time interval. This is done by substituting:

$$d\dot{\epsilon}^P = \frac{d\dot{\epsilon}^P}{dt} dt$$

$$d\epsilon^P = \frac{d\dot{\epsilon}^P}{dt} dt \quad (26)$$

Using equations (25) and (26) in (21) and cancelling dt:

$$\frac{\partial K}{\partial \dot{\epsilon}^P} \ddot{\epsilon}^P + 3/2 C \sqrt{2/3 \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}}} \dot{\epsilon}^P = \frac{\partial f}{\partial \sigma_{ij}} \dot{\sigma}_{ij} \quad (27)$$

The solution to equation (27) can be shown to be of the following form:

$$\epsilon^P = At + k_1 + k_2 e^{-\lambda t} \quad (28)$$

where by definition:

$$A = \frac{\partial f}{\partial \sigma_{ij}} \dot{\sigma}_{ij} / \left(3/2 C \sqrt{2/3 \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}}} \right)$$

$$\lambda = 3/2 C \sqrt{2/3 \frac{\partial f}{\partial \sigma_{ij}} \frac{\partial f}{\partial \sigma_{ij}}} / \frac{\partial K}{\partial \dot{\epsilon}^P} \quad (29)$$

and k_1 and k_2 are unknown constants. Time t is measured from the start of the interval. The constants k_1 and k_2 are evaluated from the conditions at the start of the interval:

at $t = 0$

$$\begin{aligned} \epsilon^P &= \epsilon_0^P \\ \dot{\epsilon}^P &= \dot{\epsilon}_0^P \end{aligned} \quad (30)$$

Using equations (30) to evaluate k_1 and k_2 gives:

$$\begin{aligned} k_1 &= \epsilon_0^P - \frac{1}{\lambda} (A - \dot{\epsilon}_0^P) \\ k_2 &= \frac{1}{\lambda} (A - \dot{\epsilon}_0^P) \end{aligned} \quad (31)$$

Substituting into equation (28) for k_1 and k_2 results:

$$\epsilon^P = At + \epsilon_0^P - \frac{1}{\lambda} (A - \dot{\epsilon}_0^P) (1 - e^{-\lambda t}) \quad (32)$$

Differentiating equation (32) gives the rate of change:

$$\dot{\epsilon}^p = A - (A - \dot{\epsilon}_0^p) e^{-\lambda t} \quad (33)$$

Consider now a time interval $t=0$ to $t=\Delta t$ and use equation (33) to obtain the change of the effective plastic strain rate:

$$\begin{aligned} d\dot{\epsilon}^p &= \dot{\epsilon}^p - \dot{\epsilon}_0^p \\ &= (A - \dot{\epsilon}_0^p)(1 - e^{-\lambda \Delta t}) \end{aligned} \quad (34)$$

Returning to the incremental stress-strain relation and substituting first for the rate term from equation (19) and then expressing $d\dot{\epsilon}^p$ from equation (34) gives the following:

$$\begin{aligned} d\sigma_{ij} &= A_{ijkl} d\epsilon_{kl} \\ &+ DE_{ijrs} \frac{\partial f}{\partial \sigma_{rs}} \frac{\partial K}{\partial \dot{\epsilon}^p} (A - \dot{\epsilon}_0^p)(1 - e^{-\lambda \Delta t}) \end{aligned} \quad (35)$$

Equation (35) is now a suitable incremental relation over a load time step Δt which gives the change in stress in terms of change in total strain and in terms of parameters which can be calculated from previous time step. The second term on the right hand side of equation (35) represents the rate effects and, in the finite element model, it will contribute to the body force.

III. NUMERICAL CALCULATIONS

The analytical development of the previous section has been incorporated into the elastic-plastic version of the SANX computer code.³ The basic arrangement of the SANX code has been retained. The main changes to the program involve changes in the Subroutine ELPLSS which assembles the incremental plastic stress-strain relations. In the new version the incremental stress-strain relations are based on the equation (35). The basic input procedure for the new SANX is the same as the original code except for the first input card. This card has been modified to input the rate dependent material parameters b , $\dot{\epsilon}_0$ defined in equation (2), and the time interval Δt needed for the time dependent incremental solution.

The first input card in the original elastic-plastic SANX was:

Card 1 (Original)

Format (2I10)

Columns	1-10	NTOTS Number of segments (8 maximum)
	11-20	NOLINC Number of load increments

The new input card is:

Card 1 (New)

Format (2I10,3E12.6)

Columns	1-10	NTOTS
	11-20	NOLINC
	21-32	DELTIM Time increment Δt
	33-44	BVR Material parameter b
	45-56	EVR Material parameter $\dot{\epsilon}_0$

In order to check the new program constant stress rate was applied to sample examples which simulates uniaxial loading and the results were compared to those obtained from uniaxial solution of Reference 1. The uniaxial analysis of Reference 1 is in two parts. The first part is an incremental uniaxial solution and the second is an exact solution for constant stress or strain rate loading. The incremental solution of Reference 1 was first compared to the exact solution and it was found that modifications were necessary to the incremental solution to make it agree with the exact formulation. The comparison of the results from the three-dimensional SANX program are made to the modified incremental, uniaxial formulation.

The first comparison was made between the uniaxial incremental solution and the uniaxial exact solution from Reference 1. The purpose of this comparison was to establish the effect on the accuracy of the size of the load steps at various stress rates. Figure 1 shows these results where the incremental and the exact solutions are shown for a wide range of stress rates. The results are presented both in Metric and British units. The Metric units are given in parentheses directly under the corresponding British units. Figure 1 presents results for two load steps of 2×10^5 psi (13.7 GPa) and 8×10^5 psi (55.1 GPa). The time step Δt was adjusted to give the desired stress rate. These results are for the following material parameters:

rate-dependent yield parameters;

$$\begin{aligned} b &= 3.67 \times 10^{-2} \\ \epsilon_0 &= 3.0 \times 10^{-2} \end{aligned}$$

elastic modulus;

$$E = 16.8 \times 10^6 \text{ psi (115.7 GPa)}$$

strain hardening parameter;

$$C = 0.259 \times 10^6$$

yield stress;

$$\sigma_y = 133 \times 10^3 \text{ psi (0.916 GPa)}$$

Comparing solutions in Figure 1 it can be seen that relatively good agreement exists between exact and incremental solutions. This is especially true for the smaller load step of 2×10^5 psi (13.7 GPa). It is expected that smaller load steps will give more accurate results.

The next comparison involved using the new three-dimensional computer program to analyze a uniaxial situation composed of a cylindrical body subject to axial load. The results are compared to the exact uniaxial solution of Reference 1. These results are presented in Figure 2 for the two different load step sizes used in Figure 1. As in Figure 1, the results are presented in two different systems of units. It is expected that the results in Figure 2 should duplicate results in Figure 1 if the three-dimensional code is working properly. It can be seen that the results between uniaxial incremental solution and the finite element program are almost identical.

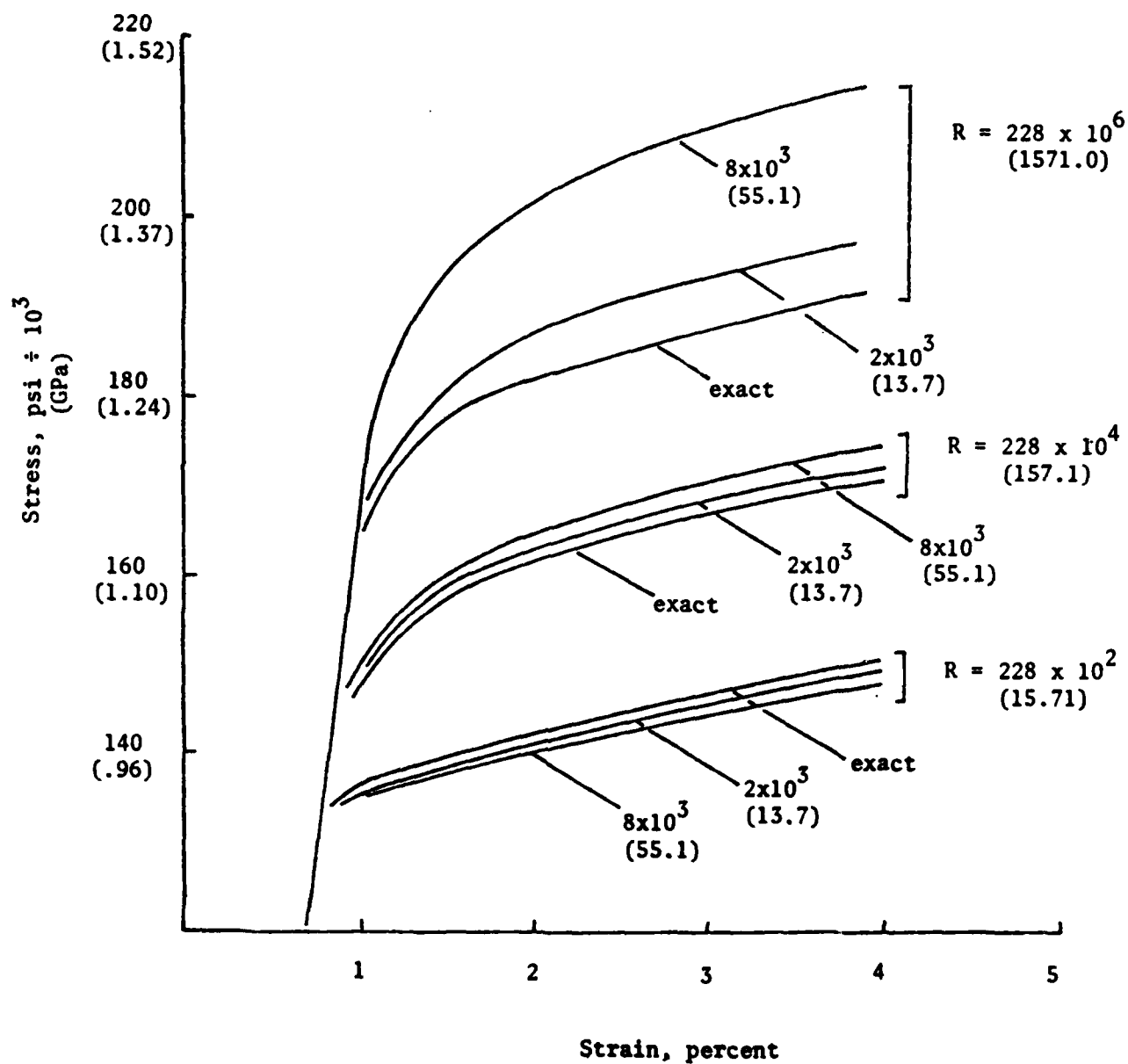


Figure 1. Comparison of uniaxial incremental and exact solutions for different stress rates and different load steps.

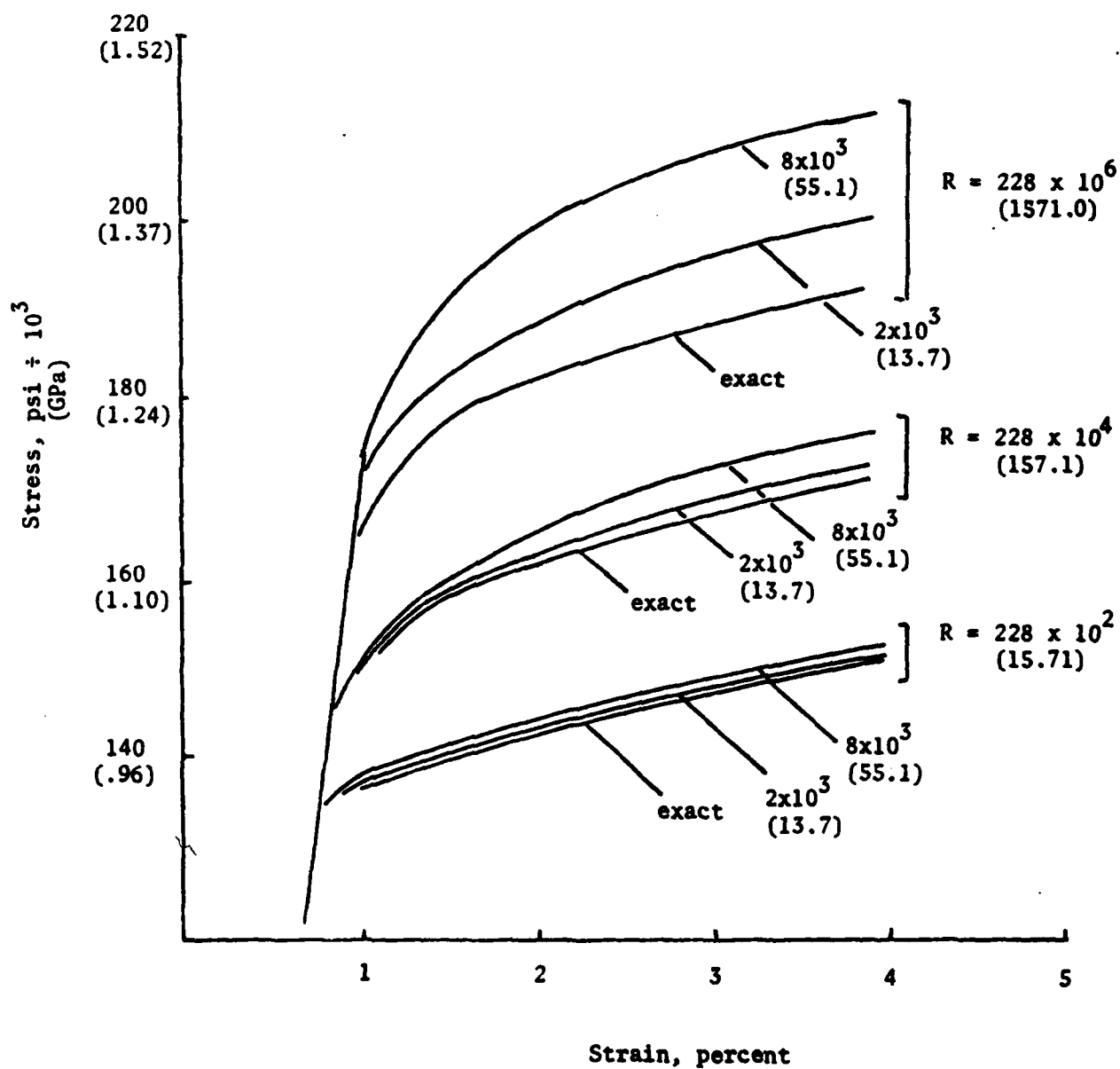


Figure 2. Comparison of three-dimensional incremental solution and exact uniaxial solution for different stress rates and load steps.

REFERENCES

1. W.H. Drysdale, "A Theory of Rate Dependent Plasticity," Ballistic Research Laboratory Report, APG, MD (Forthcoming).
2. A.R. Zak, J.N. Craddock and W.H. Drysdale, "An Elastic-Plastic Analysis of Non-Axisymmetric Structures," International Journal of Computers and Structures, vol. 10, pp. 841-846, 1979.
3. J.N. Craddock and A.R. Zak, "An Approximate Finite Element Method of Stress Analysis of Non-Axisymmetric Bodies with Elastic-Plastic Materials," Technical Rept. UILU-ENG 79 0501, Aeronautical and Astronautical Engineering Dept., University of Illinois, Urbana, March 1979.

A P P E N D I X A

Computer Listing for the Program
SANXVR for the Analysis of Nonaxi-
symmetric Configuration with Rate
Dependent Yield Criterion

```

PROGRAM SANXVR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,
1 TAPE2,TAPE3,TAPE15,
2TAPE21,TAPE25,TAPE26)
C* * * * *
C BRLESC FINITE ELEMENT STRESS ANALYSIS OF AXISYMMETRIC,
C PLANE STRAIN, AND PLANE STRESS SOLIDS WITH ORTHOTROPIC,
C TEMPERATURE-DEPENDENT MATERIAL PROPERTIES
C* * * * *
INTEGER CODE
COMMON/VISC/EPDIN(12,10,8),SIGVP(6),DEPSR(6,10,8),DELTIM
COMMON/RATE/DKPR,SIGPR,BVR,EVR,PSRATE(10,8),NRATE
COMMON/INCR/NOLINC,NOL,INERT,NUMMAT,SIGTOT(12,4,8)
1 ,EPSTOT(12,4,8)
COMMON/PLAS/ALFA(6,4,8),SIGYLD(7,6,8),IFGPL(4,8)
COMMON/BOCON/NRDF,NREQ(18),URES(18)
COMMON/NPDATA/R(10),CODE(10),XR(10),Z(10),XZ(10),
1NPNUM(4,8),T(10),XT(10)
COMMON/ARG/RRR(5),ZZZ(5),RR(4),ZZ(4),S(15,15),P(15),TT(6),
1H(6,15),CRZ(6,6),XI(10),ANGLE(4),SIG(18),EPS(18),N
COMMON/ELDATA/BETA(10),EPR(10),PR(4),SH(4),IX(8,5),
1IP(4),JP(4),IS(4),JS(4),ALPHA(10),IT(4),JT(4),
2ST(4)
COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
1NUMST
COMMON/NXMESH/THETAN(8),NPC(8,8)
COMMON/ANS1/NUMELS(8),NUMNPS(8)
COMMON/NXDATA/NTP,NTS,NTOTS,GTSIG(24,24,8)
COMMON/NOAXI/S1(30,30),P1(30),THETA,BS1(6,30)
COMMON/SOLVE/X(888),Y(888),TEM(888),NUMTC,MBAND
COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
COMMON/CONVRG/IDONE
COMMON/PLANE/NPF
COMMON/RESULT/BS(6,15),D(6,6),C(6,6),AR,BB(6,9),CNS(6,6)
COMMON/MATP/RO(6),E(12,16,6),EE(16),AOFTS(6)
COMMON/NXSOLV/SKG(36,24),FTG(132),FTOT(132),ITOT
DIMENSION TITLE(20)
C* * * * *
C READ AND WRITE CONTROL INFORMATION
C* * * * *
READ(5,3000) NTOTS,NOLINC,DELTIM,BVR,EVR
WRITE(6,3017) BVR,EVR
3017 FORMAT(1H," BVR = ",E12.4," EVR = ",E12.4)
DO 150 I=1,NTOTS
150 READ(5,3001) THETAN(I)
DO 152 I=1,NTOTS
152 READ(5,3002) (NPC(I,J),J=1,8)
3000 FORMAT(2I10,3E12.6)
3001 FORMAT(F10.5,I10)
3002 FORMAT(8I10)
REWIND 15
REWIND 26
REWIND 21
REWIND 25
WRITE(6,3010)
3010 FORMAT("1","SEGMENT DATA FOR NONAXISYMMETRIC PROBLEM")
WRITE(6,3011) NTOTS,NOLINC,DELTIM
3011 FORMAT(" ", " NUMBER OF TOTAL SEGMENTS =",I5,/,
2 " NUMBER OF LOAD INCREMENTS =",I5,/,
3 " TIME INCREMENT =",E15.8)

```

```

      DO 153 I=1,NTOTS
      WRITE(6,3012) I,THETAN(I)
3012 FORMAT(" ",///," SEGMENT TYPE =",I5/,"          THETA = ",F10.5)
      153 CONTINUE
      DO 154 I=1,NTOTS
      154 WRITE(6,3014)I,(NPC(I,J),J=1,8)
3014 FORMAT(" ", "CONNECTING NODES FOR SEGMENT",I5," ARE",8I5)
      DO 910 NOL=1,NOLINC
      WRITE(6,2030) NOL
      REWIND 15
      DO 950 NTP = 1,NTOTS
      THETA= THETAN(NTP)          /57.295780
      IF(NOL.NE.1)GO TO 525
50 READ(5,1000 )TITLE,NNLA,NUMTC,NUMMAT,NUMPC,NUMSC,NUMST,TREF
      1,INERT,NLINC,INCI,INCF,IPLT
      WRITE(6,2000)TITLE,NNLA,NUMTC,NUMMAT,NUMPC,NUMSC,NUMST,TREF,INERT,
      1NLINC
      WRITE(15)NUMTC,NUMMAT,NUMPC,NUMSC,TREF,INERT,INCI,INCF
      NPP=0
C* * * * *
C  GENERATE FINITE ELEMENT MESH
C* * * * *
      100 CALL MESH
      DO 155 I=1,NUMEL
      IFGPL(I,NTP)=0
      PSRATE(I,NTP)=0.0
      DO 155 J=1,12
      SIGTOT(J,I,NTP)=0.0
      EPSTOT(J,I,NTP)=0.0
      ALFA(J,I,NTP)=0.0
      EPSDN(J,I,NTP)=0.0
      155 CONTINUE
      WRITE(15)(R(I),I=1,NUMNP)
      WRITE(15)(Z(I),I=1,NUMNP)
      NUMELS(NTP) = NUMEL
      NUMNPS(NTP) = NUMNP
      IF (IPLT.EQ.1) CALL MPLOT
C* * * * *
C  READ AND WRITE TEMPERATURE DATA
C* * * * *
      103 IF(NUMTC.EQ.0) GO TO 440
      IF(NUMTC.GT.0) READ(5,1001) (X(I),Y(I),TEM(I),I=1,NUMTC)
      IF(NUMTC.EQ.-2) CALL TEM2(NUMNP)
      IF(NUMTC.EQ.-2) GO TO 440
      MPRINT=0
      DO 210 I=1,NUMTC
      IF(MPRINT.NE.0) GO TO 200
      WRITE(6,2001)
      MPRINT=59
      200 MPRINT=MPRINT-1
      210 WRITE(6,2002) X(I),Y(I),TEM(I)
      MPRINT=0
      DO 230 N=1,NUMNP
      IF(MPRINT.NE.0) GO TO 220
      WRITE(6,2003)
      MPRINT=59
      220 MPRINT=MPRINT-1
      CALL TEMP(R(N),Z(N),T(N))
      230 WRITE(6,2004) N,R(N),Z(N),T(N)
      440 MPRINT=0

```

```

DO 460 N=1,NUMEL
IF(MPRINT.NE.0) GO TO 450
WRITE(6,2008)
MPRINT=59
450 MPRINT=MPRINT-1
II=IX(N,1)
JJ=IX(N,2)
KK=IX(N,3)
LL=IX(N,4)

C
C TEM IS TEMPORARY STORAGE FOR ELEMENT TEMPERATURES
C
TEM(N)=(T(II)+T(JJ)+T(KK)+T(LL))/4.00
460 WRITE(6,2009) N,(IX(N,I),I=1,5),BETA(N),ALPHA(N),TEM(N)
WRITE(15)(IX(I,J),J=1,5),I=1,NUMEL)
WRITE(15)(BETA(I),I=1,NUMEL)
WRITE(15)(ALPHA(I),I=1,NUMEL)
WRITE(15)(TEM(I),I=1,NUMEL)
DO 470 K=1,NUMEL
470 T(K)=TEM(K)
C* * * * *
C READ AND WRITE MATERIAL PROPERTIES
C* * * * *
500 CONTINUE
DO 510 M=1,NUMMAT
READ(5,1004) MTYPE,(NT,RO(MTYPE),AOFTS(MTYPE))
WRITE(6,2010) MTYPE,NT,RO(MTYPE)
READ(5,1005)((E(I,J,MTYPE),J=1,14),I=1,NT)
READ(5,1005)(SIGYLD(I,MTYPE,NTP),I=1,7)
IF(AOFTS(MTYPE).NE.1.) WRITE(6,2011)((E(I,J,MTYPE),J=1,13),I=1,NT)
IF(AOFTS(MTYPE).EQ.1.) WRITE(6,2012)((E(I,J,MTYPE),J=1,13),I=1,NT)
WRITE(6,3015)(SIGYLD(I,MTYPE,NTP),I=1,7)
3015 FORMAT(1H,"YIELD STRESSES ARE :",/
11H,"Y11 =",E15.7/
21H,"Y22 =",E15.7/
31H,"Y33 =",E15.7/
41H,"Y12 =",E15.7/
51H,"Y13 =",E15.7/
61H,"Y23 =",E15.7/
71H," C =",E15.7)
WRITE(15)MTYPE,NT, RO(MTYPE)
WRITE(15)((E(I,J,MTYPE),J=1,14),I=1,NT)
DO 510 I=NT,12
DO 510 J=1,16
510 E(I,J,MTYPE)=E(NT,J,MTYPE)
GO TO 526
525 CALL DATA
526 CONTINUE
DO 900 NL=1,NLINC
ACELZ=0.00
ANGVEL=0.00
ANGACC=0.00
IF(INERT.EQ.0) GO TO 511
IF(NL.NE.1.AND.INCI.EQ.0) GO TO 511
C*****
C READ AND WRITE DYNAMIC FORCES
C*****
READ(5,1030) ACELZ, ANGVEL, ANGACC
WRITE(6,2031) ACELZ, ANGVEL, ANGACC
511 CONTINUE

```

```

C * * * * * READ AND WRITE PRESSURE AND SHEAR BOUNDARY CONDITIONS
C * * * * * IF(NL.NE.1.AND.INCF.EQ.0) GO TO 700
600 IF(NUMPC.EQ.0) GO TO 630
    MPRINT=0
    DO 620 L=1,NUMPC
        IF(MPRINT.NE.0) GO TO 610
        WRITE(6,2013)
        MPRINT=58
610 MPRINT=MPRINT-1
        READ(5,1006) IP(L),JP(L),PR(L)
620 WRITE(6,2014) IP(L),JP(L),PR(L)
630 IF(NUMSC.EQ.0) GO TO 701
    MPRINT=0
    DO 650 L=1,NUMSC
        IF(MPRINT.NE.0) GO TO 640
        WRITE(6,2015)
        MPRINT=58
640 MPRINT=MPRINT-1
        READ(5,1006) IS(L),JS(L),SH(L)
650 WRITE(6,2014) IS(L),JS(L),SH(L)
701 IF(NUMST.EQ.0) GO TO 700
    MPRINT=0
    DO 680 L=1,NUMST
        IF(MPRINT.NE.0) GO TO 670
        WRITE(6,2025)
        MPRINT=58
670 MPRINT=MPRINT-1
        READ(5,1006) IT(L),JT(L),ST(L)
680 WRITE(6,2014) IT(L),JT(L),ST(L)
C * * * * *
C DETERMINE BANDWIDTH, INITIALIZE ELASTIC-PLASTIC RATIO,
C AND CONVERT BETA FROM DEGREES TO RADIANS
C * * * * *
700 J=0
    DO 710 N=1,NUMEL
        IX(N,5)=IABS(IX(N,5))
        DO 710 I=1,4
            DO 710 L=1,4
                KK=IABS(IX(N,I)-IX(N,L))
                IF(KK.GE.J) J=KK
710 CONTINUE
        MBAND=3*J+3
        IF(NL.GT.1) GO TO 721
        DO 720 N=1,NUMEL
            EPR(N)=1.
            ALPHA(N)=ALPHA(N)/57.295780
720 BETA(N)=BETA(N)/57.295780
721 CONTINUE
C * * * * *
C SOLVE NONLINEAR PROBLEM BY SUCCESSIVE APPROXIMATIONS
C * * * * *
DO 800 NNN=1,NNLA
C
C FORM STIFFNESS MATRIX
C
C CALL STIFF
C
C SOLVE FOR DISPLACEMENTS

```

```

C      CALL SOLV
C      COMPUTE STRESSES
C      CALL STRESS
C      CALL STORE
      IF( IDONE.NE.1 ) GO TO 800
799  NITER=NNN
      IF( IDONE.EQ.1 ) GO TO 810
800  CONTINUE
810  IF( IDONE.EQ.1 ) WRITE(6,2016) NITER
      IF( IDONE.NE.1 ) WRITE(6,2017) NITER
900  CONTINUE
950  CONTINUE
      ITOT=24+12*(NTOTS-1)
                                           NEWDYN

      IF( NDL.NE.1 ) GO TO 88
                                           NEWDYN

C*****
C      INITIALIZE PREVIOUS HISTORY TOTAL DISPLACEMENTS
                                           NEWDYN
C*****
      DO 89 I=1,ITOT
                                           NEWDYN
          FTOT(I)=0.00
                                           NEWDYN
89    CONTINUE
                                           NEWDYN
88    CONTINUE
                                           NEWDYN

      CALL ASEMBL
      CALL ANSWER
910  CONTINUE
1000 FORMAT(20A4/6I5,F5.0,5I5)
1001 FORMAT(3F10.0)
1004 FORMAT(2I5,2F10.0)
1005 FORMAT(7F10.0)
1006 FORMAT(2I5,F10.0)
1030 FORMAT(3F10.0)
2000 FORMAT(2H1,20A4/
1 33H0  NUMBER OF APPROXIMATIONS-----I4/
2 33H0  NUMBER OF TEMPERATURE CARDS---I4/
3 33H0  NUMBER OF MATERIALS-----I4/
4 33H0  NUMBER OF PRESSURE CARDS-----I4/
5 33H0  NUMBER OF SHEAR CARDS-----I4/
6 33H0  NUMBER OF TORSION CARDS-----I4/
7 33H0  REFERENCE TEMPERATURE-----E12.4/
8 33H0  NUMBER OF INERTIA CARDS-----I4/
9 33H0  NUMBER OF LOAD INCREMENTS-----I4/)
2001 FORMAT(1H1,13X,1HR,14X,1HZ,14X,1HT)
2002 FORMAT(3F15.3)
2003 FORMAT(35H1  N      R      Z      T)
2004 FORMAT(15,2F10.4,F10.0)
2008 FORMAT(74H1 EL  I  J  K  L  MATERIAL  ANGLE BETA  ANGLE A
1LPHA  TEMPERATURE)
2009 FORMAT(15,4I4,I8,F11.1,2F13.3)
2010 FORMAT(1H1,"MATERIAL IDENTIFICATION NUMBER =",I2/

```

```

11H ,"NO. OF MATERIAL TEMPERATURE CARDS =",I2/
21H ,"MASS DENSITY =",E15.7)
2011 FORMAT (1H ,"TEMPERATURE =",E15.7/
11H ,"MODULUS OF ELASTICITY-EN =",E15.7/
21H ,"MODULUS OF ELASTICITY-ES =",E15.7/
31H ,"MODULUS OF ELASTICITY-ET =",E15.7/
41H ,"POISSON RATIO-NUNS =",E15.7/
51H ,"POISSON RATIO-NUNT =",E15.7/
61H ,"POISSON RATIO-NUST =",E15.7/
71H ,"SHEAR MODULUS-GNS =",E15.7/
81H ,"SHEAR MODULUS-GST =",E15.7/
91H ,"SHEAR MODULUS-GTN =",E15.7/
11H ,"COEFFICIENT OF THERMAL EXPANSION-AN =",E15.7/
21H ,"COEFFICIENT OF THERMAL EXPANSION-AS =",E15.7/
31H ,"COEFFICIENT OF THERMAL EXPANSION-AT =",E15.7/)
2012 FORMAT (1H ,"TEMPERATURE =",E15.7/
11H ,"MODULUS OF ELASTICITY-EN =",E15.7/
21H ,"MODULUS OF ELASTICITY-ES =",E15.7/
31H ,"MODULUS OF ELASTICITY-ET =",E15.7/
41H ,"POISSON RATIO-NUNS =",E15.7/
51H ,"POISSON RATIO-NUNT =",E15.7/
61H ,"POISSON RATIO-NUST =",E15.7/
71H ,"SHEAR MODULUS-GNS =",E15.7/
81H ,"SHEAR MODULUS-GST =",E15.7/
91H ,"SHEAR MODULUS-GTN =",E15.7/
11H ,"FREE THERMAL STRAIN-FN =",E15.7/
21H ,"FREE THERMAL STRAIN-FS =",E15.7/
31H ,"FREE THERMAL STRAIN-FT =",E15.7/)
2013 FORMAT (30H PRESSURE BOUNDARY CONDITIONS/20H      I      J PRESSURE)
2014 FORMAT (2I5,F10.1)
2015 FORMAT (27H SHEAR BOUNDARY CONDITIONS/17H      I      J SHEAR)
2016 FORMAT (26H THE SYSTEM CONVERGED IN I2,I1H ITERATIONS)
2017 FORMAT (33H THE SYSTEM DID NOT CONVERGE IN I2,I1H ITERATIONS)
2024 FORMAT (43H0 THE AXISYMMETRIC OPTION HAS BEEN SELECTED)
2025 FORMAT(30H TORSION BOUNDARY CONDITIONS/17H      I      J SHEAR)
2030 FORMAT(1H ,45X,"***** LOAD STEP ***** =",I4)
2031 FORMAT(1H0 ,"AXIAL ACCELERATION =",E12.4/
11H0 ,"ANGULAR VELOCITY      =",E12.4/
21H0 ,"ANGULAR ACCELERATION=",E12.4)
920 STOP
END
SUBROUTINE ANGLE (R,Z,RC,ZC,ANG)
C FIND ANGLE OF INCLINATION BETWEEN 0 AND 2*PI
C* * * * *
PI=3.1415927
D1=(Z-ZC)
D2=(R-RC)
IF(ABS(R-RC).GT.1.E-8) GO TO 100
ANG=PI/2.
IF(D1.GT.1.E-8) RETURN
ANG=-ANG
RETURN
C* * * * *
C ALLOW CIRCLE TO CROSS AXIS
C* * * * *
100 ANG=ATAN2(D1,D2)
RETURN
END
SUBROUTINE ANSWER
INTEGER CODE

```

```

COMMON/VISC/EPSON(12,10,8),SIGVP(6),DEPSR(6,10,8),DELTIM
COMMON/RESULT/BS(6,15),D(6,6),C(6,6),AR,BB(6,9),CNS(6,6)
COMMON/PLAS/ALFA(6,4,8),SIGYLD(7,6,8),IFGPL(4,8)
COMMON/INCR/NOLINC,NOL,INERT,NUMMAT,SIGTOT(12,4,8)
1, EPSTOT(12,4,8)
COMMON/ELDATA/BETA(10),EPR(10),PR(4),SH(4),IX(8,5),
1IP(4),JP(4),IS(4),JS(4),ALPHA(10),IT(4),JT(4),
2ST(4)
COMMON/ARG/RRR(5),ZZZ(5),RR(4),ZZ(4),S(15,15),P(15),TT(6),
1H(6,15),CRZ(6,6),XI(10),ANGLE(4),SIG(18),EPS(18),N
COMMON/NXSOLV/SKG(36,24),FTG(132),FTOT(132),ITOT
COMMON/ANS2/UT1(24),G(24,24),GR1(24,24),DUMM(24,24)
COMMON/ANS1/NUMELS(8),NUMNPS(8)
COMMON/NDNAXI/S1(30,30),P1(30),THETA,BS1(6,30)
COMMON/NXDATA/NTP,NTS,NTOTS,GTS1G(24,24,8)
COMMON/NXMESH/THETAN(8),NPC(8,8)
COMMON/ARG1/SIG1(18),EPS1(18),DEPSP(12),CEPSP(6,6)
COMMON/SOLVE/B(72),A(72,36),NUMBLK,MBAND
DIMENSION UT(24),UC1(24),UC(24),R1(24,24)
REWIND 25
REWIND 26
REWIND 21
KOLD=1
DO 100 K=1,NTOTS
NS=K
KNEW=K
NUMNP = NUMNPS(K)
NUMNP3 = 3*NUMNP
NUMEL = NUMELS(K)
K20 = 21
READ(26)(B(I),I=1,NUMNP3)
READ(26)((IX(I,J),J=1,5),I=1,NUMEL)
WRITE(6,1200) K
READ(25)((R1(I,J),J=1,24),I=1,24)
DO 110 KK=1,4
NP1 = NPC(NS,KK)
NP2 = NPC(NS,KK+4)
DO 110 I=1,3
UC(3*(KK-1)+I) = B(3*NP1-3+I)
UC(3*(KK-1)+I+12) = B(3*NP2-3+I)
110 CONTINUE
DO 115 KK=1,24
115 UT(KK) = FTG(KK+(NS-1)*12)
WRITE(6,900)
900 FORMAT(" ", " EL SIGMAR SIGMAZ SIGMAC SIGMARZ SIGMAZC"
1 " SIGMACR SIGMAN SIGMAS SIGMAT SIGMANS",
2 " SIGMATN", "/" EPSR EPSZ EPSC EPSRZ ",
3 "EPSZC EPSCR EPSN EPSS EPST EPSNS ",
4 " EPSST EPSTN")
IF(KOLD.EQ.KNEW) REWIND 21
IF(KOLD.NE.KNEW) KOLD=KNEW
DO 120 N=1,NUMEL
MYPE=IABS(IX(N,5))
READ(K20)((CRZ(I,J),J=1,6),I=1,6)
READ(K20)((BS1(I,J),J=1,30),I=1,6)
READ(K20)((G(I,J),J=1,24),I=1,24)
READ(K20)((CEPSP(I,J),J=1,6),I=1,6)
READ(K20)((CNS(I,J),J=1,6),I=1,6)
READ(K20)((D(I,J),J=1,6),I=1,6)
READ(K20)((C(I,J),J=1,6),I=1,6)

```

```

      DO 125 I=1,24
      DO 125 J=1,24
        GR1(I,J) = 0.00
      DO 125 M=1,24
125  GR1(I,J) = GR1(I,J) + G(I,M)*R1(M,J)
      DO 126 I=1,24
        UC1(I) = 0.00
        UT1(I) = 0.00
      DO 126 J=1,24
        UC1(I) = UC1(I) + GR1(I,J)*UC(J)
126  UT1(I) = UT1(I) + GR1(I,J)*UT(J)
      DO 130 I=1,4
        II=3*I
        JJ=3*IX(N,I)
        P1(II-2) = B(JJ-2)
        P1(II-1) = B(JJ-1)
        P1(II) = B(JJ)
        P1(II+10) = B(JJ-2)
        P1(II+11) = B(JJ-1)
        P1(II+12) = B(JJ)
130  CONTINUE
      DO 135 I=1,24
135  P1(I) = P1(I) - UC1(I) + UT1(I)
      DO 136 I=1,3
        P1(I+24) = (P1(I) + P1(I+3) + P1(I+6) + P1(I+9))/4.00
136  P1(I+27) = (P1(I+12) + P1(I+15) + P1(I+18) + P1(I+21))/4.00
      DO 140 I=1,6
        EPS1(I) = 0.00
      DO 140 J=1,30
140  EPS1(I) = EPS1(I) + BS1(I,J)*P1(J)
      DO 143 I=1,6
        EPS1(I+6) = 0.0
      DO 143 J=1,6
        DO 143 L=1,6
143  EPS1(I+6) = EPS1(I+6) + D(I,J)*C(J,L)*EPS1(L)
      DO 150 I=1,6
        SIG1(I) = EPSDN(I,N,NS)
        SIG1(I+6) = EPSDN(I+6,N,NS)
        SIGVP(I) = 0.0
      DO 150 J=1,6
        SIG1(I) = SIG1(I) + CRZ(I,J)*EPS1(J)
150  SIG1(I+6) = SIG1(I+6) + CNS(I,J)*EPS1(J+6)
      DO 151 I=1,6
        SIGVP(I) = SIG1(I+6)
151  CONTINUE
      DO 141 J=1,12
141  EPS1(J) = EPS1(J)*100.0
      DO 230 I=1,6
        DEPS1(I) = 0.0
230  DEPS1(I+6) = 0.0
      IF (IFGPL(N,NS).EQ.0) GO TO 241
      DO 250 I=1,6
        DO 250 J=1,6
150  DEPS1(I+6) = DEPS1(I+6) + CEPSP(I,J)*EPS1(J+6)/100.
      DO 251 I=1,6
151  DEPS1(I+6) = DEPS1(I+6) + DEPSR(I,N,NS)
      D(4,1) = 0.5*D(4,1)
      D(4,3) = 0.5*D(4,3)
      D(1,6) = 2.0*D(1,6)
      D(2,6) = 2.0*D(2,6)

```

```

      C(1,4)= 2.0*C(1,4)
      C(2,4)= 2.0*C(2,4)
      C(4,1)= 0.5*C(4,1)
      C(4,2)= 0.5*C(4,2)
      DO 160 I=1,6
      DEFSI(I)=0.0
      DO 160 J=1,6
      DO 160 L=1,6
160    DEFSI(I)=DEFSI(I)+C(J,I)*D(J,L)*DEFSI(L+6)
      WRITE(6,1400)(DEFSI(I),I=1,12)
1400  FORMAT(" PLASTIC STRAINS"/2X, 12E10.4)
C      DO 233 I=1,6
C      EPSIN(I,N,NS)=DEFSI(I+6)/DELTIM
C 233  CONTINUE
241  CONTINUE
      DO 240 I=1,12
      SIGTOT(I,N,NS)=SIGTOT(I,N,NS)+SIGI(I)
240  EPSTOT(I,N,NS)=EPSTOT(I,N,NS)+EPSI(I)
      WRITE(6,1000) N,(SIGTOT(I,N,NS),I=1,12)
      WRITE(6,1111)
1111  FORMAT(" ", "SIGVP ")
      WRITE(6,1000) N,(SIGVP(I),I=1,6)
      CALL YIELD(N,NS,MTYPE)
      IF(IFGPL(N,NS).EQ.1) WRITE(6,1300)N,NS
1300  FORMAT(" ", "ELEMENT",I5,"OF SEGMENT",I5,"HAS YIELDED")
      WRITE(6,1100) (EPSTOT(I,N,NS),I=1,12)
120  CONTINUE
100  CONTINUE
      REWIND 21
      REWIND 25
      REWIND 26
1000  FORMAT(" ",I5,12F9.0)
1100  FORMAT(" ",5X,12F9.5)
1200  FORMAT("1","SEGMENT TYPE",I5,/,," ", "SEGMENT NUMBER = ",I5)
      RETURN
      END
      SUBROUTINE ASEMBL
      COMMON/INCR/NOLINC,NOL,INERT,NUMMAT,SIGTOT(12, 4,8)
1    ,EPSTOT(12, 4,8)
      COMMON/BOCON/NRDF,NREQ(18),URES(18)
      COMMON/GLBSEG/FI(24,8),FE(24,8),UC(24,8),SK(24,24,8) -
      COMMON/NXDATA/NTP,NTS,NTOTS,GTSIG(24,24,8)
      COMMON/NXSOLV/SKG(36,24),FTG(132),FTOT(132),ITOT
      COMMON/ANS2/FC(24),G(24,24),GR1(24,24),DUMM(24,24)
      ITOT= 24 + 12*(NTOTS-1)
      DO 10 I=1,ITOT
      FTG(I) = 0.00
      DO 10 J = 1,24
10    SKG(I,J) = 0.00
      DO 100 M=1,NTOTS
C *****
C    COMBINE FI, FE, AND SK*UC INTO A TOTAL FORCE VECTOR FC
C *****
      DO 55 I=1,24
      FC(I) = 0.00
      DO 55 J=1,24
55    FC(I) = FC(I) + SKG(I,J,M)* UC(J,M)
      DO 60 I=1,24
60    FC(I) = FC(I) +FE(I,M) -FI(I,M)
C *****

```

```

C   NOW FILL GLOBAL FORCE AND STIFFNESS MATRICES
C   *****
      DO 70 I=1,24
        I1 = I+(M-1)*12
        FTG(I1) = FTG(I1) + FC(I)
      DO 70 J=1,24
        SKG(I1,J+1-I) = SKG(I1,J+1-I) + SK(I,J,M)
      70 CONTINUE
    100 CONTINUE
      IF (NOL.NE.1) GO TO 80
C   READ THE TOTAL NUMBER OF RESTRAINED DEGREES OF FREEDOM
      READ(5,1200) NRDF
      WRITE(6,1255) NRDF
C   IMPOSE BOUNDARY CONDITIONS ON RESTRAINED D-O-F
      DO 150 NBC=1,NRDF
C   READ THE EQUATION NUMBER AND THE IMPOSED BOUNDARY CONDITION
      READ(5,1250) NREQ(NBC),URES(NBC)
      WRITE(6,1260)NREQ(NBC),URES(NBC)
    150 CONTINUE
      80 CONTINUE
      DO 160 NBC=1,NRDF
    160 CALL XMODFY(URES(NBC),NREQ(NBC))
    1200 FORMAT(I5)
    1250 FORMAT(I5,F10.0)
    1255 FORMAT(1H1,"NUMBER OF RESTRAINED DEGREES OF FREEDOM =",I10/
      1 " EQUATION NUMBER VALUE ")
    1260 FORMAT (" ",5X,I5,5X,F10.2)
      CALL XSOLVE
      WRITE(6,1050)
      WRITE(6,1100)(FTG(I),I=1,ITOT)
      DO 200 I=1,ITOT
    200 FTOT(I)=FTOT(I)+FTG(I)
      WRITE(6,1051)
      WRITE(6,1100)(FTOT(I),I=1,ITOT)
    1050 FORMAT("1","INCREMENTAL DISPLACEMENTS AT CONNECTING NODES"/
      1 " 18X,2HUR,18X,2HUZ,18X,2HUT )
    1100 FORMAT(" ",3E20.7)
    1051 FORMAT("1","TOTAL DISPLACEMENTS AT CONNECTING NODES"/
      1 " 18X,2HUR,18X,2HUZ,18X,2HUT )
      RETURN
      END
      SUBROUTINE CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      INTEGER CODE
      COMMON/TO/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAYO,NMTL,RBC
      COMMON/NPDATA/R(10 ),CODE(10 ),XR(10 ),Z(10 ),XZ(10 ),
      INPNUM( 4, 8),T(10 ),XT(10 )
      DIMENSION AR( 4, 8),AZ( 4, 8)
      EQUIVALENCE (R(1),AR),(Z(1),AZ)
C* * * * *
C   FIND INTERSECTION OF LINE AND CIRCLE = NEW R AND Z
C* * * * *
      ANG1=ANG1+DELPHI
      RR=SQRT((RSTRT-RC)**2+(ZSTRT-ZC)**2)
      AR(I,J)=RC+RR*COS(ANG1)
      AZ(I,J)=ZC+RR*SIN(ANG1)
      RETURN
      END
      SUBROUTINE DATA
      INTEGER CODE
      COMMON/INCR/NOLINC,NOL,INERT,NUMMAT,SIGTOT(12, 4,8)

```

```

1  ,EPSTOT(12, 4,8)
COMMON/NPDATA/R(10 ),CODE(10 ),XR(10 ),Z(10 ),XZ(10 ),
1 NPNUM( 4, 8),T(10 ),XT(10 )
COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC
1NUMST
COMMON/ELDATA/BETA(10 ),EPR(10 ),PR( 4 ),SH( 4 ),IX( 8 ,5),
1IP( 4 ),JP( 4 ),IS( 4 ),JS( 4 ),ALPHA(10 ),IT( 4 ),JT( 4 ),
2ST( 4 )
COMMON/NXDATA/NTP,NTS,NTOTS,GTS1G(24,24,8)
COMMON/MATP/RO( 6 ),E(12,16,6),EE(16),AOFTS( 6 )
COMMON/SOLVE/X(888),Y(888),TEM(888),NUMTC,MBAND
COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,
1 MAXJ,NMTL,NBC
READ(15)NUMTC,NUMMAT,NUMPC,NUMSC,TREF,INERT,
1 INCI,INCF
READ(15)NBC,NMTL
READ(15)NUMEL,NUMNP
READ(15)(CODE(I),I=1,NUMNP)
READ(15)(XR(I),I=1,NUMNP)
READ(15)(XT(I),I=1,NUMNP)
READ(15)(XZ(I),I=1,NUMNP)
READ(15)(R(I),I=1,NUMNP)
READ(15)(Z(I),I=1,NUMNP)
READ(15)((IX(I,J),J=1,5),I=1,NUMEL)
READ(15)(BETA(I),I=1,NUMEL)
READ(15)(ALPHA(I),I=1,NUMEL)
READ(15)(TEM(I),I=1,NUMEL)
DO 200 I=1,NUMMAT
READ(15)MTYPE,NT,RO(MTYPE)
READ(15)((E(II,J,MTYPE),J=1,14),II=1,NT)
DO 200 K=NT,12
DO 200 L=1,6
200 E(K,L,MTYPE)=E(NT,L,MTYPE)
RETURN
END
SUBROUTINE INTER
COMMON/ARG/RRR( 5 ),ZZZ( 5 ),RR( 4 ),ZZ( 4 ),S(15,15),P(15),TT( 6 ),
1H( 6,15),CRZ( 6,6),XI(10),ANGLE( 4 ),SIG(18),EPS(18),N
COMMON/PLANE/NPP
DIMENSION XM(7),R(7),Z(7),XX(9)
DATA XX/3*.1259391805448,3*.1323941527884,.225,
1 .696140478028,.410426192314/
R(7)=(RR(1)+RR(2)+RR(3))/3.0
Z(7)=(ZZ(1)+ZZ(2)+ZZ(3))/3.0
DO 100 I=1,3
J=I+3
R(I)=XX(8)*RR(I)+(1.00-XX(8))*R(7)
R(J)=XX(9)*RR(I)+(1.00-XX(9))*R(7)
Z(I)=XX(8)*ZZ(I)+(1.00-XX(8))*Z(7)
100 Z(J)=XX(9)*ZZ(I)+(1.00-XX(9))*Z(7)
DO 200 I=1,7
200 XM(I)=XX(I)*R(I)
DO 300 I=1,10
300 XI(I)=0.00
AREA=.50*(RR(1)*(ZZ(2)-ZZ(3))+RR(2)*(ZZ(3)-ZZ(1))+RR(3)*(ZZ(1)
1 -ZZ(2)))
IF(NPP.NE.0) GO TO 600
DO 400 I=1,7
XI(1)=XI(1)+XM(I)
XI(2)=XI(2)+XM(I)/R(I)

```

```

      XI(3)=XI(3)+XM(I)/(R(I)**2)
      XI(4)=XI(4)+XM(I)*Z(1)/R(I)
      XI(5)=XI(5)+XM(I)*Z(I)/(R(I)**2)
      XI(6)=XI(6)+XM(I)*(Z(I)**2)/(R(I)**2)
      XI(7)=XI(7)+XM(I)*R(I)
      XI(8)=XI(8)+XM(I)*Z(I)
      XI(9)=XI(9)+XM(I)*(R(I)**2)
400  XI(10)=XI(10)+XM(I)*R(I)*Z(I)
      DO 500 I=1,10
500  XI(I)=XI(I)*AREA
      RETURN
600  XI(1)=AREA
      XI(7)=R(7)*AREA
      XI(8)=Z(7)*AREA
      RETURN
      END
      SUBROUTINE MESH
      INTEGER CODE
      DIMENSION AR( 4, 8),AZ( 4, 8),NCODE( 4, 8)
      COMMON/TO/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/NPDATA/R(10 ),CODE(10 ),XR(10 ),Z(10 ),XZ(10 ),
1NPNUM( 4, 8),T(10 ),XT(10 )
      COMMON/ELDATA/BETA(10 ),EPR(10 ),PR( 4 ),SH( 4 ),IX( 8 ,5),
1IP( 4 ),JP( 4 ),IS( 4 ),JS( 4 ),ALPHA(10 ),IT( 4 ),JT( 4 ),
2ST( 4 )
      EQUIVALENCE (R(1),AR),(Z(1),AZ),(IX(1,1),NCODE)
C* * * * *
C  MESH CONTROL INFORMATION
C* * * * *
      READ (5,1000) MAXI,MAXJ,NSEG,NBC,NMTL
      WRITE(6,2000) MAXI,MAXJ,NSEG,NBC,NMTL
C* * * * *
C  INITIALIZE
C* * * * *
      ISEG=-1
      PI=3.1415927
      DO 110 J=1,8
      DO 100 I=1,4
      NCODE(I,J)=0
      AR(I,J)=0.
      AZ(I,J)=0.
      JMAX(I)=0
100  JMIN(I)=MAXI
      IMIN(J)=MAXJ
110  IMAX(J)=0
C* * * * *
C  LINE SEGMENT CARDS
C* * * * *
150  ISEG=ISEG+1
159  IF(ISEG.EQ.NSEG) GO TO 400
      READ(5,1001) I1,J1,R1,Z1,I2,J2,R2,Z2,I3,J3,R3,Z3,IPTION
      WRITE(6,2001) I1,J1,R1,Z1,I2,J2,R2,Z2,I3,J3,R3,Z3,IPTION
      IPTION=IPTION+1
      AR(I1,J1)=R1
      AZ(I1,J1)=Z1
      NCODE(I1,J1)=1
      CALL MNIMX(I1,J1)
      GO TO (150,200,200,300,300,200,200), IPTION
C* * * * *
C  GENERATE STRAIGHT LINES ON BOUNDARY

```

```

C* * * * *
200  DI= ABS(FLOAT(I2-I1))
    DJ= ABS(FLOAT(J2-J1))
    AR(I2,J2)=R2
    AZ(I2,J2)=Z2
    NCODE(I2,J2)=1
    CALL MNIMX(I2,J2)
    ISTRT=I1
    ISTP=I2
    JSTRT=J1
    JSTP=J2
    DIFF=MAX1(DI,DJ)
    ITER=DIFF-1.
    IINC=0
    JINC=0
    IF(I2.NE.I1) IINC=(I2-I1)/IABS(I2-I1)
    IF(J2.NE.J1) JINC=(J2-J1)/IABS(J2-J1)
    KAPPA=1
    IF(I2.NE.I1.AND.J2.NE.J1.AND.IPTION.NE.3) KAPPA=2
    IF(KAPPA.EQ.2) DIFF=2.*DIFF
    RINC=(R2-R1)/DIFF
    ZINC=(Z2-Z1)/DIFF
    WRITE(6,2002) DI,DJ,DIFF,RINC,ZINC,ITER,IINC,JINC,KAPPA

C
C   CHECK FOR INPUT ERROR
C
    IF(KAPPA.NE.2.OR.DI.EQ.DJ) GO TO 210
    WRITE(6,2003)
    GO TO 150

C
C   INTERPOLATE
C
210  I=I1
    J=J1
    WRITE(6,2004)
    DO 230 M=1,ITER
    IF(ITER.EQ.0.AND.IPTION.EQ.2) GO TO 230
    IF(ITER.EQ.0.AND.IPTION.EQ.6) GO TO 230
    IF(ITER.EQ.0.AND.IPTION.EQ.7) GO TO 230
    IF(KAPPA.EQ.2) GO TO 220
    IOLD=I
    I=I+IINC
    JOLD=J
    J=J+JINC
    AR(I,J)=AR(IOLD,JOLD)+RINC
    AZ(I,J)=AZ(IOLD,JOLD)+ZINC
    WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
    CALL MNIMX(I,J)
    NCODE(I,J)=1
    GO TO 230
220  CONTINUE
    IF(I1.GT.I2.AND.IPTION.EQ.7) GO TO 221
    IF(I1.LT.I2.AND.IPTION.EQ.6) GO TO 221
    IOLD=I
    I=I+IINC
    AR(I,J)=AR(IOLD,J)+RINC
    AZ(I,J)=AZ(IOLD,J)+ZINC
    WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
    NCODE(I,J)=1
    CALL MNIMX(I,J)

```

```

JOLD=J
J=J+JINC
AR(I,J)=AR(I,JOLD)+RINC
AZ(I,J)=AZ(I,JOLD)+ZINC
NCODE(I,J)=1
WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
CALL MNIMX(I,J)
GO TO 230
221 JOLD=J
J=J+JINC
AR(I,J)=AR(I,JOLD)+RINC
AZ(I,J)=AZ(I,JOLD)+ZINC
NCODE(I,J)=1
WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
CALL MNIMX(I,J)
IOLD=I
I=I+IINC
AR(I,J)=AR(IOLD,J)+RINC
AZ(I,J)=AZ(IOLD,J)+ZINC
NCODE(I,J)=1
WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
CALL MNIMX(I,J)
230 CONTINUE
IF(KAPPA.EQ.1) GO TO 150
IF(I1.GT.I2.AND.IPTION.EQ.7) GO TO 231
IF(I1.LT.I2.AND.IPTION.EQ.6) GO TO 231
IOLD=I
I=I+IINC
AR(I,J)=AR(IOLD,J)+RINC
AZ(I,J)=AZ(IOLD,J)+ZINC
GO TO 232
231 CONTINUE
JOLD=J
J=J+JINC
AR(I,J)=AR(I,JOLD)+RINC
AZ(I,J)=AZ(I,JOLD)+ZINC
232 CONTINUE
NCODE(I,J)=1
WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
CALL MNIMX(I,J)
GO TO 150
C* * * * *
C GENERATE CIRCULAR ARCS ON BOUNDARY
C* * * * *
300 AR(I2,J2)=R2
AZ(I2,J2)=Z2
NCODE(I2,J2)=1
CALL MNIMX(I2,J2)
IF(IPTION.EQ.5) GO TO 320
C
C FIND CENTER OF CIRCLE
C
AR(I3,J3)=R3
AZ(I3,J3)=Z3
NCODE(I3,J3)=1
CALL MNIMX(I3,J3)
SLAC=(Z2-Z1)/(R2-R1)
SLBF=-1./SLAC
SLCE=(Z3-Z2)/(R3-R2)
SLDF=-1./SLCE

```

```

C
C   CHECK FOR INPUT ERROR
C
    IF(ABS(SLAC-SLCE).GT..001) GO TO 310
    WRITE(6,2006) R1,Z1,R2,Z2,R3,Z3,SLAC,SLCE
    GO TO 150
310  R4=R1+(R2-R1)/2.
    Z4=Z1+(Z2-Z1)/2.
    R5=R2+(R3-R2)/2.
    Z5=Z2+(Z3-Z2)/2.
    BBF=Z4-SLBF*R4
    BDF=Z5-SLDF*R5
    RC=(BBF-BDF)/(SLDF-SLBF)
    ZC=SLBF*RC+BBF
    WRITE(6,2007) RC,ZC
    KAPPA=1
    GO TO 330
320  KAPPA=2
    RC=R3
    ZC=Z3
330  ISTRT=I1
    ISTOP=I2
    JSTRT=J1
    JSTOP=J2
    RSTRT=R1
    RSTOP=R2
    ZSTRT=Z1
    ZSTOP=Z2
340  CALL ANGLE(RSTRT,ZSTRT,RC,ZC,ANG1)
    CALL ANGLE(RSTOP,ZSTOP,RC,ZC,ANG2)
    IF(ANG2.LE.ANG1) ANG2=2.0*PI+ANG2
C
C   FIND ANGULAR INCREMENT
C
    DI=ABS(FLOAT(ISTOP-ISTRT))
    DJ=ABS(FLOAT(JSTOP-JSTRT))
    IINC=0
    JINC=0
    IF(ISTRT.NE.ISTOP) IINC=(ISTOP-ISTRT)/IABS(ISTOP-ISTRT)
    IF(JSTRT.NE.JSTOP) JINC=(JSTOP-JSTRT)/IABS(JSTOP-JSTRT)
    LAMDA=1
    IF(IINC.NE.0.AND.JINC.NE.0) LAMDA=2
    DIFF=MAX1(DI,DJ)
    ITER=DIFF-1.
    IF(LAMDA.EQ.2) DIFF=2.*DIFF
    DELPHI=(ANG2-ANG1)/DIFF
    WRITE(6,2008) ANG1,ANG2,DIFF,DELPHI
C
C   CHECK FOR INPUT ERROR
C
    IF(LAMDA.NE.2.OR.DI.EQ.DJ) GO TO 350
    WRITE(6,2003)
    GO TO 150
350  IO=ISTRT
    JO=JSTRT
    WRITE(6,2004)
C
C   INTERPOLATE
C
    NPT=IABS(I2-I1)+IABS(J2-J1)-1

```

```

      DO 380 N=1,ITER
359 IF(LAMDA.EQ.2) GO TO 360
      I=IO+IINC
      J=JO+JINC
      CALL MNIMX(I,J)
      NCODE(I,J)=1
      CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      GO TO 370
360 I=IO+IINC
      J=JO
      NCODE(I,J)=1
      CALL MNIMX(I,J)
      CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      J=JO+JINC
      NCODE(I,J)=1
      CALL MNIMX(I,J)
      CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
370 IO=I
380 JO=J
      IF(LAMDA.NE.2) GO TO 390
      I=IO+IINC
      NCODE(I,J)=1
      CALL MNIMX(I,J)
      CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
390 IF(KAPPA.EQ.2) GO TO 150
      ISTRT=I2
      ISTOP=I3
      JSTRT=J2
      JSTOP=J3
      RSTRT=R2
      RSTOP=R3
      ZSTRT=Z2
      ZSTOP=Z3
      KAPPA=2
399 GO TO 340
C* * * * *
C   CALCULATE COORDINATES OF INTERIOR POINTS
C* * * * *
400 IF(MAXJ.LE.2) GO TO 430
      J2=MAXJ-1
      DO 420 N=1,500
      RESID=0.
      DO 410 J=2,J2
      I1=IMIN(J)+1
      I2=IMAX(J)-1
      DO 410 I=I1,I2
      IF(NCODE(I,J).EQ.1) GO TO 410
      DR=(AR(I+1,J)+AR(I-1,J)+AR(I,J+1)+AR(I,J-1))/4.-AR(I,J)
      DZ=(AZ(I+1,J)+AZ(I-1,J)+AZ(I,J+1)+AZ(I,J-1))/4.-AZ(I,J)
      RESID=RESID+ABS(DR)+ABS(DZ)
      AR(I,J)=AR(I,J)+1.8*DR
      AZ(I,J)=AZ(I,J)+1.8*DZ
410 CONTINUE
      IF(N.EQ.1) RES1=RESID
      IF(N.EQ.1.AND.RESID.EQ.0.)GO TO 430
      IF(RESID/RES1.LT.1.E-5) GO TO 430

```

```

420 CONTINUE
430 WRITE(6,2009) N
    WRITE(15)NBC,NMTL
C* * * * *
CALL POINTS
C* * * * *
1000 FORMAT (5I5)
1001 FORMAT (3(2I3,2F8.3),I5)
2000 FORMAT (30H1 MESH GENERATION INFORMATION//
1 41H0 MAXIMUM VALUE OF I IN THE MESH-----I3/
2 41H0 MAXIMUM VALUE OF J IN THE MESH-----I3/
3 41H0 NUMBER OF LINE SEGMENT CARDS-----I3/
4 41H0 NUMBER OF BOUNDARY CONDITION CARDS-----I3/
5 41H0 NUMBER OF MATERIAL BLOCK CARDS-----I3///)
2001 FORMAT (//88H INPUT I1 J1 R1 Z1 I2 J2 R2 Z
12 I3 J3 R3 Z3 IPTION/8X,3(2I4,2F8.4),I6)
2002 FORMAT (5H DI=F4.0,5H DJ=F4.0,7H DIFF=F4.0,7H RINC=F8.3,7H ZI
INC=F8.3,7H ITER=I3,7H IINC=I3,7H JINC=I3,8H KAPPA=I1)
2003 FORMAT(1X,38H**BAD INPUT--THIS LINE IS NOT DIAGONAL)
2004 FORMAT (30H I J AR AZ)
2005 FORMAT (2I5,2F11.6)
2006 FORMAT (51H ** BAD INPUT - THESE POINTS DO NOT DEFINE A CIRCLE,/,
13X,6F12.4,10X,2E20.8)
2007 FORMAT(19H CENTER COORDINATE,(F11.6,1X,F11.6,1X))
2008 FORMAT (7H ANG1=F9.6,7H ANG2=F9.6,7H DIFF=F3.0,9H DELPHI=F9.6)
2009 FORMAT (//30H COORDINATES CALCULATED AFTER I3,11H ITERATIONS)
RETURN
END

```

```

C
C
C      SUBROUTINE MINV
C
C      PURPOSE
C      INVERT A MATRIX
C
C      USAGE
C      CALL MINV(A,N,D,L,M)
C
C      DESCRIPTION OF PARAMETERS
C      A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C      RESULTANT INVERSE.
C      N - ORDER OF MATRIX A
C      D - RESULTANT DETERMINANT
C      L - WORK VECTOR OF LENGTH N
C      M - WORK VECTOR OF LENGTH N
C
C      REMARKS
C      MATRIX A MUST BE A GENERAL MATRIX
C
C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      NONE
C
C      METHOD
C      THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
C      IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C      THE MATRIX IS SINGULAR.
C
C      .....

```

```

SUBROUTINE MINV(A,N,D,L,M)

```

DIMENSION A(1),L(1),M(1)

.....
IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
STATEMENT WHICH FOLLOWS.

DOUBLE PRECISION A,D,BIGA,HOLD

THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
ROUTINE.

THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS. ABS IN STATEMENT
10 MUST BE CHANGED TO DABS.

.....
SEARCH FOR LARGEST ELEMENT

D=1.0
NK=-N
DO 80 K=1,N
NK=NK+N
L(K)=K
M(K)=K
KK=NK+K
BIGA=A(KK)
DO 20 J=K,N
IZ=N*(J-1)
DO 20 I=K,N
IJ=IZ+I
10 IF(ABS(BIGA)-ABS(A(IJ))) 15,20,20
15 BIGA=A(IJ)
L(K)=I
M(K)=J
20 CONTINUE

INTERCHANGE ROWS

J=L(K)
IF(J-K) 35,35,25
25 KI=K-N
DO 30 I=1,N
KI=KI+N
HOLD=-A(KI)
JI=KI-K+J
A(KI)=A(JI)
30 A(JI)=HOLD

INTERCHANGE COLUMNS

35 I=M(K)
IF(I-K) 45,45,38
38 JP=N*(I-1)
DO 40 J=1,N
JK=NK+J
JI=JP+J

```

        HOLD=-A(JK)
        A(JK)=A(JI)
40  A(JI)=HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
45  IF(BIGA) 43,46,48
46  D=0.0
    RETURN
48  DO 55 I=1,N
    IF(I-K) 50,55,50
50  IK=NK+I
    A(IK)=A(IK)/(-BIGA)
55  CONTINUE
C
C        REDUCE MATRIX
C
    DO 65 I=1,N
    IK=NK+I
    HOLD=A(IK)
    IJ=I-N
    DO 65 J=1,N
    IJ=IJ+N
    IF(I-K) 60,65,60
60  IF(J-K) 62,65,62
62  KJ=IJ-I+K
    A(IJ)=HOLD*A(KJ)+A(IJ)
65  CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
    KJ=K-N
    DO 75 J=1,N
    KJ=KJ+N
    IF(J-K) 70,75,70
70  A(KJ)=A(KJ)/BIGA
75  CONTINUE
C
C        PRODUCT OF PIVOTS
C
    D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
    A(KK)=1.0/BIGA
80  CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
    K=N
100 K=(K-1)
    IF(K) 150,150,105
105 I=L(K)
    IF(I-K) 120,120,108
108 JQ=N*(K-1)
    JR=N*(I-1)
    DO 110 J=1,N
    JK=JQ+J
    HOLD=A(JK)

```

```

      JI=JR+J
      A(JK)=-A(JI)
110  A(JI)=HOLD
120  J=M(K)
      IF(J-K) 100,100,125
125  KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
130  A(JI)=HOLD
      GO TO 100
150  RETURN
      END
      SUBROUTINE MNIMX(I,J)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      IF(J.LT.JMIN(I)) JMIN(I)=J
      IF(J.GT.JMAX(I)) JMAX(I)=J
      IF(I.LT.IMIN(J)) IMIN(J)=I
      IF(I.GT.IMAX(J)) IMAX(J)=I
      RETURN
      END
      SUBROUTINE MODIFY(NEQ,N,U)
      COMMON/SOLVE/B(72),A(72,36),NUMBLK,MBAND
      DO 10 M=2,MBAND
      K=N-M+1
      IF(K.LE.0) GO TO 5
      B(K)=B(K)-A(K,M)*U
      A(K,M)=0.00
5    K=N+M-1
      IF(NEQ.LT.K) GO TO 10
      B(K)=B(K)-A(N,M)*U
      A(N,M)=0.00
10   CONTINUE
      A(N,1)=1.00
      B(N)=U
      RETURN
      END
      SUBROUTINE MPLOT
      INTEGER CODE
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/NPDATA/R(10),CODE(10),XR(10),Z(10),XZ(10),
      INPNUM(4,8),T(10),XT(10)
      REAL X(100),Y(100),TX(2),TY(2),TITLE(20),ZMAX
      READ(5,1000) TITLE,RMAX,ZMAX
C     CALL CCP2SY(0.7,0.2,0.2,TITLE,0.0,30)
C     CALL CCP1PL(0.7,0.7,-3)
      TX(1)=0.00
      TY(1)=0.00
      TX(2)=RMAX/9.0
      TY(2)=RMAX/9.0
      ZMAX=ZMAX*TY(2)+2.0
      IF(ZMAX.LT.17.0) ZMAX=17.0
      DO 100 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      N=0
      DO 101 I=NSTART,NSTOP
      N=N+1

```

```

      NP=NPNUM(I,J)
      Y(N)=R(NP)
- 101 X(N)=Z(NP)
C    CALL CCP6LN(X,Y,N,1,TX,TY)
- 100 CONTINUE
      DO 102 I=1,MAXI
      NSTART=JMIN(I)
      NSTOP=JMAX(I)
      N=0
      DO 103 J=NSTART,NSTOP
      N=N+1
      NP=NPNUM(I,J)
      Y(N)=R(NP)
- 103 X(N)=Z(NP)
C    CALL CCP6LN(X,Y,N,1,TX,TY)
- 102 CONTINUE
C    CALL CCP1PL(ZMAX,-0.7,-3)
- 1000 FORMAT(20A4/2F10.0)
      RETURN
      END
      SUBROUTINE NAXSTF(II,JJ,KK)
      INTEGER CODE
      COMMON/VISC/EPSON(12,10,8),SIGVP(6),DEPSR(6,10,8),DELTIM
      COMMON/PLAS/ALFA(6,4,8),SIGYLD(7,6,8),IFGPL(4,8)
      COMMON/NXDATA/NTP,NTS,NTOTS,GTSIG(24,24,8)
      COMMON/MATP/RO(6),E(12,16,6),EE(16),AOFTS(6)
      COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
      1NUMST
      COMMON/ARG/RRR(5),ZZZ(5),RR(4),ZZ(4),S(15,15),P(15),TT(6),
      1H(6,15),CRZ(6,6),XI(10),ANGLE(4),SIG(18),EPS(18),N
      COMMON/NPDATA/R(10),CODE(10),XR(10),Z(10),XZ(10),
      1NPNUM(4,8),T(10),XT(10)
      COMMON/ELDATA/BETA(10),EPR(10),PR(4),SH(4),IX(8,5),
      1IP(4),JP(4),IS(4),JS(4),ALPHA(10),IT(4),JT(4),
      2ST(4)
      COMMON/NXQUAD/AR1
      COMMON/NONAXI/S1(30,30),P1(30),THETA,BS1(6,30)
      DIMENSION C(18,18),B(18,18),B1(6,18),B2(6,18),B3(6,18),B4(6,18),
      1B5(6,18),B6(6,18),B1A(6,18),B1B(6,18),B2A(6,18),B2B(6,18),
      2B3A(6,18),B3B(6,18),B4A(6,18),B4B(6,18),B5A(6,18),
      3B5B(6,18),B6A(6,18),B6B(6,18),TVP(18)
C    ZERO MATRICES
      DO 100 I=1,18
      DO 100 J=1,18
- 100 C(I,J)=0.0
      DO 110 I=1,6
      DO 110 J=1,18
      B1(I,J)=0.0
      B2(I,J)=0.0
      B3(I,J)=0.0
      B4(I,J)=0.0
      B5(I,J)=0.0
- 110 B6(I,J)=0.0
      RR(1)=RRR(II)
      RR(2)=RRR(JJ)
      RR(3)=RRR(KK)
      ZZ(1)=ZZZ(II)
      ZZ(2)=ZZZ(JJ)
      ZZ(3)=ZZZ(KK)
      COMM=RR(2)*(ZZ(3)-ZZ(1))+RR(1)*(ZZ(2)-ZZ(3))+RR(3)*(ZZ(1)-ZZ(2))

```

C FILL C INVERSE

```

C(1,1)= ( RR(2)*ZZ(3) -RR(3)* ZZ(2)) / COMM
C(1,4)= ( RR(3)*ZZ(1) -RR(1)* ZZ(3)) / COMM
C(1,7)= ( RR(1)*ZZ(2) -RR(2)* ZZ(1)) / COMM
C(2,1)= ( ZZ(2) - ZZ(3)) / COMM
C(2,4)= ( ZZ(3) - ZZ(1)) / COMM
C(2,7)= ( ZZ(1) - ZZ(2)) / COMM
C(3,1)= ( RR(3) - RR(2)) / COMM
C(3,4)= ( RR(1) - RR(3)) / COMM
C(3,7)= ( RR(2) - RR(1)) / COMM
C(4,2)= C(1,1)
C(4,5)= C(1,4)
C(4,8)= C(1,7)
C(5,2)= C(2,1)
C(5,5)= C(2,4)
C(5,8)= C(2,7)
C(6,2)= C(3,1)
C(6,5)= C(3,4)
C(6,8)= C(3,7)
C(7,3)= C(1,1)
C(7,6)= C(1,4)
C(7,9)= C(1,7)
C(8,3)= C(2,1)
C(8,6)= C(2,4)
C(8,9)= C(2,7)
C(9,3)= C(3,1)
C(9,6)= C(3,4)
C(9,9)= C(3,7)
DO 120 I=10,18
DO 120 J= 1,9
I1 = I-9
J1=J+9
C(I,J)=(-1./THETA) * C(I1,J)
C(I,J1)=( 1./THETA) * C(I1,J)

```

120 CONTINUE

C FILL B MATRICES

```

C B1 CONSTANT TERMS
C B2 THETA TERMS
C B3 1/R TERMS
C B4 THETA/ R TERMS
C B5 Z/R TERMS
C B6 THETA *Z/R TERMS

```

```

DO 130 J=1,18
B1(1,J) = C(2,J)
B1(2,J) = C(6,J)
B1(3,J) = C(2,J)+C(17,J)
B1(4,J) = C(3,J)+C(5,J)
B1(5,J) = C(9,J) +C(14,J)
B1(6,J) = C(11,J)
B2(1,J) = C(11,J)
B2(2,J) = C(15,J)
B2(3,J) = C(11,J)
B2(4,J) = C(12,J)+C(14,J)
B2(5,J) = C(18,J)
B3(3,J) = C(1,J)+ C(16,J)
B3(5,J) = C(13,J)
B3(6,J) = C(10,J) - C(7,J)
B4(3,J) = C(10,J)
B4(6,J) = -C(16,J)
B5(3,J) = C(3,J) +C(18,J)

```

```

      B5(5,J) = C(15,J)
      B5(6,J) = C(12,J)-C(9,J)
      B6(3,J) = C(12,J)
      B6(6,J) = -C(18,J)
130  CONTINUE
C NOW CALCULATE BT * D * B
      CALL INTER
      THETA2 = (THETA **2)/2.0
      THETA3 = (THETA **3)/3.0
      DO 140 I=1,6
      DO 140 J=1,18
        B1A(I,J)=(B1(I,J)*XI(1) +B3(I,J)* XI(2) + B5(I,J)* XI(4))* THETA +
1          (B2(I,J)*XI(1) +B4(I,J)* XI(2) + B6(I,J)* XI(4))* THETA2
        B2A(I,J)=(B1(I,J)*XI(1) +B3(I,J)* XI(2) + B5(I,J)* XI(4))* THETA2
1          + (B2(I,J)*XI(1) +B4(I,J)* XI(2) + B6(I,J)* XI(4))* THETA3
        B3A(I,J)=(B1(I,J)*XI(2) +B3(I,J)* XI(3) + B5(I,J)* XI(5))* THETA
1          + (B2(I,J)*XI(2) +B4(I,J)* XI(3) + B6(I,J)* XI(5))* THETA2
        B4A(I,J)=(B1(I,J)*XI(2) +B3(I,J)* XI(3) + B5(I,J)* XI(5))* THETA2
1          + (B2(I,J)*XI(2) +B4(I,J)* XI(3) + B6(I,J)* XI(5))* THETA3
        B5A(I,J)=(B1(I,J)*XI(4) +B3(I,J)* XI(5) + B5(I,J)* XI(6))* THETA
1          + (B2(I,J)*XI(4) +B4(I,J)* XI(5) + B6(I,J)* XI(6))* THETA2
        B6A(I,J)=(B1(I,J)*XI(4) +B3(I,J)* XI(5) + B5(I,J)* XI(6))* THETA2
1          + (B2(I,J)*XI(4) +B4(I,J)* XI(5) + B6(I,J)* XI(6))* THETA3
140  CONTINUE
      DO 150 I=1,6
      DO 150 K=1,18
        B1B(I,K)= 0.0
        B2B(I,K)= 0.0
        B3B(I,K)= 0.0
        B4B(I,K)= 0.0
        B5B(I,K)= 0.0
        B6B(I,K)= 0.0
      DO 150 J=1,6
        B1B(I,K) = B1B(I,K) + CRZ(I,J) * B1A(J,K)
        B2B(I,K) = B2B(I,K) + CRZ(I,J) * B2A(J,K)
        B3B(I,K) = B3B(I,K) + CRZ(I,J) * B3A(J,K)
        B4B(I,K) = B4B(I,K) + CRZ(I,J) * B4A(J,K)
        B5B(I,K) = B5B(I,K) + CRZ(I,J) * B5A(J,K)
        B6B(I,K) = B6B(I,K) + CRZ(I,J) * B6A(J,K)
150  CONTINUE
      DO 160 I=1,18
      DO 160 K=1,18
        B(I,K)=0.0
      DO 160 J=1,6
        B(I,K) = B(I,K) + B1(J,I)* B1B(J,K)+B2(J,I)*B2B(J,K)+B3(J,I)*
1          B3B(J,K)+B4(J,I)*B4B(J,K)+B5(J,I)*B5B(J,K)+B6(J,I)*B6B(J,K)
160  CONTINUE
250  CONTINUE
C B(I,K) NOW CONTAINS THE STIFFNESS MATRIX FOR ONE TRIANGULAR ELEMENT
      AR1 = AR1 + XI(1) *THETA
      DO 235 K=1,6
      DO 235 I=1,3
        BS1(K,3*II-3+I) = BS1(K,3*II-3+I) +B1A(K,I )
        BS1(K,3*JJ-3+I) = BS1(K,3*JJ-3+I) +B1A(K,I+3 )
        BS1(K,3*KK-3+I) = BS1(K,3*KK-3+I) +B1A(K,I+6 )
        BS1(K,3*II+I+12)= BS1(K,3*II+12+I)+B1A(K,I+9 )
        BS1(K,3*JJ+I+12)= BS1(K,3*JJ+12+I)+B1A(K,I+12)
        BS1(K,3*KK+I+12)= BS1(K,3*KK+12+I)+B1A(K,I+15)
235  CONTINUE
      IIM = 3* II -3

```

```

JJM = 3* JJ -3
KKM = 3* KK -3
DO 170 K=1,4
DO 170 I=1,3
DO 170 J=1,3
IF(K.EQ.1 .OR. K.EQ.2) I1=I
IF(K.EQ.3 .OR. K.EQ.4) I1=I +9
IF(K.EQ.1 .OR. K.EQ.3) J1=J
IF(K.EQ.2 .OR. K.EQ.4) J1=J      +9
IF(K.EQ.1 .OR. K.EQ.2) K1=0
IF(K.EQ.3 .OR. K.EQ.4) K1=15
IF(K.EQ.1 .OR. K.EQ.3) K2=0
IF(K.EQ.2 .OR. K.EQ.4) K2=15
182 KK2=KKM
    II2=IIM
    JJ2=JJM
180 KK1=KKM
    JJ1=JJM
    II1=IIM
    S1(II1+I+K1,II2+J+K2) = S1(II1+I+K1,II2+J+K2) +B(I1,J1)
    S1(II1+I+K1,JJ2+J+K2) = S1(II1+I+K1,JJ2+J+K2) +B(I1,J1+3)
    S1(II1+I+K1,KK2+J+K2) = S1(II1+I+K1,KK2+J+K2) +B(I1,J1+6)
    S1(JJ1+I+K1,II2+J+K2) = S1(JJ1+I+K1,II2+J+K2) +B(I1+3,J1)
    S1(JJ1+I+K1,JJ2+J+K2) = S1(JJ1+I+K1,JJ2+J+K2) +B(I1+3,J1+3)
    S1(JJ1+I+K1,KK2+J+K2) = S1(JJ1+I+K1,KK2+J+K2) +B(I1+3,J1+6)
    S1(KK1+I+K1,II2+J+K2) = S1(KK1+I+K1,II2+J+K2) + B(I1+6,J1)
    S1(KK1+I+K1,JJ2+J+K2) = S1(KK1+I+K1,JJ2+J+K2) + B(I1+6,J1+3)
    S1(KK1+I+K1,KK2+J+K2) = S1(KK1+I+K1,KK2+J+K2) + B(I1+6,J1+6)
170 CONTINUE
    IF(IFGFL(N,NTP).EQ.0) GO TO 190
    DO 174 I=1,18
        TVP(I)=0.0
    DO 174 J=1,6
174 TVP(I)=TVP(I)+B1A(J,I)*EPSDN(J,N,NTP)
        K=3*II-2
        L=3*JJ-2
        M=3*KK-2
        DO 179 I=1,3
            J=I-1
            P1(K+J)=P1(K+J)-TVP(I)
            P1(K+J+15)=P1(K+J+15)-TVP(I+9)
            P1(L+J)=P1(L+J)-TVP(I+3)
            P1(L+J+15)=P1(L+J+15)-TVP(I+12)
            P1(M+J)=P1(M+J)-TVP(I+6)
179 P1(M+J+15)=P1(M+J+15)-TVP(I+15)
190 CONTINUE
    RETURN
    END
    FUNCTION NODE(I,J)
    COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
    NODE=0
    DO 100 JJ=1,J
        NSTART=IMIN(JJ)
        NSTOP=IMAX(JJ)
        DO 100 II=NSTART,NSTOP
            NODE=NODE+1
            IF(JJ.EQ.J.AND.II.EQ.I) RETURN
100 CONTINUE
    RETURN
    END

```

```

SUBROUTINE POINTS
INTEGER CODE
COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
1NUMST
COMMON/MATP/RO(6),E(12,16,6),EE(16),AOFTS(6)
COMMON/NPDATA/R(10 ),CODE(10 ),XR(10 ),Z(10 ),XZ(10 ),
1NPNUM( 4, 8),T(10 ),XT(10 )
COMMON/ELDATA/BETA(10 ),EPR(10 ),PR(4 ),SH(4 ),IX(8 ,5),
1IP(4 ),JP(4 ),IS(4 ),JS(4 ),ALPHA(10 ),IT(4 ),JT(4 ),
2ST(4 )
COMMON/SOLVE/X(888),Y(888),TEM(888),NUMTC,MBAND
COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
COMMON/PLANE/NPP
DIMENSION AR( 4, 8),AZ( 4, 8),MATRIL(100,5),BLKANG(100),BLKALF(1
100)
DIMENSION IBNG(100),NBNG(100)
EQUIVALENCE (R(1),AR),(Z(1),AZ)
C ESTABLISH NODAL POINT INFORMATION
C* * * * *
NEL=0
NODSUM=0
DO 100 J=1,MAXJ
NSTART=IMIN(J)
NSTOP=IMAX(J)
DO 100 I=NSTART,NSTOP
100 NODSUM=NODSUM+1
NELSUM=0
JJMAX=MAXJ-1
DO 110 JJ=1,JJMAX
NSTOP=MINO(IMAX(JJ),IMAX(JJ+1))-1
NSTART=MAXO(IMIN(JJ),IMIN(JJ+1))
DO 110 II=NSTART,NSTOP
110 NELSUM=NELSUM+1
NUMNP=NODSUM
NUMEL=NELSUM
WRITE(15)NUMEL,NUMNP
DO 120 J=1,MAXJ
NSTART=IMIN(J)
NSTOP=IMAX(J)
DO 120 I=NSTART,NSTOP
NPNUM(I,J)=NODE(I,J)
NP=NPNUM(I,J)
R(NP)=AR(I,J)
120 Z(NP)=AZ(I,J)
C* * * * *
C READ AND ASSIGN BOUNDARY CONDITIONS
C* * * * *
C INITIALIZE
C* * * * *
DO 130 I=1,NUMNP
CODE(I)=0
IF(R(I).EQ.0..AND.NPP.EQ.0) CODE(I)=1.
XR(I)=0.
XZ(I)=0.
XT(I)=0.0
130 T(I)=0.
IF(NBC.EQ.0) GO TO 210
DO 200 IBCON=1,NBC
READ(5,1002) I1,I2,J1,J2,ICN,RCON,ZCON,TCON
DO 200 I=I1,I2

```

```

DO 200 J=J1,J2
NP=NPNUM(I,J)
CODE(NP)=ICN
XR(NP)=RCON
XT(NP)=TCON
XZ(NP)=ZCON
200 CONTINUE
210 MPRINT=0
WRITE(15)(CODE(I),I=1,NUMNP)
WRITE(15)(XR(I),I=1,NUMNP)
WRITE(15)(XT(I),I=1,NUMNP)
WRITE(15)(XZ(I),I=1,NUMNP)
DO 230 J=1,MAXJ
NSTART=IMIN(J)
NSTOP=IMAX(J)
DO 230 I=NSTART,NSTOP
NP=NPNUM(I,J)
IF(MPRINT.NE.0) GO TO 220
WRITE(6,2000)
MPRINT=59
220 MPRINT=MPRINT-1
230 WRITE(6,2001) I,J,NP,CODE(NP),R(NP),Z(NP),XR(NP),XZ(NP),XT(NP)
C* * * * *
C ASSIGN MATERIALS IN BLOCKS
C* * * * *
DO 300 M1=1,NUMEL
300 IX(M1,5)=0
DO 310 IMTL=1,NMTL
READ(5,1000) MTL,(MATRIL(IMTL,IM),IM=2,5),BLKANG(IMTL),BLKALF(IMT
1L),IBNG(IMTL),NBNG(IMTL)
310 MATRIL(IMTL,1)=MTL
C* * * * *
C ESTABLISH ELEMENT INFORMATION
C* * * * *
JJMAX=MAXJ-1
N=0
MTL=1
KTL=1
DO 440 JJ=1,JJMAX
NSTOP=MINO(IMAX(JJ),IMAX(JJ+1))-1
NSTART=MAXO(IMIN(JJ),IMIN(JJ+1))
DO 440 II=NSTART,NSTOP
NEL=NEL+1
DO 400 IMTL=1,NMTL
IF(II.LT.MATRIL(IMTL,2)) GO TO 400
IF(II.GE.MATRIL(IMTL,3)) GO TO 400
IF(JJ.LT.MATRIL(IMTL,4)) GO TO 400
IF(JJ.GE.MATRIL(IMTL,5)) GO TO 400
KAT=IMTL
MAT=MATRIL(IMTL,1)
400 CONTINUE
IF(KAT.EQ.KTL) GO TO 410
KTL=KAT
MTL=MAT
GO TO 420
410 IF(II.EQ.NSTART) GO TO 420
IF(JJ.NE.JJMAX.OR.II.NE.NSTOP) GO TO 440
M=NEL+1
IANG=ICNG
NANG=NCNG

```

```

      GO TO 421
420  I=NPNUM( II,JJ )
      J=I+1
      K=NPNUM( II+1,JJ+1 )
      L=K-1
      M=NEL
      IX( M,1 )=I
      IX( M,2 )=J
      IX( M,3 )=K
      IX( M,4 )=L
      IX( M,5 )=MTL
      BETA( M )=BLKANG( KTL )
      ALPHA( M )=BLKALF( KTL )
      IANG=ICNG
      NANG=NCNG
      ICNG=IBNG( KTL )
      NCNG=NBNG( KTL )
421  NC=2
430  N=N+1
      IF( M.LE.N ) GO TO 440
      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 )
      BETA( N )=BETA( N-1 )
      IF( IANG.EQ.1 ) GO TO 442
      ALPHA( N )=ALPHA( N-1 )
      GO TO 443
442  CONTINUE
      IF( NC.GT.NANG ) GO TO 444
      ALPHA( N )=ALPHA( N-1 )
      GO TO 443
444  NC=1
      ALPHA( N )=-ALPHA( N-1 )
443  CONTINUE
      NC=NC+1
      IF( M.GT.N ) GO TO 430
440  CONTINUE
      IF( NUMNP.GT.2000 ) WRITE( 6,2002 )
C * * * * *
C   SET NODAL POINT TEMPERATURE TO REFERENCE TEMPERATURE
C * * * * *
      IF( NUMTC.NE.0 ) RETURN
      DO 500 N=1,NUMNP
500  T( N )=TREF
1000 FORMAT ( 5I5,2F10.0,2I5 )
1002 FORMAT( 4I5,I10,3F10.0 )
2000 FORMAT ( 128H1   I   J   NP           TYPE   R-ORDINATE   Z-ORDINA
1TE  R LOAD OR DISPLACEMENT  Z LOAD OR DISPLACEMENT  T LOAD OR DISP
2LACEMENT )
2001 FORMAT ( 2I5,I6,I12,F13.6,F14.6,E26.7,E24.7,E24.7 )
2002 FORMAT ( 35H  BAD INPUT - TOO MANY NODAL POINTS )
      RETURN
      END
      SUBROUTINE QUAD
      INTEGER CODE
      REAL NUSN,NUTN,NUTS,NUNS,NUNT,NUST
      DIMENSION DUMMY( 6,6 ),DUMMY1( 6,6 )
      COMMON/PLAS/ALFA( 6, 4,8 ),SIGYLD( 7,6,8 ),IFGPL( 4,8 )

```

```

COMMON/ARG1/SIG1(18),EPS1(18),DEFSF(12),CEFSF(6,6)
COMMON/NXIATA/NTP,NTS,NTOTS,GTSG(24,24,8)
COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
1NUMST
COMMON/NXQUAD/AR1
COMMON/NONAXI/S1(30,30),P1(30),THETA,BS1(6,30)
COMMON/MATP/RO(6),E(12,16,6),EE(16),ADFTS(6)
COMMON/NPDATA/R(10),CODE(10),XR(10),Z(10),XZ(10),
1INPNUM(4,8),T(10),XT(10)
COMMON/ELDATA/BETA(10),EPR(10),PR(4),SH(4),IX(8,5),
1IP(4),JP(4),IS(4),JS(4),ALPHA(10),IT(4),JT(4),
2ST(4)
COMMON/ARG/RRR(5),ZZZ(5),RR(4),ZZ(4),S(15,15),P(15),TT(6),
1H(6,15),CRZ(6,6),XI(10),ANGLE(4),SIG(18),EPS(18),N
COMMON/INCR/NOLINC,NOL,INERT,NUMMAT,SIGTOT(12,4,8)
COMMON/RESULT/BS(6,15),D(6,6),C(6,6),AR,BB(6,9),CNS(6,6)
COMMON/PLANE/NPP
COMMON/DUM1/SITEM(3,30),S1T(24,24),TS(6,24)
DIMENSION S2T(24,6)
DIMENSION BS1T(6,3),P1T(3),P1TT(24)
I1=IX(N,1)
J1=IX(N,2)
K1=IX(N,3)
L1=IX(N,4)
M1=IX(N,5)
IX(N,5)=-IX(N,5)
C* * * * *
C INTERPOLATE MATERIAL PROPERTIES
C* * * * *
DO 100 I=1,12
100 EE(I)=E(1,I+1,M1)
DO 110 I=1,6
DO 110 J=1,6
CNS(I,J)=0.00
C(I,J)=0.00
110 D(I,J)=0.00
C* * * * *
C FORM STRESS-STRAIN RELATIONSHIP IN N-S-T SYSTEM
C* * * * *
NUNS=EE(4)
NUNT=EE(5)
NUST=EE(6)
NUSN=(EE(2)*NUNS)/EE(1)
NUTN=(EE(3)*NUNT)/EE(1)
NUTS=(EE(3)*NUST)/EE(2)
DIV=1.00-NUNS*NUSN-NUST*NUTS-NUNT*NUTN-NUSN*NUNT*NUTS
1-NUNS*NUTN*NUST
CNS(1,1)=EE(1)*(1.00-NUST*NUTS)/DIV
CNS(1,2)=EE(2)*(NUNS+NUNT*NUTS)/DIV
CNS(1,3)=EE(3)*(NUNT+NUNS*NUST)/DIV
CNS(2,1)=CNS(1,2)
CNS(2,2)=EE(2)*(1.00-NUNT*NUTN)/DIV
CNS(2,3)=EE(3)*(NUST+NUSN*NUNT)/DIV
CNS(3,1)=CNS(1,3)
CNS(3,2)=CNS(2,3)
CNS(3,3)=EE(3)*(1.00-NUNS*NUSN)/DIV
CNS(4,4)=EE(7)
CNS(5,5)=EE(8)
CNS(6,6)=EE(9)
DO 162 I=1,6

```

```

      DO 162 J=1,6
162  CEPSP(I,J)=0.0
C    IF (IFGPL(N,NTP).NE.0)CALL ELPLSS(MTYPE)
C    SET UP STRAIN TRANSFORM TO N-S-T SYSTEM
      SINA=SIN( ALPHA(N))
      COSA=COS( ALPHA(N))
      S2=SINA**2
      C2=COSA**2
      SC=SINA*COSA
      D( 1,1)=C2
      D( 1,3)=S2
      D( 1,6)=-SC
      D( 2,1)=S2
      D( 2,3)=C2
      D( 2,6)=SC
      D( 3,2)=1.00
      D( 4,1)=2.00*SC
      D( 4,3)=-2.00*SC
      D( 4,6)=C2-S2
      D( 5,4)=SINA
      D( 5,5)=COSA
      D( 6,4)=COSA
      D( 6,5)=-SINA
C    SET UP STRAIN TRANSFORMATION TO R-Z-T SYSTEM
      SINB=SIN( BETA(N))
      COSB=COS( BETA(N))
      S2=SINB**2
      C2=COSB**2
      SC=SINB*COSB
      C( 1,1)=S2
      C( 1,2)=C2
      C( 1,4)=SC
      C( 2,1)=C2
      C( 2,2)=S2
      C( 2,4)=-SC
      C( 3,3)=1.00
      C( 4,1)=-2.00*SC
      C( 4,2)=2.00*SC
      C( 4,4)=S2-C2
      C( 5,5)=SINB
      C( 5,6)=-COSB
      C( 6,5)=COSB
      C( 6,6)=SINB
      IF (IFGPL(N,NTP).NE.0)CALL ELPLSS(MTYPE)
C    CALCULATE CRZ MATRIX
      DO 120 I=1,6
      DO 120 J=1,6
      DUMMY(I,J)=0.00
      DO 120 K=1,6
120  DUMMY(I,J)=DUMMY(I,J)+D( I,K)*C( K,J)
      DO 130 I=1,6
      DO 130 J=1,6
      DUMMY1(I,J)=0.00
      DO 130 K=1,6
130  DUMMY1(I,J)=DUMMY1(I,J)+CNS( I,K)*DUMMY(K,J)
      DO 140 I=1,6
      DO 140 J=1,6
      DUMMY(I,J)=0.00
      DO 140 K=1,6
140  DUMMY(I,J)=DUMMY(I,J)+D( K,I)*DUMMY1(K,J)

```

```

      DO 160 I=1,6
      DO 150 J=1,6
      CRZ(I,J)=0.00
      DO 150 K=1,6
150  CRZ(I,J)=CRZ(I,J)+C(K,I)*DUMMY(K,J)
      TT(I)=0.00
      DO 160 M=1,6
      P(M)=0.00
      DO 161 II=1,3
      IF(AOFTS(MTYPE).EQ.1.) P(M)=CNS(M,II)*EE(II+9)
161  P(M)=P(M)+(T(N)-TREF)*CNS(M,II)*EE(II+9)
      DO 160 K=1,6
160  TT(I)=TT(I)+C(K,I)*D(M,K)*P(M)
C
C   FORM QUADRILATERAL STIFFNESS MATRIX
      RRR(5)=(R(I1)+R(J1)+R(K1)+R(L1))/4.
      ZZZ(5)=(Z(I1)+Z(J1)+Z(K1)+Z(L1))/4.
      DO 200 M=1,4
      MM=IX(N,M)
      IF(NPP.NE.0) GO TO 190
      IF(R(MM).EQ.0..AND.CODE(MM).EQ.0.)CODE(MM)=1.
190  RRR(M)=R(MM)
200  ZZZ(M)=Z(MM)
      DO 220 II=1,15
      P1(II)=0.0
      P1(II+15)=0.0
      P(II)=0.00
      DO 220 JJ=1,15
220  S(II,JJ)=0.00
      VOL=0.
      DO 90 I=1,6
      DO 90 J=1,15
      BS1(I,J)=0.0
      BS1(I,J+15)=0.0
90  BS(I,J)=0.00
      AR=0.00
240  CALL TRISTF(4,1,5)
      CALL TRISTF(1,2,5)
      CALL TRISTF(2,3,5)
      CALL TRISTF(3,4,5)
      DO 91 I=1,6
      DO 91 J=1,15
91  BS(I,J)=BS(I,J)/AR
      DO 300 I=1,30
      DO 300 J=1,30
300  S1(I,J)=0.0
      AR1=0.0
      CALL NAXSTF(4,1,5)
      CALL NAXSTF(1,2,5)
      CALL NAXSTF(2,3,5)
      CALL NAXSTF(3,4,5)
      DO 310 I=1,6
      DO 310 J=1,30
310  BS1(I,J)=BS1(I,J)/AR1
      DO 320 I=1,6
      DO 320 J=1,3
320  BS1T(I,J)=BS1(I,J+12)
      DO 325 I=1,6
      DO 325 J=1,12
325  BS1(I,J+12)=BS1(I,J+15)

```

```

      DO 330 I=1,6
      DO 330 J=1,3
330  BS1(I,J+24) = BS1T(I,J)
      DO 340 I=1,3
340  P1T(I) = P1(I+12)
      DO 341 I=1,12
341  P1(I+12) = P1(I+15)
      DO 342 I=1,3
342  P1(I+24) = P1T(I)
      DO 149 I=1,3
      DO 149 J=1,30
149  S1TEM(I,J) = S1(I+12,J)
      DO 151 I=1,12
      DO 151 J=1,30
151  S1(I+12,J) = S1(I+15,J)
      DO 152 I=1,3
      DO 152 J=1,30
152  S1(I+24,J) = S1TEM(I,J)
      DO 153 I=1,3
      DO 153 J=1,30
153  S1TEM(I,J) = S1(J,I+12)
      DO 154 J=1,12
      DO 154 I=1,30
154  S1(I,J+12) = S1(I,J+15)
      DO 155 I=1,3
      DO 155 J=1,30
155  S1(J,I+24) = S1TEM(I,J)
      DO 251 I=1,6
      DO 251 J=1,24
251  TS(I,J) = 0.0
      DO 252 I=1,3
      DO 252 J=1,4
      TS(I,I+(J-1)*3) = 0.250
252  TS(I+3,I+12+(J-1)*3) = 0.250
      DO 253 I=1,24
      DO 253 J=1,24
      S1T(I,J) = 0.00
      DO 253 K=1,6
253  S1T(I,J) = S1T(I,J) + S1(I,24+K)*TS(K,J)
      DO 254 I=1,24
      DO 254 J=1,24
254  S1(I,J) = S1(I,J) + S1T(I,J) + S1T(J,I)
      DO 255 I=1,24
      DO 255 J=1,6
      S2T(I,J) = 0.0
      DO 255 K=1,6
255  S2T(I,J) = S2T(I,J) + TS(K,I)*S1(K+24,J+24)
      DO 256 I=1,24
      DO 256 J=1,24
      S1T(I,J) = 0.0
      DO 256 K=1,6
256  S1T(I,J) = S1T(I,J) + S2T(I,K)*TS(K,J)
      DO 257 I=1,24
      DO 257 J=1,24
257  S1(I,J) = S1(I,J) + S1T(I,J)
      DO 258 I=1,24
      P1TT(I) = 0.0
      DO 258 K=1,6
258  P1TT(I) = P1TT(I) + TS(K,I)*P1(K+24)
      DO 259 I=1,24

```

```

259 P1(I)=P1(I)+P1TT(I)
      RETURN
      END
      SUBROUTINE SOLV
      COMMON/ELDATA/BETA(10 ),EPR(10 ),PR(4 ),SH(4 ),IX(8 ),5),
      1IP(4 ),JP(4 ),IS(4 ),JS(4 ),ALPHA(10 ),IT(4 ),JT(4 ),
      2ST(4 )
      COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
      1NUMST
      COMMON/NONAXI/S1(30,30),P1(30),THETA,BS1(6,30)
      COMMON/NXDATA/NTP,NTS,NTOTS,GTS1G(24,24,8)
      COMMON/SOLVE/B( 72),A( 72,36),NUMBLK,MBAND
      MM=MBAND
      NN=36
      NL=NN+1
      NH=NN+NN
      REWIND 1
      REWIND 2
      NB=0
      GO TO 150
C* * * * *
C      REDUCE EQUATIONS BY BLOCKS
C* * * * *
C      1. SHIFT BLOCK OF EQUATIONS
C
      100 NB=NB+1
      DO 125 N=1,NN
      NM=NN+N
      B(N)=B(NM)
      B(NM)=0.00
      DO 125 M=1,MM
      A(N,M)=A(NM,M)
      125 A(NM,M)=0.00
C
C      2. READ NEXT BLOCK OF EQUATIONS INTO CORE
C
      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
C
C      3. REDUCE BLOCK OF EQUATIONS
C
      200 DO 300 N=1,NN
      IF(A(N,1).EQ.0.00) GO TO 300
      B(N)=B(N)/A(N,1)
      DO 275 L=2,MM
      IF(A(N,L).EQ.0.00) 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
C
C      4. WRITE BLOCK OF REDUCED EQUATIONS ON FORTRAN UNIT 1

```

```

C
  IF(NUMBLK.EQ.NB) GO TO 400
  WRITE (1) (B(N),(A(N,M),M=2,MM),N=1,NN)
  GO TO 100
C* * * * *
C  BACK-SUBSTITUTION
C* * * * *
  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
C* * * * *
C  ORDER FORMER UNKNOWN IN B ARRAY
C* * * * *
  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)
C* * * * *
C  WRITE SOLUTION
C* * * * *
      NN12 = 3*NUMNP
  1500 FORMAT(" ",5I10)
      WRITE(26) (B(I),I=1,NN12)
      WRITE(26)((IX(I,J),J=1,5),I=1,NUMEL)
      MPRINT=0
      DO 710 N=1,NUMNP
          IF(MPRINT.NE.0) GO TO 700
          WRITE (6,2000)
          MPRINT=59
  700 MPRINT=MPRINT-1
  710 WRITE (6,2001) N,B(3*N-2),B(3*N-1),B(3*N)
  2000 FORMAT (13H1 NODAL POINT,18X,2HUR,18X,2HUZ,18X,2HUT)
  2001 FORMAT (I13,3E20.7)
      RETURN
      END
      SUBROUTINE STIFF
      INTEGER CODE
      COMMON/RESULT/BS(6,15),D(6,6),C(6,6),AR,BB(6,9),CNS(6,6)
      COMMON/ARG1/SIG1(18),EPS1(18),DEPSP(12),CEPSP(6,6)
      COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
      1NUMST
      COMMON/ELDATA/BETA(10 ),EPR(10 ),PR(4 ),SH(4 ),IX(8 ,5),
      1IP(4 ),JP(4 ),IS(4 ),JS(4 ),ALPHA(10 ),IT(4 ),JT(4 ),
      2ST(4 )
      COMMON/NPDATA/R(10 ),CODE(10 ),XR(10 ),Z(10 ),XZ(10 ),
      1NPNUM( 4, 8),T(10 ),XT(10 )
      COMMON/SOLVE/B( 72),A( 72,36),NUMBLK,MBAND

```

```

COMMON/NXDATA/NTP,NTS,NTOTS,GTS1G(24,24,8)
COMMON/ANS4/FT(24,4),GTS1U(24),GTS1UT(24,4)
COMMON/ARG/RRR(5),ZZZ(5),RR(4),ZZ(4),S(15,15),P(15),TT(6),
1H(6,15),CRZ(6,6),XI(10),ANGLE(4),SIG(18),EPS(18),N
COMMON/NONAXI/S1(30,30),P1(30),THETA,BS1(6,30)
COMMON/PLANE/NPP
COMMON/ANS2/GTP1(24),G(24,24),GTS1(24,24),GTS1GE(24,24)
COMMON/EUM1/SITEM(3,30),S1T(24,24),TS(6,24)
COMMON/RATE/DKPR,SIGPR,BVR,EVR,PSRATE(10,8),NRATE
DIMENSION LM(4),S2(12,3),S3(3,12),S4(3,3),S5(12,3),S6(12,12)
C* * * * *
C  INITIALIZATION
NRATE=1
REWIND 2
REWIND 3
NB=12
ND=3*NB
ND2=2*ND
STOP=0.
NUMBLK=0
DO 100 N=1,ND2
B(N)=0.00
DO 100 M=1,ND
100 A(N,M)=0.00
DO 50 I=1,24
FT(I,NTP)=0.0
GTS1UT(I,NTP)=0.0
DO 50 J=1,24
50 GTS1G(I,J,NTP)=0.0
C* * * * *
C  FORM STIFFNESS MATRIX IN BLOCKS
C* * * * *
200 NUMBLK=NUMBLK+1
NH=NB*(NUMBLK+1)
NM=NH-NB
NL=NM-NB+1
KSHIFT=3*NL-3
DO 340 N=1,NUMEL
IF(IX(N,5).LE.0) GO TO 340
DO 210 I=1,4
IF(IX(N,I).LT.NL) GO TO 210
IF(IX(N,I).LE.NM) GO TO 220
210 CONTINUE
GO TO 340
220 CALL QUAD
IF(VOL.GT.0.) GO TO 230
WRITE(6,2000) N
STOP=1.
230 IF(IX(N,3).EQ.IX(N,4)) GO TO 300
DO 231 II=1,3
DO 231 JJ=1,3
231 S4(II,JJ)=S(II+12,JJ+12)
CALL SYMINV(S4,3)
DO 232 II=1,12
DO 232 JJ=1,3
232 S2(II,JJ)=S(II,JJ+12)
DO 233 II=1,3
DO 233 JJ=1,12
233 S3(II,JJ)=S(II+12,JJ)
DO 240 I=1,12

```

```

DO 240 J=1,3
S5(I,J)=0.00
DO 240 K=1,3
240 S5(I,J) = S5(I,J) + S2(I,K) * S4(K,J)
DO 241 I=1,12
DO 241 J=1,12
S6(I,J)=0.00
DO 241 K=1,3
241 S6(I,J) = S6(I,J) + S5(I,K) * S3(K,J)
DO 234 II=1,12
DO 234 JJ=1,3
234 P(II)=P(II)-S5(II,JJ)*P(JJ+12)
DO 235 II=1,12
DO 235 JJ=1,12
235 S(II,JJ)=S(II,JJ)-S6(II,JJ)
DO 259 I=1,24
DO 259 J=1,24
259 G(I,J) = 0.0
DO 260 K=1,4
DO 260 I=1,3
G(K*3-3+I,I*4-3) = 1.0
G(K*3-3+I,I*4-2) = RRR(K)
G(K*3-3+I,I*4-1) = ZZZ(K)
260 G(K*3-3+I,I*4 ) = ZZZ(K) *RRR(K)
DO 262 I=1,12
DO 262 J=1,12
262 G(I+12,J+12) = G(I,J)
NTP20 = 21
WRITE(NTP20)(( CRZ(I,J),J=1,6),I=1,6)
WRITE(NTP20)(( BS1(I,J),J=1,30),I=1,6)
WRITE(NTP20)(( G(I,J),J=1,24),I=1,24)
WRITE(NTP20)(( CEPSP(I,J),J=1,6),I=1,6)
WRITE(NTP20)(( CNS(I,J),J=1,6),I=1,6)
WRITE(NTP20)(( D(I,J),J=1,6),I=1,6)
WRITE(NTP20)(( C(I,J),J=1,6),I=1,6)
DO 280 I=1,24
GTP1(I)=0.0
DO 280 K=1,24
GTS1(I,K) = 0.0
GTP1(I)= GTP1(I)+ G(K,I)*P1(K)
DO 280 J=1,24
280 GTS1(I,K) = GTS1(I,K) + G(J,I) * S1(J,K)
WRITE(3)(( GTS1(I,J),J=1,24),I=1,24)
DO 281 I=1,24
FT(I,NTP) =FT(I,NTP) + GTP1(I)
DO 281 J=1,24
GTS1GE(I,J) = 0.0
DO 281 K=1,24
281 GTS1GE(I,J) = GTS1GE(I,J)+ GTS1(I,K) *G(K,J)
DO 282 I=1,24
DO 282 J=1,24
282 GTS1G(I,J,NTP) = GTS1G(I,J,NTP) + GTS1GE(I,J)
C* * * * *
C ADD ELEMENT STIFFNESS MATRIX TO BODY STIFFNESS MATRIX
C* * * * *
300 DO 310 I=1,4
310 LM(I)=3*IX(N,I)-3
DO 330 I=1,4
DO 330 K=1,3
II=LM(I)+K-KSHIFT

```

```

KK=3*I-3+K
B(II)=B(II)+P(KK)
DO 330 J=1,4
DO 330 L=1,3
JJ=LM(J)+L-II+1-KSHIFT
LL=3*J-3+L
IF(JJ.LE.0) GO TO 330
IF(ND.GE.JJ) GO TO 320
WRITE(6,2001) N
STOP=1.
GO TO 340
320 A(II,JJ)=A(II,JJ)+S(KK,LL)
330 CONTINUE
340 CONTINUE
C* * * * *
C   ADD CONCENTRATED FORCES
C* * * * *
DO 400 N=NL,NM
IF(N.GT.NUMNP) GO TO 500
K=3*N-KSHIFT
B(K)=B(K)+XT(N)
B(K-1)=B(K-1)+XZ(N)
400 B(K-2)=B(K-2)+XR(N)
C* * * * *
C   ADD FRESSURE BOUNDARY CONDITIONS
C* * * * *
500 IF(NUMPC.EQ.0) GO TO 600
DO 540 L=1,NUMPC
I=IP(L)
J=JP(L)
PP=PR(L)/6.
DR=(R(J)-R(I))*PP
DZ=(Z(I)-Z(J))*PP
RX=2.*R(I)+R(J)
ZX=R(I)+2.*R(J)
II=3*I-KSHIFT-1
JJ=3*J-KSHIFT-1
IF(II.LE.0.OR.II.GT.ND) GO TO 520
SINA=0.
COSA=1.
510 B(II-1)=B(II-1)+RX*(COSA*DZ+SINA*DR)
GR=RX*(COSA*DZ+SINA*DR)*THETA/2.0
FT(1,NTP)=FT(1,NTP)+GR
FT(2,NTP)=FT(2,NTP)+R(I)*GR
FT(3,NTP)=FT(3,NTP)+Z(I)*GR
FT(4,NTP)=FT(4,NTP)+R(I)*Z(I)*GR
FT(14,NTP)=FT(14,NTP)+R(I)*GR
FT(13,NTP)=FT(13,NTP)+GR
FT(15,NTP)=FT(15,NTP)+Z(I)*GR
FT(16,NTP)=FT(16,NTP)+Z(I)*R(I)*GR
B(II)=B(II)-RX*(SINA*DZ-COSA*DR)
GZ=-RX*(SINA*DZ-COSA*DR)*THETA/2.0
FT(5,NTP)=FT(5,NTP)+GZ
FT(6,NTP)=FT(6,NTP)+R(I)*GZ
FT(7,NTP)=FT(7,NTP)+Z(I)*GZ
FT(8,NTP)=FT(8,NTP)+Z(I)*R(I)*GZ
FT(17,NTP)=FT(17,NTP)+GZ
FT(18,NTP)=FT(18,NTP)+R(I)*GZ
FT(19,NTP)=FT(19,NTP)+Z(I)*GZ
FT(20,NTP)=FT(20,NTP)+Z(I)*R(I)*GZ

```

```

520 IF(JJ.LE.0.OR.JJ.GT.ND) GO TO 540
    SINA=0.
    COSA=1.
530 B(JJ-1)=B(JJ-1)+ZX*(COSA*DZ+SINA*DR)
    GR= ZX *(COSA*DZ+SINA*DR) *THETA/2.0
    FT(1,NTP)=FT(1,NTP)+GR
    FT(2,NTP)=FT(2,NTP)+R(J)*GR
    FT(3,NTP)=FT(3,NTP)+Z(J)*GR
    FT(4,NTP)=FT(4,NTP)+Z(J)*R(J)*GR
    FT(13,NTP)=FT(13,NTP)+GR
    FT(14,NTP)=FT(14,NTP)+R(J)*GR
    FT(15,NTP)=FT(15,NTP)+Z(J)*GR
    FT(16,NTP)=FT(16,NTP)+Z(J)*R(J)*GR
    B(JJ)=B(JJ)-ZX*(SINA*DZ-COSA*DR)
    GZ= -ZX*(SINA*DZ-COSA*DR) *THETA/2.0
    FT(5,NTP)=FT(5,NTP)+GZ
    FT(6,NTP)=FT(6,NTP)+R(J)*GZ
    FT(7,NTP)=FT(7,NTP)+Z(J)*GZ
    FT(8,NTP)=FT(8,NTP)+Z(J)*R(J)*GZ
    FT(17,NTP)=FT(17,NTP)+GZ
    FT(18,NTP)=FT(18,NTP)+R(J)*GZ
    FT(19,NTP)=FT(19,NTP)+Z(J)*GZ
    FT(20,NTP)=FT(20,NTP)+Z(J)*R(J)*GZ
540 CONTINUE
1100 FORMAT(" ",12E10.3)
C* * * * *
C  ADD SHEAR BOUNDARY CONDITIONS
C* * * * *
600 IF(NUMSC.EQ.0) GO TO 701
    DO 640 L=1,NUMSC
        I=IS(L)
        J=JS(L)
        SS=SH(L)/6.
        DZ=(Z(I)-Z(J))*SS
        DR=(R(J)-R(I))*SS
        RX=2.*R(I)+R(J)
        ZX=R(I)+2.*R(J)
        II=3*I-KSHIFT-1
        JJ=3*J-KSHIFT-1
        IF(II.LE.0.OR.II.GT.ND) GO TO 620
        SINA=0.
        COSA=1.
610 B(II-1)=B(II-1)+RX*(SINA*DZ+COSA*DR)
        GR= RX*(SINA*DZ+COSA*DR) *THETA/2.0
        FT(1,NTP)=FT(1,NTP)+GR
        FT(2,NTP)=FT(2,NTP)+R(I)*GR
        FT(3,NTP)=FT(3,NTP)+Z(I)*GR
        FT(4,NTP)=FT(4,NTP)+Z(I)*R(I)*GR
        FT(13,NTP)=FT(13,NTP)+GR
        FT(14,NTP)=FT(14,NTP)+R(I)*GR
        FT(15,NTP)=FT(15,NTP)+Z(I)*GR
        FT(16,NTP)=FT(16,NTP)+Z(I)*R(I)*GR
        B(II)=B(II)-RX*(COSA*DZ-SINA*DR)
        GZ= -RX*(COSA*DZ-SINA*DR) *THETA/2.0
        FT(5,NTP)=FT(5,NTP)+GZ
        FT(6,NTP)=FT(6,NTP)+R(I)*GZ
        FT(7,NTP)=FT(7,NTP)+Z(I)*GZ
        FT(8,NTP)=FT(8,NTP)+Z(I)*R(I)*GZ
        FT(17,NTP)=FT(17,NTP)+GZ
        FT(18,NTP)=FT(18,NTP)+R(I)*GZ

```



```

C      ADD DISPLACEMENT BOUNDARY CONDITIONS
C* * * * *
700 DO 750 M=NL,NH
      IDM=0
      IF(M.GT.NUMNP) GO TO 750
      IF(CODE(M).GT.3) GO TO 751
      U=XR(M)
      N=3*M-2-KSHIFT
752 IF(CODE(M)) 740,750,710
710 IF(CODE(M).EQ.1) GO TO 720
      IF(CODE(M).EQ.2) GO TO 740
      IF(CODE(M).EQ.3) GO TO 730
      GO TO 740
720 CALL MODIFY(ND2,N,U)
      CODE(M)=CODE(M)+IDM
      GO TO 750
730 CALL MODIFY(ND2,N,U)
740 U=XZ(M)
      N=N+1
      CALL MODIFY(ND2,N,U)
      CODE(M)=CODE(M)+IDM
      GO TO 750
751 IDM=IDM+4
      U=XT(M)
      N=3*M-KSHIFT
      CALL MODIFY(ND2,N,U)
      U=XR(M)
      N=3*M-2-KSHIFT
      IF(CODE(M).EQ.4) GO TO 750
      CODE(M)=CODE(M)-4
      GO TO 752
750 CONTINUE
C* * * * *
C      WRITE BLOCK OF EQUATIONS ON FORTRAN UNIT AND SHIFT UP LOWER BLOCK
C* * * * *
      WRITE(2)(B(N),(A(N,M),M=1,MBAND),N=1,ND)
      DO 800 N=1,ND
        K=N+ND
        B(N)=B(K)
        B(K)=0.00
        DO 800 M=1,ND
          A(N,M)=A(K,M)
800 A(K,M)=0.00
C* * * * *
C      CHECK FOR LAST BLOCK
C* * * * *
      IF(NM.LT.NUMNP) GO TO 200
      IF(STOP.NE.0.) STOP
2000 FORMAT(27H NEGATIVE AREA ELEMENT NO.,I4)
2001 FORMAT(46H BAND WIDTH EXCEEDS ALLOWABLE FOR ELEMENT NO.,I4)
      RETURN
      END
      SUBROUTINE STORE
      INTEGER CODE
      COMMON/ANS4/ FT(24,4),GTS1U(24),GTS1UT(24,4)
      COMMON/NXDATA/NTP,NTS,NTOTS,GTS1G(24,24,8)
      COMMON/NXMESH/THETAN(8),NPC(8,8)
      COMMON/NPDATA/R(10),CODE(10),XR(10),Z(10),XZ(10),
1NPNUM(4,8),T(10),XT(10)
      COMMON/SOLVE/B(72),A(72,36),NUMBLK,MBAND

```

```

COMMON/BLRSEG/FI(24,8),FE(24,8),UC(24,8),SK(24,24,8)
COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
1NUMST
COMMON/ANS2/LW(24),R1(24,24),SK1(24,24),DUMH(24,24)
DIMENSION MW(24)
DO 50 I=1,24
DO 50 J=1,24
50 R1(I,J) = 0.0
NS = NTP
DO 110 KK = 1,4
NP1 = NPC(NS,KK)
NP2 = NPC(NS,KK+4)
DO 110 I= 1,3
R1(3*(KK-1)+I ,I*4-3 ) = 1.0
R1(3*(KK-1)+I ,I*4-2 ) = R(NP1)
R1(3*(KK-1)+I ,I*4-1 ) = Z(NP1)
R1(3*(KK-1)+I ,I*4 ) = R(NP1) * Z(NP1)
R1(3*(KK-1)+I+12,I*4+9 ) = 1.0
R1(3*(KK-1)+I+12,I*4+10) = R(NP2)
R1(3*(KK-1)+I+12,I*4+11) = Z(NP2)
R1(3*(KK-1)+I+12,I*4+12) = R(NP2)* Z(NP2)
UC(3*(KK-1)+I,NS) = B(3*NP1-3+I)
UC(3*(KK-1)+I+12,NS) = B(3*NP2-3+I)
110 CONTINUE
CALL MINV(R1,24,D1,LW,MW)
WRITE(25) ((R1(I,J),J=1,24),I=1,24)
DO 115 I=1,24
FE(I,NS)=0.0
FI(I,NS)=0.0
DO 115 J=1,24
FE(I,NS) = FE(I,NS)+ R1(J,I)*FT(J,NTP)
FI(I,NS) = FI(I,NS) + R1(J,I)*GTS1UT(J,NTP)
SK1(I,J) = 0.00
DO 115 K=1,24
SK1(I,J) = SK1(I,J) + R1(K,I) * GTS1G(K,J,NTP)
115 CONTINUE
DO 120 I=1,24
DO 120 J=1,24
SK(I,J,NS) =0.0
DO 120 K=1,24
SK(I,J,NS) = SK(I,J,NS) + SK1(I,K) * R1(K,J)
120 CONTINUE
RETURN
END
SUBROUTINE STRESS
INTEGER CODE
COMMON/VISC/EPSON(12,10,8),SIGVP(6),DEPSR(6,10,8),DELTIM
COMMON/ANS4/FT(24,4),GTS1U(24),GTS1UT(24,4)
COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
1NUMST
COMMON/MATP/RO(6),E(12,16,6),EE(16),ADFTS(6)
COMMON/NPDATA/R(10),CODE(10),XR(10),Z(10),XZ(10),
1NPNUM(4,8),T(10),XT(10)
COMMON/ELDATA/BETA(10),EPR(10),PR(4),SH(4),IX(8,5),
1IP(4),JP(4),IS(4),JS(4),ALPHA(10),IT(4),JT(4),
2ST(4)
COMMON/ARG/RRR(5),ZZZ(5),RR(4),ZZ(4),S(15,15),P(15),TT(6),
1H(6,15),CRZ(6,6),XI(10),ANGLE(4),SIG(18),EPS(18),N
COMMON/NONAXI/S1(30,30),P1(30),THETA,BS1(6,30)
COMMON/NXDATA/NTP,NTS,NTOTS,GTS1G(24,24,8)

```

```

COMMON/NXMESH/THETAN(8),NPC(8,8)
COMMON/ARG1/SIG1(18),EPS1(18),DEPSP(12),CEPSP(6,6)
COMMON/SOLVE/B(72),A(72,36),NUMBLK,MBAND
COMMON/CONVRG/IDONE
COMMON/RATE/DKPR,SIGPR,BVR,EVR,PSRATE(10,8),NRATE
COMMON/PLANE/NPP
COMMON/RESULT/BS(6,15),D(6,6),C(6,6),AR,BB(6,9),CNS(6,6)
DIMENSION LM(4),TP(6),TR(3,3),Q(3)
DIMENSION QQ(3)
COMMON/DUM1/SITEM(3,30),GTS1(24,24),TS(6,24)
DIMENSION P11(24)
C* * * * *
C  INITIALIZE
C* * * * *
  REWIND 3
  XKE=0.
  XPE=0.
  MPRINT=0
  ERROR=.005
  IDONE=1
  NRATE=0
  DO 200 N=1,NUMEL
    IX(N,5)=IABS(IX(N,5))
    CALL QUAD
    MTYPE=IABS(IX(N,5))
    DO 100 I=1,4
      II=3*I
      JJ=3*IX(N,I)
      P1(II-2)=B(JJ-2)
      P1(II-1)=B(JJ-1)
      P1(II)=B(JJ)
      P1(II+10)=B(JJ-2)
      P1(II+11)=B(JJ-1)
      P1(II+12)=B(JJ)
      P(II-2)=B(JJ-2)
      P(II-1)=B(JJ-1)
100    P(II)=B(JJ)
      READ(3)((GTS1(I,J),J=1,24),I=1,24)
      DO 115 I=1,24
        GTS1U(I)=0.0
        DO 115 J=1,24
          115 GTS1U(I)=GTS1U(I)+GTS1(I,J)*P1(J)
        DO 116 I=1,24
          116 GTS1UT(I,NTP)=GTS1UT(I,NTP)+GTS1U(I)
        DO 110 I=1,3
          110 Q(I)=P(I+12)
          DO 120 I=1,3
            DO 120 J=1,3
              120 TR(I,J)=S(I+12,J+12)
              CALL SYMINV(TR,3)
              DO 125 J=1,3
                QQ(J)=0.00
                DO 125 K=1,12
                  QQ(J)=QQ(J)+S(J+12,K)*P(K)
              125 CONTINUE
              DO 130 I=1,3
                P(I+12)=0.00
                DO 130 J=1,3
                  130 P(I+12)=P(I+12)+TR(I,J)*(Q(J)-QQ(J))
              500 CONTINUE

```

```

      RETURN
      END
      SUBROUTINE SYMINV(A,NMAX)
      DIMENSION A(NMAX,NMAX)
      DO 300 N=1,NMAX
      D=A(N,N)
      DO 100 J=1,NMAX
100  A(N,J)=-A(N,J)/D
      DO 210 I=1,NMAX
      IF(N.EQ.I) GO TO 210
      DO 200 J=1,NMAX
      IF(N.NE.J) A(I,J)=A(I,J)+A(I,N)*A(N,J)
200  CONTINUE
210  A(I,N)=A(I,N)/D
300  A(N,N)=1.00/D
      RETURN
      END
      SUBROUTINE TEMP(R,Z,T)
      COMMON/SOLVE/X(888),Y(888),TEM(888),NUMTC,MBAND
      DIMENSION SMALL(20),ISM(20)
C* * * * *
C  INITIALIZE
C* * * * *
      J=1
      JMAX=16
      IF(NUMTC.LT.JMAX) JMAX=NUMTC
      DO 10 I=1,JMAX
      SMALL(I)=0.
10  ISM(I)=0
C* * * * *
C  FIND THE JMAX CLOSEST POINTS
C* * * * *
      DO 50 I=1,NUMTC
      DSQ=(X(I)-R)**2+(Y(I)-Z)**2
      IF(DSQ.GT..1E-4) GO TO 20
      T=TEM(I)
      RETURN
20  IF(I.EQ.1) SMALL(1)=DSQ
      IF(I.EQ.1) ISM(1)=1
      IF(I.EQ.1) GO TO 50
      IF(SMALL(J).LE.DSQ.AND.J.LT.JMAX) SMALL(J+1)=DSQ
      IF(SMALL(J).LE.DSQ.AND.J.LT.JMAX) ISM(J+1)=I
      IF(SMALL(J).LE.DSQ) GO TO 40
      DO 30 K=1,J
      JB=J-K +1
      IF(JB.EQ.0) GO TO 40
      SMALL(JB+1)=SMALL(JB)
      ISM(JB+1)=ISM(JB)
      SMALL(JB)=DSQ
      ISM(JB)=I
      IF(JB.EQ.1) GO TO 40
      IF(SMALL(JB-1).LE.DSQ) GO TO 40
30  CONTINUE
40  IF(J.LT.JMAX) J=J+1
50  CONTINUE
C* * * * *
C  FIND THE THIRD TEMPERATURE POINT BY AREA TEST
C* * * * *
      JCHK=JMAX-2
      J=0

```

```

C
C   MATRIX P NOW CONTAINS 15 DISPLACEMENTS FOR QUADRILATERAL ELEMENT
C
C   CALCULATE AVERAGE STRAINS
C
      DO 140 I=1,6
      EPS(I)=0.00
      DO 140 J=1,15
140    EPS(I)=EPS(I)+BS(I,J)*P(J)
C
C   CALCULATE AVERAGE STRESSES
C
      DO 151 I=1,6
      SIG(I)=EPSDN(I,N,NTP)
      DO 151 J=1,6
151    SIG(I)=SIG(I)+CRZ(I,J)*EPS(J)
      DO 152 I=1,6
152    SIG(I)=SIG(I)-TT(I)
C
C   CALCULATE STRAINS IN N-S-T COORDINATES
C
      DO 150 I=1,6
      EPS(I+6)=0.00
      DO 150 J=1,6
      DO 150 K=1,6
150    EPS(I+6)=EPS(I+6)+D(I,J)*C(J,K)*EPS(K)
C
C   CALCULATE STRESSES IN N-S-T COORDINATES
C
      DO 160 I=1,6
      SIG(I+6)=EPSDN(I+6,N,NTP)
      DO 160 J=1,6
160    SIG(I+6)=SIG(I+6)+CNS(I,J)*EPS(J+6)
      DO 161 M=1,6
      P(M)=0.00
      DO 161 II=1,3
      IF(AOFTS(MTYPE),EQ.1.) P(M)=CNS(M,II)*EE(II+9)
161    P(M)=P(M)+(T(N)-TREF)*CNS(M,II)*EE(II+9)
      DO 162 I=1,6
162    SIG(I+6)=SIG(I+6)-P(I)
C
C
      DO 300 I=1,12
300    EPS(I)=100.0*EPS(I)
      IF(MPRINT.NE.0) GO TO 210
      WRITE(6,2000)
      WRITE(6,2002)
      MPRINT=19
      210 MPRINT=MPRINT-1
      WRITE(6,2001) N,RRR(5),ZZZ(5),(SIG(I),I=1,12)
      WRITE(6,2003) T(N),(EPS(I),I=1,12)
      200 CONTINUE
2000  FORMAT(129H1    EL      R      Z      SIGMAR  SIGMAZ  SIGMAC  SIGMA
      1RZ  SIGMAZC  SIGMACR  SIGMAN  SIGMAS  SIGMAT  SIGMANS  SIGMAST
      2  SIGMATN)
2001  FORMAT(1H0,I5,1X,2F7.4,12F9.0)
2002  FORMAT(128H0      TEMPERATURE      EPSR      EPSZ      EPSC      EPSR
      1Z      EPSZC      EPSCR      EPSN      EPSS      EPST      EPSNS      EPSST
      2  EPSTN)
2003  FORMAT(6X,F13.0,2X,12F9.5)

```

```

      I1=ISM(1)
      I2=ISM(2)
60   I3=ISM(J+3)
      AREA=.50*(Y(I1)*X(I3)-Y(I3)*X(I1)+Y(I3)*X(I2)-Y(I2)*X(I3)+
1     Y(I2)*X(I1)-Y(I1)*X(I2))
      D1=(X(I2)-X(I1))*2+(Y(I2)-Y(I1))*2
C    IF D1 IS APPROXIMATELY 0. IT IS ASSUMED THAT THERE EXISTS A
C    DUPLICATION OF INPUT
      IF(D1.GT..1E-3) GO TO 70
      I2=I3
      J=J+1
      GO TO 60
70   IF(AREA**2.GT..1*D1*SMALL(1)) GO TO 80
      J=J+1
      IF(J.LT.JCHK) GO TO 60
      WRITE(6,2000) I1,I2,I3,J
      T=TEM(I1)
      RETURN
C* * * * *
C    FIND TEMPERATURE INTERCEPT
C* * * * *
80   DETA=Y(I1)*(TEM(I3)-TEM(I2))+Y(I2)*(TEM(I1)-TEM(I3))
1     +Y(I3)*(TEM(I2)-TEM(I1))
      DETB=X(I1)*(TEM(I2)-TEM(I3))+X(I2)*(TEM(I3)-TEM(I1))
1     +X(I3)*(TEM(I1)-TEM(I2))
      DETC=TEM(I1)*(X(I2)*Y(I3)-X(I3)*Y(I2))+TEM(I2)*(X(I3)*Y(I1)-X(I1)*
1     Y(I3))+TEM(I3)*(X(I1)*Y(I2)-X(I2)*Y(I1))
      T=(DETA*R+DETB*K+DETC)/(2.*AREA)
2000 FORMAT(28H ERROR IN TEMPERATURE INPUT,5H I1=I4,5H I2=I4,
15H I3=I4,4H J=I4)
      RETURN
      END
      SUBROUTINE TEM2(NUMNP)
      INTEGER CODE
      COMMON/NPDATA/R(10 ),CODE(10 ),XR(10 ),Z(10 ),XZ(10 ),
1NPNUM( 4, 8),T(10 ),XT(10 )
      READ(5,1000) TCONST
      DO 100 N=1,NUMNP
100   T(N)=TCONST
1000  FORMAT(F10.0)
      RETURN
      END
      SUBROUTINE TRISTF (II,JJ,KK)
      INTEGER CODE
      COMMON/VISC/EPSON(12,10,8),SIGVP(6),DEFSR(6,10,8),DELTIM
      COMMON/PLAS/ALFA(6, 4,8),SIGYLD(7,6,8),IFGPL( 4,8)
      COMMON/NXDATA/NTP,NTS,NTOTS,GTS1G(24,24,8)
      COMMON/MATP/RO(6),E(12,16,6),EE(16),AOFTS(6)
      COMMON/BASIC/ACELZ,ANGVEL,ANGACC,TREF,VOL,NUMNP,NUMEL,NUMPC,NUMSC,
1NUMST
      COMMON/ARG/RRR(5),ZZZ(5),RR(4),ZZ(4),S(15,15),P(15),TT(6),
1H(6,15),CRZ(6,6),XI(10),ANGLE(4),SIG(18),EPS(18),N
      COMMON/NPDATA/R(10 ),CODE(10 ),XR(10 ),Z(10 ),XZ(10 ),
1NPNUM( 4, 8),T(10 ),XT(10 )
      COMMON/ELDATA/BETA(10 ),EPR(10 ),PR(4 ),SH(4 ),IX(8 ,5),
1IP(4 ),JP(4 ),IS(4 ),JS(4 ),ALPHA(10 ),IT(4 ),JT(4 ),
2ST(4 )
      COMMON/NONAXI/S1(30,30),P1(30),THETA,BS1(6,30)
      COMMON/RESULT/BS(6,15),D(6,6),C(6,6),AR,BB(6,9),CNS(6,6)
      DIMENSION B1A(6,9),B1B(6,9),B2A(6,9),B2B(6,9),B3A(6,9),B3B(6,9)

```

```

DIMENSION B1(6,9),B2(6,9),B3(6,9),F(6,7),G(9,6),V(9,9)
DIMENSION BF(3),BFR(3),BFZ(3),TF(9),B(9,9),TVF(9)
MTYPE=IABS(IX(N,5))
RR(1)=RRR(II)
RR(2)=RRR(JJ)
RR(3)=RRR(KK)
ZZ(1)=ZZZ(II)
ZZ(2)=ZZZ(JJ)
ZZ(3)=ZZZ(KK)
CALL INTER
VOL=VOL+XI(1)
COMM=RR(2)*(ZZ(3)-ZZ(1))+RR(1)*(ZZ(2)-ZZ(3))+RR(3)*(ZZ(1)-ZZ(2))
DO 10 I=1,6
DO 10 J=1,9
B1(I,J)=0.00
B2(I,J)=0.00
10 B3(I,J)=0.00
C FILL B1 MATRIX-CONSTANT TERMS
B1(1,1)=(ZZ(2)-ZZ(3))/COMM
B1(1,4)=(ZZ(3)-ZZ(1))/COMM
B1(1,7)=(ZZ(1)-ZZ(2))/COMM
B1(3,1)=B1(1,1)
B1(3,4)=B1(1,4)
B1(3,7)=B1(1,7)
B1(2,2)=(RR(3)-RR(2))/COMM
B1(2,5)=(RR(1)-RR(3))/COMM
B1(2,8)=(RR(2)-RR(1))/COMM
B1(4,1)=B1(2,2)
B1(4,4)=B1(2,5)
B1(4,7)=B1(2,8)
B1(4,2)=B1(1,1)
B1(4,5)=B1(1,4)
B1(4,8)=B1(1,7)
B1(5,3)=B1(4,1)
B1(5,6)=B1(4,4)
B1(5,9)=B1(4,7)
C FILL B2 MATRIX-1/R TERMS
B2(3,1)=(1/COMM)*((ZZ(3)-ZZ(2))*RR(2)+(RR(2)-RR(3))*ZZ(2))
B2(3,4)=(1/COMM)*((ZZ(1)-ZZ(3))*RR(3)-(RR(1)-RR(3))*ZZ(3))
B2(3,7)=(1/COMM)*((ZZ(2)-ZZ(1))*RR(1)+(RR(1)-RR(2))*ZZ(1))
B2(6,3)=-B2(3,1)
B2(6,6)=-B2(3,4)
B2(6,9)=-B2(3,7)
C FILL B3 MATRIX-Z/R TERMS
B3(3,1)=(RR(3)-RR(2))/COMM
B3(3,4)=(RR(1)-RR(3))/COMM
B3(3,7)=(RR(2)-RR(1))/COMM
B3(6,3)=(RR(2)-RR(3))/COMM
B3(6,6)=(RR(3)-RR(1))/COMM
B3(6,9)=(RR(1)-RR(2))/COMM
AR=AR+XI(1)
DO 80 I=1,6
DO 80 J=1,9
80 BB(I,J)=B1(I,J)*XI(1)+B2(I,J)*XI(2)+B3(I,J)*XI(4)
DO 81 K=1,6
DO 81 I=1,3
BS(K,3*JJ-3+I)=BB(K,I+3)+BS(K,3*JJ-3+I)
BS(K,3*II-3+I)=BB(K,I)+BS(K,3*II-3+I)
81 BS(K,3*KK-3+I)=BB(K,I+6)+BS(K,3*KK-3+I)
DO 220 I=1,6

```

```

      DO 220 J=1,9
      B1A(I,J)=B1(I,J)*XI(1)+B2(I,J)*XI(2)+B3(I,J)*XI(4)
      B2A(I,J)=B1(I,J)*XI(2)+B2(I,J)*XI(3)+B3(I,J)*XI(5)
      B3A(I,J)=B1(I,J)*XI(4)+B2(I,J)*XI(5)+B3(I,J)*XI(6)
* 220  CONTINUE
      DO 200 I=1,6
      DO 200 K=1,9
      B1B(I,K)=0.0
      B2B(I,K)=0.0
      B3B(I,K)=0.0
      DO 200 J=1,6
      B1B(I,K)=B1B(I,K)+CRZ(I,J)*B1A(J,K)
      B2B(I,K)=B2B(I,K)+CRZ(I,J)*B2A(J,K)
      B3B(I,K)=B3B(I,K)+CRZ(I,J)*B3A(J,K)
200  CONTINUE
      DO 230 I=1,9
      DO 230 K=1,9
      B(I,K)=0.0
      DO 230 J=1,6
      B(I,K)=B(I,K)+B1(J,I)*B1B(J,K)+B2(J,I)*B2B(J,K)+B3(J,I)*B3B(J,K)
230  CONTINUE
C    ASSEMBLE QUADRILATERAL STIFFNESS MATRIX, S, FROM TRIANGULAR
C    STIFFNESS MATRIX, B.
      IIM=3*I-3
      JJM=3*J-3
      KKM=3*K-3
      DO 120 I=1,3
      DO 120 J=1,3
      S(IIM+I,IIM+J)=B(I,J)+S(IIM+I,IIM+J)
      S(IIM+I,JJM+J)=B(I,J+3)+S(IIM+I,JJM+J)
      S(IIM+I,KKM+J)=B(I,J+6)+S(IIM+I,KKM+J)
      S(JJM+I,IIM+J)=B(I+3,J)+S(JJM+I,IIM+J)
      S(JJM+I,JJM+J)=B(I+3,J+3)+S(JJM+I,JJM+J)
      S(JJM+I,KKM+J)=B(I+3,J+6)+S(JJM+I,KKM+J)
      S(KKM+I,IIM+J)=B(I+6,J)+S(KKM+I,IIM+J)
      S(KKM+I,JJM+J)=B(I+6,J+3)+S(KKM+I,JJM+J)
      S(KKM+I,KKM+J)=B(I+6,J+6)+S(KKM+I,KKM+J)
120  CONTINUE
C    ASSEMBLE BODY FORCES MATRIX
      BF(1)=(ZZ(3)*RR(2)-RR(3)*ZZ(2))/COMM
      BF(2)=(ZZ(1)*RR(3)-RR(1)*ZZ(3))/COMM
      BF(3)=(ZZ(2)*RR(1)-RR(2)*ZZ(1))/COMM
      BFR(1)=(ZZ(2)-ZZ(3))/COMM
      BFR(2)=(ZZ(3)-ZZ(1))/COMM
      BFR(3)=(ZZ(1)-ZZ(2))/COMM
      BFZ(1)=(RR(3)-RR(2))/COMM
      BFZ(2)=(RR(1)-RR(3))/COMM
      BFZ(3)=(RR(2)-RR(1))/COMM
C    BODY FORCE IN Z-DIRECTION
      COMM=-ACELZ*RO(MTYPE)
      DO 140 I=1,3
      IIK=3*I-1
140  TP(IIK)=COMM*(BF(I)*XI(1)+BFR(I)*XI(7)+BFZ(I)*XI(8))
C    BODY FORCE IN R-DIRECTION
      COMM=ANGVEL**2*RO(MTYPE)
      DO 150 I=1,3
      L=3*I-2
150  TP(L)=COMM*(BF(I)*XI(7)+BFR(I)*XI(9)+BFZ(I)*XI(10))
C    BODY FORCES IN TANG. DIRECTION
      COMM=-ANGACC*RO(MTYPE)

```

```

      DO 160 I=1,3
      IIM=3*I
      160 TP(IIM)=COMM*(BF(I)*XI(7)+BFR(I)*XI(9)+BFZ(I)*XI(10))
      C   ADD THERMAL EFFECTS
      DO 161 J=1,9
      DO 161 K=1,6
      161 TP(J)=TP(J)+(XI(1)*B1(K,J)+XI(2)*B2(K,J)
      1+XI(4)*B3(K,J))*TT(K)
      C   REARRANGE TP INTO P-MATRIX, THE BODY FORCES MATRIX
      K=3*I-2
      L=3*JJ-2
      M=3*KK-2
      DO 170 I=1,3
      J=I-1
      P1(K+J) = P1(K+J) + TP(I) * THETA/2.0 + TP(I+6) * THETA/4.0
      P1(K+J+15) = P1(K+J+15) + TP(I) * THETA/2.0 + TP(I+6) * THETA/4.0
      P1(L+J) = P1(L+J) + TP(I+3) * THETA/2.0 + TP(I+6) * THETA/4.0
      P1(L+J+15) = P1(L+J+15) + TP(I+3) * THETA/2.0 + TP(I+6) * THETA/4.0
      P1(M+J) = P1(M+J) + TP(I+6) * THETA/2.0
      P1(M+J+15) = P1(M+J+15) + TP(I+6) * THETA/2.0
      P(K+J)=P(K+J)+TP(I)
      P(L+J)=P(L+J)+TP(I+3)
      170 P(M+J)=P(M+J)+TP(I+6)
      IF(IFGPL(N,NTP).EQ.0) GO TO 190
      DO 174 I=1,9
      TVP(I)=0.0
      DO 174 J=1,6
      TVP(I)=TVP(I)+BB(J,I)*EPSDN(J,N,NTP)
      174 CONTINUE
      DO 180 I=1,3
      J=I-1
      P(K+J)=P(K+J)-TVP(I)
      P(L+J)=P(L+J)-TVP(I+3)
      180 P(M+J)=P(M+J)-TVP(I+6)
      190 CONTINUE
      RETURN
      END
      SUBROUTINE XMODFY(U,N)
      COMMON/NXSOLV/SK(36,24),R1(132),FTOT(132),NSZF
      NBAND=24
      DO 10 M=2,NBAND
      K=N-M+1
      IF(K.LE.0) GO TO 5
      R1(K)=R1(K)-SK(K,M)*U
      SK(K,M)=0.
      5 K=N+M-1
      IF(NSZF.LT.K) GO TO 10
      R1(K)=R1(K)-SK(N,M)*U
      SK(N,M)=0.
      10 CONTINUE
      SK(N,1)=1.
      R1(N)=U
      RETURN
      END
      SUBROUTINE XSOLVE
      COMMON/NXSOLV/SK(36,24),R1(132),FTOT(132),NSZF
      NBAND=24
      DO 300 N=1,NSZF
      I=N
      DO 290 L=2,NBAND

```

```

      I=I+1
      IF(SK(N,L)) 240,290,240
240 AC=SK(N,L)/SK(N,1)
      J=0
      DO 270 K=L,NBAND
      J=J+1
      IF(SK(N,K)) 260,270,260
260 SK(I,J)=SK(I,J)-AC*SK(N,K)
270 CONTINUE
280 SK(N,L)=AC
C
      R1(I)=R1(I)-AC*R1(N)
290 CONTINUE
300 R1(N)=R1(N)/SK(N,1)
C
      N=NSZF
350 N=N-1
      IF(N) 500,500,360
360 L=N
      DO 400 K=2,NBAND
      L=L+1
      IF(SK(N,K)) 370,400,370
370 R1(N)=R1(N)-SK(N,K)*R1(L)
400 CONTINUE
      GO TO 350
C
500 RETURN
      END
      SUBROUTINE YIELD(N,NS,MTYPE)
      DIMENSION DALFA(6),SIGYB(3)
      COMMON/PLAS/ALFA(6, 4,8),SIGYLD(7,6,8),IFGPL( 4,8)
      COMMON/INCR/NOLINC,NOL,INERT,NUMMAT,SIGTOT(12, 4,8)
1,EPSTOT(12, 4,8)
      COMMON/ARG1/SIG1(18),EPS1(18),DEPSP(12),CEPSP(6,6)
      C=SIGYLD(7,MTYPE,NS)
      DO 50 I=1,6
50 ALFA(I,N,NS)=ALFA(I,N,NS)+C*DEPSP(I+6)
C
      WRITE(6,1000)N,NS
C1000 FORMAT(" ", " ALFA FOR EL ",I5," SEGMENT", I5)
C
      WRITE(6,1100)(ALFA(I,N,NS),I=1,6)
1100 FORMAT(" ",6E12.6)
      DO 100 I=1,6
100 SIG1(I)=SIGTOT(I+6,N,NS)-ALFA(I,N,NS)
C GET COMBINATION YIELD STRESSES
      SIGYB(1)=1./SIGYLD(1,MTYPE,NS)**2-1./SIGYLD(2,MTYPE,NS)**2
1
      -1./SIGYLD(3,MTYPE,NS)**2
      SIGYB(2)= 1./SIGYLD(2,MTYPE,NS)**2-1./SIGYLD(1,MTYPE,NS)**2
1
      -1./SIGYLD(3,MTYPE,NS)**2
      SIGYB(3)= 1./SIGYLD(3,MTYPE,NS)**2-1./SIGYLD(2,MTYPE,NS)**2
1
      -1./SIGYLD(1,MTYPE,NS)**2
C
      TEST YIELD CRITERION *****
      TEST=0.0
      DO 200 I=1,6
200 TEST=TEST+SIG1(I)**2/SIGYLD(I,MTYPE,NS)**2
      TEST=TEST+ SIGYB(1)*SIG1(2)*SIG1(3) +SIGYB(2)*SIG1(1)*SIG1(3)
1
      +SIGYB(3)*SIG1(1)*SIG1(2)
      IFGPL(N,NS)=0
      IF (TEST.LE.1.0) GO TO 500
      IFGPL(N,NS)=1
500 RETURN

```

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314	1	Commander US Army Materiel Development and Readiness Command ATTN: DRCMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Director of Defense Research & Engineering (OSD) ATTN: R. Thorkildsen Washington, DC 20301	1	Commander US Army Materiel Development and Readiness Command ATTN: DRCLDC 5001 Eisenhower Avenue Alexandria, VA 22333
1	Director of Defense Research & Engineering (OSD) ATN: J. Persh Washington, DC 20301	1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDE Deputy Director 5001 Eisenhower Avenue Alexandria, VA 22333
1	Director Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209	1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDE-R 5001 Eisenhower Avenue Alexandria, VA 22333
1	Director Institute of Defense Analysis ATTN: Documents Acquisition 1801 Beauregard Street Alexandria, VA 22311	1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDE-R 5001 Eisenhower Avenue Alexandria, VA 22333
1	HQDA (DAMA-MS) Washington, DC 20310	1	Commander US Army Materiel Development and Readiness Command 5001 Eisenhower Avenue Alexandria, VA 22333
1	HQDA (DAMA-ZA) Washington, DC 20310	8	Commander USA AMCCOM ATTN: DRSMC-LCA(D), A. Moss DRSMC-LC(D), DRSMC-SE(D) DRSMC-SA(D) DRSMC-AC(D) DRSMC-LCU-SE(D), J. Pearson DRSMC-LCA-M(D) F. Saxe DRSMC-LCU-SS(D), R. Botticelli Dover, NJ 07801
1	HQDA (DAMA-ZD, H. Woodall) Washington, DC 20310		
1	HQDA (DAMA-ARZ-A, M.E. Lasser) Washington, DC 20310		
2	HQDA (DAMA-CSM-VA) (DAMA-CSM-CA) Washington, DC 20310		
1	HQDA (DAMA-CSS-T, J. Bryant) Washington, DC 20310		

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
5	Commander USA AMCCOM ATTN: DRSMC-SCS(D), D. Brandt DRSMC-SCS-E(D), J. Blumer DRSMC-SCF(D), G. Del Coco DRSMC-SCS(D), S. Jacobson DRSMC-SCF(D), K. Pflieger Dover, NJ 07801	7	Commander USA AMCCOM ATTN: DRSMC-LCA(D), B. Knutelski DRSMC-LCR-R(D), E.H. Moore III DRSMC-LCS(D), J. Gregorits DRSMC-LCS-D(D), K. Rubin DRSMC-LCA(D), T. Davidson DRSMC-LCA(D), E. Friedman DRSMC-LCA(D), A. Lehberger Dover, NJ 07801
2	Commander USA AMCCOM ATTN: DRSMC-TSS(D) Dover, NJ 07801		
4	Commander USA AMCCOM ATTN: DRSMC-TD(D) DRSMC-TDA(D) DRSMC-TDS(D) DRSMC-TDC(D), D. Gyorog Dover, NJ 07801	7	Commander USA AMCCOM ATTN: DRSMC-SCA(D), C.J. McGee DRSMC-SCA(D), S. Goldstein DRSMC-SCA(D), F.P. Puzychki DRSMC-SCA(D), E. Jeeter DRSMC-SCF(D), M.J. Schmitz DRSMC-SCF(D), L. Berman DRSMC-SCZ(D), P. Petrella Dover, NJ 07801
5	Commander USA AMCCOM ATTN: DRSMC-LCU(D), E. Barrieres DRSMC-LCU(D), R. Davitt DRSMC-LCU-M(D), D. Robertson DRSMC-LCU-M(D), M. Weinstock DRSMC-LCA-M(D), C. Larson Dover, NJ 07801	9	Commander USA AMCCOM ATTN: DRSMC-SCM(D) DRSMC-SCM(D), E. Bloore DRSMC-SCM(D), J. Mulherin DRSMC-SCS(D), B. Brodman DRSMC-SCS(D), T. Hung DRSMC-SCA(D), W. Gadomski DRSMC-SCA(D), E. Malatesta DRSMC-SCA-T(D) P. Benzkofer DRSMC-SCA-T(D), F. Dahdouh Dover, NJ 07801
1	Director Lawrence Livermore Laboratory ATTN: D. Burton, L200 P.O. Box 808 Livermore, CA 94550		
1	Director Lawrence Livermore Laboratory ATTN: M. L. Wilkins P.O. Box 808 Livermore, CA 94550		

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
3	Commander USA AMCCOM ATTN: DRSMC-LCA(D), W. Williver DRSMC-LCA(D), S. Bernstein DRSMC-LCA(D), G. Demitrack Dover, NJ 07801	2	Commander USA AMCCOM ATTN: DRSMC-SC(D) DRSMC-SC(D), B. Shulman Dover, NJ 07801
4	Commander USA AMCCOM ATTN: DRSMC-LCA(D), S. Yim DRSMC-LCA(D), L. Rosendorf DRSMC-LCA(D), S.H. Chu DRSMC-LCW(D), R. Wrenn Dover, NJ 07801	2	Commander USA AMCCOM ATTN: Army Fuze MgtProject Office DRSMC-FU(D) Dover, NJ 07801
6	Director Benet Weapons Laboratory Armament R&D Center USA AMCCOM ATTN: DRSMC-LCB-TL(D) DRSMC-LCB(D) DRSMC-LCB-RA(D), R. Scanlon DRSMC-LCB-RM(D), M. Scarullo DRSMC-LCB-RA(D), R. Soanes, Jr. DRSMC-LCB-RA(D), J. Vasilakis Watervliet, NY 12189	2	Commander USA AMCCOM ATTN: Development Project Office for Selected Ammo, DRSMC-DP(D) Dover, NJ 07801
7	Director USA AMCCOM Benet Weapons Laboratory ATTN: DRSMC-LCB-RA(D), T. Simkins DRSMC-LCB-D(D), J. Zweig DRSMC-LCB-RA(D), G. Pflegl DRSMC-LCB-M(D), J. Purtell DRSMC-LCB-RA(D), R. Racicot DRSMC-LCB-DS(D), J. Santini DRSMC-LCB-RA, J. Wu Watervliet, NY 12189	2	Commander USA AMCCOM ATTN: Product Assurance Directorate DRSMC-QA(D) Dover, NJ 07801
		1	Commander USA AMCCOM ATTN: DRSMC-NS(D) Dover, NJ 07801
		1	Commander USA AMCCOM ATTN: L. Goldsmith Dover, NJ 07801
		1	Commander USA Armament, Munitions & Chem Cmd ATTN: DRSMC-TSE-SW(R) R. Radkiewicz Rock Island, IL 61299
		2	Commander USA Armament, Munitions & Chem Cmd ATTN: DRSMC-LEP-L(R) DRSMC-TSE-SW, G. Strahl Rock Island, IL 61299

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Commander USA Armament, Munitions & Chem Cmd ATTN: SARRI-RLS Rock Island, IL 61299	1	Commander Atmospheric Sciences Lab USA Electronics Research and Development Command ATTN: DELAS-EO-MO, R.B. Gomez White Sands Missile Range NM 88002
1	Commander USA Aviation Research and Development Command ATTN: DRDAV-E 4300 Goodfellow Blvd St. Louis, MO 63120	3	Commander USA Harry Diamond Lab ATTN: DELHD-I-TR, H.D. Curchak DELHD-I-TR, H. Davis DELHD-S-QE-ES, Ben Banner 2800 Powder Mill Road Adelphi, MD 20783
3	Director USA Air Mobility Research and Development Laboratory ATTN: Hans Mark R.L. Cohen Ames Research Center Moffett Field, CA 94035	1	Commander USA Harry Diamond Labs ATTN: DELHD-TA-L 2800 Powder Mill Road Adelphi, MD 20783
2	Director USA Research and Technology Laboratories (AVRADCOM) Ames Research Center Moffett Field, CA 94035	1	Director Night Vision Laboratory Fort Belvoir, VA 22060
2	Director USA Research and Technology Laboratories ATTN: DAVDL-AS DAVDL-AS Ames Research Center Moffett Field, CA 94035	1	Commander USA Missile Command ATTN: DRSMI-R Dover, NJ 07801
1	Commander USA Communications Research and Development Command ATTN: DRSEL-ATDD Fort Monmouth, NJ 07703	2	Commandant USA Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905
1	Commander USA Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703	1	Commander USA Missile Command ATTN: DRCPM-TO Dover, NJ 07801
		1	Commander USA Mobility Equipment Research & Development Cnd Fort Belvoir, VA 22060

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
3	Commander USA Tank Automotive Cnd ATTN: DRSTA-TSL DRSTA-ZSA, R. Beck DRSTA-NS Warren, MI 48090	4	Director USA Materials and Mechanics Research Center ATTN: Director (3 cys) DRXMR-ATL (1 cy) Watertown, MA 02172
1	Commander USA Natick Research and Development Laboratory ATTN: DRDNA-DT Natick, MA 01760	2	Commander USA Materials and Mechanics Research Center ATTN: J. Mescall Tech. Library Watertown, MA 02172
1	Director USA TRADOC Systems Analysis Activity ATTN: ATAA-SL White Sands Missile Range NM 88002	1	Commander USA Training and Doctrine Command ATTN: TRADOC Lib Fort Monroe, VA 23651
2	President USA Armor and Engineering Board ATTN: ATZK-AE-CV ATZK-AE-IN, L. Smith Fort Knox, KY 40121	1	Commandant USA Armor School ATTN: Armor Agency, MG Brown Fort Knox, KY 40121
3	Commander USA Research Office ATTN: R. Weigle E. Saibel J. Chandra P.O. Box 12211 Research Triangle Park NC 27709	1	Commandant USA Field Artillery School ATTN: Field Artillery Agency Fort Sill, OK 73503
3	Commander USA Research Office P.O. Box 12211 ATTN: Technical Director Engineering Division Metallurgy & Materials Division Research Triangle Park NC 27709	1	Superintendent Naval Postgraduate School ATTN: Dir of Lib Monterey, CA 93940
		1	Commander USA Combined Arms Combat Development Activity Fort Leavenworth, KS 66027
		1	Commander USA Combat Development Experimentation Command ATTN: Tech Info Center Bldg. 2925, Box 22 Fort Ord, CA 93941

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
3	Commander Naval Sea Systems Command ATTN: SEA-62R, SEA-62Y, SEA-9961 Washington, DC 20360	5	Commander Naval Surface Weapons Center ATTN: Code G-33, T.N. Tschirn Code N-43, J.J. Yagla L. Anderson G. Soo Hoo Code TX, W.G. Soper Dahlgren, VA 22448
1	Commander Naval Sea Systems Command (SEA-62R41) ATTN: L. Pasiuk Washington, DC 20360	1	Commander Naval Weapons Center China Lake, CA 93555
1	Commander Naval Research Laboratory ATTN: Commander H. Peritt, Code R31 Washington, DC 20375	1	Commander Naval Weapons Center ATTN: J. O'Malley China Lake, CA 93555
1	Commander David W. Taylor Naval Ship Research and Development Center Bethesda, MD 20084	2	Commander Naval Weapons Center ATTN: Code 3835, R. Sewell Code 3431, Tech Lib China Lake, CA 93555
4	Commander Naval Research Laboratory ATTN: W.J. Ferguson C. Sanday H. Pusey Tech Library Washington, DC 20375	2	Commander Naval Weapons Center ATTN: Code 608, R. Derr Code 4505, C. Thelen China Lake, CA 93555
6	Commander Naval Surface Weapons Center ATTN: Code X211, Lib E. Zimet, R13 R.R. Bernecker, R13 J.W. Forbes, R13 S.J. Jacobs, R10 K. Kim, R13 Silver Springs, MD 20910	3	Commander Naval Weapons Center ATTN: Code 4057 Code 3835, B. Lundstrom Code 3835, M. Backman China Lake, CA 93555
3	Commander Naval Surface Weapons Center ATTN: Code E-31, R.C. Reed M.T. Walchak Code V-14, W.M. Hinckley Silver Springs, MD 20910	1	Commander Naval Ordnance Station Indian Head, MD 20640
		2	Commander Naval Ordnance Station ATTN: Code 5034, Ch. Irish T.C. Smith Indian Head, MD 20640

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Office of Naval Research ATTN: Code ONR 439, N. Perrone Department of the Navy 800 North Quincy Street Arlington, VA 22217	3	Battelle Pacific Northwest Laboratory ATTN: E.M. Patton F.A. Simonen L.A. Strobe P.O. Box 999 Richland, WA 99352
1	Commandant US Marine Corps ATTN: AX Washington, DC 20380	1	Bell Telephone Labs. Inc. Mountain Avenue Murray Hill, NJ 07971
1	AFATL/DLXP ATTN: W. Dittrich Eglin AFB, FL 32542	1	Director Lawrence Livermore Laboratory P.O. Box 808 Livermore, CA 94550
1	AFATL/DLD ATTN: D. Davis Eglin AFB, FL 32542	1	Director Lawrence Livermore Laboratory ATTN: E. Farley, L9 P.O. Box 808 Livermore, CA 94550
2	ADTC/DLJW Eglin AFB, FL 32542		
1	ADTC/DLODL, Tech Lib Eglin AFB, FL 32542	1	Director Lawrence Livermore Laboratory ATTN: R.H. Toland, L-424 P.O. Box 808 Livermore, CA 94550
1	AFWL/SUL Kirtland AFB, NM 87117		
1	AFWL/SUL ATTN: J.L. Bratton Kirtland AFB, NM 87117	1	Director NASA - Ames Research Center Moffett Field, CA 94035
1	AFML/LLN (T. Nicholas) Wright-Patterson AFB OH 45433	2	Forrestal Research Center Aeronautical Engineering Lab. ATTN: S. Lam A. Eringen Princeton, NJ 08540
1	ASD (XROT, G. Bennett) Wright-Patterson AFB OH 45433		
2	Battelle Memorial Institute ATTN: L.E. Hulbert J.E. Backofen, Jr. 505 King Avenue Columbus, OH 43201	1	Director Sandia National Laboratory ATTN: D. E. Wayne Livermore, CA 94550
		1	ASD (EMFTV, Martin Lentz) Wright-Patterson AFB OH 45433

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Aircraft Armaments Inc. ATTN: John Hebert P.O. Box 126 Cockeysville, MD 21030	2	General Electric Armament and Electrical Systems ATTN: D.A. Graham M.J. Bulman Lakeside Avenue Burlington, VT 05401
2	ARES, Inc. ATTN: Duane Summers Phil Conners Port Clinton, OH 43452	1	Olin Corporation Badger Army Ammunition Plant ATTN: R.J. Thiede Baraboo, WI 53913
2	AVCO System Division Structures and Mechanics Dept ATTN: W. Broding, J. Gilmore 201 Lowell Street Wilmington, MA 01887	1	S&D Dynamics, Inc. ATTN: M. Soifer 755 New York Avenue Huntington, NY 11743
3	BLM Applied Mechanics Consultants ATTN: A. Boresi R. Miller H. Langhaar 3310 Willett Drive Laramie, WY 82070	1	Southwest Research Institute ATTN: P.A. Cox 8500 Culebra Road San Antonio, TX 78228
1	Martin Marietta Laboratories ATTN: J.I. BACILE P.O. Box 5837 Orlando, FL 32805	1	Southwest Research Institute ATTN: T. Jeter 8500 Culebra Road San Antonio, TX 78228
1	H.P. White Laboratory 3114 Scarboro Road Street, MD 21154	1	University of Dayton Research Institute ATTN: S.J. Bless Dayton, OH 45406
1	CALSPAN Corporation ATTN: E. Fisher P.O. Box 400 Buffalo, NY 14225	1	University of Delaware Department of Mechanical Engineering ATTN: H. Kingsbury Newark, DE 19711
1	FMC Corporation Ordnance Engineering Division San Jose, CA 95114	3	University of Illinois College of Engineering ATTN: A. Zak (2) D. C. Drucker Urbana, IL 61801
1	Kaman-TEMPO ATTN: E. Bryant 715 Shamrock Road Bel Air, MD 21014	1	Director Lawrence Livermore Laboratory ATTN: R. Werne P.O. Box 808 Livermore, CA 94550

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	University of Iowa College of Engineering ATTN: R. Benedict E.J. Haug Iowa City, IA 52240	4	Director Sandia National Laboratories ATTN: L. Davison P. Chen L. Bertholf W. Herrmann Albuquerque, NM 87115
1	Rutgers University Dept. of Mechanical and Aerospace Engineering P.O. Box 909 ATTN: T.W. Lee New Brunswick, NJ 08903	4	SRI International ATTN: D. Curran L. Seaman Y. Gupta G.R. Abrahamson 333 Ravenswood Avenue Menlo Park, CA 94025
1	University of Wisconsin Mechanical Engineering Dept ATTN: S.M. Wu Madison, WI 53706	5	Brown University Division of Engineering ATTN: R. Clifton H. Kolsky A. Pipkin P. Symonds J. Martin Providence, RI 02912
1	University of Wisconsin Mathematics Research Center 610 Walnut Street ATTN: B. Noble Madison, WI 53706	3	California Institute of Technology Division of Engineering and Applied Science ATTN: J. Miklowitz E. Sternberg J. Knowles Pasadena, CA 91102
1	AFFDL/FB (J. Halpin) Wright-Patterson AFB OH 45433	1	University of California Department of Mechanics ATTN: W. Goldsmith 504 Hilgard Avenue Los Angeles, CA 90024
3	Honeywell, Inc. Government and Aerospace Products Division ATTN: G. Johnson J. Blackburn R. Simpson 600 Second Street, NE Hopkins, MN 55343	1	Drexel Institute of Technology Wave Propagation Research Center ATTN: P.C. Chou 32nd and Chestnut Streets Philadelphia, PA 19104
4	Director Los Alamos Scientific Lab ATTN: Tech Lib J. Taylor R. Karpp U. F. Kocks P.O. Box 1663 Los Alamos, NM 87544		

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Harvard University Division of Engineering and Applied Physics ATTN: G. Carrier Cambridge, MA 02138	1	Rensselaer Polytechnic Inst Department of Mechanical Engineering ATTN: E. Krempf Troy, NY 12181
1	Iowa State University Department of Engineering Science and Mechanics ATTN: C.P. Burger Ames, IA 50010	2	Southwest Research Institute Department of Mechanical Sciences ATTN: U. Lindholm W. Baker 8500 Culebra Road San Antonio, TX 78228
2	Iowa State University Engineering Research Lab. ATTN: G. Nariboli A. Sedov Ames, IA 50010	2	University of Delaware Department of Mechanical Engineering ATTN: J. Vinson M. Taya Newark, DE 19711
4	The Johns Hopkins University ATTN: J. Bell R. Green R. Pond, Sr. C. Truesdell 34th and Charles Streets Baltimore, MD 21218	3	University of Florida Department of Engineering Science and Mechanics ATTN: C. Sciammarilla L. Malvern E. Walsh Gainesville, FL 32601
1	Louisiana State University Department of Mechanical Engineering ATTN: W. Sharpe Baton Rouge, LA 70803	2	University of Illinois at Chicago Circle College of Engineering Dept. of Materials Engineering ATTN: A. Schultz T.C.T. Ting P.O. Box 4348 Chicago, IL 60680
4	Massachusetts Institute of Technology ATTN: R. Probst A.S. Argon F. McClintock K.J. Bathe 77 Massachusetts Avenue Cambridge, MA 02139	2	University of Minnesota Department of Engineering Mechanics ATTN: R. Fosdick J. Erickson Minneapolis, MN 55455
1	New Mexico Institute Mining and Technology Terra Group Socorro, NM 87801	1	Commander US Army Missile Command ATTN: DRSMI-YDL Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: DRSMI-YDL Redstone Arsenal, AL 35898	1	Commander US Army Missile Command ATTN: DRCPM-HD Redstone Arsenal, AL 35898

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>
1	Yale University Dept of Engr & Applied Science ATTN: A. Phillips 400 Temple Street New Haven, CT 06520
2	Washington State University Department of Physics ATTN: G.E. Duvall R. Fowles Pullman, WA 99163

Aberdeen Proving Ground

Dir, USAMSAA

ATTN: DRXSY-D
DRXSY-MP, H. Cohen
DRXSY-G, E. Christman
DRXSY-OSD, H. Burke
DRXSY-G, R.C. Conroy
DRXSY-LM, J.C.C. Fine

Dir, USAHEL

ATTN: DRXHE, A.H. Eckles, III

Cdr, USATECOM

ATTN: DRSTE-CE
DRSTE-TO-F

Cdr, CRDC, AMCCOM

ATTN: DRSMC-CLB-PA
DRSMC-CL
DRSMC-CLB
DRSMC-CLD
DRSMC-CLY
DRSMC-CLN
DRSMC-CLN, L. Shaff
DRSMC-CLN-D, F. Dagostin
DRSMC-CLN-D, C. Hughes
DRSMC-CLN, J. McKivirigan
DRSMC-CLJ-L

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number _____
2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) _____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) _____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: _____

Telephone Number: _____

Organization Address: _____

----- FOLD HERE -----

Director
US Army Ballistic Research Laboratory
ATTN: DRSMC-BLA-S (A)
Aberdeen Proving Ground, MD 21005

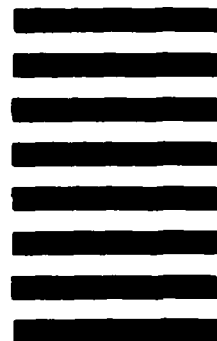


NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, \$300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC
POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
US Army Ballistic Research Laboratory
ATTN: DRSMC-BLA-S (A)
Aberdeen Proving Ground, MD 21005



----- FOLD HERE -----