SLAMCODE: FINITE ELEMENT STRESS WAVE ANALYSIS

D. W. McCowan

Teledyne Geotech

Prepared for:

Air Force Office of Scientific Research

14 September 1972

AL-72-1

.....contributing to man's understanding of the environment world

31 OCT 1972

SLAMCODE: FINITE ELEMENT STRESS WAVE ANALYSIS

BY

D.W. McCOWAN

SPONSORED BY
ADVANCED RESEARCH PROJECTS AGENCY

MONITORED BY
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

ARPA ORDER NO. 1357

14 SEPTEMBER 1972

TELEDYNE GEOTECH
ALEXANDRIA LABORATORIES

Approved for public release;
distribution unlimited.

60
R

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

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

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Teledyne Geotech Alexandria, Virginia	Unclassified
	2b. GROUP

3. REPORT TITLE

SLAMCODE: FINITE ELEMENT STRESS WAVE ANALYSIS

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

Scientific .....Interim

5. AUTHOR(S) (Last name, first name, initial)

McCowan, D.W.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
14 September 1972	60	3

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
F44620-69-C-0082	AL-72-1
b. PROJECT NO. AO 1357	
c. 62701D	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFOSR - TR - 73 - 0120
d.	

10. AVAILABILITY/LIMITATION NOTICES

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
TECH, OTHER	Air Force Office of Scientific Res. 1400 Wilson Boulevard NPG Arlington, Virginia

13. ABSTRACT

The program SLAMCODE developed in this study and described in this Volume is a large Finite Element Method (FEM) code capable of dynamic analysis of transient problems. It was adapted from the original SLAMCODE written by C. J. Costantino (Costantino, 1968) by incorporating the implicit Newmark B-integration (Newmark, 1959) and by constraining the free surface to be stress free. The present code also differs from the original by not permitting plastic elements. It calculates solutions for axisymmetric, plane stress or plane strain problems, as in the original code. It is designed to run on the IBM 360 family of computers under ØS/360. Being written in Fortran, it can be easily adapted to other computer systems. In fact, it was written with this requirement in mind by minimizing the number of system-dependent Fortran peculiarities.

14. KEY WORDS

Finite Element
Stress Waves

Unclassified

Security Classification

SLAMCODE: FINITE ELEMENT STRESS WAVE ANALYSIS

ALEXANDRIA LABORATORIES REPORT NO. AL-72-1

Effective Date of Contract:	1 March 1969
Contract Expiration Date:	31 March 1972
Amount of Contract Dollars:	$ 590,572
Program Code:	1F10
Contract Number:	F~~1~~4620-69-C-0082
ARPA Order No.:	1357
Principal Investigator:	Carl A. Newton
Period Covered:	1 March 1969 through 31 March 1972

TABLE OF CONTENTS

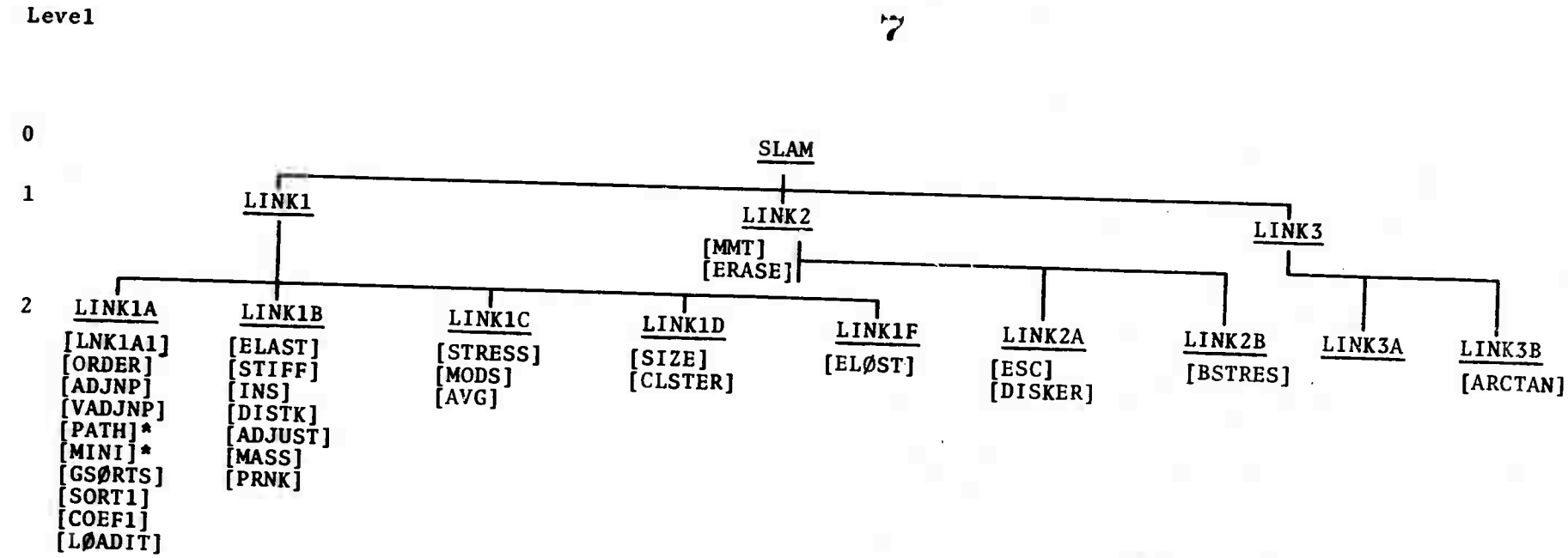
LIST OF TABLES

FIGURE TITLE

SLAMCODE Computer Program

The program SLAMCODE developed in this study and described in this Volume is a large Finite Element Method (FEM) code capable of dynamic analysis of transient problems. It was adapted from the original SLAMCODE written by C. J. Costantino (Costantino, 1968) by incorporating the implicit Newmark β-integration (Newmark, 1959) and by constraining the free surface to be stress free. The present code also differs from the original by not permitting plastic elements. It calculates solutions for axisymmetric, plane stress or plane strain problems, as in the original code. It is designed to run on the IBM 360 family of computers under ØS/360. Being written in Fortran, it can be easily adapted to other computer systems. In fact, it was written with this requirement in mind by minimizing the number of system-dependent Fortran peculiarities.

The Fortran source cards are listed in Appendix A. The Link Editor control cards are listed in Appendix B. These control cards are appended to the object module for execution of the program. Appendix C contains examples of Job Control Language cards for the Fortran H computer using either two or four 9-track tape drives. Tables 1-4 describe the operation of the program and Figure 1 is a block diagram of the overlay structure. Finally, detailed descriptions of the input parameters and operation of the program can be gleaned by reading the comments in the source listing. All input data is described



*[PATH] and [MINI] may be replaced by Rosen's matrix bandwidth minimization.

Figure 1. SLAMCODE overlay structure. Brackets indicate subroutines.

where it is read in the program listing.

The bandwidths of the diagonal matrices are controlled by the numbering of the nodal points. The bandwidth minimization algorithms listed in Appendix A are those used by Costantino, (PATH) and (MINI). A third algorithm written by Rosen (1968) and adapted for SLAMCØDE is listed in Appendix A1, (MINBND). The listing of the original Rosen algorithm contained some typographical errors which were corrected for (MINBND), but the comment cards which we have not reproduced in Appendix A1 are very useful. (PATH) is much faster than (MINBND) but its effectiveness depends upon the selection of starting node points. (MINI) has not decreased the bandwidth for any of our element arrays.

In general, large problems are solved by several runs of the program, all output being saved on magnetic tape or cataloged disk files. A wide range in input/output (I/Ø) flexibility is available with ØS/360 and this should be exploited to maximize program efficiency. For example, tape units should only be used in preference to disks when the time reduction made possible by saving intermediate results exceeds the time lost by the tape drives.

The program in its present form is I/Ø bound, i.e., a large amount of the computer time is spent spinning tapes and disks. In a typical 360/67 run there are approximately the same number of I/Ø seconds as there are CPU seconds. This performance

might be improved on larger machines where I/Ø buffers and program dimensions could be increased. The 360/67 version runs in a 280K byte region of core memory.

The number of I/Ø devices required depends on the operating system. The 360/67 system can assign many files to one device. A general rule of thumb for a system with 280K bytes of available core memory is that the program requires at least four tape drives and a hundred cylinders of disk file space.

9

REFERENCES

Costantino, C. J., 1968, Stress Waves in Layered Arbitrary Media, Final Report to Space and Missile Systems Organization (SAMSO) Norton Air Force Base, California: SAMSO TR 68-181.

Newmark, M., 1959, A Method of Computation for Structural Dynamics, J. of Engineering Mechanics Division, ASCE, 85, EM 3, 67-94.

Rosen, R., 1968, Matrix Bandwidth Minimization, Proceedings of 23d National Conference, ACM, Brandon/Systems Press, Inc., New Jersey, pp 585-595.

10

Table 1

STRESS-CONSTRAINED SLAMCODE PARAMETER LIMITS

MACHINE: 360/67
WORD LENGTH: FLOATING POINT: 8 BYTES = 64 BITS
FIXED POINT: 4 BYTES = 32 BITS
CYCLE TIME: 0.75 MICROSEC.
MEMORY SIZE: 280K BYTES = 35K FLOATING POINT WORDS

PARAMETER	DESCRIPTION	MAXIMUM VALUE	COMMENT
NUMNP	NUMBER OF NODE POINTS	1000	
NUMEL	NUMBER OF ELEMENTS	1250	
--	NUMBER OF ADJACENT NODES TO ANY NODE POINT	8	NEVER ACTUALLY SET BUT IMPLIED IN A DIMENSION STATEMENT
LINES	NUMBER OF STRESSED LINES	1	≤ 6 FOR STRESSED NODES
LØADNP	NUMBER OF NODES PER STRESSED LINE	50	
NZØNES	NUMBER OF MATERIAL ZONES	20	
NUMST	NUMBER OF STARTING NODES	20	
--	MATRIX BLOCK SIZE=MESH BANDWIDTH+1 NODES	30	LINK1 LIMIT ON BANDWIDTH IS 100; HOWEVER, LINK2 LIMIT IS 29
--	PLASTIC ELEMENT TAPE BLOCKING FACTOR	26	NOT A CONSIDERATION IN THIS VERSION OF SLAM
--	NUMBER OF GRAND PARTITIONS IN PATH PROCEDURE	100	PARAMETER INVOLVED IN BANDWIDTH MINIMIZATION
--	NUMBER OF BLOCKS OF NODES ON STIFF./STRESS TAPE	100	e.g., MINIMUM BANDWIDTH FOR 1000 NODE PROBLEM = 9
NUMØUT	NUMBER OF OUTPUT NODES	100	

Table 2

360 STRESS-CONSTRAINED SLAMCODE RUN CHART

KRUN	CARD GROUPS SKIPPED	OUTPUT SKIPPED	TAPE ASSIGNMENTS	LINK 2 RUN TIME	LINK 3 RUN TIME	DESCRIPTION & COMMENTS
0	NONE	NONE	STIFF.&STRESS TAPE OUT ON #10 UTR TAPE OUT ON #20 IF TMAX ≠0 DTH TAPE OUT ON #8 IF TMAX ≠0 OTH TAPE OUT ON #3 IF TMAX ≠0	0 → TMAX	TSTRESS → MAX	INITIAL RUN TRERUN MUST BE 0 MESH TITLE WRITTEN ON STIFF & STRESS,UTR,DTH,& OTH TAPES
1	NONE	ALL LINK1 STIFF.,MASS & STRESS TABLES	STIFF.&STRESS TAPE IN ON #10 UTR TAPE OUT ON #20 DTH TAPE OUT ON #8 OTH TAPE OUT ON #3 IF TMAX ≠0	0 → TMAX	TSTRSS → TMAX	LINK1 RERUN TRERUN MUST BE 0 PROBLEM TITLE WRITTEN ON UTR,DTH,& OTH TAPES
2	ALL LINK1 INPUT 1.1 - 1.6	ALL LINK1 OUTPUT	STIFF.& STRESS TAPE IN ON #10 UTR TAPE IN ON #20 DTH TAPE IN/OUT ON #8 OTH TAPE OUT ON #3	TRERUN→TMAX	TSTRSS→TMAX	LINK 2 RERUN
3	ALL LINK1 INPUT 1.1 - 1.6	ALL LINK1 OUTPUT LINK3 STRESS OUT-PUT	STIFF.& STRESS TAPE IN ON #10 UTR TAPE IN ON #20 DTH TAPE IN/OUT #8 OTH TAPE OUT ON #3	TRERUN → TMAX	TSTRSS → TMAX	LINK2 SUMMARY OUTPUT SUMMARY≡NO LINK3 STRESS OUTPUT
4	ALL LINK1 INPUT ALL LINK2 INPUT 1.1-2.2	ALL LINK1 OUTPUT ALL LINK2 OUT-PUT LINK3 STRESS OUT-PUT	STIFF.&STRESS TAPE IN ON #10 DTH TAPE IN ON #8 OTH TAPE OUT ON #3	NOT RUN	TSTRSS → TMAX	LINK3 SUMMARY OUTPUT
5	ALL LINK1 INPUT ALL LINK2 INPUT 1.1-2.2	ALL LINK1 OUTPUT ALL LINK2 OUT-PUT	STIFF.&STRESS TAPE IN ON #10 DTH TAPE IN ON #8 OTH TAPE OUT ON #3	NOT RUN	TSTRSS → TMAX	LINK3 RERUN

Table 3

CARD INPUT CHART FOR 360 STRESS-CONSTRAINED SLAMCODE

CARD GROUP	READING ROUTINE	CONDX. OF READ	READ LIST	FORMAT	PARAMETERS
0.1	SLAM		KRUN,ANAME	(I5,9A8)	KRUN - RUN TYPE SWITCH ANAME - 72 CHAR.RUN TYPE IDENTIFIER
1.1	LNK1AI		TITLE NUMNP,NUMEL,ISTRES,IPRINT	(10A8) (4I5)	TITLE - MESH TITLE/PROBLEM TITLE NUMNP - # NODES NUMEL - # ELEMENTS ISTRES - GEOMETRY SWITCH IPRINT - PRINT SWITCH
1.2	LNK1A1	NPN ≤ NUMNP	ANAME NPN,R(NPN),Z(NPN),ITYPE(NPN),THETA(NPN)	(10A8) (I5,2E10.4,I10,E10.4)	ANAME - CARD GROUP IDENTIFIER (NOT PRINTED) NPN - NODE # R - RADIUS Z - DEPTH ITYPE - NODE TYPE THETA - ROLLER ANGLE
1.3	LNK1A1		ANAME LINES,(LØADNP(I),I=1,LINES)	(10A8) (14I5)	ANAME - LOADING TYPE IDENTIFIER LINES - #STRESSED LINES LØADNP(I) - #STRESSED LINES ON LINE I
		I≤LINES	(NPLØAD(I,J),SNØRM(I,J),J=1,LØADNP(I))	(I5,E10.2)	NPLØAD(I,J) - NODE#of Jth NODE ON STRESSED LINE I SNØRM(I,J) - SURFACE NORMAL ANGLE
1.4	LNK1A1		ANAME NZONES	(10A8) (I5)	ANAME - CARD GROUP IDENTIFIER (NOT PRINTED) NZONES - #MATERIAL ZONES
		IZ≤ LINES	IZ,ANAME	(I5,9A8)	IZ - ZONE # ANAME - ZONE MATERIAL IDENTIFIER
			IELAST,IPLAST,WGT,E1,E2,E3,E4,E5	(2I5,6E10.4)	IELAST - ELASTICITY SWITCH IPLAST - PLASTICITY SWITCH WGT - DENSITY E1 - ELASTIC MODULUS E2 - POISSON'S RATIO E3,E4,E5 - ANISOTROPIC PARAMETERS
1.5	LNK1A1	NUME≤ NUMEL	ANAME NUME,IZONE,NPI,NPJ,NPK,NPL	(10A8) (6I5)	ANAME - CARD GROUP IDENTIFIER (NOT PRINTED) NUME - ELEMENT # IZONE - ZONE # NPI-NPL - NODE#'s OF VERTICES

Table 3 (Cont'd.)

1.6	LNK1A1	ANAME	(10A8)	ANAME - CARD GROUP IDENTIFIER (NOT PRINTED)
		NUMST	(I5)	NUMST - #STARTING NODES
		(NSTART(I),I=1,NUMST)	(14I5)	NSTART - STARTING NODE #'s
2.1	L.NK2	TMAX,TRERUN,ET,KDT,KINT,BETA,PDAMP	(3E10.0,2I5,2E10.0)	TMAX - MAX.RUN TIME FOR PRESENT RUN TRERUN - RESTART TIME ET - TIME INCREMENT KDT - TIME INCREMENT SWITCH KINT - DIVISOR FOR CHOOSING DT BETA - INTEGRATION SCHEME PARAMETER PDAMP - CRITICAL DAMPING FOR ARTIFICIAL VISCOSITY
2.2	BSTRES	IPULSE,IDIREC,N1,N2,STAMP	(4I5,E10.2)	IPULSE - STRESS PULSE TYPE SWITCH IDIREC - STRESS DIRECTION SWITCH N1 - FIRST STRESSED NODE N2 - LAST STRESSED NODE STAMP - STRESS AMPLITUDE
3.1	LINK3A	TSTRSS,TMAX,NUMOUT,IØUT,JØUT	(2E10.0,3I5)	TSTRSS - OUTPUT START TIME TMAX - OUTPUT STOP TIME NUMØUT - # OUTPUT NODES IØUT - #INTEGRATION POINTS/PRINTED OUTPUT JØUT - #INTEGRATION POINTS/TAPE OUTPUT
		(NPØUT(I),I-1,NUMØUT)	(14I5)	NPØUT - OUTPUT NODES #'s

Table 4
DATA SET CHART FOR 360 STRESS-CONSTRAINED SLAMCODE

PROGRAM	DSRN	I/O		DEVICE	SAVED	COMMENT
LINK1A	1	O	SORTED ELEMENT DATA	DISK	NEVER	LINK1 SCRATCH
	4	O	SORTED NODE DATA	DISK	NEVER	LINK1 SCRATCH
	8	O	ADJACENCY TABLES	DISK	NEVER	LINK1 SCRATCH
	14	O	PLASTIC ELEMENT DATA	DUMMY		NEVER USED IN THIS VERSION
LINK1B	1	I	AS ABOVE			
	3	O	UNBLOCKED STIFFNESS TABLES	DISK	NEVER	LINK1 SCRATCH
	4	I	AS ABOVE			
	8	I	AS ABOVE			
	12	O	LINK1C ELEMENT DATA	DISK	NEVER	LINK1 SCRATCH
LINK1C	3	O	UNBLOCKED STIFFNESS AND STRESS TABLES	DISK	NEVER	CONTINUATION OF DATA SET ABOVE
	4	I	AS ABOVE			
	8	I	AS ABOVE			
	12	I	AS ABOVE			
LINK1D	3	I	AS ABOVE			
	10	IO	BLOCKED STIFFNESS AND STRESS TAPE	TAPE	AFTER KRUN=0 RUN	OUTPUT FOR KRUN=0, INPUT FOR KRUN=1
LINK1F	12	O	PLASTIC SAVE TAPE	TAPE		THIS VERSION OF SLAMCODE IS INCAPABLE OF PLASTIC ANALYSIS; HENCE, LINK1F SHOULD NOT BE EXECUTED.
LINK2	8	IO	DISPLACEMENT TIME HISTORY TAPE	TAPE	AFTER KRUN=0(TMAX≠0)	WRITES TITLE AND COMMON BLOCK RECORDS FOR KRUN=0,1 READS TITLE AND COMMON BLOCK RECORDS FOR KRUN=2,3
LINK2A	10	I	AS ABOVE			INPUT FOR KRUN=0,1
	20	O	UPPER TRIANGULARIZATION TAPE	TAPE	AFTER KRUN=0(TMAX≠0) AND KRUN=1 RUNS	OUTPUT FOR KRUN=0,1
	22	IO	NON-SEQUENTIAL SCRATCH	DISK	NEVER	LINK2A SCRATCH
LINK2B	8	IO	AS ABOVE	TAPE	AFTER KRUN=0(TMAX≠0) 1(TMAX≠0),2,3RUNS	WRITES PROBLEM INTEGRATION OUTPUT FOR KRUN=0,1,2,3 CONTINUATION OF DATA SET ABOVE INPUT FOR KRUN=2,3
LINK3	8	I	AS ABOVE			READS TITLE AND COMMON BLOCK RECORDS FOR ALL KRUN
LINK3A	10	I	AS ABOVE			READS BLOCKED STRESS TABLES FOR ALL KRUN
LINK3B	3	O	OUTPUT TIME HISTORY TAPE	TAPE	AFTER ALL KRUN	PROBLEM OUTPUT FOR ALL KRUN
	8	I	AS ABOVE			READS INTEGRATION OUTPUT FOR ALL KRUN

```
C     PROGRAM SLAM
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1       ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(1,50),
     2       RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXBD,
     3       MBLOCK,NREADS,MAXMBK,TREAL,POAMP,BETA,KRUN,
     4       MXZONE,NZONES,IPRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
      DIMENSION NPTN(1000),IELAST(20),WGT(20),E1(20),E2(20),E3(20),
     1E4(20),E5(20)
      DIMENSION ANAME(9)
C
C     IBM 360/67 VERSION OF VISCO-ELASTIC SLAM CODE
C     CONVERTED APRIL, 1971 BY D.W. MCCOWAN WITH CONSULTATION FROM
C     C.J. COSTANTINO
C
C     UPDATED TO DO IMPLICIT TIME INTEGRATION  OCTOBER, 1971
C     UPDATED TO CONSTRAIN BOUNDARY STRESS  MARCH, 1972
C
C
C     SET LIMITS
C
      CALL NOTIFY
      MAXNP=1000
      MXADJP=8
      MXLINE=1
      MXLOAD=50
      MXNPB=100
      MAXELB=26
      MXZONE=20
      MAXMBK=30
C
C     MAXNP  = MAXIMUM NUMBER OF NODE POINTS
C     MXADJP = MAXIMUM NUMBER OF NODE POINTS ADJACENT TO ANY NODE POINT
C     MXLINE = MAXIMUM NUMBER OF LOADED LINES OR SURFACES
C     MXLOAD = MAXIMUM NUMBER OF LOADED NODES PER LOADED LINE
C     MXNPB  = MAXIMUM NUMBER OF NODE POINTS PER BUFFER
C     MAXELB = MAXIMUM PLASTIC ELEMENTS PER BLOCK
C     MXZONE = MAXIMUM NUMBER OF MATERIAL ZONES
C     MAXMBK = MAXIMUM BANDWIDTH FOR INTEGRATION SCHEME
C
C     READ AND PRINT RUN SWITCH
C
      READ(5,1) KRUN,ANAME
    1 FORMAT(I5,9A8)
      WRITE(6,4) KRUN,ANAME
    4 FORMAT(1H1,5HKRUN=,I5/1X,9A8)
C
C     KRUN  =SWITCH FOR RERUN
C     KRUN = 0 INITIAL RUN
C          = 1 LINK1 RERUN
C          = 2 LINK2 RERUN
C          = 3 LINK2 SUMMARY RERUN
C          = 4 LINK3 SUMMARY RERUN
C          = 5 LINK3 RERUN
C     ANAME =DESCRIPTOR ON DATA CARD, FOR RUN TYPE ONLY
C
      IF(KRUN.GT.1) GO TO 3
      CALL LINK1
    3 IF(KRUN.GT.3) GO TO 5
      CALL LINK2
    5 CONTINUE
      CALL LINK3
      CALL EXIT
      STOP
      END
```

```
      SUBROUTINE LINK1
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1       ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(1,50),
     2       RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXBD,
     3       MBLOCK,NREADS,MAXMBK,TREAL,POAMP,BETA,KRUN,
     4       MXZONE,NZONES,IPRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
      DIMENSION NPTN(1000),IELAST(20),WGT(20),E1(20),E2(20),E3(20),
     1E4(20),E5(20)
C
C     LINK1 PREPARES THE STIFFNESS, MASS, STRESS, AND PLASTIC TABL
C     FOR THE RUN
C
      CALL LINK1A
      IF(KRUN.EQ.1) GO TO 10
      CALL LINK1B
      CALL LINK1C
   10 CALL LINK1D
      IF(KRUN.EQ.1.OR.NUMPEL.EQ.0) RETURN
      CALL LINK1E
      RETURN
      END
```

```
      SUBROUTINE LINK1A
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1       ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(1,50),
     2       RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXBD,
     3       MBLOCK,NREADS,MAXMBK,TREAL,POAMP,BETA,KRUN,
     4       MXZONE,NZONES,IPRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
      DIMENSION NPADJ(1000,8),NADJNP(1000),NADJEL(1000),
     1NPTP(1000),S(1000),NSTART(20),IGP(100),IELAST(20),IPLAST(20),
     2WGT(20),E1(20),E2(20),E3(20),E4(20),E5(20),R(1000),Z(1000),
     3ITYPE(1000),THETA(1000),TMP(5000),NTMP(8,1250)
      DIMENSION NPTN(1000)
      EQUIVALENCE (TMP(1),NTMP(1,1))
C
C     THIS LINK READS THE MESH DATA CARDS, RENUMBERS THE MESH,
C     SORTS THE MESH DATA, AND PREPARES TABLES FOR THE STIFFNESS,
C     STRESS, AND PLASTIC MATRICES CALCULATIONS
C
C     SET MAXIMUM NUMBER OF STARTING NODES
C
      MXSTRT=20
```

```
C
C     READ AND STORE CARD INPUT
C
      CALL LNK1A1(MAXNP,MXLINE,MXLOAD,MXZONE,NUMNP,NUMEL,IPRINT,
     1ISTRES,NUMPEL,NZONES,R,Z,ITYPE,THETA,LINES,LOADNP,NPLOAD,
     2RAD,ZAO,SNORM,IELAST,IPLAST,WGT,E1,E2,E3,E4,E5,NUMST,NSTART,
     3MXSTRT,NTMP,TITLE)
C
C     ZERO OUT EVERYTHING
C
      DO 5 I=1,NUMNP
      NADJNP(I)=0
      NADJEL(I)=0
      DO 5 J=1,MXADJP
    5 NPADJ(I,J)=0
C
C     CALCULATE NPADJ, NADJNP, NADJEL
C
C     NPADJ  = ADJACENT NODE POINT NUMBERS ARRAY
C     NADJNP = NUMBER OF ADJACENT NODE POINTS ARRAY
C     NADJEL = NUMBER OF ADJACENT ELEMENTS ARRAY
C
      DO 7 M=1,NUMEL
    7 CALL ADJNP(MXADJP,MAXNP,NUMNP,NPADJ,NADJEL,NTMP(2,M),NTMP(5,M),
     1NTMP(6,M),NTMP(7,M),NTMP(8,M))
      CALL VADJNP(MXADJP,NADJNP,MAXNP,NUMNP,NPADJ)
C
C     PRINT
C
      IF(((IPRINT.NE.1).AND.((PRINT.NE.8)) GO TO 4
      WRITE(6,1)
    1 FORMAT(1H1,38HTABLE OF ORIGINAL ADJACENT NODE POINTS//
     1 4X,4HNODE,13X,6HNO. OF,4X,6HNO. OF,27X,20HADJACENT NODE POINTS/
     2 4X,5HPOINT,11X,9HADJ. PTS.,1X,9HADJ. ELS.,5X,1H(,9X,1H2,9X,(H3,
     3 9X,1H4,9X,1H5,9X,1H9,9X,1H7,9X,1H8//)
      DO 2 I=1,NUMNP
    2 WRITE(6,3) I,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP)
    3 FORMAT(I8,8X,2I10,8I10)
C
C     CALCULATE ORIGINAL BANDWIDTH
C
    4 CONTINUE
      MAXBD=0
      DO 6 I=1,NUMNP
      NUM=NADJNP(I)
      DO 6 J=1,NUM
      NP=NPADJ(I,J)
      NBAN=IABS(NP-I)
      IF(NBAN.LE.MAXBD) GO TO 6
      MAXBD=NBAN
    6 CONTINUE
C
C     RENUMBER WITH PATH PROCEDURE
C
      CALL PATH(MAXNP,NUMNP,NUMST,NSTART,NPTN,NPTP,MXADJP,NADJNP,NPADJ,
     1IGP,NUMGP)
C
C     NPTP   = NEW NODE POINT NUMBERS IN OLD NODE POINT ORDER
C     NPTN   = OLD NODE POINT NUMBERS IN NEW NODE POINT ORDER
C
C     PRINT
C
      WRITE(6,8) NUMGP
    8 FORMAT(1H1,24HNO. OF GRAND PARTITIONS=,I5)
      WRITE(6,9) (I,IGP(I),I=1,NUMGP)
    9 FORMAT(//23H PARTITION NEW NODE NO./(2X,I5,10X,I5))
  450 WRITE(6,10) MAXBD
   10 FORMAT(1H1//20H ORIGINAL BANDWIDTH=,I5)
C
C     CALCULATE AND PRINT NEW BANDWIDTH
C
      MAXBD=0
      DO 11 I=1,NUMNP
      NPNEW=NPTP(I)
      NUM=NADJNP(I)
      DO 11 J=1,NUM
      MP=NPADJ(I,J)
      MPNEW=NPTP(MP)
      NBAN=IABS(NPNEW-MPNEW)
      IF(NBAN.LE.MAXBD) GO TO 11
      MAXBD=NBAN
   11 CONTINUE
      WRITE(6,12) MAXBD
   12 FORMAT(20H NEW       BANDWIDTH=,I5)
C
C     MINIMIZE BANDWIDTH WITH MINIMAX PROCEDURE AND CHECK
C
      CALL MINI(MAXNP,NUMNP,NADJNP,MXADJP,NPADJ,NPTN,NPTP,S,
     1MAXBD)
      IF(MAXBD.LT.MXNP8) GO TO 14
      WRITE(6,13)
   13 FORMAT(//20H BANDWIDTH TOO LARGE)
      CALL EXIT
C
C     RENUMBER NPADJ
C
   14 DO 15 I=1,NUMNP
      KN=NADJNP(I)
      DO 15 J=1,KN
      KT=NPADJ(I,J)
   15 NPADJ(I,J)=NPTP(KT)
C
C     PRINT
C
      IF(((IPRINT.NE.1).AND.((PRINT.NE.8)) GO TO 19
      WRITE(6,16)
   16 FORMAT(1H1,38HTABLE OF NEW       ADJACENT NODE POINTS//
     14X,4HNEW ,3X,4HOLD ,6X,6HNO. OF,4X,6HNO. OF,27X,
     22HADJACENT NODE POINTS/4X,4HNODE,3X,4HNODE,5X,9HADJ. PTS.,
     31X,9HADJ. ELS.,5X,1H1,9X,1H2,9X,1H3,
     4 9X,1H4,9X,1H5,9X,(H9,9X,1H7,9X,1H8//)
      DO 17 I=1,NUMNP
```

A-2

```
      KP=NPTN(I)
   17 WRITE(6,18) I,KP,NADJNP(KP),NADJEL(KP),(NPADJ(KP,J),J=1,MXADJP)
   18 FORMAT(2I8,2I10,8I10)
      GO TO 26
   19 IF(IPR(NT.EQ.0) GO TO 26
      WRITE(6,20)
   20 FORMAT(1H1,21H NEW NUMBERING SCHEME//)
      DO 21 I=1,NUMNP,10
      IF((NUMNP-I).LT.10) IOUM=NUMNP-I
      IF((NUMNP-I).GE.10) IOUM=10
   23 WRITE(6,24) (J,J=I,IOUM)
   24 FORMAT(/20H ORIGINAL NODES    =,10I8)
   21 WRITE(6,25) (NPTP(J),J=I,IOUM)
   25 FORMAT( 20H NEW      NODES    =,10I8)
C
C     WRITE ADJACENCY TABLES ON TAPE
C
   26 CONTINUE
  500 IF(KRUN.EQ.1) GO TO 501
      REWIND 8
      DO 22 I=1,NUMNP
      KP=NPTN(I)
   22 WRITE(8) I,NADJNP(KP),NADJEL(KP),(NPADJ(KP,J),J=1,MXADJP)
      REWIND 8
C
C     RENUMBER NPLOAD
C
  501 DO 32 I=1,LINES
      NUM=LOADNP(I)
      DO 32 J=1,NUM
      NPOLD=NPLOAD(I,J)
   32 NPLOAD(I,J)=NPTP(NPOLD)
C
C     RENUMBER ELEMENT VERTICES AND SORT ELEMENT DATA ONTO TAPE 1 IN
C     INCREASING LOWEST NEW NODE POINT ORDER
C
C     REMEMBER COLUMNS OF NTMP ARE - KEY,NUME,IZONE,KASE,NPI,NPJ,NPK,NPL
C
      IF(KRUN.EQ.1) GO TO 631
      DO 27 I=1,NUMEL
      NTMP(5,I)=NPTP(NTMP(5,I))
      NTMP(6,I)=NPTP(NTMP(6,I))
      NTMP(7,I)=NPTP(NTMP(7,I))
      IF(NTMP(8,I).EQ.0) GO TO 29
      NTMP(8,I)=NPTP(NTMP(8,I))
      NTMP(1,I)=MIN0(NTMP(5,I),NTMP(6,I)),NTMP(7,I),NTMP(8,I))
      GO TO 27
   29 NTMP(1,I)=MIN0(NTMP(5,I),NTMP(6,I),NTMP(7,I))
   27 CONTINUE
      CALL GSORTS(NTMP,NUMEL,8,1,I)
C
C     SORT NODE DATA ONTO TAPE 4 IN INCREASING NEW NODE POINT ORDER
C
      DO 30 N=1,NUMNP
      NTMP(1,N)=NPTP(N)
      CALL LOAD(TIRIN),NTMP(2,N))
      CALL LOADIT(Z(N),NTMP(4,N))
      NTMP(6,N)=ITYPE(N)
   30 CALL LOAD(T(THETA(N),NTMP(7,N))
      CALL GSORTS(NTMP,NUMNP,8,1,4)
C
  631 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOADIT(A,B)
      LOGICAL*1 A(8),B(8)
C
C     LOAD A NUMBER
C
      DO 10 I=1,8
   10 B(I)=A(I)
      RETURN
      END
```

```
      SUBROUTINE LNK1A1(MAXNP,MXLINE,MXLOAD,MXZONE,NUMNP,NUMEL,IPR
     1ISTRES,NUMPEL,NZONES,R,Z,ITYPE,THETA,LINES,LOADNP,NPLOAD,RAD
     2SNORM,IELAST,IPLAST,WGT,E1,E2,E3,E4,E5,NUMST,NSTART,MXSTRT,M
     3TITLE)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(MAXNP),Z(MAXNP),ITYPE(MAXNP),THETA(MAXNP),
     1LOADNP(MXLINE),NPLOAD(MXLINE,MXLOAD),RAD(MXLINE,MXLOAD),
     2ZAD(MXLINE,MXLOAD),IELAST(MXZONE),IPLAST(MXZONE),WGT(MXZONE)
     3E1(MXZONE),E2(MXZONE),E3(MXZONE),E4(MXZONE),E5(MXZONE),
     4NSTART(MXSTRT),NTMP(8,1250),TITLE(10),SNORM(MXLINE,MXLOAD)
      DIMENSION ANAME(10),SSTAR(10),HSTAR(10)
C
C     MESH DATA CARD INPUT ROUTINE
C
C     READ AND PRINT MESH SIZE PARAMETER CARD
C
      READ(5,100) TITLE,NUMNP,NUMEL,ISTRES,IPRINT
  100 FORMAT(10A8/1415)
C
C     TITLE =PROBLEM TITLE
C     NUMNP =NO. OF NODE POINTS
C     NUMEL =NO. OF ELEMENTS
C     ISTRES=0  AXISYMMETRIC PROBLEM
C           =1  PLANE STRAIN PROBLEM
C           =2  PLANE STRESS PROBLEM
C     IPRINT=0  ONLY ECHO PRINT INPUT DATA
C           =1  ONLY PRINT ADJACENCY TABLE
C           =2  ONLY PRINT STIFFNESS TABLE BEFORE BOUNDARY ALTERAT
C           =3  ONLY PRINT STIFFNESS TABLE AFTER  BOUNDARY ALTERAT
C           =4  ONLY PRINT STRESS    TABLE BEFORE BOUNDARY ALTERAT
C           =5  ONLY PRINT STRESS    TABLE AFTER  BOUNDARY ALTERAT
C           =6  ONLY PRINT MASS VECTOR
C           =7  ONLY PRINT LOAD TABLES
C           =8  PRINT ALL OF ABOVE TABLES
C
```

```
      WRITE(6,101) TITLE,NUMNP,NUMEL,IPRINT
  101 FORMAT(1H1,10A8//20H NO. OF NODE POINTS=,I5/
     1                 20H NO. OF ELEMENTS  =,I5/
     2                 20H IPRINT           =,I5)
      IF(ISTRES.NE.0) GO TO 103
      WRITE(6,104) ISTRES
  104 FORMAT(20H ISTRES            =,I5,3X,20HAXISYMMETRIC PROBLEM)
      GO TO 102
  103 IF(ISTRES.NE.1) GO TO 105
      WRITE(6,106) ISTRES
  106 FORMAT(20H ISTRES            =,I5,3X,20HPLANE STRAIN PROBLEM)
      GO TO 102
  105 IF(ISTRES.NE.2) GO TO 107
      WRITE(6,108) ISTRES
  108 FORMAT(20H ISTRES            =,I5,3X,20HPLANE STRESS PROBLEM)
      GO TO 102
  107 WRITE(6,109) ISTRES
  109 FORMAT(20H ISTRES            =,I5,3X,20HERROR IN ISTRES DATA)
      CALL EXIT
C
C     READ, PRINT, AND CONVERT NODE DATA
C
  102 P=3.1415926500
      WRITE(6,110)
  110 FORMAT(1H1,15HNODE POINT DATA//12X,7HNODE PT,4X,4HTYPE,13X,
     1 15HTHETA,15X,6HRADIUS,14X,5HDEPTH/14X,3HNO.,21X,9H(DEGREES),
     2 14X,4H(FT),16X,4H(FT) //)
      READ(5,138) ANAME
  138 FORMAT(10A8)
      DO 111 I=1,NUMNP
  111 READ(5,1234)NPN,R(NPN),Z(NPN),ITYPE(NPN),THETA(NPN)
 1234 FORMAT(I5,2E10.4,I10,E10.4)
C
C     NPN    =NODE POINT NO.
C     R      =RADIUS (FT)
C     Z      =DEPTH  (FT)
C     ITYPE =0 FREE NODE
C           =1 FIXED NODE IN ONE DIRECTION (EXCEPT Z-AXIS)
C           =2 FIXED NODE IN TWO DIRECTIONS(INCLUDING Z-AXIS)
C           =3 FREE NODE ON Z-AXIS (NOT NECESSARY TO SPECIFY)
C     THETA =CLOCKWISE ANGLE OF ROLLER DIRECTION FROM R-AXIS (DEGREES)
C
      DO 16 NPN=1,NUMNP
      IF(R(NPN).NE.0.0) GO TO 17
      IF(ITYPE(NPN).EQ.2) GO TO 17
      IF(ISTRES.NE.0) GO TO 17
      ITYPE(NPN)=3
   17 WRITE(6,113)  NPN,ITYPE(NPN),THETA(NPN),R(NPN),Z(NPN)
  113 FORMAT(I16,(9,5X,(P3F20.7)
      R(NPN)=R(NPN)*12.
      Z(NPN)=Z(NPN)*12.
      THETA(NPN)=THETA(NPN)*P/180.
   16 CONTINUE
C
C     READ, PRINT AND SAVE LOADED LINE DATA
C
      READ(5,100) ANAME,LINES,(LOADNP(I),I=1,LINES)
      DO 20 I=1,LINES
      NUM=LOADNP(I)
   20 READ(5,8532)(NPLOAD(I,J),SNORM(I,J),J=1,NUM)
 8532 FORMAT(I5,E10.2)
  114 FORMAT(14I5)
C
C     ANAME =DESCRIPTOR FOR LOADED SURFACE
C     LINES =NO. OF LOADED SURFACES
C     LOADNP=NO. OF NODE POINTS ON SURFACE
C     NPLOAD=NODE NUMBERS ON LOADED SURFACE
C     SNORM =CLOCKWISE ANGLE OF SURFACE NORMAL FROM R-AXIS (DEGR
C
      WRITE(6,115) ANAME,LINES
  115 FORMAT(1H1,10A8//24H NO. OF LOADED SURFACES=,(5)
      DO 21 I=1,LINES
      WRITE(6,116) LOADNP(I),I
  116 FORMAT(//24H NO. OF NODE POINTS    =,(5,2X,14HON SURFACE N
      NUM=LOADNP(I)
      WRITE(6,117)(NPLOAD(I,J),SNORM(I,J),J=1,NUM)
  117 FORMAT(//53H LOADED NODE POINT NO.  NORMAL VECTOR ANGLE (D
     1(17X,(5,16X,1PE15.5))
      DO 21 J=1,NUM
      NP=NPLOAD(I,J)
      RAD(I,J)=R(NP)
      ZAD(I,J)=Z(NP)
      SNORM(I,J)=SNORM(I,J)*PI/180.D
   21 CONTINUE
C
C     READ, PRINT, AND WRITE ON TAPE MATERIAL ZONE DATA
C
      REWIND 14
      READ(5,138) ANAME
      READ(5,114) NZONES
      WRITE(6,118)NZONES
  118 FORMAT(1H1,18HZONE PROPERTY DATA/14H NO. OF ZONES=,(5)
      WRITE(14) NZONES
      DO 1 I=1,NZONES
      READ(5,119) IZ,ANAME
  119 FORMAT(I5,12A6)
C
C     NZONES=NO. OF MATERIAL ZONES
C     IZ     =ZONE NUMBER
C     ANAME =ZONE DESCRIPTOR
C
      WRITE(6,120) IZ,ANAME
  120 FORMAT(//13H ZONE NUMBER=,(5,2X,12A6)
C
      READ(5,121) IELAST(IZ),(PLAST(IZ),WGT(IZ),E1(IZ),E2(IZ),
     1E3(IZ),E4(IZ),E5(IZ)
  121 FORMAT(2(5,6E10.4)
C
C     WGT    =UNIT WEIGHT (LB/FT3)
C     E1     =ELASTIC MODULUS (PSI) FOR ISOTROPIC MATERIAL
C     E2     =POISSON,S RATIO FOR ISOTROPIC MATERIAL
C     E3,4,5=PARAMETERS FOR ANISOTROPIC ELASTICITY
```

A-4

```
C     IELAST=1, ISOTROPIC ELASTICITY
C           =2, ANISOTROPIC ELASTICITY
C           =3, COMPRESSIBLE FLUID
C     IPLAST=0, LINEAR MATERIAL
C           =1, VON MISES PLASTICITY
C           =3, SIMPLE COULOMB-MOHR MATERIAL
C
      WRITE(6,122) IELAST(IZ),IPLAST(IZ),WGT(IZ)
  122 FORMAT(10X,20HIELAST                =,I5/
     1       10X,20HIPLAST                =,I5/
     2       10X,20HUNIT WEIGHT           =,1PE15.5,2X,3HPCF)
C
      IF((IELAST(IZ).EQ.0).OR.(IELAST(IZ).GT.3)) GO TO 400
      IF(IELAST(IZ).GT.1) GO TO 123
      WRITE(6,124) E1(IZ),F2(IZ)
  124 FORMAT(10X,20HELASTIC MODULUS        =,1PE15.5,2X,3HPSI/
     1       10X,20HPOISSON'S RATIO        =,1PE15.5)
      GO TO 500
  123 IF(IELAST(IZ).GT.2) GO TO 125
      WRITE(6,197) E1(IZ),E2(IZ),E3(IZ),E4(IZ),E5(IZ)
  197 FORMAT(10X,20HE1                     =,1PE15.5/
     1       10X,20HE2                     =,1PE15.5/
     2       10X,20HE3                     =,1PE15.5/
     3       10X,20HE4                     =,1PE15.5/
     4       10X,20HE5                     =,1PE15.5)
      GO TO 500
  125 WRITE(6,126) E1(IZ)
  126 FORMAT(10X,20HBULK MODULUS           =,1PE15.5,2X,3HPSI)
      GO TO 500
C
  400 WRITE(6,401)
  401 FORMAT(19H ERROR IN ZONE DATA)
      CALL EXIT
C
  500 WRITE(14) IZ,IELAST(IZ),IPLAST(IZ),WGT(IZ),E1(IZ),F2(IZ)
     1E3(IZ),E4(IZ),E5(IZ)
C
      IF(IPLAST(IZ).GT.2) GO TO 400
      IF(IPLAST(IZ).EQ.0) GO TO 1
      IF(IPLAST(IZ).GT.1) GO TO 200
C
      IF(IELAST(IZ).NE.1) GO TO 400
      READ(5,114) NOYILD
      READ(5,127) (SSTAR(J),J=1,NOYILD)
      READ(5,127) (HSTAR(J),J=1,NOYILD)
  127 FORMAT(7E10.4)
C
C     NOYILD=NO. OF NONELASTIC STRAIGHT LINE SEGMENTS ON
C            STRESS-STRAIN CURVE (TENSION TEST)
C     SSTAR =STRESS AT BEGINNING OF SEGMENT (PSI)
C     HSTAR =SLOPE OF SEGMENT (PSI)
C
      WRITE(6,128) NOYILD
  128 FORMAT(10X,20HNO. OF P'. SEGMENTS=,I5)
      WRITE(6,139) SSTAR(1)
  139 FORMAT(10X,20HSTRESS AT START    =,1PE15.5,2X,3HPSI)
      IF(NOYILD.EQ.1) GO TO 141
      WRITE(6,140) (SSTAR(J),J=2,NOYILD)
  140 FORMAT(29X,1H=,1PE15.5,2X,3HPSI)
  141 WRITE(6,142) HSTAR(1)
  142 FORMAT(10X,20HSLOPE OF EL. CURVE =,1PE15.5,2X,3HPSI)
      IF(NOYILD.EQ.1) GO TO 143
      WRITE(6,140) (HSTAR(J),J=2,NOYILD)
  143 DO 129 J=1,NOYILD
      IF(HSTAR(J).GE.E1(IZ)) GO TO 400
  129 HSTAR(J)=E1(IZ)*HSTAR(J)/(E1(IZ)-HSTAR(J))
      WRITE(6,144) HSTAR(1)
  144 FORMAT(10X,20HSLOPE OF PL. CURVE =,1PE15.5,2X,3HPSI)
      IF(NOYILD.EQ.1) GO TO 145
      WRITE(6,140) (HSTAR(J),J=2,NOYILD)
  145 WRITE(14) NOYILD,(SSTAR(J),J=1,NOYILD),(HSTAR(J),J=1,NOYILD)
      GO TO 1
C
  200 IF(IELAST(IZ).NE.1) GO TO 400
      READ(5,127) COHESN,FRCTAN
C
C     COHESN=SOIL COHESION (PSI) FROM TRIAXIAL TEST
C     FRCTAN=FRICTION ANGLE (DEGREES)
C
      WRITE(6,201) COHESN,FRCTAN
  201 FORMAT(10X,20HCOHESION, TRIAXIAL =,1PE15.5,2X,3HPSI/
     1       10X,20HFRICTION ANGLE      =,1PE15.5,2X,7HDEGREES)
      FRCTAN=FRCTAN*PI/180.
      ALPHA=(2./DSQRT(3.D00))*DSIN(FRCTAN)/(3.-DSIN(FRCTAN))
      CAPPA=(6./DSQRT(3.D00))*COHESN*DCOS(FRCTAN)/(3.-DSIN(FRCTAN))
      WRITE(6,202) ALPHA,CAPPA
  202 FORMAT(10X,20HYIELD COEF, ALPHA  =,1PE15.5/
     1       10X,20HYIELD COEF, K      =,1PE15.5,2X,3HPSI)
      COSTH=DSQRT(ALPHA**2/(ALPHA**2+(1./6.)))
      IF((ISTRES.EQ.2).AND.(CAPPA.EQ.0.D0)) GO TO 400
      WRITE(14) ALPHA,CAPPA,COSTH
    1 CONTINUE
C
C     READ ELEMENT DATA, PRINT, AND REORDER VERTICES
C
      NUMPEL=0
      WRITE(6,131)
  131 FORMAT(1H1,12HELEMENT DATA//12X,7HELEMENT,5X,4HZONE,6X,3HNPI,
     17X,3HNPJ,7X,3HNPK,7X,3HNPL/14X,3HNO.,8X,3HNO.//)
      READ(5,138) ANAME
C
      DO 7 N=1,NUMEL
      NPL=0
      READ(5,5678)NUME,IZONE,NPI,NPJ,NPK,NPL
 5678 FORMAT(6I5)
C
C     NUME  =ELEMENT NUMBER
C     IZONE =ZONE NUMBER
C     NP    =NODE POINT NUMBERS
C
      CALL ORDER(NPI,NPJ,NPK,NPL,R,Z,ISTRES,KASE,MAXNP)
      NTMP(1,N)=0
```

```
      NTMP(2,M)=NUME
      NTMP(3,M)=IZONE
      NTMP(4,M)=KASE
      NTMP(5,M)=NPI
      NTMP(6,M)=NPJ
      NTMP(7,M)=NPK
      NTMP(8,M)=NPL
      WRITE(6,132) NUME,IZONE,NPI,NPJ,NPK,NPL
  132 FORMAT(I16,I11,3X,I6,4X,I6,4X,I6,4X,I6)
C
      IF(IPLAST(IZONE).EQ.0) GO TO 7
      NUMPEL=NUMPEL+1
      IF(NPL.NE.0) GO TO 133
      ITL=0
      THL=0.0
      RL=0.0
      ZL=0.0
      GO TO 134
  133 ITL=ITYPE(NPL)
      THL=THETA(NPL)
      RL=R(NPL)
      ZL=Z(NPL)
  134 WRITE(14) NUME,IZONE,IPLAST(IZONE),NPI,NPJ,NPK,NPL,
     1ITYPE(NPI),ITYPE(NPJ),ITYPE(NPK),ITL,
     2THETA(NPI),THETA(NPJ),THETA(NPK),THL,
     3     R(NPI),     R(NPJ),     R(NPK),RL,
     4     Z(NPI),     Z(NPJ),     Z(NPK),ZL
    7 CONTINUE
C
C     READ AND PRINT STARTING NODE DATA
C
      READ(5,100) ANAME,NUMST
      READ(5,114) (NSTART(I),I=1,NUMST)
C
C     NUMST =NO. OF STARTING NODES (LE.20)
C     NSTART=STARTING NODE NUMBERS
C
      WRITE(6,135) NUMST
  135 FORMAT(1H1,18HSTARTING NODE DATA//2CH NO. OF START NODES=,I5//)
      WRITE(6,136) (NSTART(I),I=1,NUMST)
  136 FORMAT(22H STARTING NODE NUMBERS/(19X,I5))
C
      PENINC 14
      RETURN
      END
```

```
      SUBROUTINE ORDER(NPI,NPJ,NPK,NPL,R,Z,ISTRES,KASE,MAXNP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(MAXNP),Z(MAXNP)
C
C     ORDER NODE POINT LEFTFRING FOR ELEMENT AND DEFINE CASE
C
C     R       = RADIAL COORDINATE OF NODE POINT
C     Z       = VERTICAL COORDINATE OF NODE POINT
C     KASE    = 1 GENERAL TRIANGLE
C             = 2 TRIANGLE, ONE NODE ON Z-AXIS
C             = 3 TRIANGLE, TWO NODES ON Z-AXIS
C             = 4 GENERAL RECTANGLE
C             = 5 RECTANGLE, ONE NODE ON Z-AXIS
C             = 6 RECTANGLE, TWO NODES ON Z-AXIS
C     ISTRES  = 0 AXISYMMETRIC PROBLEM
C             = 1 PLANE STRAIN PROBLEM
C             = 2 PLANE STRESS PROBLEM
C
      NI=NPI
      NJ=NPJ
      NK=NPK
      NL=NPL
      IF(R(NI).LT.R(NJ)) GO TO 1
      IF(R(NI).NE.R(NJ)) GO TO 2
      IF(Z(NI).LT.Z(NJ)) GO TO 1
    2 NI=NPJ
      NJ=NPI
      NPI=NI
      NPJ=NJ
    1 IF(R(NI).LT.R(NK)) GO TO 3
      IF(R(NI).NE.R(NK)) GO TO 4
      IF(Z(NI).LT.Z(NK)) GO TO 3
    4 NI=NPK
      NK=NPI
      NPI=NI
      NPK=NK
    3 IF(NPL.EQ.0) GO TO 5
      IF(R(NI).LT.R(NL)) GO TO 5
      IF(R(NI).NE.R(NL)) GO TO 6
      IF(Z(NI).LT.Z(NL)) GO TO 5
    6 NI=NPL
      NL=NPI
      NPI=NI
      NPL=NL
C
C     NODE POINT I IN PROPER LOCATION (CLOSEST TO ORIGIN)
C
    5 AJ=R(NJ)-R(NI)
      BJ=Z(NJ)-Z(NI)
      AK=R(NK)-R(NI)
      BK=Z(NK)-Z(NI)
      HJK=AJ*BK-AK*BJ
      IF(NPL.NE.0) GO TO 7
C
C     TRIANGULAR ELEMENT
C
      IF(HJK.GT.0.) GO TO 8
      NPK=NJ
      NPJ=NK
    8 IF(R(NPI).EQ.0.) GO TO 9
   10 KASE=1
      RETURN
    9 IF(ISTRES.NE.0) GO TO 10
      IF(R(NPK).EQ.0.) GO TO 11
      KASE=2
```

```
      RETURN
   11 KASE=3
      RETURN
C
C     RECTANGULAR ELEMENT
C
    7 AL=R(NL)-R(NI)
      BL=Z(NL)-Z(NI)
      HJL=AJ*BL-AL*BJ
      HKL=AK*BL-AL*BK
      IF(HJK.GT.0.) GO TO 12
      IF(HJL.GT.0.) GO TO 13
      IF(HKL.GT.0.) GO TO 14
      NPJ=NL
      NPL=NJ
      GO TO 15
   14 NPJ=NK
      NPK=NL
      NPL=NJ
      GO TO 15
   13 NPJ=NK
      NPK=NJ
      GO TO 15
   12 IF(HJL.GT.0.) GO TO 16
      NPJ=NL
      NPK=NJ
      NPL=NK
      GO TO 15
   16 IF(HKL.GT.0.) GO TO 15
      NPK=NL
      NPL=NK
   15 IF(R(NP)).EQ.0.) GO TO 17
   18 KASE=4
      RETURN
   17 IF(ISTRES.NE.0) GO TO 18
      IF(P(NPL).EQ.0.) GO TO 19
      KASE=5
      RETURN
   19 KASE=6
      RETURN
      END
```

```
      SUBROUTINE ADJNP(MXADJP,MAXNP,NUMNP,NPADJ,NADJEL,NUME,
     1NPI,NPJ,NPK,NPL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NPADJ(MAXNP,MXADJP),NADJEL(MAXNP),NA(4)
C
C     FORM TABLE OF ADJACENT NODAL POINTS
C     (TABLE ASSUMED TO BE ALREADY ZEROED OUT)
C
      NA(1)=NPI
      NA(2)=NPJ
      NA(3)=NPK
      NA(4)=NPL
      ICOUNT=1
    9 NPNUM=NA(1)
      NADJEL(NPNUM)=NADJEL(NPNUM)+1
      MX=NA(2)
      JCOUNT=1
    5 DO 1 I=1,MXADJP
      J=I
      IF(NPADJ(NPNUM,I).EQ.MX) GO TO 2
      IF(NPADJ(NPNUM,I).EQ.0)  GO TO 3
    1 CONTINUE
      WRITE (6,101)  NUME,NPNUM,MX,(NPADJ(NPNUM,I),I=1,MXADJP)
      CALL EXIT
C
    3 NPADJ(NPNUM,J)=MX
    2 JCOUNT=JCOUNT+1
      IF(JCOUNT.GT.3) GO TO 4
      IF(JCOUNT.GT.2) GO TO 102
      MX=NA(3)
      GO TO 5
  102 MX=NA(4)
      IF(MX.EQ.0) GO TO 4
      GO TO 5
C
    4 GO TO (6,7,8,103),ICOUNT
    6 ICOUNT=2
      NA(1)=NPJ
      NA(2)=NPK
      NA(3)=NPI
      NA(4)=NPL
      GO TO 9
C
    7 ICOUNT=3
      NA(1)=NPK
      NA(2)=NPI
      NA(3)=NPJ
      NA(4)=NPL
      GO TO 9
C
    8 ICOUNT=4
      NA(1)=NPL
      IF(NA(1).EQ.0) GO TO 103
      NA(2)=NPI
      NA(3)=NPJ
      NA(4)=NPK
      GO TO 9
C
  101 FORMAT(1H1,43HERROR IN FORMING ADJACENT NODAL POINT ARRAY/
     121H ELEMENT NUMBER      =,I5/21H NODE POINT NUMBER  =,I5/
     221H ADJACENT NODE POINT=,I5//15H NPADJ(NPNUM,I)/(21X,I5))
C
  103 RETURN
      END
```

```
      SUBROUTINE VADJNP(MXADJP,NADJNP,MAXNP,NUMNP,NPADJ)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP)
```

```
C
C     FORM NUMBER OF ADJACENT NODE POINTS ARRAY
C
      DO 12 M=1,NUMNP
      DO 10 I=1,MXADJP
      J=I
      IF (NPADJ(M,I).EQ.0) GO TO 11
   10 CONTINUE
      NADJNP(M)=MXADJP
      GO TO 12
   11 NADJNP(M)=J-1
   12 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PATH(MAXNP,NUMNP,NUMST,NSTART,NPTN,NPTP,MXADJP,
     1NADJNP,NPADJ,IGP,NUMGP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NSTART(NUMST),NPTN(MAXNP),NPTP(MAXNP),NADJNP(MAXNP),
     1NPADJ(MAXNP,MXADJP),IGP(1)
C
C     MINIMIZE BANDWIDTH WITH PATH PROCEDURE
C
C     NSN    = NUMBER OF STARTING NODES
C     NSNR   = STARTING NODE ARRAY
C     IGP    = GRAND PARTITION ARRAY
C     NUMGP  = NUMBER OF GRAND PARTITIONS
C
      KOUNT=0
      IN=1
      DO 1 I=1,NUMNP
      NPTN(I)=0
    1 NPTP(I)=0
C
      DO 2 I=1,NUMST
      NP=NSTART(I)
      NPTP(NP)=IN
      KOUNT=KOUNT+1
    2 NPTN(KOUNT)=NP
C
      IGP(IN)=KOUNT
C
    4 DO 7 I=1,NUMNP
      IF(NPTP(I).NE.IN) GO TO 7
      NUM=NADJNP(I)
      DO 3 J=1,NUM
      NP=NPADJ(I,J)
      IF(NPTP(NP).NE.0) GO TO 3
      NPTP(NP)=IN+1
      KOUNT=KOUNT+1
      NPTN(KOUNT)=NP
      IF(KOUNT .EQ.NUMNP) GO TO 5
    3 CONTINUE
    7 CONTINUE
C
      IN=IN+1
      IGP(IN)=KOUNT
      GO TO 4
C
    5 IGP(IN+1)=KOUNT
      NUMGP=IN+1
C
      DO 6 I=1,NUMNP
      NPOLD=NPTN(I)
    6 NPTP(NPOLD)=I
C
      RETURN
      END
```

```
      SUBROUTINE MINI(MAXNP,NUMNP,NADJNP,MXADJP,NPADJ,NPTN,NPTP,S,
     1MAXBDP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP),NPTN(MAXNP),
     1NPTP(MAXNP),S(MAXNP)
C
C     MINIMIZE BANDWIDTH WITH MINIMAX PROCEDURE
C
    3 DO 1 I=1,NUMNP
      S(I)=FLOAT(I)
      NPOLD=NPTN(I)
      NUM=NADJNP(NPOLD)
      DO 2 J=1,NUM
      NADJ=NPADJ(NPOLD,J)
      NPNEW=NPTP(NADJ)
    2 S(I)=S(I)+FLOAT(NPNEW)
    1 S(I)=S(I)/FLOAT(NUM+1)
C
      CALL SORT1(S,NPTN,NUMNP,2,1,1,1)
C
      MAXBD=0
      DO 14 I=1,NUMNP
      NPOLD=NPTN(I)
      NUM=NADJNP(NPOLD)
      DO 10 J=1,NUM
      NADJ=NPADJ(NPOLD,J)
      DO 11 K=1,NUMNP
      KK=K
      IF(NADJ.EQ.NPTN(K)) GO TO 15
   11 CONTINUE
      WRITE(6,100) I,NPOLD,(NPTN(L),L=1,NUMNP)
  100 FORMAT(1H1,13HERROR IN MINI//10X,2I10//(10X,10I10))
      CALL EXIT
   15 NPNEW=KK
      NBAN=IABS(NPNEW-I)
      IF(MAXBD.LT.NBAN) MAXBD=NBAN
   10 CONTINUE
   14 CONTINUE
C
      WRITE(6,6) MAXBD
    6 FORMAT(20H NEW       BANDWIDTH=,I5)
```

```
C
      IF(MAX8DP.LE.MAX8D) GO TO 12
C
      DO 5 I=1,NUMNP
      NPOLD=NPTN(I)
    5 NPTP(NPOLD)=I
      MAX8DP=MAX8D
      GO TO 3
C
   12 DO 16 I=I,NUMNP
      NPNEW=NPTP(I)
   16 NPTN(NPNEW)=I
      RETURN
      END
```

```
      SUBROUTINE GSORTS(IARRAY,NRCDS,NWRDS,NKEY,IOUTAP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IARRAY(NWRDS,NRCDS)
C
C     SORT DATA AND WRITE ONTO TAPE
C
      REWIND IOUTAP
      CALL SORT1(IARRAY,IARRAY,NRCDS,NWRDS,NWRDS,NKEY,0)
      DO 3 I=1,NRCDS
    3 WRITE(IOUTAP)(IARRAY(J,I),J=1,NWRDS)
      REWIND IOUTAP
      RETURN
      END
```

```
      SUBROUTINE SORT1(IARRAY,JARRAY,NRECDS,IWRDS,JWRDS,IKEY,ISWT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IARRAY(IWRDS,NRECDS),JARRAY(JWRDS,NRECDS)
C
C     GENERAL PURPOSE SORTER
C
      LOGICAL CHECK
      M=NRECDS-1
C
    1 CHECK=.FALSE.
      DO 6 I=1,2
      DO 2 J=I,M,2
      IF(IARRAY(IKEY,J).LE.IARRAY(IKEY,J+1)) GO TO 2
      DO 3 K=1,IWRDS
      ITEMP=IARRAY(K,J)
      IARRAY(K,J)=IARRAY(K,J+1)
    3 IARRAY(K,J+1)=ITEMP
      IF(ISWT.EQ.0) GO TO 5
      DO 4 K=I,JWRDS
      JTEMP=JARRAY(K,J)
      JARRAY(K,J)=JARRAY(K,J+1)
    4 JARRAY(K,J+1)=JTEMP
    5 CHECK=.TRUE.
    2 CONTINUE
    6 CONTINUE
C
      IF(CHECK) GO TO 1
      RETURN
      END
```

```
      SUBROUTINE LINK1B
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1       ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(I,50),
     2       RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAX8D,
     3       M8LOCK,NREADS,MAXM8K,TREAL,PDAMP,BETA,KRUN,
     4       MXZONE,NZONES,IPRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
      DIMENSION NPTN(1000),NADJNP(100),NPADJ(100,8),NADJEL(100),R(1
     1Z(100),ITYPE(100),THETA(100),XMASS(100)
      DIMENSION C(4,4),CK(8,8),AINT(20)
      DIMENSION IELAST(20),WGT(20),E1(20),E2(20),E3(20),E4(20),E5(20
      DIMENSION SNPUU(100),SNPUW(100),SNPWW(100),SADUU(100,8),
     1SADUW(100,8),SADWU(100,8),SADWW(100,8)
C
C     THIS LINK CALCULATES THE MASS AND STIFFNESS MATRICES FOR EACH
C     ELEMENT, MAKES THE ROLLER SUPPORT MODIFICATIONS, AND DISTRIBU
C     THE ELEMENT STIFFNESSES TO NODE POINT STIFFNESSES
C
C     KEND   = CONTROL CONSTANT TO DETERMINE END OF PROCESSING
C     NPOUT  = FIRST NODE POINT MINUS ONE IN BUFFERS
C     NUMCP  = NUMBER OF COMPLETED NODE POINTS IN BUFFERS
C     NUMNPB = NUMBER OF NODE POINTS IN BUFFERS (COMPLETED OR INITI
C     NPR    = NUMBER OF UNCOMPLETED NODE POINTS IN BUFFERS
C     KX     = WHERE TO START ERASING BUFFERS
C
C     INITIALIZE
C
      REWIND 1
      REWIND 3
      REWIND 4
      REWIND 8
      REWIND 12
C
      KEND=0
      NPOUT=0
      NUMCP=0
      NUMNPB=0
      NPR=MXNPB
      KX=1
      IU=0
      DUMU=0.0
      IW=0
      DUMW=0.0
C
C     GO TO ZERO OUT ALL OF NODE POINT ARRAYS
C
    1 ISWTCH=1
      GO TO 900
C
C     FILL UP NODE POINT ARRAYS INITIALLY
```

```
C
    2 IF(NUMNP.LT.MXNPB) NUMNPB=NUMNP
      IF(NUMNP.GE.MXNPB) NUMNPB=MXNPB
      DO 3 I=1,NUMNPB
      READ(8) NPN,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP)
      READ(4) NPN,R(I),Z(I),ITYPE(I),THETA(I)
    3 CONTINUE
C
C     READ ANOTHER ELEMENT DATA RECORD
C
      ICOUNT=0
    4 READ(1) KEY,NUME,IZONE,KASE,NTI,NTJ,NTK,NTL
      ICOUNT=ICOUNT+1
      LNP=MAXO(NTI,NTJ,NTK,NTL)
C
C     CHECK TO SEE IF LARGEST NODE POINT IN THIS ELEMENT FALLS OUTSIDE
C     BUFFER
C
      IF((LNP-NPOUT).GT.MXNPB) GO TO 100
C
C     PROCESS THIS ELEMENT
C
    6 NPI=NTI-NPOUT
      NPJ=NTJ-NPOUT
      NPK=NTK-NPOUT
      IF(NTL.EQ.0) NPL=0
      IF(NTL.NE.0) NPL=NTL-NPOUT
      S1=0.0
      C1=0.0
      IE=IELAST(IZONE)
      A1=E1(IZONE)
      A2=E2(IZONE)
      A3=E3(IZONE)
      A4=E4(IZONE)
      A5=E5(IZONE)
      RHO=WGT(IZONE)/(386.4*1728.)
      CALL ELAST(IE,ISTRES,A1,A2,A3,A4,A5,C,NUME)
      CALL STIFF(KASE,NPI,NPJ,NPK,NPL,NUME,MXNPB,ISTRES,C,R,Z,CK,AINT,
     1S1,C1)
      CALL ADJUST(MXNPB,CK,ITYPE,THETA,NADJEL,NPI,NPJ,NPK,NPL)
      CALL OSTK(MXNPB,MXADJP,CK,SNPUU,SNPUW,SNPWW,SADUU,SADUW,SADWU,
     1SADWW,NPI,NPJ,NPK,NPL,NPADJ,NPOUT)
      CALL MASS(MXNPB,RHO,R,Z,AINT,XMASS,S1,C1,NPI,NPJ,NPK,NPL,ISTRES)
      WRITE(12) KEY,NUME,IZONE,NTI,NTJ,NTK,NTL,((C(I,J),I=1,4),J=1,4),
     1KASE,S1,C1
C
C     GO TO READ ANOTHER ELEMENT RECORD OR SET STOP SWITCH
C
      IF(ICOUNT.LT.NUMEL) GO TO 4
      KEND=1
      KEY=NUMNP+1
C
C     SET-UP FOR DUMPING BUFFERS
C
  100 NUMCP=KEY-1
      NUMNPB=NUMCP-NPOUT
C
C     PRINT AND WRITE FINISHED NODE DATA ON TAPE
C
  113 CALL        PRNK(MXNPB,MXADJP,NADJNP,NPADJ,NADJEL,IPRINT,SNPUU,
     1SNPUW,SNPWW,SADUU,SADUW,SADWU,SADWW,THETA,ITYPE,XMASS,NPOUT,
     2NUMNPB)
      DO 101 I=1,NUMNPB
  101 WRITE( 3) I  ,NADJNP(I),ITYPE(I),THETA(I),XMASS(I),SNPUU(I),
     1SNPUW(I),SNPWW(I),(NPADJ(I,J),SADUU(I,J),SADUW(I,J),SADWU(I,J),
     2SADWW(I,J),J=1,MXADJP)
C
C     UPDATE PERIOD DATA
C
  104 DO 105 I=1,NUMNPB
      IF(ITYPE(I).EQ.2) GO TO 106
      DUM=SNPUU(I)/XMASS(I)
      IF(DUM.LT.DUMU) GO TO 106
      DUMU=DUM
      IU=NPOUT+I
  106 IF(ITYPE(I).EQ.2.OR.ITYPE(I).EQ.1) GO TO 105
      DUM=SNPWW(I)/XMASS(I)
      IF(DUM.LT.DUMW) GO TO 105
      DUMW=DUM
      IW=NPOUT+I
  105 CONTINUE
C
C     SKIP OUT FOR LAST ELEMENT COMPLETED OR GO TO REFILL NODE ARRAYS
C
      IF(KEND.EQ.1) GO TO 300
      NPR=MXNPB-NUMNPB
      GO TO 902
C
C     GO TO ERASE REMAINDER OF BUFFERS
C
  107 KX=NPR+1
      ISWTCH=2
      GO TO 900
C
C     FILL UP REMAINDER OF ARRAYS WITH NEW NODE DATA
C
  108 IF((NUMNP-NUMCP).LT.MXNPB) KNP=NUMNP-NUMCP-NPR
      IF((NUMNP-NUMCP).GE.MXNPB) KNP=MXNPB-NPR
      DO 109 I=1,KNP
      L=NPR+I
      READ(8) NPN,NADJNP(L),NADJEL(L),(NPADJ(L,J),J=1,MXADJP)
      READ(4) NPN,R(L),Z(L),(TYPE(L),THETA(L)
  109 CONTINUE
      NPOUT=NUMCP
      GO TO 6
C
C     MOVE UNPROCESSED DATA TO TOP OF BUFFERS
C
  902 DO 903 K=1,NPR
      L=NUMNPB+K
      NADJNP(K)=NADJNP(L)
      NADJEL(K)=NADJEL(L)
```

```
      ITYPE(K)=ITYPE(L)
      THETA(K) =THETA(L)
      R(K)     =R(L)
      Z(K)     =Z(L)
      XMASS(K) =XMASS(L)
      SNPUU(K) =SNPUU(L)
      SNPUW(K) =SNPUW(L)
      SNPWW(K) =SNPWW(L)
      DO 903 J=1,MXADJP
      NPADJ(K,J)=NPADJ(L,J)
      SADUU(K,J)=SADUU(L,J)
      SADUW(K,J)=SADUW(L,J)
      SADWU(K,J)=SADWU(L,J)
  903 SADWW(K,J)=SADWW(L,J)
      GO TO 107
C
C     ZERO OUT ALL OR PART OF BUFFERS
C
  900 DO 901 L=KK,MXNPR
      NADJNP(L)=0
      NADJEL(L)=0
      ITYPE(L)=0
      THETA(L)=0.0
      R(L)=0.0
      Z(L)=0.0
      XMASS(L)=0.0
      SNPUU(L)=0.0
      SNPUW(L)=0.0
      SNPWW(L)=0.0
      DO 901 J=1,MXADJP
      NPADJ(L,J)=0
      SADUU(L,J)=0.0
      SADUW(L,J)=0.0
      SADWU(L,J)=0.0
  901 SADWW(L,J)=0.0
      GO TO (2,108),ISWTCH
C
C     CALCULATE MINIMUM DIAGONAL PERIOD OF OSCILLATION
C
  300 PI=3.1415927
      IF(DUMU.EQ.0.0) GO TO 330
      OUMU=2.*PI/DSQRT(DUMU)
  330 IF(DUMW.EQ. 0.0) GO TO 331
      OUMW=2.*PI/DSQRT(DUMW)
  331 CONTINUE
      IU=NPTN(IU)
      IW=NPTN(IW)
      WRITE(6,301) IU,OUMU,IW,OUMW
  301 FORMAT(1H1,45H SMALLEST MAIN DIAGONAL FREQUENCY (SEC/CYCLE)//
     1 12H NODE POINT=,I5,5X,8HPERIODU=,1PE15.5/
     2 12H NODE POINT=,I5,5X,8HPERIODW=,1PE15.5/)
      PERIOD=OMIN1(OUMU,OUMW)
      IF(DUMU.EQ.0.0) PERIOD=OUMW
      IF(OUMW.EQ.0.0) PERIOD=OUMU
C
      REWIND 1
      REWIND 4
      REWIND 8
      REWIND 12
      RETURN
      END
```

```
      SUBROUTINE ELAST(IELAST,ISTRES,E1,E2,E3,E4,E5,C,NUME)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(4,4)
C
C     FORM STRESS-STRAIN MATRIX, F IN C
C
      DO 1 I=1,4
      DO 1 J=1,4
    1 C(I,J)=0.0
C
      IF(IELAST.NE.1) GO TO 20
C
      IF(ISTRES.EQ.2) GO TO 4
C
C     ISOTROPIC AXISYMMETRIC OR PLANE STRAIN PROBLEM
C
      EBAR=E1/((1.+E2)*(1.-2.*E2))
      C(1,1)=EBAR*(1.-E2)
      C(1,2)=EBAR*E2
      C(1,3)=C(1,2)
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=C(1,2)
      C(3,1)=C(1,2)
      C(3,2)=C(1,2)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-2.*E2)/2.
      RETURN
C
C     ISOTROPIC PLANE STRESS PROBLEM
C
    4 EBAR=E1/(1.-E2*E2)
      C(1,1)=EBAR
      C(3,1)=EBAR*E2
      C(1,3)=C(3,1)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-E2)/2.
      RETURN
C
   20 IF(IELAST.NE.2) GO TO 30
C
      IF(ISTRES.EQ.2) GO TO 2
C
C     ANISOTROPIC AXISYMMETRIC OR PLANE STRAIN PROBLEM
C
      C(1,1)=E1
      C(1,2)=E1-2.*E5
      C(1,3)=E3
      C(2,1)=C(1,2)
```

```
      C(2,2)=C(1,1)
      C(2,3)=C(1,3)
      C(3,1)=C(1,3)
      C(3,2)=C(2,3)
      C(3,3)=E2
      C(4,4)=E4
      RETURN
C
C     ANISOTROPIC PLANE STRESS PROBLEM
C
    2 C(1,1)=2.*E5*(E1-2.*E5)/E1
      C(1,3)=2.*E3*E5/E1
      C(3,1)=C(1,3)
      C(3,3)=E2-E3**2/E1
      C(4,4)=E4
      RETURN
C
   21 WRITE(6,3) IELAST,NUME,ISTRES
    3 FORMAT(1H1/31H ERROR IN ELASTIC CONSTANT DATA/
     113H IELAST      =,I5/13H ELEMENT NO.=,I5/
     213H ISTRES      =,I5)
      CALL EXIT
C
C     COMPRESSIBLE FLUID
C
   30 IF(IELAST.NE.3) GO TO 21
C
      IF(ISTRES.EQ.2) GO TO 21
C
      DO 31 I=1,3
      DO 31 J=1,3
   31 C(I,J)=E1
C
      RETURN
      END
```

```
      SUBROUTINE STIFF(KASE,NPI,NPJ,NPK,NPL,NUME,MAXNP,ISTRES,C,R,Z,
     1CK,AI,S1,C1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(4,4),R(MAXNP),Z(MAXNP),CK(8,8),AI(23),D(8,8),G(8,8),
     1VEC(8)
C
C     COMPUTE ELEMENT STIFFNESS MATRIX
C
C     KASE    = 1 GENERAL TRIANGLE
C             = 2 NODE I ON Z-AXIS
C             = 3 NODES I, K ON Z-AXIS
C             = 4 GENERAL RECTANGLE
C             = 5 NODE I ON Z-AXIS
C             = 6 NODES I, L ON Z-AXIS
C     C       = ELASTIC MODULI MATRIX
C     CK      = STIFFNESS MATRIX
C     AI      = INTEGRALS FOR COMPUTING K AND M
C
      DO 1 I=1,8
      DO 1 J=I,8
      D(I,J)=0.0
      G(I,J)=0.0
    1 CK(I,J)=0.0
      CALL INS(KASE,NPI,NPJ,NPK,NPL,ISTRES,R,Z,AI,S1,C1,MAXNP)
C
      IF(NPL.NE.0) GO TO 300
C
C     TRIANGULAR ELEMENT
C
      AJ=R(NPJ)-R(NPI)
      AK=R(NPK)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      BK=Z(NPK)-Z(NPI)
      H=AJ*BK-AK*BJ
      B=BJ-BK
      A=AJ-AK
      IF(KASE.NE.1) GO TO 2
      D(1,1)=1.0
      D(2,1)=B/H
      D(3,1)=-A/H
    2 D(4,2)=1.0
      D(5,2)=B/H
      D(6,2)=-A/H
      D(2,3)=BK/H
      D(3,3)=-AK/H
      D(5,4)=D(2,3)
      D(6,4)=D(3,3)
      IF(KASE.EQ.3) GO TO 3
      D(2,5)=-BJ/H
      D(3,5)=AJ/H
    3 D(5,6)=-BJ/H
      D(6,6)=AJ/H
      NORD=6
C
      IF(KASE.NE.1) GO TO 4
      IF(ISTRES.NE.0) GO TO 4
      G(1,1)=C(2,2)*AI(5)
      G(2,1)=C(1,2)*AI(1)+C(2,2)*AI(7)
      G(3,1)=C(2,2)*AI(6)
      G(6,1)=C(2,3)*AI(1)
    4 G(2,2)=C(1,1)*AI(4)+2.*C(1,2)*AI(3)+C(2,2)*AI(10)
      IF(KASE.EQ.3) GO TO 5
      G(3,2)=C(1,2)*AI(2)+C(2,2)*AI(9)
      G(3,3)=C(2,2)*AI(8)+C(4,4)*AI(4)
      G(5,3)=C(4,4)*AI(4)
      G(6,3)=C(2,3)*AI(2)
    5 G(6,2)=C(1,3)*AI(4)+C(2,3)*AI(3)
      G(5,5)=C(4,4)*AI(4)
      G(6,6)=C(3,3)*AI(4)
      GO TO 301
C
C     RECTANGULAR ELEMENT
C
  300 AJ=R(NPJ)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
```

```
      A=DSQRT(AJ*AJ+BJ*BJ)
      AL=R(NPL)-R(NPI)
      BL=Z(NPL)-Z(NPI)
      B=DSQRT(AL*AL+BL*BL)
      H=A*B
      IF(KASE.NE.4) GO TO 6
      D(1,1)=1.0
      D(2,1)=-B/H
      D(3,1)=1./H
      D(4,1)=-A/H
    6 D(5,2)=1.0
      D(6,2)=-B/H
      D(7,2)=1./H
      D(8,2)=-A/H
      D(2,3)=-D(6,2)
      D(3,3)=-D(7,2)
      D(6,4)=D(2,3)
      D(7,4)=D(3,3)
      D(3,5)=D(7,2)
      D(7,6)=D(7,2)
      IF(KASE.EQ.6) GO TO 7
      D(3,7)=D(3,3)
      D(4,7)=-D(8,2)
    7 D(7,8)=D(7,4)
      D(8,8)=-D(8,2)
      NORD=8
C
      IF(KASE.NE.4) GO TO 8
      IF(ISTRES.NE.0) GO TO 8
      G(1,1)=C(2,2)*AI(5)
      G(2,1)=C(1,2)*C1*AI(1)+C(2,2)*AI(7)
      G(3,1)=C(1,2)*(C1*AI(2)+S1*AI(3))+C(2,2)*A((9)
      G(4,1)=C(1,2)*S1*A((1)+C(2,2)*AI(6)
      G(6,1)=-C(2,3)*S1*A((1)
      G(7,1)=C(2,3)*(C1*AI(3)-S1*A((2))
      G(8,1)=C(2,3)*(C1*AI(1))
C
    8 DUM1=C1*AI(14)+S1*A((13)
      DUM2=C1*AI(13)-S1*AI(14)
      DUM3=C1*C1*AI(12)+2.*S1*C1*A((15)+S1*S1*AI((1)
      DUM4=S1*S1*AI(12)-2.*S1*C1*AI(15)+C1*C1*A((11)
      G(2,2)=C1*(C(1,1)*C1*A((4)+2.*C(1,2)*AI(3))+C(2,2)*AI(10)
     1+C(4,4)*S1*S1*AI(4)
      G(3,2)=C(1,1)*C1*DUM1+C(1,2)*(2.*C1*A1(16)+S1*AI(18))
     1+C(2,2)*AI(17)-C(4,4)*S1*DUM2
      G(7,2)=C(1,3)*C1*DUM2+C(2,3)*(C1*A((18)-S1*AI(16))
     1-C(4,4)*S1*DUM1
      G(8,2)=AI(4)*(C(1,3)*C1*C1-C(4,4)*S1*S1)+C(2,3)*C1*AI(3)
      G(3,3)=C(1,1)*DUM3+2.*C(1,2)*(C1*AI(20)+S1*A((23))
     1+C(2,2)*AI(21)+C(4,4)*DUM4
      G(6,3)=-C(1,3)*S1*DUM1-C(2,3)*S1*AI(16)+C(4,4)*C1*DUM2
      G(7,3)=(C(1,3)+C(4,4))*(S1*C1*(AI(11)-AI(12))+AI(15)*(C1*C1-S(*S1)
     1)+C(2,3)*(C1*A((23)-S1*AI(20))
      G(8,3)=C(1,3)*C1*DUM1+C(2,3)*C1*AI(16)+C(4,4)*S1*DUM2
      G(6,6)=(C(3,3)*S1*S1+C(4,4)*C1*C1)*AI(4)
      G(7,6)=-C(3,3)*S1*DUM2+C(4,4)*C1*DUM1
      G(7,7)=C(3,3)*DUM4+C(4,4)*DUM3
      G(8,7)=C(3,3)*C1*DUM2+C(4,4)*S1*DUM1
      G(8,8)=(C(3,3)*C1*C1+C(4,4)*S1*S1)*AI(4)
      IF(KASE.EQ.6) GO TO 301
      G(4,2)=(C(1,1)-C(4,4))*S1*C1*AI(4)+C(1,2)*(C1*AI(2)+S1*AI(3))
     1+C(2,2)*AI(9)
      G(6,2)=-(C(1,3)+C(4,4))*S1*C1*A((4)-C(2,3)*S1*AI(3)
      G(4,3)=C(1,1)*S1*DUM1+C(1,2)*(C1*AI(19)+2.*S1*AI(16))
     1+C(2,2)*AI(22)+C(4,4)*C1*DUM2
      G(4,4)=C(1,1)*S1*S1*AI(4)+2.*C(1,2)*S1*A1(2)+C(2,2)*AI(8)
     1+C(4,4)*C1*C1*AI(4)
      G(6,4)=AI(4)*(C(4,4)*C1*C1-C(1,3)*S1*S1)-C(2,3)*S1*A1(2)
      G(7,4)=C(1,3)*S1*DUM2+C(2,3)*(C1*AI(16)-S1*AI(19))+C(4,4)*C1*DU
      G(8,4)=(C(1,3)+C(4,4))*S1*C1*AI(4)+C(2,3)*C1*AI(2)
      G(8,6)=S1*C1*(C(4,4)-C(3,3))*A1(4)
C
C     FILL IN REST OF G AND CALCULATE K
C
  301 DO 201 I=2,NORD
      K=I-1
      DO 201 J=1,K
  201 G(J,I)=G(I,J)
C
      DO 51 J=1,NORD
      DO 50 L=1,NORD
      VEC(L)=0.0
      DO 50 K=1,NORD
   50 VEC(L)=VEC(L)+G(L,K)*D(K,J)
      DO 51 I=1,NORD
      CK(I,J)=0.0
      DO 51 L=1,NORD
   51 CK(I,J)=CK(I,J)+D(L,I)*VEC(L)
C
      RETURN
      END
```

```
      SUBROUTINE INS(KASE,NPI,NPJ,NPK,NPL,ISTRES,R,Z,AI,S1,CI,MAXNP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AI(23),R(MAXNP),Z(MAXNP)
C
C     COMPUTE ELEMENT INTEGRALS
C
      DO 1 I=1,23
    1 AI(I)=0.
C
      IF(NPL.NE.0) GO TO 300
C
C     TRIANGULAR ELEMENT
C
      AJ=R(NPJ)-R(NPI)
      AK=R(NPK)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      BK=Z(NPK)-Z(NPI)
      H=AJ*BK-AK*BJ
      B=BJ-BK
```

```
      A=AJ-AK
      R1=R(NP1(
C
      IF((STRES.EQ.0( GO TO 2
      A((4)=H/2.
      A1(13)=H*(AJ+AK)/6.
      A((14)=H*(BJ+BK(/6.
      RETURN
C
    2 A1(1)=H/2.
      A1(2)=H*(BJ+BK)/6.
      A((3)=H*(AJ+AK(/6.
      A1(4)=R(*A((1(+A1(3(
      (F(KASE.EQ.1( GO TO 3
      A((9(=A((2(
      A((10(=A((4(
C
    3 ICOUNT=1
      RA=R(
      RB=R(NPJ)
      C=BJ/AJ
      D=0.
      DUM=-1.
      (F(C.EQ.0.( GO TO 100
  101 (F(KASE.EQ.1( GO TO 102
      (F(KASE.NE.2( GO TO 104
      (F(RA.NE.0.( GO TO 102
      FC=OLOG(RB)
      GO TO 104
  102 FC=OLOG(RB/RA)
  104 DUM1=RB-RA
      DUM2=RB*RB-RA*RA
      DUM3=RB*RB*RB-RA*RA*RA
      DUM4=DUM2*(RB*RB+RA*RA)
      (F(KASE.EQ.3( GO TO 103
      F1=DUM1-R(*FO
      F2=DUM2/2.-2.*R(*DUM1+R(*R(*FO
      F3=DUM3/3.-1.5*R1*DUM2+3.*R(*R(*DUM1-R(*R(*R(*FO
  103 GO=DUM2/2.
      G1=DUM3/3.-R(*DUM2/2.
      G2=DUM4/4.-2.*R1*DUM3/3.+R(*R(*DUM2/2.
C
      (F(KASE.NE.1) GO TO 105
      A1(5(=A((5)+DUM*(C*F1+D*FO(
      A((6(=A1(6(+DUM*(C*C*F2/2.+C*D*F1+D*D*FO/2.(
      A((7(=A((7)+DUM*(C*F2+D*F1)
      A((9(=A((9)+DUM*(C*C*F3/2.+C*D*F2+D*D*F(/2.(
      A((10)=A((10)+DUM*(C*F3+D*F2)
      GO TO 106
  105 (F(KASE.NE.2) GO TO 107
  106 A((8(=A((8)+DUM*(C*C*C*F3/3.+C*C*D*F2+C*D*D*F1+D*D*D*FO/3.)
  107 A((13(=A((13)+DUM*(C*G2+D*G1(
      A((14(=A((14(+DUM*(C*C*G2/2.+C*D*G1+D*D*GO/2.)
C
  100 GO TO(201,202,203),(COUNT
  201 (COUNT=2
      (F(A.EQ.0.( GO TO 100
      RB=R(NPJ(
      RA=R(NPK)
      C=B/A
      D=H/A
      DUM=+1.
      GO TO 101
  202 ICOUNT=3
      (F(AK.EQ.0.( GO TO 100
      (F(BK.EQ.0.( GO TO 100
      RB=R(NPK(
      RA=RI
      C=BK/AK
      D=0.
      DUM=+1.
      GO TO (01
  203 RETURN
C
C     RECTANGULAR ELEMENT
C
  300 AJ=R(NPJ(-R(NP((
      BJ=Z(NPJ)-Z(NP((
      A=OSQRT(AJ*AJ+BJ*BJ)
      AL=R(NPL)-R(NP((
      BL=Z(NPL)-Z(NP((
      B=OSQRT(AL*AL+BL*BL)
      H=A*B
      S(=-BJ/A
      C(=AJ/A
      R(=R(NP((
      (F(S1.GT..01) GO TO 301
      S1=0.
      C1=1.
      GO TO 302
  301 (F(C1.GT..01) GO TO 302
      S1=1.
      C1=0.
  302 S2=S1*S1
      S3=S2*S1
      C2=C1*C1
      C3=C2*C1
C
      (F((STRES.EQ.0) GO TO 303
      A((4(=H
      A((11)=A*A*H/3.
      A((12(=B*B*H/3.
      A((13(=A*H/2.
      A((14(=B*H/2.
      A1(15)=H*H/4.
      RETURN
C
  303 A((1(=H
      A((2)=B*H/2.
      A((3)=A*H/2.
      A((4(=R(*A((1(+C(*A((3)+S1*A((2)
      A((8(=A*A*H/3.
```

```
      A(19)=B*B*H/3.
      A(20)=B*H*H/6.
      A(23)=A*H*H/6.
      A(16)=H*H/4.
      A(15)=R(A(16)+C1*A(23)+S1*A(20)
      A(14)=R(A(2)+C1*A(16)+S1*A(19)
      A(13)=R(A(3)+C1*A(18)+S1*A(16)
      A(12)=R1*A(19)+C1*A(20)+S1*B**3*H/4.
      A(11)=R1*A(18)+S1*A(23)+C1*A**3*H/4.
      IF(KASE.NE.6) GO TO 304
      A(2)=0.
      A(10)=A(4)
      A(17)=A(14)
      A(19)=0.
      A(21)=A(12)
      RETURN
C
  304 A2=A*A
      A3=A2*A
      B2=B*B
      B3=B2*B
      R12=R1*R1
      R13=R12*R1
      IF(S1.NE.0.) GO TO 305
      D1=DLOG(1.+A/R1)
      A(15)=R*D1
      A(6)=B2*D1/2.
      A(17)=H-B*R1*D1
      A(8)=B3*D1/3.
      A(9)=B2*(A-R1*D1)/2.
      A(10)=B*(A2/2.-A*R1+R12*D1)
      A(17)=B2*R12*D1/2.-B*H*R1/2.+H*H/4.
      A(21)=B3*(R12*D1+A2/2.-A*R1)/3.
      A(22)=B2*(H-B*R1*D1)/3.
      RETURN
C
  305 IF(C1.NE.0.) GO TO 306
      D1=DLOG(1.+B/R1)
      A(15)=A*D1
      A(16)=H-A*R1*D1
      A(17)=A2*D1/2.
      A(18)=A*R12*D1-H*R1+B*H/2.
      A(19)=A2*(B-R1*D1)/2.
      A(10)=A3*D1/3.
      A(17)=A2*H/3.-A3*R1*D1/3.
      A(21)=A3*(R12*D1+R2/2.-R*R1)/3.
      A(22)=H*H/4.-A*H*R1/2.+A2*R12*D1/2.
      RETURN
C
  306 D1=0.0
      IF(KASE.NE.5) D1=DLOG((1.-H*S1*C1/(1)R1+R*S1)*(R1+A*C1)))
      D2=DLOG(1.+A*C1/(R1+R*S1))
      D3=DLOG(1.+R*S1/(R1+A*C1))
      A(15)=(R1*D1+B*D2*S1+A*D3*C1)/(S1*C1)
      A(16)=(B2*D2*S2-R(2*D1-A*C)*D3*(2.*R(+A*C1)+H*S1*C1)/(2.*S2*C1)
      A(17)=(A2*D3*C2-R12*D1-B*S1*D2*(2.*R1+R*S1)+H*S1*C))/2.*S1*C2)
      A(18)=(B*H*S2*C1/6.-2.*H*R)*S1*C1/3.-A*H*S1*C2/3.+R(3*D1/3.
     1+B3*S3*D2/3.+C1*A*(R12+A*R)*C1+A2*C2/3.)*D3)/(S3*C1)
      A(19)=(A*H*S1*C2/3.+R*H*S2*C1/3.+H*R(*S1*C1/6.+R)3*D1/6.
     1-S2*)B2*R1/2.+B3*S1/3.)*D2-C2*)A2*R1/2.+A3*C1/3.)*D3)/(S2*C2)
      A(10)=(A*H*S1*C2/6.-2.*H*R)*S1*C1/3.-B*H*S2*C1/3.+R)3*D1/3.
     1+A3*C3*D3/3.+S1*R*(R(2+B*R)*S1+B2*S2/3.)*D2)/(S1*C3)
      A(17)=(A2*H*S1*C3/4.-R1*R13*D1/12.+S2*R2*)R12/2.+2.*B*R1*S1/3.
     1+B2*S2/4.)*D2-C3*A3*(R1/3.+A*C1/4.)*D3-S1*C1*H*)R12/12.+5.*R*R)
     2*S1/12.+R2*S2/4.)+S1*C2*H*(A*R)/12.+H*S1/8.))/(S2*C3)
      A(21)=-H*(A3*C1/S2+B3*S1/C2)/5.+H*H*)A/S1+B/C1)/10.-R12*H*)R/C1
     1+A/S1))/30.*S1*C1)-.3*R1*H*)A2/S2+B2/C2)+R13*H/)30.*S2*C2)
     2+R1*H*H/)20.*S1*C1)+R12*R)3*D1/)30.*S3*C3)+R3*D2*(R12/3.+B*R1*S1
     3/2.+B2*S2/5.)/C3+A3*D3*(R12/3.+A*R1*C1/2.+A2*C2/5.)/S3
      A(22)=H*H/)8.*S1)-5.*A*H*R1/)12.*S2)-A2*H*C1/)4.*S2)-H*R12/)12.*
     1S2*C1)+B*H*R1/(12.*S1*C1)+B2*H/(4.*C1)+A2*D3*)R12/2.+2.*A*R)*C1
     2/3.+A2*C2/4.)/S3-R)*R13*D1/(12.*S3*C2)-B3*D2*)R1/3.+R*S1/4.)/C2
C
      RETURN
      END
```

```
      SUBROUTINE DISTK(MAXNP,MXADJP,BK,SNPUU,SNPUW,SNPWW,SADUU,
     1SADUW,SADWU,SADWW,NPI,NPJ,NPK,NPL,NPADJ,NPOUT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BK(8,8),SNPUU(MAXNP),SNPUW(MAXNP),SADUU(MAXNP,MXADJP),
     1SADUW(MAXNP,MXADJP),SNPWW(MAXNP),SADWU(MAXNP,MXADJP),
     2SADWW(MAXNP,MXADJP),NPADJ(MAXNP,MXADJP)
C
C     DISTRIBUTE ELEMENT STIFFNESS TO NODE POINT STIFFNESS
C
C     BK     = ELEMENT STIFFNESS, 6*6 FOR TRIANGLE, 8*8 FOR RECTANGLE
C     SNPUU  = NODE PT. STIFF, U-DIRECTION, U-DISPL.
C     SNPUW  = NODE PT. STIFF, U-DIRECTION, W-DISPL.
C     SADUU  = ADJ. PT. STIFF, U-DIRECTION, U-DISPL.
C     SADUW  = ADJ. PT. STIFF, U-DIRECTION, W-DISPL.
C     SNPWW  = NODE PT. STIFF, W-DIRECTION, W-DISPL.
C     SADWW  = ADJ. PT. STIFF, W-DIRECTION, W-DISPL.
C     SADWU  = ADJ. PT. STIFF, W-DIRECTION, U-DISPL.
C
      ICOUNT=1
    1 GO TO (2,3,4,5,6),ICOUNT
C
    2 NI=NPI
      NJ=NPJ
      NK=NPK
      NL=NPL
      MX=1
      LX=3
      LY=5
      LZ=7
      GO TO 100
C
    3 NI=NPJ
      NJ=NPK
      NK=NPL
      NL=NPI
```

```
      MX=3
      LX=1
      LY=5
      LZ=7
      GO TO 100
C
    4 NI=NPK
      NJ=NPI
      NK=NPJ
      NL=NPL
      MX=5
      LX=1
      LY=3
      LZ=7
      GO TO 100
C
    5 IF(NPL.EQ.0) GO TO 6
      NI=NPL
      NJ=NPI
      NK=NPJ
      NL=NPK
      MX=7
      LX=1
      LY=3
      LZ=5
      GO TO 100
C
    6 RETURN
C
  100 SNPUU(NI)=SNPUU(NI)+BK(MX,MX)
      SNPUW(NI)=SNPUW(NI)+BK(MX,MX+1)
      SNPWW(NI)=SNPWW(NI)+BK(MX+1,MX+1)
      DO 101 I=1,MXADJP
      J=I
      IF(NPADJ(NI,I)-NPOUT).EQ.NJ) GO TO 102
  101 CONTINUE
  205 NPI=NPI+NPOUT
      NPJ=NPJ+NPOUT
      NPK=NPK+NPOUT
      IF(NPL.EQ.0) GO TO 204
      NPL=NPL+NPOUT
  204 WRITE(6,203) NPI,NPJ,NPK,NPL,NI,NJ,NK,NL,NPOUT,
     1(NPADJ(NI,II),I=1,MXADJP)
  203 FORMAT(1H1/32H ERROR IN STIFFNESS DISTRIBUTION//
     1   13H NPI         =,I5/13H NPJ         =,I5/13H NPK         =,I5/
     213H NPL         =,I5/10X,5I5/10X,8I5)
      CALL EXIT
C
  102 SADUU(NI,J)=SADUU(NI,J)+BK(MX,LX)
      SADUW(NI,J)=SADUW(NI,J)+BK(MX,LX+1)
      SADWU(NI,J)=SADWU(NI,J)+BK(MX+1,LX)
      SADWW(NI,J)=SADWW(NI,J)+BK(MX+1,LX+1)
C
      DO 103 I=1,MXADJP
      J=I
      IF((NPADJ(NI,I)-NPOUT).EQ.NK) GO TO 104
  103 CONTINUE
      GO TO 205
  104 SADUU(NI,J)=SADUU(NI,J)+BK(MX,LY)
      SADUW(NI,J)=SADUW(NI,J)+BK(MX,LY+1)
      SADWU(NI,J)=SADWU(NI,J)+BK(MX+1,LY)
      SADWW(NI,J)=SADWW(NI,J)+BK(MX+1,LY+1)
C
      IF(NL.EQ.0) GO TO 105
      DO 106 I=1,MXADJP
      J=I
      IF((NPADJ(NI,II-NPOUT).EQ.NL) GO TO 107
  106 CONTINUE
      GO TO 205
  107 SADUU(NI,J)=SADUU(NI,J)+BK(MX,LZ)
      SADUW(NI,J)=SADUW(NI,J)+BK(MX,LZ+1)
      SADWU(NI,J)=SADWU(NI,J)+BK(MX+1,LZ)
      SADWW(NI,J)=SADWW(NI,J)+BK(MX+1,LZ+1)
  105 ICOUNT=ICOUNT+1
      GO TO 1
C
      END
```

```
      SUBROUTINE ADJUST(MAXNP,BK,ITYPE,THETA,NADJEL,NPI,NPJ,NPK,NPL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BK(8,8),ITYPE(MAXNP),THETA(MAXNP),NADJEL(MAXNP),NODE(4)
C
C     DO ROLLER AND PINNED NODE MODIFICATIONS
C
      LIM=4
      IF(NPL.EQ.0) LIM=3
      NODE(1)=NPI
      NODE(2)=NPJ
      NODE(3)=NPK
      NODE(4)=NPL
C
      DO 100 I=1,LIM
      IN=NODE(I)
      IND=2*I-1
      SFAC=1.0/NADJEL(IN)
C
      IF(ITYPE(IN).NE.2) GO TO 10
C
      BK(IND,IND)=SFAC
      BK(IND+1,IND)=0.0
      BK(IND,IND+1)=0.0
      BK(IND+1,IND+1)=SFAC
      GO TO 20
C
   10 IF(ITYPE(IN).NE.1) GO TO 20
C
      S=DSIN(THETA(IN))
      C=DCOS(THETA(IN))
      BK(IND,IND)=BK(IND,IND)*C*C+2.0*BK(IND,IND+1)*C*S+BK(IND+1,IND+1)
     1*S*S
      BK(IND+1,IND)=0.0
```

```
      BK(INC,IND+1)=0.0
      BK(IND+1,IND+1)=SFAC
C
   20 DO 100 J=I,L(M
      IF(I.EQ.J) GO TO 100
      JN=NODE(J)
      JND=2*J-1
C
      IF(.NOT.(ITYPE(IN).EQ.2.OR.ITYPE(JN).EQ.2)) GO TO 50
C
      BK(INC,JND)=0.0
      BK(IND+1,JND)=0.0
      BK(INC,JND+1)=0.0
      BK(IND+1,JND+1)=0.0
      GO TO 100
C
   50 IF(ITYPE(IN).NE.1) GO TO 75
C
      S=DSIN(THETA(IN))
      C=DCOS(THETA(IN))
      BK(IND,JND)=BK(IND,JND)*C+BK(IND+1,JND)*S
      BK(INC,JND+1)=BK(IND,JND+1)*C+BK(IND+1,JND+1)*S
      BK(IND+1,JND)=0.0
      BK(IND+1,JND+1)=0.0
C
   75 IF(ITYPE(JN).NE.1) GO TO 100
C
      S=DSIN(THETA(JN))
      C=DCOS(THETA(JN))
      BK(INC,JND)=BK(IND,JND)*C+BK(IND,JND+1)*S
      BK(IND+1,JND)=BK(IND+1,JND)*C+BK(IND+1,JND+1)*S
      BK(IND,JND+1)=0.0
      BK(IND+1,JND+1)=0.0
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE MASS(MAXNP,RHO,R,Z,AI,XMASS,SI,CI,NPI,NPJ,NPK,NPL,
     1ISTRES)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(MAXNP),Z(MAXNP),XMASS(MAXNP),AI(23)
C
C     COMPUTE AND DISTRIBUTE MASS TO NODE POINTS
C
      IF(NPL.NE.0) GO TO 2
C
      AJ=R(NPJ)-R(NPI)
      AK=R(NPK)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      BK=Z(NPK)-Z(NPI)
      H=AJ*BK-AK*BJ
      B=BJ-BK
      A=AJ-AK
C
      DUMI=RHO/H
      AMI=DUMI*(H*AI(4)+B*AI(13)-A*AI(14))
      AMJ=DUMI*(BK*AI(13)-AK*AI(14))
      AMK=DUMI*(AJ*AI(14)-BJ*AI(13))
      GO TO 3
C
    2 AJ=R(NPJ)-R(NPI)
      BJ=Z(NPJ)-Z(NPI)
      A=DSQRT(AJ*AJ+BJ*BJ)
      AL=R(NPL)-R(NPI)
      BL=Z(NPL)-Z(NPI)
      B=DSQRT(AL*AL+BL*BL)
C
      IF(ISTRES.EQ.0) GO TO 4
C
      AMI=RHO*AI(4)/4.
      AMJ=AMI
      AMK=AMI
      AML=AMI
      GO TO 3
C
    4 DUMI=RHO/(1.+SI+CI)
      AMI=DUMI*((SI+CI)*AI(4)-SI*AI(14)/B-CI*AI(13)/A)
      AMJ=DUMI*(AI(4)-(1.+CI)*AI(14)/B+CI*AI(13)/A)
      AMK=DUMI*(-AI(4)+(1.+CI)*AI(14)/B+(1.+SI)*AI(13)/A)
      AML=DUMI*(AI(4)+SI*AI(14)/B-(1.+SI)*AI(13)/A)
C
C
    3 XMASS(NPI)=XMASS(NPI)+AMI
      XMASS(NPJ)=XMASS(NPJ)+AMJ
      XMASS(NPK)=XMASS(NPK)+AMK
      IF(NPL.EQ.0) RETURN
      XMASS(NPL)=XMASS(NPL)+AML
      RETURN
      END
```

```
      SUBROUTINE PRNK(MAXNP,MXADJP,NADJNP,NPADJ,NADJEL,IPRINT,SNPUU,
     1SNPUW,SNPWW,SADUU,SADUW,SADWU,SADWW,THETA,ITYPE,XMASS,NPOUT,
     2NUMNP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AI(40)
      DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP),NADJEL(MAXNP),
     1XMASS(MAXNP),          SNPUU(MAXNP),SNPUW(MAXNP),SNPWW(MAXNP),
     2SADUU(MAXNP,MXADJP),SADUW(MAXNP,MXADJP),SADWU(MAXNP,MXADJP),
     3SADWW(MAXNP,MXADJP),THETA(MAXNP),ITYPE(MAXNP)
C
C     PRINT STIFFNESSES AND MASSES
C
      DATA AI/6HBEFORE,6HAFTER ,6HSTIFFN,6HESS   ,6HTABLES,
     16HBOUNDA,6HRY ALT,6HERATIO,6HN     ,6H  NODE,6HSNPUU ,6HSNPUW ,
     26HSNPWW ,6HADJACE,6HNT    ,6HSADUU ,6HSADUW ,6HSADWU ,6HSADWW ,
     36HSTRESS,6HSTRNPU,6HSTRNPW,6HSTTNPU,6HSTTNPW,6HSTZNPU,6HSTZNPW,
     46HSTSNPU,6HSTSNPW,6HSTRADU,6HSTRADW,6HSTTADU,6HSTTADW,6HSTZADU,
     56HSTZADW,6HSTSADU,6HSTSADW,6HMASS  ,6HVECTOR,6HP-S2/I,6H      /
```

```
C
      GO TO 25
   14 WRITE(6,112) A1(3),A1(4),A1(5),A1(4C),A1(ISTIFF),A1(40),
     1(A1(K),K=6,9)
  112 FORMAT(1HI,12A6)
      WRITE(6,113) A1(10),(A1(K),K=11,13)
  113 FORMAT(//A6,81X,A6///)
      DO 15 I=1,NUMNP
      K=I+NPOUT
   15 WRITE(6,114) K,SNPUU(I),SNPUW(I),SNPWW(I)
  114 FORMAT(I5,3X,1P8E14.4)
C
      ICOUNT=1
      JCOUNT=16
   16 WRITE(6,112) A1(14),A1(15),A1(3),A1(4),A1(JCOUNT)
      WRITE(6,113) A1(10)
      GO TO (17,19,21,23),ICOUNT
   17 DO 18 I=1,NUMNP
      NUM=NADJNP(I)
      K=I+NPOUT
   18 WRITE(6,114) K,(SADUU(I,J),J=1,NUM)
      ICOUNT=2
      JCOUNT=17
      GO TO 16
   19 DO 20 I=1,NUMNP
      NUM=NADJNP(I)
      K=I+NPOUT
   20 WRITE(6,114) K,(SADUW(I,J),J=1,NUM)
      ICOUNT=3
      JCOUNT=18
      GO TO 16
   21 DO 22 I=1,NUMNP
      NUM=NADJNP(I)
      K=I+NPOUT
   22 WRITE(6,114) K,(SADWW(I,J),J=1,NUM)
      ICOUNT=4
      JCOUNT=19
      GO TO 16
   23 DO 24 I=1,NUMNP
      NUM=NADJNP(I)
      K=I+NPOUT
   24 WRITE(6,114) K,(SADWW(I,J),J=1,NUM)
      GO TO 56
C
   25 ISTIFF=2
C
   30 IF(IPRINT.EQ.3) GO TO 14
      IF(IPRINT.EQ.8) GO TO 14
C
   56 IF(IPRINT.EQ.6) GO TO 58
      IF(IPRINT.EQ.8) GO TO 58
      GO TO 60
C
C
   58 WRITE(6,112) A1(37),A1(38),A1(4C),A1(39)
      WRITE(6,113) A1(10)
      DO 59 I=1,NUMNP,8
      L=I+NPOUT
      NUM=I+7
      IF(NUM.GT.NUMNP) NUM=NUMNP
   59 WRITE(6,114) L,(XMASS(J),J=I,NUM)
C
   60 RETURN
      END
```

```
      SUBROUTINE LINK1C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPR,MAXELR,NUMNP,NUMEL,
     1        ISTRES,LINES,NUMPEL,LOADNP(1),PER100,DT,NPLOAD(1,50),      3/4/72
     2        RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXBD,           3/4/72
     3        NBLOCK,NREADS,MAXMBK,TREAL,POAMP,BETA,KRUN,
     4        MXZONE,NZONES,IPRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
      DIMENSION NPTN(1000),IELAST(20),WGT(20),E1(20),E2(20),E3(20),E4(20
     1),E5(20)
      DIMENSION C(4,4)
      DIMENSION NADJNP(100),NPADJ(100,8),NADJEL(100)
      DIMENSION R(100),Z(100),ITYPE(100),THETA(100)
      DIMENSION STNPU(4,100),STNPW(4,100),STADU(4,100,8),STADW(4,100,8)
C
C     THIS LINK COMPUTES ELEMENT STRESSES, MODIFIES THEM FOR ROLLER
C     SUPPORTS, AND DISTRIBUTES THEM TO NODE POINT STRESSES
C
C     FLOW OF THIS LINK IS SAME AS LINK1A
C
      REWIND 4
      REWIND 8
      REWIND 12
C
      KEND=0
      NPOUT=0
      NUMCP=0
      NUMNPB=0
      NPR=MXNPP
      KK=1
    1 ISWTCH=1
      GO TO 900
C
    2 IF(NUMNP.LT.MXNPB) NUMNPB=NUMNP
      IF(NUMNP.GE.MXNPB) NUMNPB=MXNPB
      DO 3 I=1,NUMNPB
      READ(8) NPN,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP)
      READ(4) NPN,R(I),Z(I),ITYPE(I),THETA(I)
    3 CONTINUE
C
      ICOUNT=0
    4 READ(12) KEY,NUMSE,IZONE,NTI,NTJ,NTK,NTL,((C(I,J),I=1,4),J=1,4),
     1KASE,S1,C1
      ICOUNT=ICOUNT+1
      LNP=MAX0(NTI,NTJ,NTK,NTL)
      IF((LNP-NPOUT).GT.MXNPB) GO TO 100
```

```
    6 NPL=NT)-NPOUT
      NPJ=NTJ-NPOUT
      NPK=NTK-NPOUT
      IF(NTL.EQ.0) NPL=0
      IF(NTL.NE.0) NPL=NTL-NPOUT
      CALL STRESS(MXNPR,MXADJP,C,NP),NPJ,NPK,NPL,R,Z,KASE,NUMF,NPADJ,
     1 STNPU,STNPW,STADU,STADW,ISTRES,SI,CI,NPOUT,ITYPE,THETA)
C
      IF(ICOUNT.LT.NUMEL) GO TO 4
      KEND=1
      KEY=NUMNP+1
C
  100 NUMCP=KEY-1
      NUMNPR=NUMCP-NPOUT
  113 CALL        MODS)MXNPB,MXADJP,NADJNP,NPADJ,NADJEL,STNPU,STNPW,
     1STADU,STADW,IPRINT,THETA,ITYPE,NPOUT,NUMNPR)
      DO 101 I=1,NUMNPR
  101 WRITE(3) I,NADJNP(I),NADJEL(I),(NPADJ(I),J),J=1,MXADJP),(STNPU(K,I)
     1,STNPW(K,I),K=1,4),((STADU(K,I),J),STADW(K,I),J),K=1,4),J=1,MXADJP)
C
      IF(KEND.EQ. 1) GO TO 300
C
      NPR=MXNPB-NUMNPR
      GO TO 902
C
  107 KX=NPR+1
      ISWTCH=2
      GO TO 900
C
  108 IF((NUMNP-NUMCP).LT.MXNPB) KNP=NUMNP-NUMCP-NPR
      IF((NUMNP-NUMCP).GE.MXNPB) KNP=MXNPB-NPR
      DO 109 I=1,KNP
      L=NPR+I
      READ(8) NPN,NADJNP(L),NADJEL(L),(NPADJ(L,J),J=1,MXADJP)
      READ(4) NPN,R(L),Z(L),)TYPE(L),THETA(L)
  109 CONTINUE
      NPOUT=NUMCP
      GO TO 6
C
  902 DO 903 K=1,NPR
      L=NUMNPR+K
      NADJNP(K)=NADJNP(L)
      NADJEL(K)=NADJEL(L)
      ITYPE(K)=ITYPE)L)
      THETA(K)=THETA(L)
      R(K)=R(L)
      Z(K)=Z(L)
      DO 904 J=1,4
      STNPU(J,K)=STNPU(J,L)
  904 STNPW(J,K)=STNPW(J,L)
      DO 903 J=1,MXADJP
      NPADJ(K,J)=NPADJ(L,J)
      DO 903 I=1,4
      STADU(I,K,J)=STADU(I,L,J)
  903 STADW(I,K,J)=STADW(I,L,J)
      GO TO 107
C
  900 DO 901 L=KX,MXNPB
      NADJNP(L)=0
      NADJEL(L)=0
      ITYPE(L)=0
      THETA(L)=0.0
      R(L)=0.0
      Z(L)=0.0
      DO 905 J=1,4
      STNPU(J,L)=0.0
  905 STNPW(J,L)=0.0
      DO 901 J=1,MXADJP
      NPADJ(L,J)=0
      DO 901 I=1,4
      STADU(I,L,J)=0.0
  901 STADW(I,L,J)=0.0
      GO TO (2,100),ISWTCH
C
  300 CONTINUE
      REWIND 3
      REWIND 4
      REWIND 8
      REWIND 12
      RETURN
      END
```

```
      SUBROUTINE STRESS(MXNP,MXADJP,C,NP),NPJ,NPK,NPL,RA,ZA,KASE,NUMF,
     1NPADJ,STNPU,STNPW,STADU,STADW,)STRES,SI,CI,NPOUT,)TYPE,THETA)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(4,4),RA(MAXNP),ZA(MAXNP),NPADJ(MAXNP,MXADJP),S(4,8),
     1STNPU(4,MAXNP),STNPW(4,MAXNP),STADU(4,MAXNP,MXADJP),
     2STADW(4,MAXNP,MXADJP),ITYPE(MAXNP),THETA(MAXNP)
C
C     COMPUTE ELEMENT STRESSES, MODIFY THEM FOR ROLLER SUPPORTS, AND
C     DISTRIBUTE THEM TO NODE POINT STRESSES
C
C       C     = STRESS-STRAIN MATRIX
C       S     = STRESS-DISPLACEMENT MATRIX FOR ELEMENT
C       ST(1) = RADIAL STRESS,  SIGMA R
C       ST(2) = MERIDIONAL STRESS, SIGMA THETA
C       ST(3) = VERTICAL STRESS, SIGMA Z
C       ST(4) = SHEAR STRESS, TAU
C
C       ST(I) = STNPU(I,NU+STNPW(I)*W+SUM(STADU(I))*U+STADW(I)*W)
C
      DO 1 I=1,4
      DO 1 J=1,8
    1 S(I,J)=0.0
C
      RR=RA(NP(1)
      R=0.
      RP=0.
      Z=0.
      ZP=0.
      ICOUNT=1
```

```
      IF(NPL.NE.0) GO TO 200
C
C     TRIANGULAR ELEMENT
C
      AJ=RA(NPJ)-RA(NPI)
      AK=RA(NPK)-RA(NPI)
      BJ=ZA(NPJ)-ZA(NPI)
      BK=ZA(NPK)-ZA(NPI)
      H=AJ*BK-AK*BJ
      B=BJ-BK
      A=AJ-AK
C
   25 IF(KASE.NE.1) GO TO 26
      IF(ISTRES.NE.0) GO TO 27
      AOR=1./RR
      ROR=R/RR
      ZOR=Z/RR
      GO TO 28
   27 IF(ICOUNT.NE.1) GO TO 100
      AOR=0.0
      ROR=0.0
      ZOR=0.0
   28 DUM1=B/H
      DUM2=AOR*(B*ROR-A*ZOR)/H
      DO 35 I=1,3
   35 S(I,1)=C(1,I)*DUM1+C(I,2)*DUM2
      S(4,1)=-C(4,4)*A/H
      GO TO 29
   26 ROR=1.0
      IF(KASE.NE.2) GO TO 30
      IF(ICOUNT.NE.1) GO TO 31
      ZOR=0.0
      GO TO 29
   31 ZOR=Z/RR
      GO TO 29
   30 ZOR=0.0
      IF(ICOUNT.NE.1) GO TO 100
   29 DO 36 I=1,3
   36 S(I,2)=-C(I,3)*A/H
      S(4,2)=C(4,4)*B/H
      DUM1=BK/H
      DUM2=(BK*ROR-AK*ZOR)/H
      DO 37 I=1,3
   37 S(I,3)=C(1,I)*DUM1+C(I,2)*DUM2
      DUM1=AK/H
      S(4,3)=-C(4,4)*DUM1
      DO 38 I=1,3
   38 S(I,4)=-C(I,3)*DUM1
      S(4,4)=C(4,4)*BK/H
      DUM1=AJ/H
      DO 39 I=1,3
   39 S(I,6)=C(I,3)*DUM1
      S(4,6)=-C(4,4)*BJ/H
      IF(KASE.EQ.3) GO TO 300
      DUM1=-BJ/H
      DUM2=(AJ*ZOR-BJ*ROR)/H
      DO 40 I=1,3
   40 S(I,5)=C(1,I)*DUM1+C(I,2)*DUM2
      S(4,5)=C(4,4)*AJ/H
      GO TO 300
C
C     RECTANGULAR ELEMENT
C
  200 AJ=RA(NPJ)-RA(NPI)
      BJ=ZA(NPJ)-ZA(NPI)
      A=DSQRT(AJ*AJ+BJ*BJ)
      AL=RA(NPL)-RA(NPI)
      BL=ZA(NPL)-ZA(NPI)
      B=DSQRT(AL*AL+BL*BL)
      H=A*B
C
   50 IF(KASE.NE.4) GO TO 51
      IF(ISTRES.NE.0) GO TO 52
      AOR=1./RR
      ROR=RP/RR
      ZOR=ZP/RR
      GO TO 53
   52 AOR=0.0
      ROR=0.0
      ZOR=0.0
   53 DUM1=(RP*S1+ZP*C1-A*SI-B*C1)/H
      DUM2=AOR*(ROR*ZP-B*ROR-A*ZOR)/H
      DO 65 I=1,3
   65 S(I,1)=C(1,I)*DUM1+C(I,2)*DUM2
      S(4,1)=C(4,4)*(B*S1-A*C1+RP*C1-ZP*S1)/H
      GO TO 54
   51 IF(KASE.EQ.5) GO TO 73
      ROR=1.
      ZOR=0.
      GO TO 54
   73 IF(ICOUNT.NE.1) GO TO 74
      ROR=C1
      ZOR=S1
      GO TO 54
   74 ROR=RP/RR
      ZOR=ZP/RR
   54 DUM1=(R*S1-A*C1+RP*C1-ZP*S1)/H
      DO 66 I=1,3
   66 S(I,2)=C(I,3)*DUM1
      S(4,2)=C(4,4)*(RP*S1+ZP*C1-R*C1-A*S1)/H
      DUM1=(R*C1-RP*S1-ZP*C1)/H
      DUM2=(R*ROR-ROR*ZP)/H
      DO 67 I=1,3
   67 S(I,3)=C(1,I)*DUM1+C(I,2)*DUM2
      DUM1=(-R*S1-RP*C1+ZP*S1)/H
      S(4,3)=C(4,4)*DUM1
      DO 68 I=1,3
   68 S(I,4)=C(I,3)*DUM1
      S(4,4)=C(4,4)*(R*C)-RP*SI-ZP*C1)/H
      DUM1=(RP*SI+ZP*C1)/H
      DUM2=ROR*ZP/H
      DO 69 I=1,3
```

```
   69 S(I,5)=C(1,I)*DUM1+C(I,2)*DUM2
      DUM1=(RP*C1-ZP*S1)/H
      S(4,5)=C(4,4)*DUM1
      DO 70 I=1,3
   70 S(I,6)=C(I,3)*DUM1
      S(4,6)=C(4,4)*(ZP*C1+RP*S1)/H
      DUM1=(A*C1-RP*C1+ZP*S1)/H
      DO 71 I=1,3
   71 S(I,8)=C(I,3)*DUM1
      S(4,8)=C(4,4)*(A*S1-RP*S(-ZP*C1)/H
      IF(KASE.EQ.6) GO TO 300
      S(4,7)=C(4,4)*DUM1
      DUM1=(A*S1-RP*S1-ZP*C1)/H
      DUM2=(A*ZOR-ROR*ZP)/H
      DO 72 I=1,3
   72 S(I,7)=C(1,I)*DUM1+C(I,2)*DUM2
C
C     MODIFY FOR ROLLER SUPPORT
C
  300 IF(ITYPE(NPI).NE.1) GO TO 300
      NP=NPI
      KCOUNT=1
      MX=1
      GO TO 350
  301 IF(ITYPE(NPJ).NE.1) GO TO 302
      NP=NPJ
      KCOUNT=2
      MX=3
      GO TO 350
  302 IF(ITYPE(NPK).NE.1) GO TO 303
      NP=NPK
      KCOUNT=3
      MX=5
      GO TO 350
  303 IF(NPL.EQ.0) GO TO 100
      IF(ITYPE(NPL).NE.1) GO TO 100
      NP=NPL
      KCOUNT=4
      MX=7
C
  350 NX=MX+1
      CSN=DCOS(THETA(NP))
      SSN=DSIN(THETA(NP))
      DO 351 J=1,4
      S(J,MX)=S(J,MX)*CSN+S(J,NX)*SSN
  351 S(J,NX)=0.0
C
      GO TO (301,302,303,100),KCOUNT
C
C     DISTRIBUTE ELEMENT STRESS TO NODE POINT STRESS
C
  100 IF(ICOUNT.NE.1) GO TO 101
      MI=1
      MJ=3
      MK=5
      ML=7
      NI=NPI
      NJ=NPJ
      NK=NPK
      NL=NPL
      GO TO 103
C
  101 IF(ICOUNT.NE.2) GO TO 102
      MI=3
      MJ=1
      MK=5
      ML=7
      NI=NPJ
      NJ=NPI
      NK=NPK
      NL=NPL
      GO TO 103
C
  102 IF(ICOUNT.NE.3) GO TO 107
      MI=5
      MJ=1
      MK=3
      ML=7
      NI=NPK
      NJ=NPI
      NK=NPJ
      NL=NPL
      GO TO 103
C
  107 MI=7
      MJ=1
      MK=3
      ML=5
      NI=NPL
      NJ=NPI
      NK=NPJ
      NL=NPK
C
  103 DO 114 I=1,4
      STNPU(I,NI)=STNPU(I,NI)+S(I,MI)
  114 STNPW(I,NI)=STNPW(I,NI)+S(I,MI+1)
      N=1
  155 GO TO (150,151,152,153),N
  150 NN=NJ
      MN=MJ
      GO TO 154
  151 NN=NK
      MN=MK
      GO TO 154
  152 IF(NPL.EQ.0) GO TO 153
      NN=NL
      MN=ML
C
  154 DO 104 K=1,MXADJP
      J=K
      IF(INPADJ(N,K)-NPOUT).EQ.NN) GO TO 105
  104 CONTINUE
```

```
  109 WRITE(6,106) NUMF,NI,N,NPOUT
  106 FORMAT(1H1/30H ERROR IN STRSS, STATEMENT 104/
     113H ELEMENT NO.=,I5/13H NODE POINT =,I5/13H N          =,(5/
     213H NPOUT      =,I5)
      CALL EXIT
  105 DO 115 I=1,4
      STADU(I,NI,J)=STADU(I,NI,J)+S(I,MN)
  115 STADW(I,NI,J)=STADW(I,NI,J)+S(I,MN+1)
      N=N+1
      GO TO 155
C ..
C
  153 GO TO (110,111,112,113),ICOUNT
  110 ICOUNT=2
      RR=RA(NPJ)
      R=AJ
      Z=BJ
      RP=A
      ZP=0.0
      GO TO 120
C
  111 ICOUNT=3
      RR=RA(NPK)
      R=AK
      Z=BK
      RP=A
      ZP=B
  120 IF(NPL.EQ.0) GO TO 25
      GO TO 50
C
  112 IF(NPL.EQ.0) RETURN
      ICOUNT=4
      RR=RA(NPL)
      RP=0.0
      ZP=B
      GO TO 50
C
C
  113 RETURN
      END
```

```
      SUBROUTINE MODS(MAXNP,MXADJP,NADJNP,NPADJ,NADJEL,STNPU,STNPW,
     1STADU,STADW,IPRINT,THETA,ITYPE,NPOUT,NUMNP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A1(40)
      DIMENSION NADJNP(MAXNP),NPADJ(MAXNP,MXADJP),NADJEL(MAXNP),
     1STNPU(4,MAXNP),STNPW(4,MAXNP),STADU(4,MAXNP,MXADJP),
     2STADW(4,MAXNP,MXADJP),THETA(MAXNP),ITYPE(MAXNP)
      DATA A1/6HBEFORE,6HAFTER ,6HSTIFFN,6HESS   ,6HTABLES,
     16HBOUNDA,6HRY ALT,6HERATO,6HN     ,6H  NODE,6HSNPUU ,6HSNPUW ,
     26HSNPWW ,6HADJACE,6HNT    ,6HSADUU ,6HSADUW ,6HSADWU ,6HSADWW ,
     36HSTRESS,6HSTRNPU,6HSTRNPW,6HSTTNPU,6HSTTNPW,6HSTZNPU,6HSTZNPW,
     46HSTSNPU,6HSTSNPW,6HSTRADU,6HSTRADW,6HSTTADU,6HSTTADW,6HSTZADU,
     56HSTZADW,6HSTSADU,6HSTSADW,6HMASS  ,6HVECTOR,6HP-S2/I,6H      /
C
```

```
C     AVERAGE AND PRINT STRESS
C
  112 FORMAT(1H1,12A6)
  113 FORMAT(//A6,8(8X,A6)//)
  114 FORMAT(I5,3X,1P8E14.4)
C
      CALL      AVG(MAXNP,MXADJP,NADJNP,NADJEL,STNPU,STNPW,STADU,
     1STADW,NUMNP)
C
      GO TO 51
   32 WRITE(6,112) A1(20),A1(40),A1(5),A1(40),A1(ISTIFF),A1(40),
     1(A1(K) ,K=6,9)
      WRITE(6,113) A1(10),(A1(K),K=21,28)
      DO 33 I=1,NUMNP
      L=1+NPOUT
   33 WRITE(6,114) L,(STNPU(K,I),STNPW(K,I),K=1,4)
C
      ICOUNT=1
      JCOUNT=29
   34 WRITE(6,112) A1(14),A1(5),A1(20),A1(40),A1(JCOUNT)
      WRITE(6,113) A1(10)
   35 DO 36 I=1,NUMNP
      NUM=NADJNP(I)
      L=I+NPOUT
   36 WRITE(6,114) L,(STADU(ICOUNT,I,J),J=1,NUM)
      JCOUNT=JCOUNT+1
      WRITE(6,112) A1(14),A1(15),A1(20),A1(40),A1(JCOUNT)
      WRITE(6,113) A1(10)
      DO 37 I=1,NUMNP
      NUM=NADJNP(I)
      L=I+NPOUT
   37 WRITE(6,114) L,(STADW(ICOUNT,I,J),J=1,NUM)
      ICOUNT=ICOUNT+1
      JCOUNT=JCOUNT+1
      IF(ICOUNT.LE.4) GO TO 34
      GO TO 60
C
   51 ISTIFF=2
      IF(IPRINT.EQ.5) GO TO 32
      IF(IPRINT.EQ.8) GO TO 32
   60 RETURN
      END
```

```
      SUBROUTINE AVG(MAXNP,MXADJP,NADJNP,NADJEL,STNPU,STNPW,STADU,
     1STADW,NUMNP)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NADJNP(MAXNP),NADJEL(MAXNP),STNPU(4,MAXNP),
     1STNPW(4,MAXNP),STADU(4,MAXNP,MXADJP),STADW(4,MAXNP,MXADJP)
C
C     AVERAGE STRESS
C
      DO 8 I=1,NUMNP
      DUM=NADJEL(I)
      DO 1 IST=1,4
      STNPU(IST,I)=STNPU(IST,I)/DUM
```

```
     1 STNPW(IST,II=STNPW(IST,I)/DUM
       NUM=NADJNP(I)
       DO 8 J=1,NUM
       DO 8 IST=1,4
       STADU(IST,I,J)=STADU(IST,I,J)/DUM
     8 STADW(IST,I,J)=STADW(IST,I,J)/DUM
C
       RETURN
       END
```

```
       SUBROUTINE LINK10
       IMPLICIT REAL*8(A-H,O-Z)
       COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
      1        ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(1,50),
      2        RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXBO,
      3        MBLOCK,NREADS,MAXMBK,TREAL,PDAMP,BETA,KRUN,
      4        MXZONE,NZONES,IPRINT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
       DIMENSION NPTN(1000),IELAST(20),WGT(20),E1(20),E2(20),E3(20),E4(20
      1),E5(20)
       DIMENSION NPLOW(100),NPHIGH(100)
C
C      BLOCK THE STIFFNESS AND STRESS TABLES
C
C      SET MAXIMUM NUMBER OF CLUSTERS AND CALCULATE BLOCK SIZE
C
       MXCLS=100
       IF(NUMNP.LE.MAXMBK) MBLOCK=NUMNP
       IF(NUMNP.GT.MAXMBK) MBLOCK=MAXBO+1
C
       IF(KRUN.EQ.1) GO TO 1
C
       CALL SIZE(MXCLS,NUMCLS,NPLOW,NPHIGH,NUMNP,MBLOCK)
C
C      WRITE TITLE RECORD ON STIFFNESS TAPE
C
       REWIND 10
       WRITE(10)TITLE
C
     1 CALL CLSTER(MXCLS,NUMCLS,NPLOW,NPHIGH,NUMNP,MXADJP,NPTN,
      1MAXNP,PERIOD,KRUN)
       WRITE(6,1000)MBLOCK
  1000 FORMAT(21H1MATRIX BLOCK SIZE IS,I5,6H NODES)
C
       RETURN
       END
```

```
       SUBROUTINE SIZE(MXCLS,NUMCLS,NPLOW,NPHIGH,NUMNP,MBLOCK)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION NPLOW(MXCLS),NPHIGH(MXCLS)
C
C      COMPUTE THE SIZES OF THE BLOCKS
C
       NUMCLS=1
       NODES=0
     1 NPR=NUMNP-NODES
       NPLOW(NUMCLS)=NODES+1
       IF(NPR.GT.MBLOCK) GO TO 2
       NPHIGH(NUMCLS)=NUMNP
       RETURN
     2 NPHIGH(NUMCLS)=NODES+MBLOCK
       NODES=NODES+MBLOCK
       NUMCLS=NUMCLS+1
       IF(NUMCLS.LE.MXCLS) GO TO 1
       WRITE(6,1000)MBLOCK
  1000 FORMAT(64H1ERROR IN LINK10 IN SUBROUTINE SIZE, TOO MANY CLUSTERS.
      1 MBLOCK=,I5,6H NODES)
       STOP
C
       END
```

```
       SUBROUTINE CLSTER(MXCLS,NUMCLS,NPLOW,NPHIGH,NUMNP,
      1MXADJP,NPTN,MAXNP,PERIOD,KRUN)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION NPLOW(MXCLS),NPHIGH(MXCLS),NPTN(MAXNP)
       DIMENSION NADJNP(100),ITYPE(100),THETA(100),XMASS(100),SNPUU(100),
      1SNPUW(100),SNPWW(100),NPADJ(100,8),SADUU(100,8),SADUW(100,8),
      2SADWW(100,8)
      3,SADWU(100,8)
       DIMENSION NADJEL(100),STNPU(4,100),STNPW(4,100),STADU(4,100,8),
      1STADW(4,100,8)
       EQUIVALENCE (STNPU(1,1),ITYPE(1)),(STNPU(1,26),THETA(1)),
      1(STNPU(1,51),XMASS(1)),(STNPU(1,76),SNPUU(1)),(STNPW(1,1),
      2SNPUW(1)),(STNPW(1,26),SNPWW(1)),(STADU(1,1,1),SADUU(1,1)),
      3(STADU(1,1,3),SADUW(1,1)),(STADU(1,1,5),SADWW(1,1)),
      4(STADU(1,1,7),SADWU(1,1))
C
C      PERFORM THE BLOCKING OPERATION
C
C      INITIALIZE AND SKIP TO SECOND PART FOR PRIMARY RERUN
C
       IF(KRUN.EQ.1) GO TO 300
C
C      READ STIFFNESS TABLES FROM TAPE 3 AND WRITE BLOCKS ONTO TAPE 10
C
       REWIND 3
       ICOUNT=1
  101  NLOW=NPLOW(ICOUNT)
       NHGH=NPHIGH(ICOUNT)
       DO 100 L=NLOW,NHGH
       I=L-NLOW+1
  100  READ( 3) NPN,NADJNP(I),ITYPE(I),THETA(I),XMASS(I),SNPUU(I),
      1SNPUW(I),SNPWW(I),(NPADJ(I,J),SADUU(I,J),SADUW(I,J),SADWU(I,J),
      2SADWW(I,J),J=1,MXADJP)
C
       NUMNPB=NHGH-NLOW+1
       WRITE(10)NLOW,NHGH,NUMNPB,
      1(NADJNP(I),ITYPE(I),THETA(I),XMASS(I),SNPUU(I),SNPUW(I),SNPWW(I),
      2(NPADJ(I,J),SADUU(I,J),SADUW(I,J),SADWU(I,J),SADWW(I,J),J=1,MXADJP
      3),I=1,NUMNPB)
```

```
C
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LE.NUMCLS) GO TO 101
C
C     WRITE NPTN ONTO TAPE 10
C
      WRITE(10) (NPTN(I),I=1,NUMNP)
C
C     READ STRESS TABLES FROM TAPE 3 AND WRITE BLOCKS ONTO TAPE 10
C
      ICOUNT=1
  201 NLOW=NPLOW(ICOUNT)
      NHGH=NPHIGH(ICOUNT)
      DO 200 L=NLOW,NHGH
      I=L-NLOW+1
  200 READ(3) NPN,NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP),
     1(STNPUK,I),STNPWK,I),K=1,4),((STADUK,I,J),STADWK,I,J),
     2K=1,4),J=1,MXADJP)
C
      NUMNPB=NHGH-NLOW+1
      WRITE(10)NLOW,NHGH,NUMNPB,
     1(NADJNP(I),NADJEL(I),(NPADJ(I,J),J=1,MXADJP),(STNPU(K,I),
     2STNPW(K,I),K=1,4),((STADU(K,I,J),STADW(K,I,J),K=1,4),J=1,MXADJP),
     3I=1,NUMNPB)
C
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LE.NUMCLS) GO TO 201
C
  203 CONTINUE
      END FILE 10
      REWIND 10
      REWIND 3
      RETURN
C
C     CALCULATE MINIMUM DIAGONAL PERIOD OF OSCILLATION FOR RERUN
C
  300 DUMMY=0.0
      REWIND 10
      READ(10)
  305 READ(10) N1,N2,      N5,(NADJNP(I),ITYPE(I),THETA(I),XMASS(I),
     1SNPUU(I),SNPUW(I),SNPWW(I),(NPADJ (I,J),SADUU(I,J),SADUW(I,J),
     XSADWU(I,J),
     2SADWW(I,J),J=1,MXADJP),I=1,N5)
      DO 301 I=1,N5
      DUMMY1=0.0
      IF(ITYPE(I).NE.2) DUMMY1=SNPUU(I)/XMASS(I)
      DUMMY2=0.0
      IF(ITYPE(I).NE.2.AND.ITYPE(I).NE.1) DUMMY2=SNPWW(I)/XMASS(I)
      DUMMY=DMAX1(DUMMY,DUMMY1,DUMMY2)
  301 CONTINUE
      IF(N2.LT.NUMNP) GO TO 305
      REWIND 10
      PI=3.1415927
      PERIOD=2.*PI/DSQRT(DUMMY)
      WRITE(6,303) PERIOD
  303 FORMAT(1HI,7HPERIOD=,1PE15.5,2X,9HSEC/CYCLE)
C
      RETURN
      END
```

```
      SUBROUTINE LINK1F
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 DUM
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1       ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(1,50),
     2       RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXRD,
     3       MBLOCK,NREADS,MAXMBK,TREAL,POAMP,BETA,KRUN,
     4       MXZONE,NZONES,IPR(NT,NPTN,IELAST,WGT,E1,E2,E3,E4,E5
      DIMENSION NPTN(1000)
      DIMENSION IELAST(20),JPLAST(20),WGT(20),E1(20),E2(20),E3(20),
     1E4(20),E5(20),CC(4,4),NPTP(1000)
      DIMENSION NOOFEL(26),IPLAST(26),NP(26,4),ITYPE(26,4),THETA(26,4),
     1C(26,4,4),B(26,4,8),P(26,8,4),EPSTI(26,4),EPSPI(26,4),
     2SIGI(26,4),DUM(26,44),IDUM(26,44)
      EQUIVALENCE (DUM(1,1),IDUM(1,1))
      DIMENSION NOYILO(20),SSTAR(20,10),HSTAR(20,10)
      DIMENSION ALPHA(20),CAPPA(20),COSTH(20)
C
C     THIS LINK SETS UP THE PLASTIC TABLES
C
      DO 1 I=1,NUMNP
      NPOLD=NPTN(I)
    1 NPTP(NPOLD)=I
C
      REWIND 12
      REWIND 14
C
      READ(14) NZONES
      DO 3 I=1,NZONES
      READ(14) IZ,IELAST(IZ),JPLAST(IZ),WGT(IZ),E1(IZ),E2(IZ),E3(IZ),
     1E4(IZ),E5(IZ)
C
      IF(JPLAST(IZ).EQ.0) GO TO 3
      IF(JPLAST(IZ).GT.2) GO TO 400
      IF(JPLAST(IZ).EQ.2) GO TO 4
C
      READ(14) KYILO,(SSTAR(IZ,J),J=1,KYILO),(HSTAR(IZ,J),J=1,KYILO)
      NOYILO(IZ)=KYILO
      GO TO 3
C
    4 READ(14) ALPHA(IZ),CAPPA(IZ),COSTH(IZ)
      GO TO 3
C
  400 WRITE(6,401) IZ,JPLAST(IZ)
  401 FORMAT(1H1,24HERROR IN LNK1F ZONE DATA//10X,7HZONE  =,I5/
     1 10X,7HJPLAST=,I5)
      CALL EXIT
C
    3 CONTINUE
C
C
```

```
      NUMCEL=0
      NCLUST=0
C
  100 IF((NUMPEL-NUMCEL).LT.MAXELB) NUMELB=NUMPEL-NUMCEL
      IF((NUMPEL-NUMCEL).GE.MAXELB) NUMELB=MAXELB
C
      DO 200 KK=1,NUMELB
C
      READ(14) NODEEL(KK),IZONE,(PLAST(KK),NP(KK,1),NP(KK,2),NP(KK,3),
     1NP(KK,4),ITYPE(KK,1),ITYPE(KK,2),ITYPE(KK,3),ITYPE(KK,4),
     2          THETA(KK,1),THETA(KK,2),THETA(KK,3),THETA(KK,4),
     3          R(,RJ,RK,RL,ZI,ZJ,ZK,ZL
C
      II=1
  201 NODE=NP(KK,(I)
      NP(KK,(I)=NPTP(NODE)
      II=II+1
      IF(II.LT.4) GO TO 201
      IF(II.GT.4) GO TO 202
      IF(NP(KK,4).EQ.0) GOTO 202
      GO TO 201
C
  202 IE=IELAST(IZONE)
      A1=E1((ZONE)
      A2=E2(IZONE)
      A3=E3(IZONE)
      A4=E4(IZONE)
      A5=E5((ZONE)
      NUME=NODEEL(KK)
      CALL ELOST(IE,ISTRES,A1,A2,A3,A4,A5,CC,NUME)
      DO 203 I=1,4
      DO 203 J=1,4
  203 C(KK,I,J)=CC(I,J)
C
      DO 204 I=1,4
      DO 204 J=1,8
      B(KK,1,J)=0.0
  204 P(KK,J,I)=0.0
C
      IF(NP(KK,4).NE.0) GO TO 208
C
      AJ=RJ-R(
      AK=RK-RI
      BJ=ZJ-ZI
      BK=ZK-ZI
      HH=AJ*BK-AK*BJ
      AA=AJ-AK
      BB=BJ-BK
C
      B(KK,1,1)=BB/HH
      B(KK,1,3)=BK/HH
      B(KK,1,5)=-BJ/HH
      IF(ISTRES.NE.0) GO TO 205
      RO=(AJ+AK)/3.
      ZO=(BJ+BK)/3.
      CAPRO=R(+RO
      B(KK,2,1)=(HH+BB*RO-AA*ZO)/(HH*CAPRO)
      B(KK,2,3)=(BK*RO-AK*ZO)/(HH*CAPRO)
      B(KK,2,5)=(-BJ*RO+AJ*ZO)/(HH*CAPRO)
  205 B(KK,3,2)=-AA/HH
      B(KK,3,4)=-AK/HH
      B(KK,3,6)=AJ/HH
      B(KK,4,1)=B(KK,3,2)
      B(KK,4,2)=B(KK,1,1)
      B(KK,4,3)=B(KK,3,4)
      B(KK,4,4)=B(KK,1,3)
      B(KK,4,5)=B(KK,3,6)
      B(KK,4,6)=B(KK,1,5)
C
      IF(ISTRES.NE.0) CONST=HH/2.
      IF(ISTRES.EQ.0) CONST=HH*CAPRO/2.
C
  212 DO 206 I=1,8
      DO 206 J=1,4
      DO 207 N=1,4
  207 P(KK,I,J)=P(KK,I,J)+CONST*B(KK,N,I)*CC(N,J)
  206 CONTINUE
      GO TO 231
C
C
  208 AJ=RJ-R(
      BJ=ZJ-ZI
      AA=DSQRT(AJ*AJ+BJ*BJ)
      AL=RL-RI
      BL=ZL-ZI
      BB=DSQRT(AL*AL+BL*BL)
      HH=AA*BB
      S1=-BJ/AA
      C1=AJ/AA
C
      IF(ISTRES.EQ.0) GO TO 209
      RO=AA/2.
      ZO=BB/2.
      GO TO 210
  209 AINT1=HH
      AINT2=HH*BB/2.
      AINT3=HH*AA/2.
      A(NT4=HH*(RI+(AA*C1+BB*S1)/2.)
      AINT13=AA*AINT4/2.+HH*AA**2*C1/12.
      AINT14=BB*A(NT4/2.+HH*BB**2*S1/12.
      A(NT16=(HH/2.)**2
      RO=AINT13/A(NT4
      ZO=AINT14/AINT4
  210 DUMMY=RO*S1+ZO*C1
      B(KK,1,1)=(-AA*S1-BB*C1 +DUMMY)/HH
      B(KK,1,3)=(BB*C1-DUMMY)/HH
      B(KK,1,5)=DUMMY/HH
      B(KK,1,7)=(AA*S1-DUMMY)/HH
      IF(ISTRES.NE.0) GO TO 211
      B(KK,2,1)=(HH*AINT1-BB*A(NT3+AINT16-AA*AINT2)/(HH*AINT4)
      B(KK,2,3)=(BB*A(NT3-A(NT16)/(HH*AINT4)
      B(KK,2,5)=A(NT16/(HH*A(NT4)
```

```
      B(KK,2,7)=(AA*AINT2-A(NT16)/(HH*AINT4)
  211 OUMMY=RO*C1-ZO*S1
      B(KK,3,2)=((BB*S1-AA*C1)+DUMMY)/HH
      B(KK,3,4)=(-BB*S1-OUMMY)/HH
      B(KK,3,6)=DUMMY /HH
      B(KK,3,8)=(AA*C1-DUMMY)/HH
      B(KK,4,1)=B(KK,3,2)
      B(KK,4,2)=B(KK,1,1)
      B(KK,4,3)=B(KK,3,4)
      B(KK,4,4)=B(KK,1,3)
      B(KK,4,5)=B(KK,3,6)
      B(KK,4,6)=B(KK,1,5)
      B(KK,4,7)=B(KK,3,8)
      B(KK,4,8)=B(KK,1,7)
      IF(ISTRES.NE.0) CONST=HH
      IF(ISTRES.EQ.0) CONST=AINT4
      GO TO 212
C
C
  231 DO 232 I=1,4
      EPSTI1(KK,I)=0.0
      EPSPI1(KK,I)=0.0
  232 S(GI1(KK,I)=0.0
C
      IF((PLAST(KK).NE.1) GO TO 237
C
      IDUM(KK,21)=NOYILO(IZONE)
      KYILO=NOYILD(IZONE)
      DO 233 I=1,KYILD
      DUM(KK,1)=SSTAR(IZONE,1)
  233 DUM(KK,I+10)=HSTAR(IZONE,I)
      DO 234 I=1,8
  234 DUM(KK,I+21)=0.0
      DUM(KK,27)=SSTAR(IZONE,1)
      GO TO 200
C
  237 IF(IPLAST(KK).NE.2) GO TO 235
C
      DUM(KK,1)=ALPHA(IZONE)
      DUM(KK,2)=CAPPA(IZONE)
      DUM(KK,3)=COSTH(IZONE)
      IDUM(KK,4)=0
      IF(CAPPA(IZONE).EQ.0.0) IDUM(KK,4)=1
      DUM(KK,5)=0.0
      DO 238 I=1,4
  238 DUM(KK,I+5)=0.0
      GO TO 200
C
  235 WRITE(6,236) NOOFEL(K),IZONE,IPLAST(KK)
  236 FORMAT(1H1,28HERROR IN ELEMENT DATA, LNK1F//
     110X,12HELEMENT NO.=,I5/10X,12HZONE NO.    =,I5/
     210X,12H(PLAST      =,I5)
      CALL EXIT
C
C
  200 CONTINUE
C
C
      NUMCEL=NUMCEL+NUMELB
      WRITE(12) NUMCEL,NUMELB,(NOOFEL(K),IPLAST(K),(NP(K,J),(TYPE(K,J),
     1THETA(K,J),J=1,4),((C(K,J,I),I=1,4),(B(K,J,I),I=1,8),J=1,4),
     2((P(K,J,I),J=1,8),EPSTI1(K,1),EPSPI1(K,I),S(GI1(K,I),I=1,4),
     3(DUM(K,I),I=1,44),K=1,NUMELB)
      NCLUST=NCLUST+1
C
      IF(NUMCEL.LT.NUMPEL) GO TO 100
C
      END FILE 12
      REWIND 12
      REWIND 14
C
  311 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELOST(IELAST,ISTRES,E1,E2,E3,E4,E5,C,NUME)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(4,4)
C
C     CALCULATE THE PLASTIC STRESS-STRAIN MATRIX
C
      DO 1 I=1,4
      DO 1 J=1,4
    1 C(I,J)=0.0
C
      IF(IELAST.NE.1) GO TO 20
C
      IF(ISTRES.EQ.2) GO TO 4
C
      EBAR=E1/((1.+E2)*(1.-2.*E2))
      C(1,1)=EBAR*(1.-E2)
      C(1,2)=EBAR*E2
      C(1,3)=C(1,2)
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=C(1,2)
      C(3,1)=C(1,2)
      C(3,2)=C(1,2)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-2.*E2)/2.
      RETURN
C
    4 EBAR=E1/(1.-E2*E2)
      C(1,1)=EBAR
      C(3,1)=EBAR*E2
      C(1,3)=C(3,1)
      C(3,3)=C(1,1)
      C(4,4)=EBAR*(1.-E2)/2.
      RETURN
C
   20 IF(IELAST.NE.2) GO TO 30
```

```
   21 WRITE(6,3) IELAST,NUME,ISTRES
    3 FORMAT(1H1/31H ERROR IN ELASTIC CONSTANT DATA/
     113H IELAST      =,I5/13H ELEMENT NO.=,I5/
     213H ISTRES      =,I5)
      CALL EXIT
C
   30 IF(IELAST.NE.3) GO TO 21
C
      IF(ISTRES.EQ.2) GO TO 21
C
      DO 31 I=1,3
      DO 31 J=1,3
   31 C(I,J)=E1
C
      RETURN
      END
```

```
      SUBROUTINE LINK2
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1       ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(1,50),
     2       RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXBD,
     3       MBLOCK,NREADS,MAXMBK,TREAL,PDAMP,BETA,KRUN,
     4       IPELTP,KOUNT,IFILLA,T,TMAX,TRERUN
      DIMENSION COM(198)
      EQUIVALENCE (MAXNP,COM(1))
C
C     THIS LINK INTEGRATES THE EQUATIONS OF MOTION
C
C     READ AND PRINT INTEGRATION DATA
C
      READ(5,1000)TMAX,TRERUN,ET,KDT,KINT,BETA,PDAMP
 1000 FORMAT(3E10.0,2I5,2E10.0)
C
C     TMAX   =MAX. RUN TIME FOR PROBLEM (SEC.)
C     TRERUN=BEGINNING TIME FOR RESTART (SEC.)
C     ET     =TIME INCREMENT (SEC.)
C     KDT    =0 USES DT AS READ IN
C            =1 USES KINT TO CHOOSE DT
C     KINT   =DIVISOR FOR CHOOSING DT
C     BETA   =INTEGRATION SCHEME PARAMETER
C     PDAMP =PERCENT CRITICAL DAMPING FOR ARTIFICIAL VISCOSITY
C
      IF(BETA.LE.0.0.OR.BETA.GT.0.25) BETA=1.0/6.0
      REWIND 8
      IF(KRUN.LT.2) GO TO 10
      READ(8)TITLE
      READ(8)COM
      GO TO 20
   10 WRITE(8)TITLE
      DT=ET
      IF(KDT.NE.0) DT=PERIOD/FLOAT(KINT)
      NREADS=1
      IF(NUMPEL.NE.0) NREADS=2+(NUMPEL-1)/MAXELB
      TREAL=0.0
      WRITE(8)COM
   20 WRITE(6,2000)TMAX,TRERUN,DT,BETA,PDAMP
 2000 FORMAT(1H1,30HMAX. TIME DURATION              =,1PE15.5,2X,3HSEC/
     1       1X,30HINITIAL TIME FOR INTEGRATION =,1PE15.5,2X,3HSEC/
     3       1X,30HTIME INCREMENT               =,1PE15.5,2X,3HSEC/
     7       1X,30HBETA                         =,1PE15.5/
     8       1X,30HPDAMP                        =,1PE15.5,2X,3HPCT)
C
C     CHECK LIMITS
C
      IF(KRUN.EQ.0.AND.TMAX.EQ.0.0) STOP
   40 IF(MBLOCK.LE.MAXMBK) GO TO 50
      WRITE(6,3000)MBLOCK,MAXMBK
 3000 FORMAT(49H1BANDWIDTH TOO LARGE FOR INTEGRATION CALCULATION./
     114H BANDWIDTH IS ,I5,31H, MAXIMUM BANDWIDTH ALLOWED IS ,I5)
      STOP
C
C     CALCULATE MATRICES FOR RECURSION
C
   50 IF(KRUN.GT.1) GO TO 60
      CALL LINK2A
C
C     INTEGRATE PROBLEM
C
   60 IF(TMAX.NE.0.0) GO TO 70
      END FILE 8
      REWIND 8
      STOP
   70 CONTINUE
      CALL LINK2B
C
      RETURN
      END
```

```
      SUBROUTINE MMTIA(L,M,N,B,IA,IB,C)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(L,M),B(M,N),C(L,N)
C
C     MATRIX MULTIPLY WITH TRANSPOSE OPTION
C
      DO 10 IL=1,L
      DO 10 IN=1,N
      DO 10 IM=1,M
      I1=IL
      IF(IA.LT.0) I1=IM
      I2=IL+IM-I1
      I3=IM
      IF(IB.LT.0) I3=IN
      I4=IM+IN-I3
   10 C(IL,IN)=C(IL,IN)+A(I1,I2)*B(I3,I4)
      RETURN
      END


      SUBROUTINE ERASE(N,X)
```

```
      REAL*8 X(N)
C
C     ERASE N WORDS IN X
C
      DO 10 I=1,N
   10 X(I)=0.0
      RETURN
      END
```

```
      SUBROUTINE LINK2A
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXFLB,NUMNP,NUMFL,
     1       ISTRES,LINES,NUMPFL,LOADNP(1),PERIOD,DT,NPLOAD(1,50),
     2       RAO(1,50),ZAO(1,50),SNORM(1,50),TITLE(10),MAXBO,
     3       MBLOCK,NREAOS,MAXMBK,TREAL,POAMP,BETA,KRUN,
     4       IPELTP,KOUNT,IFILLA,T,TMAX,TPERUN
      DIMENSION NAOJNP(30),ITYPE(30),THETA(30),XMASS(30),SNP(30,3),
     1NPAOJ(30,8),SAO(30,8,4),FI(4,30,30),FI(4,30,30),TMP(4),OMP(2,30)
      DIMENSION STRESS(4,6,50),NPADJL(50,6),NADJPL(50),QI(4,50,50),
     1R(4,30,100),Q(4,50,50),SPACE(20000)
      DIMENSION KEYS(3,100)
      EQUIVALENCE (FI(1,1,1),SPACE(1)),(FI(1,1,1),SPACE(4001)),
     1(QI(1,1,1),SPACE(1)),(Q(1,1,1),SPACE(10001)),
     2(R(1,1,1),SPACE(4001))
C
C     CALCULATE MATRICES FOR RECURSION AND SAVE ON TAPE
C
C
C     INITIALIZE
C
      REWIND 10
      READ(10)
      REWIND 20
      WRITE(20)TITLE
      NRECS=0
C
C     READ ANOTHER BLOCK FROM STIFFNESS TAPE
C
  100 READ(10)N1,N2,N5,(NADJNP(I),ITYPE(I),THETA(I),XMASS(I),(SNP(I,J),
     1J=1,3),(NPAOJ(I,J),(SAO(I,J,K),K=1,4),J=1,MXADJP),I=1,N5)
C
C     ZERO OUT FI
C
      CALL ERASE(4*MAXMBK*MAXMBK,FI)
C
C     COPY C(N) INTO FI
C
      DO 950 I=1,N5
      INP=I+N1-1
C
C     ADD DIAGONAL SET OF EQUATIONS
C
      DAMU=2.0*DSQRT(XMASS(I)*SNP(I,1))*POAMP/100.0
      OMP(1,I)=DAMU
      DAMW=2.0*DSQRT(XMASS(I)*SNP(I,3))*POAMP/100.0
      OMP(2,I)=DAMW
      FI(1,I,I)=(0.5*DAMU*DT+XMASS(I))/(BETA*DT*DT)+SNP(I,1)
      FI(2,I,I)=SNP(I,2)
      FI(3,I,I)=SNP(I,2)
      FI(4,I,I)=(0.5*DAMW*DT+XMASS(I))/(BETA*DT*DT)+SNP(I,3)
C
C     CHECK ADJACENT STIFFNESS TABLES FOR NODES IN THIS BLOCK
C
      DO 945 K=1,MXADJP
      JNP=NPADJ(I,K)
      IF(JNP.LT.N1.OR.JNP.GT.N2) GO TO 945
      J=JNP-N1+1
C
C     COPY ADJACENT STIFFNESS INTO FI AND ZERO IT OUT IN TABLES
C
      FI(1,I,J)=SAO(I,K,1)
      FI(2,I,J)=SAO(I,K,3)
      FI(3,I,J)=SAO(I,K,2)
      FI(4,I,J)=SAO(I,K,4)
      DO 920 M=1,4
  920 SAO(I,K,M)=0.0
      NADJNP(I)=NADJNP(I)-1
      NPAOJ(I,K)=0
C
C     ZERO OUT ITS TRANSPOSE ACROSS THE MAIN DIAGONAL
C
      DO 925 K1=1,MXADJP
      KK=K1
      IF(NPADJ(J,K1).EQ.INP) GO TO 930
  925 CONTINUE
      WRITE(6,1000)INP,JNP
 1000 FORMAT(36H1TABLE SEARCH ERROR IN LINK2A.  INP=,I5,6H, JNP=,I5)
      STOP
  930 DO 935 M=1,4
  935 SAO(J,KK,M)=0.0
      NADJNP(J)=NADJNP(J)-1
      NPAOJ(J,KK)=0
  945 CONTINUE
  950 CONTINUE
C
C     SKIP FOR FIRST BLOCK
C
      IF(N1.EQ.1) GO TO 150
C
C                          T
C     SUBTRACT OFF C(N) *F(N-1)*C(N)
C
C     LOOP OVER EACH NODE IN FI
C
      DO 120 I=1,N5
      DO 120 J=I,N5
C
C     LOOP OVER EACH NODE IN C
C
      DO 120 K1=1,MXADJP
      L=NPAOJ(I,K1)
```

```
      IF(L.EQ.0.OR.L.GE.N1) GO TO 120
      L=L-N1LAST+1
      DO 121 K2=1,MXADJP
      M=NPADJ(J,K2)
      IF(M.EQ.0.OR.M.GE.N1) GO TO 121
      M=M-N1LAST+1
C
C     SUBTRACT OFF CONTRIBUTION
C
      IL=MIN0(L,M)
      IM=MAX0(L,M)
      TMP(1)=-F(1,IL,IM)
      TMP(2)=-F(2,IL,IM)
      TMP(3)=-F(3,IL,IM)
      IF((M-L).GE.0) GO TO 125
      SAVE=TMP(3)
      TMP(3)=TMP(2)
      TMP(2)=SAVE
  125 TMP(4)=-F(4,IL,IM)
      F(1,I,J)=F(1,I,J)+SAD(I,K1,1)*SAD(J,K2,1)*TMP(1)+
     1                  SAD(I,K1,2)*SAD(J,K2,1)*TMP(2)+
     2                  SAD(I,K1,3)*SAD(J,K2,2)*TMP(3)+
     3                  SAD(I,K1,4)*SAD(J,K2,2)*TMP(4)
      F(2,I,J)=F(2,I,J)+SAD(I,K1,3)*SAD(J,K2,1)*TMP(1)+
     1                  SAD(I,K1,4)*SAD(J,K2,1)*TMP(2)+
     2                  SAD(I,K1,3)*SAD(J,K2,2)*TMP(3)+
     3                  SAD(I,K1,4)*SAD(J,K2,2)*TMP(4)
      F(3,I,J)=F(3,I,J)+SAD(I,K1,1)*SAD(J,K2,3)*TMP(1)+
     1                  SAD(I,K1,2)*SAD(J,K2,3)*TMP(2)+
     2                  SAD(I,K1,1)*SAD(J,K2,4)*TMP(3)+
     3                  SAD(I,K1,2)*SAD(J,K2,4)*TMP(4)
      F(4,I,J)=F(4,I,J)+SAD(I,K1,3)*SAD(J,K2,3)*TMP(1)+
     1                  SAD(I,K1,4)*SAD(J,K2,3)*TMP(2)+
     2                  SAD(I,K1,3)*SAD(J,K2,4)*TMP(3)+
     3                  SAD(I,K1,4)*SAD(J,K2,4)*TMP(4)
  121 CONTINUE
  120 CONTINUE
C
C     INVERT FI TO GET F(N)
C
  150 CALL ESC(FI,N5,MAXMBK,F)
C
C     SKIP FOR LAST BLOCK
C
      IF(N2.EQ.NUMNP) GO TO 175
C
C     CALCULATE -F(N)*C(N+1) IN FI
C
C     LOOP OVER NODES IN FI (I AND J) AND COLUMNS OF F (ROWS OF C) (L)
C
      CALL ERASE(40*MAXMBK*MAXMBK,FI)
      N6=0
      DO 160 L=1,N5
      DO 160 K=1,MXADJP
      IF(NPADJ(L,K).LE.N2) GO TO 160
      J=NPADJ(L,K)-N2
      N6=MAX0(N6,J)
      TMP(1)=-SAD(L,K,1)
      TMP(2)=-SAD(L,K,3)
      TMP(3)=-SAD(L,K,2)
      TMP(4)=-SAD(L,K,4)
C
C     ADD NODE CONTRIBUTION TO FI
C
      DO 161 I=1,N5
      IF=MIN0(I,L)
      LF=MAX0(I,L)
      CALL MMT(F(1,(IF,LF),2,2,2,TMP,L-(L-1),FI(1,I,J))
  161 CONTINUE
  160 CONTINUE
C
C     DELETE C(N+1) ENTRIES FROM ADJACENT STIFF TABLES
C
C     LOOP OVER NODES (ROWS IN C)
C
      DO 180 I=1,N5
C
C     CHECK AND ZERO OUT ENTRIES IN ADJACENT STIFF TABLES
C
      DO 180 K=1,MXADJP
      IF(NPADJ(I,K).LE.N2) GO TO 180
      DO 185 M=1,4
  185 SAD(I,K,M)=0.0
      NADJNP(I)=NADJNP(I)-1
      NPADJ(I,K)=0
  180 CONTINUE
C
C     SKIP FOR FIRST BLOCK
C
  175 IF(N1.EQ.1) GO TO 205
C
C     CONDENSE STIFFNESS TABLES
C
      DO 190 I=1,N5
      DO 195 K=1,MXADJP
      IF(NPADJ(I,K).NE.0) GO TO 195
      NZRO=0
      DO 196 L=K,MXADJP
      IF(NPADJ(I,L).NE.0) GO TO 197
      NZRO=NZRO+1
  196 CONTINUE
  197 ISTT=K+NZRO
      IF(ISTT.GT.MXADJP) GO TO 190
      DO 198 L=ISTT,MXADJP
      NPADJ(I,L-NZRO)=NPADJ(I,L)
      DO 198 M=1,4
  198 SAD(I,L-NZRO,M)=SAD(I,L,M)
      DO 200 L=1,NZRO
      NPADJ(I,MXADJP+1-L)=0
      DO 200 M=1,4
  200 SAD(I,MXADJP+1-L,M)=0.0
  195 CONTINUE
```

A-28

```
  190 CONTINUE
C
C     WRITE E(N) ON TAPE
C
  205 WRITE(20)N1,N2,N5,(((E(I,J,K),I=1,4),K=J,N5(,J=1,N5(
C
C     WRITE REST OF TABLES ON TAPE
C
      WRITE(20)(ITYPE(I),THETA(I(,XMASS(I),(DMP(J,I),J=1,2),I=1,N5(
      IF(N1.EQ.1) GO TO 225
      DO 215 I=1,N5
      N7=N5-I+1
      IF(NADJNP(N7).NE.0) GO TO 220
  215 CONTINUE
  220 WRITE(20)N7,(NADJNP(I),(NPADJ(I,J),(SADJ(I,J,K),K=1,4),J=1,MXADJP),
     1I=1,N7)
  225 N1LAST=N1
C
C     SKIP FOR LAST BLOCK
C
      IF(N2.EQ.NUMNP) GO TO 500
C
C     WRITE E(N)*C(N+1) ON DA DEVICE
C
      CALL DISKER(2,MRECS+1,E(1,4*MAXMRK*MAXMRK(
      KEYS(1,MRECS+1)=N1
      KEYS(2,MRECS+1)=N5
      KEYS(3,MRECS+1)=N6
      MRECS=MRECS+(
      GO TO 100
C
C     READ BACK RECORDS FROM DA DEVICE IN REVERSE ORDER
C
  500 IF(MRECS.EQ.0) GO TO 501
      DO 510 M7=1,MRECS
      CALL DISKER(1,MRECS+1-M7,E(,4*MAXMRK*MAXMRK)
      N1=KEYS(1,MRECS+1-M7)
      N5=KEYS(2,MRECS+1-M7)
      N6=KEYS(3,MRECS+1-M7)
  510 WRITE(20)N1,N5,N6,(((E(I,J,K),I=1,4(,J=1,N5(,K=1,N6)
C
C     CALCULATE STRESS CONSTRAINT TABLES
C
  501 REWIND 20
      READ(10)
      NLOAD=LOADNP(1)
  600 READ(10)N1,N2,N5,(NADJNP(I),LOST,(NPADJ(I,J),J=1,MXADJP),
     1((QI(K,I,1),QI(K,I,2),K=1,4),(IP(K,I,J),R(K,I,J+1(),K=1,4),
     2J=1,MXADJP),I=1,N5)
      DO 625 I=1,N5
      INP=I+N1-1
      DO 605 J=1,NLOAD
      LNODE=J
      NODE=NPLOAD(1,J)
      IF(INP.EQ.NODE) GO TO 610
  605 CONTINUE
      GO TO 625
  610 CS=DCOS(SNORM(1,LNODE((
      SN=DSIN(SNORM(1,LNODE))
      NPADJL(LNODE,1(=NODE
      STRESS(1,1,LNODE(=QI(1,1,1)*CS+QI(4,1,1(*SN
      STRESS(2,1,LNODE(=QI(4,1,1)*CS+QI(3,1,1)*SN
      STRESS(3,1,LNODE)=QI(1,1,2)*CS+QI(4,1,2)*SN
      STRESS(4,1,LNODE)=QI(4,1,2)*CS+QI(3,1,2)*SN
      NAD=NADJNP(I)
      IF(NAD.LE.5) GO TO 615
      WRITE(6,2000)I
 2000 FORMAT(21H BAD NEWS, NODE POINT,I5,50H HAS TOO MANY ADJACENT NODES
     1 FOR STRESS CONSTRAINT)
      STOP
  615 NADJPL(LNODE)=NAD+1
      DO 620 J=1,NAD
      NPADJL(LNODE,J+1(=NPADJ(I,J)
      STRESS(1,J+1,LNODE)=R(1,I,J)*CS+R(4,I,J)*SN
      STRESS(2,J+1,LNODE)=R(4,I,J)*CS+R(3,I,J)*SN
      STRESS(3,J+1,LNODE(=R(1,I,J+1()*CS+R(4,I,J+1()*SN
  620 STRESS(4,J+1,LNODE)=R(4,I,J+1()*CS+R(3,I,J+1()*SN
  625 CONTINUE
      IF(N2.LT.NUMNP) GO TO 600
C
C                                                    -1    T
C     DO FORWARD PART OF RECURSION TO FIND IK+QI    *S
C
      REWIND 10
      READ(20)
      IHALF=1
      JHALF=51
      MRECS=0
  650 READ(20)N1,N2,N5,(((E(I,J,K),I=1,4),K=J,N5),J=1,N5)
      READ(20)(ITYPE(I),THETA(I(,XMASS(I(,(DMP(J,I(,J=1,2),I=1,N5)
      IF(N1.EQ.1) GO TO 655
      READ(20)N7,(NADJNP(I(,((NPADJ(I,J),(SADJ(I,J,K),K=1,4),J=(,MXADJP),
     1I=1,N7)
  655 CALL ERASE(4*MAXMRK*MXLOAD,R(1,1,JHALF))
C
C     PUT IN G(N)
C
      DO 671 I=1,NLOAD
      IPJHM1=1+JHALF-1
      NUM=NADJPL(I)
      DO 670 J=1,NUM
      JNODE=NPADJL(I,J)
      IF(JNODE.LT.N1.OR.JNODE.GT.N2( GO TO 670
      R(1,JNODE-N1+1,IPJHM1    (=STRESS(1,J,I)
      R(2,JNODE-N1+1,IPJHM1    I=STRESS(3,J,I(
```

```
      R(3,JNODE-N1+1,IPJHM1    )=STRESS(2,J,I)
      R(4,JNODE-N1+1,IPJHM1    )=STRESS(4,J,I)
  670 CONTINUE
  671 CONTINUE
C
C                          T
C     SUBTRACT OFF C(N) *X(N-1)
C
      IF(N1.EQ.1) GO TO 675
      DO 680 I=1,N7
      NUM=NADJNP(I)
      IF(NUM.EQ.0) GO TO 680
      DO 685 KK=1,NUM
      JPOS=LOADJ(I,KK)-N1LAST+1
      DO 685 L=1,NLOAD
      LPI=L+IHALF-1
      LPJ=L+JHALF-1
      R(1,I,LPJ)=R(1,I,LPJ)-(SAD(I,KK,1)*R(1,JPOS,LPI)+
     1                       SAD(I,KK,2)*R(2,JPOS,LPI))
      R(2,I,LPJ)=R(2,I,LPJ)-(SAD(I,KK,3)*R(1,JPOS,LPI)+
     1                       SAD(I,KK,4)*R(2,JPOS,LPI))
      R(3,I,LPJ)=R(3,I,LPJ)-(SAD(I,KK,1)*R(3,JPOS,LPI)+
     1                       SAD(I,KK,2)*R(4,JPOS,LPI))
      R(4,I,LPJ)=R(4,I,LPJ)-(SAD(I,KK,3)*R(3,JPOS,LPI)+
     1                       SAD(I,KK,4)*R(4,JPOS,LPI))
  685 CONTINUE
  680 CONTINUE
C
C     MULTIPLY BY F(N)
C
  675 CALL ERASE(4*MAXMRK*NLOAD,R(1,1,IHALF))
      DO 690 I=1,N5
      DO 690 J=1,N5
      IF=MIN0(I,J)
      JF=MAX0(I,J)
      DO 690 L=1,NLOAD
  690 CALL MMT(F(I,IF,JF),2,2,2,R(1,J,L+JHALF-1),J-1,1,R(1,I,L+IHALF-1))
      IF(NUMNP.EQ.MBLOCK) GO TO 725
      CALL DISKEP(2,MRECS+1,R(1,1,IHALF),4*MAXMRK*NLOAD)
      KEYS(1,MRECS+1)=N1
      KEYS(2,MRECS+1)=N2
      KEYS(3,MRECS+1)=N5
      MRECS=MRECS+1
      NILAST=N1
      IF(N2.LT.NUMNP) GO TO 650
      IHALF=52-IHALF
      JHALF=52-JHALF
C
C     DO BACKWARD PART OF RECURSION
C
      IDR=MRECS-1
  720 READ(20)NINEXT,N5,N6,(((F(I,J,K,L),K=1,4),J=1,N5),L=1,NA)
      N1=NINEXT+N5
      CALL DISKFR(1,IDR,R(1,1,IHALF),4*MAXMRK*NLOAD)
      DO 710 I=1,N5
      DO 710 J=1,NA
      DO 710 L=1,NLOAD
  710 CALL MMT(F(I,1,J),2,2,2,R(1,J,L+JHALF-1),1,1,R(1,I,L+IHALF-1))
      CALL DISKFR(2,IDR,R(1,1,IHALF),4*MAXMRK*NLOAD)
      IDR=IDR-1
      IHALF=52-IHALF
      JHALF=52-JHALF
      IF(N1NEXT.GT.1) GO TO 720
C
C                                  -1  T
C     MULTIPLY BY S TO FORM Q = S*(E*D)  *S
C
      CALL ERASE(4*MAXLOAD*NLOAD,Q(1))
      IHALF=51
      IDR=0
  730 IDR=IDR+1
      CALL DISKFR(1,IDR,R(1,1,IHALF),4*MAXMRK*NLOAD)
      N1=KEYS(1,IDR)
      N2=KEYS(2,IDR)
      N5=KEYS(3,IDR)
  735 DO 730 I=1,NLOAD
      NUM=NADJPL(I)
      DO 730 J=1,NUM
      JNODE=NPADJL(I,J)
      IF(JNODE.LT.N1.OR.JNODE.GT.N2) GO TO 730
      JPOS=JNODE-N1+1
      DO 740 L=1,NLOAD
  740 CALL MMT(STRESS(I,J,1),2,2,2,R(1,JPOS,L+IHALF-1),1,1,Q(I,1,L))
  730 CONTINUE
      IF(N2.LT.NUMNP) GO TO 725
C
C     INVERT Q1 TO GET Q
C
      CALL PSC(Q1,NLOAD,NLOAD,Q)
C
C     WRITE S TABLES ON TAPE
C
      WRITE(20)NLOAD,(NADJPL(I),(NPADJL(I,J),J=1,(STRESS(K,J,I),K=1,4),
     1J=1,6),I=1,NLOAD)
C
C                  T          T          -1  T -1
C     MULTIPLY BY S  TO GET S *(S*(E*D)  *S )   AND WRITE ON TAPE
C
      IHALF=51
      DO 755 I=1,NLOAD
      DO 755 J=1,NLOAD
      DO 755 K=1,4
  755 Q(K,I,J)=Q(K,I,J)
      N1=1
  760 N2=N1+MBLOCK-1
      IF(N2.GT.NUMNP) N2=NUMNP
      N5=N2-N1+1
      CALL ERASE(4*MAXMRK*NLOAD,R(1,1,IHALF))
      N7=0
      DO 765 I=1,N5
      INDOF=I+N1-1
      ISW=0
```

```
      DO 770 J=1,NLOAD
      NUM=NADJPL(J)
      DO 775 K=1,NUM
      KK=K
      IF(NPADJL(J,K).EQ.INODE) GO TO 780
  775 CONTINUE
      GO TO 770
  780 ISW=1
      DO 785 L=1,NLOAD
      JO=MIN0(J,L)
      LO=MAX0(J,L)
  785 CALL MMTISTRESS1(KK,J),2,2,2,0(1),JO,LO),-1,L-J,
     1P(1,N7+1,L+(HALF-1))
  770 CONTINUE
      IF(ISW.EQ.0) GO TO 765
      N7=N7+1
      NADJNP(N7)=INODE
  765 CONTINUE
      IF(N7.EQ.0) GO TO 790
      WRITE(20)N7,(NADJNP(J),(P(J,K+(HALF-1)),I=1,4),K=1,NLOAD),
     1J=1,N7)
  790 N1=N2+1
      IF(N2.LT.NUMNP) GO TO 760
      END FILE 20
      REWIND 20
C
      RETURN
      END
```

```
      SUBROUTINE DISKE(IIROW,IREC,X,NWRDS)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(NWRDS)
      DEFINE FILE 22(195,3200,L,IGXO)
C
C     DISC READ/WRITE ROUTINE FOR LINK2A
C
      NBLK=400
      NFIN=0
      IDR=(IREC-1)*15
      DO 100 NZ=1,15
      N=NBLK
      IF(N+NFIN.GT.NWRDS) N=NWRDS-NFIN
      IF(IROW.EQ.1) READ (22'NZ+IDR)(X(I+NFIN),I=1,N)
      IF(IROW.EQ.2) WRITE(22'NZ+IDR)(X(I+NFIN),I=1,N)
   90 NFIN=NFIN+N
      IF(NFIN.GE.NWRDS) RETURN
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE ESC(A,N,ND,A1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,ND,ND),A1(4,ND,ND),F(14)
C
C     ESCALATOR MATRIX INVERSION IN 2X2 BLOCKS
C
      CALL ERASE(16*ND*ND,A1)
C
      DET=A(1,1,1)*A(4,1,1)-A(3,1,1)*A(3,1,1)
      A1(1,1,1)=A(4,1,1)/DET
      A1(3,1,1)=-A(3,1,1)/DET
      A1(2,1,1)=A1(3,1,1)
      A1(4,1,1)=A(1,1,1)/DET
C
      DO 500 I=2,N
      IM1=I-1
      DO 105 J=1,4
  105 F(J)=A(J,I,1)
      DO 110 M=1,IM1
      Q=2.D
      IF(L.EQ.M) Q=1.D
      F(1)=F(1)-Q*(A(1,L,1)*A(1,L,M)*A(1,M,1)+
     1                A(2,L,1)*A(2,L,M)*A(1,M,1)+
     2                A(1,L,1)*A(1,L,M)*A(2,M,1)+
     3                A(2,L,1)*A(4,L,M)*A(2,M,1))
      F(2)=F(2)-Q*(A(3,L,1)*A(1,L,M)*A(1,M,1)+
     1                A(4,L,1)*A(2,L,M)*A(1,M,1)+
     2                A(3,L,1)*A(1,L,M)*A(2,M,1)+
     3                A(4,L,1)*A(4,L,M)*A(2,M,1))
      F(3)=F(3)-Q*(A(1,L,1)*A(1,L,M)*A(3,M,1)+
     1                A(2,L,1)*A(2,L,M)*A(3,M,1)+
     2                A(1,L,1)*A(3,L,M)*A(4,M,1)+
     3                A(2,L,1)*A(4,L,M)*A(4,M,1))
      F(4)=F(4)-Q*(A(3,L,1)*A(1,L,M)*A(3,M,1)+
     1                A(4,L,1)*A(2,L,M)*A(3,M,1)+
     2                A(3,L,1)*A(3,L,M)*A(4,M,1)+
     3                A(4,L,1)*A(4,L,M)*A(4,M,1))
  110 CONTINUE
      F(3)=0.5*(F(2)+F(3))
      F(2)=F(3)
C
      DET=F(1)*F(4)-F(3)*F(3)
      A1(1,1,I)=F(4)/DET
      A1(3,1,I)=-F(3)/DET
      A1(2,1,I)=A1(3,1,I)
      A1(4,1,I)=F(1)/DET
C
      DO 120 L=1,IM1
      DO 120 M=1,IM1
      LX=MIN0(L,M)
      MX=MAX0(L,M)
      CALL MMT(A1(1,LX,MX),2,2,2,A(1,M,1),M-L,1,A1(1,L,I))
  120 CONTINUE
      DO 130 L=1,IM1
      T1=A1(1,L,I)*A(1,1,1)+A1(3,L,I)*A(3,1,1)+A1(2,L,I)
      T2=A1(2,L,I)*A(1,1,1)+A1(4,L,I)*A(3,1,1)+A1(2,L,I)
      T3=A1(1,L,I)*A(3,1,1)+A1(3,L,I)*A(3,1,1)+A1(4,L,I)
      T4=A1(2,L,I)*A(3,1,1)+A1(4,L,I)*A(3,1,1)+A1(4,L,I)
```

```
      A[1,L,I)=-T1
      A[2,L,I)=-T2
      A[3,L,I)=-T3
      A[4,L,I)=-T4
  130 CONTINUE
C
      DO 140 L=1,IM1
      DO 140 M=L,IM1
      A[1,L,M)=A[1,L,M)+A[1,L,I)*A[1,M,I)*F[(1(+
     1                  A[2,L,I)*A[1,M,I)*F[(2)+
     2                  A[1,L,I(*A[2,M,I)*F[(3)+
     3                  A[3,L,I)*A[3,M,I)*F[(4)
      A[2,L,M)=A[2,L,M)+A[2,L,I)*A[1,M,I)*F[(1)+
     1                  A[4,L,I)*A[1,M,I)*F[(2)+
     2                  A[2,L,I)*A[3,M,I)*F[(3)+
     3                  A[4,L,1)*A[3,M,I)*F[(4)
      A[3,L,M)=A[3,L,M)+A[1,L,I)*A[2,M,I)*F[(1)+
     1                  A[3,L,I)*A[2,M,I)*F[(2)+
     2                  A[1,L,I)*A[4,M,I)*F[(3)+
     3                  A[3,L,I)*A[4,M,I)*F[(4)
      A[4,L,M)=A[4,L,M)+A[2,L,I)*A[2,M,I)*F[(1)+
     1                  A[4,L,I)*A[2,M,I)*F[(2)+
     2                  A[2,L,I)*A[4,M,I)*F[(3)+
     3                  A[4,L,I)*A[4,M,I)*F[(4)
  140 CONTINUE
  500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LINK28
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPR,MAXELR,NUMNP,NUMEL,
     1       ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(1,50),
     2       RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXRD,
     3       MBLOCK,NREADS,MAXMBK,TREAL,PDAMP,BETA,KRUN,
     4       IPELTP,KOUNT,IFILLA,T,TMAX,TREPUN
      DIMENSION XNP1(2,1000),XN(2,1000),VEL(2,1000),ACL(2,1000),
     1SIGMAU(1,50),SIGMAW(1,50)
      DIMENSION NADJNP(30),ITYPE(30),THETA(30),XMASS(30),DMP(2,30),
     1NPADJ(30,8),SAD(30,8,4),F(4,30,30),PHS(2,30)
      DIMENSION STRESS(4,6,50),NPADJL(50,6),NADJPL(50),XLOAD(2,50),
     1STOINV(4,30,50),NUMARY(30)
C
C     INTEGRATE THE PROBLEM AND WRITE THE DTH TAPE
C
C     INITIALIZE
C
      REWIND 20
      CALL RSTRES(MXLINE,MXLOAD,LOADNP,LINES,RAD,ZAD,SNORM,SIGMAU,
     1SIGMAW,T,TREAL)
      WRITE(6,9)
    9 FORMAT(1H1)
      READ(20)
      ICP=0
      IF(NUMNP.EQ.MBLOCK) ICP=1
      IREAD=0
      IF(TREPUN.NE.0.0) GO TO 70
C
C     ZERO OUT EVERYTHING AND START AT T=0
C
      CALL ERASE(2*NUMNP,XN)
      CALL ERASE(2*NUMNP,VEL)
      CALL ERASE(2*NUMNP,ACL)
      KOUNT=0
      T=0.0
      GO TO 110
C
C     POSITION DTH TAPE TO START TIME AND CONTINUE
C
   70 IF(NREADS.EQ.1) GO TO 80
      IDUM=NREADS-1
      DO 75 I=1,IDUM
   75 READ(8)
   80 READ(8)T
      IF((TREPUN-T).GT.(1.5*DT)) GO TO 7
      IDUM=LOADNP(1)
      READ (8)T,KOUNT,TM,(XN(1,I),VEL(1,I),ACL(1,I),XN(2,I),VEL(2,I),
     1ACL(2,I),I=1,NUMNP),(SIGMAU(1,I),SIGMAW(1,I),I=1,IDUM)
      GO TO 105
C
C     WRITE DTH TAPE
C
  100 TM=T+TREAL
      IDUM=LOADNP(1)
      WRITE(8)T,KOUNT,TM,(XN(1,I),VEL(1,I),ACL(1,I),XN(2,I),VEL(2,I),
     1ACL(2,I),I=1,NUMNP),(SIGMAU(1,I),SIGMAW(1,I),I=1,IDUM)
      WRITE(6,101)T,KOUNT,TM
  101 FORMAT( 5X,3HT =,1PE15.5,2X,6HSEC ,,3X,7HKOUNT =,I10,3X,
     1 18HREAL T =,1PE15.5,2X,3HSEC)
  105 T=T+DT
  110 IF(T.GT.TMAX) GO TO 900
C
C     ZERO OUT X(N+1) ARRAY
C
      CALL ERASE(2*NUMNP,XNP1)
C
C     READ ANOTHER BLOCK OF TABLES
C
  200 IF(ICP.EQ.1.AND.IREAD.EQ.1) GO TO 210
      READ(20)N1,N2,N5,(((F(I,J,K),I=1,4),K=J,N5),J=1,N5)
```

```
      READ (20)((ITYPE(I),THETA(I),XMASS(I,I),DMP(J,I),J=1,2),I=1,N5)
      IF(N1.EQ.1) GO TO 210
      READ (20)N7,(NADJNP(I),(NPADJ(I,J),(SAD(I,J,K),K=1,4),J=1,MXADJP),
     1I=1,N7)
C
C     CALCULATE RHS VECTOR FOR THIS BLOCK
C
  210 DO 225 I=1,N5
      NP=I+N1-1
      DAMU=DMP(1,I)
      DAMW=DMP(2,I)
      QU=(0.5*DAMU*DT+XMASS(I))/(BETA*DT*DT)
      QW=(0.5*DAMW*DT+XMASS(I))/(BETA*DT*DT)
      RHS(1,I)=QU*XN(1,NP)+(DT*QU-DAMU)*VEL(1,NP)-(0.5*DAMU)*DT*
     1(BETA-0.5)*DT*DT*QU*ACL(1,NP)
      RHS(2,I)=QW*XN(2,NP)+(DT*QW-DAMW)*VEL(2,NP)-(0.5*DAMW*DT*
     1(BETA-0.5)*DT*DT*QW*ACL(2,NP)
  225 CONTINUE
C
C     DO FORWARD PART OF RECURSION FOR THIS BLOCK
C
      IF(N1.EQ.1) GO TO 260
C
      DO 250 I=1,N7
      NUM=NADJNP(I)
      IF(NUM.EQ.0) GO TO 250
      DO 255 KK=1,NUM
      JNP=NPADJ(I,KK)
      RHS(1,I)=RHS(1,I)-(SAD(I,KK,1)*XNP1(1,JNP)+
     1                   SAD(I,KK,2)*XNP1(2,JNP))
      RHS(2,I)=RHS(2,I)-(SAD(I,KK,3)*XNP1(1,JNP)+
     1                   SAD(I,KK,4)*XNP1(2,JNP))
  255 CONTINUE
  250 CONTINUE
C
  260 DO 265 I=1,N5
      DO 265 J=1,N5
      IF=MIN0(I,J)
      JF=MAX0(I,J)
  265 CALL MMT(F(1,IF,JF),2,2,1,RHS(1,J),J-1,I,XNP1(1,1+N1-1))
      IF(N2.LT.NUMNP) GO TO 20
C
C     DO BACKWARD PART OF RECURSION
C
  300 IF(ICP.EQ.1) GO TO 315
      READ(20)N1NEXT,N5,N6,(((F(I,J,K),I=1,4),J=1,N5),K=1,N6)
      N1=N1NEXT+N5
      DO 310 I=1,N5
      DO 310 J=1,N6
  310 CALL MMT(F(1,1,JI,2,2,1,XNP1(1,J+N1-1),1,1,XNP1(1,I+N1NEXT-1))
      IF(N1NEXT.GT.1) GO TO 300
C
C     MULTIPLY BY S
C
  315 IF(ICP.EQ.1.AND.IREAD.EQ.1) GO TO 312
      READ(20)NLOAD,(NADJPL(I),(NPADJL(I,JI,(STRESS(K,J,I),K=1,4),
     IJ=1,6),I=1,NLOAD)
  312 CALL ERASE(2*NLOAD,XLOAD)
      DO 400 L=1,NLOAD
      NUM=NADJPL(L)
      DO 400 K=1,NUM
      J=NPADJL(L,K)
  400 CALL MMT(STRESS(I,K,L),2,2,1,XNP1(1,J),1,I,XLOAD(1,L))
C
C     CALCULATE BOUNDARY STRESSES
C
      CALL BSTRES(MXLINE,MXLOAD,LOADNP,LINES,RAD,ZAD,SNORM,SIGMAU,
     1SIGMAW,T,TREAL)
C
C     SUBTRACT FROM SIGMAIN*1)
C
      DO 405 L=1,NLOAD
      XLOAD(1,L)=SIGMAU(1,L)-XLOAD(1,L)
  405 XLOAD(2,L)=SIGMAW(1,L)-XLOAD(2,L)
C
C                    T          -1  T -1
C     MULTIPLY BY S *IS*(K+QI  *S )
C
      CALL ERASE(2*NUMNP,XNP1)
  410 IF(ICP.EQ.1.AND.IREAD.EQ.1) GO TO 412
      READ(20,END=425)N7,(NUMARY(J),((STQINV(I,J,K),I=1,4),K=1,NLOAD),
     1J=1,N7)
  412 DO 415 I=1,N7
      INODE=NUMARY(I)
      DO 415 L=1,NLOAD
  415 CALL MMT(STQINV(1,1,L),2,2,1,XLOAD(1,L),1,I,XNP1(1,INODE))
      IREAD=1
      IF(ICP.EQ.0) GO TO 410
  425 REWIND 20
C
C     READ ANOTHER BLOCK OF TABLES
C
      READ(20)
  500 IF(ICP.EQ.1.AND.IREAD.EQ.1) GO TO 510
      READ(20)N1,N2,N5,(((F(I,J,K),I=1,4),K=J,N5),J=1,N5)
      READ (20)((ITYPE(I),THETA(I),XMASS(I,I),DMP(J,I),J=1,2),I=1,N5)
      IF(N1.EQ.1) GO TO 510
      READ (20)N7,(NADJNP(I),(NPADJ(I,JI,(SAD(I,J,K),K=1,4),J=1,MXADJP),
     1I=1,N7)
C
C     CALCULATE RHS VECTOR FOR THIS BLOCK
C
  510 DO 525 I=1,N5
      NP=I+N1-1
      DAMU=DMP(1,I)
      DAMW=DMP(2,I)
      QU=(0.5*DAMU*DT+XMASS(I))/(BETA*DT*DT)
      QW=(0.5*DAMW*DT+XMASS(I))/(BETA*DT*DT)
      RHS(1,I)=QU*XN(1,NP)+(DT*QU-DAMU)*VEL(1,NP)-(0.5*DAMU*DT*
     1(BETA-0.5)*DT*DT*QU)*ACL(1,NP)+XNP1(1,NP)
      RHS(2,I)=QW*XN(2,NP)+(DT*QW-DAMW)*VEL(2,NP)-(0.5*DAMW*DT*
     1(BETA-0.5)*DT*DT*QW(*ACL(2,NP)+XNP1(2,NP)
```

```
  525 CONTINUE
      CALL ERASE(2*N5,XNP1(1,NI))
C
C     DO FORWARD PART OF RECURSION FOR THIS BLOCK
C
      IF(NI.EQ.1) GO TO 560
C
      DO 550 I=1,N7
      NUM=NAOJNP(I)
      IF(NUM.EQ.0) GO TO 550
      DO 555 KK=1,NUM
      JNP=NPADJ(I,KK)
      RHS(1,I)=RHS(1,I)-(SAO(I,KK,1)*XNP1(1,JNP)+
     1                   SAO(I,KK,2)*XNP1(2,JNP))
      RHS(2,I)=RHS(2,I)-(SAO(I,KK,3)*XNP1(1,JNP)+
     I                   SAO(I,KK,4)*XNP1(2,JNP))
  555 CONTINUE
  550 CONTINUE
C
  560 DO 565 I=1,N5
      DO 565 J=1,N5
      IF=MIN0(I,J)
      JF=MAX0(I,J)
  565 CALL MMT(F(1,IF,JF),2,2,1,RHS(1,J),J-I,I,XNPI(I,I+N1-1))
      IF(N2.LT.NUMNP) GO TO 500
C
C     DO BACKWARD PART OF RECURSION
C
  600 IF(ICP.EQ.1) GO TO 615
      READ(2D)NINEXT,N5,N6,((F(I,J,K),I=1,4),J=I,N5),K=1,N6)
      N1=N1NEXT+N5
      DO 610 I=1,N5
      DO 610 J=1,N6
  610 CALL MMT(F(1,I,J),2,2,1,XNP1(1,J+N1-1),1,1,XNPI(1,I+N1NEXT-1))
      IF(NINEXT.GT.1) GO TO 600
  615 REWIND 2D
C
C     CALCULATE REST OF DIFFERENCE SCHEME
C
      DO 325 I=1,NUMNP
      ACLU=ACL(1,I)
      ACL(1,I)=(XNP1(1,I)-XN(1,I)-VEL(1,I)*DT+(BETA-0.5)*DT*DT*ACL(I,I))
     1/(BETA*DT*DT)
      ACLW=ACL(2,I)
      ACL(2,I)=(XNP1(2,I)-XN(2,I)-VEL(2,I)*DT+(BETA-0.5)*DT*DT*ACL(2,I))
     I/(BETA*DT*DT)
      VEL(I,I)=VEL(1,I)+0.5*DT*(ACLU+ACL(I,I))
      VEL(2,I)=VEL(2,I)+0.5*DT*(ACLW+ACL(2,I))
      XN(1,I)=XNP1(1,I)
      XN(2,I)=XNP1(2,I)
  325 CONTINUE
      KOUNT=KOUNT+1
      READ(20)
      GO TO 100
  900 END FILE 8
      REWIND 8
      REWIND 20
      RETURN
      END
```

```
      SUBROUTINE BSTRES(MXL(NE,MXLOAD,LOADNP,LINES,RAD,ZAD,SNORM,
     ISIGMAU,SIGMAW,T,TX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LOADNP(MXLINE),RAD(MXL(NE,MXLOAD),ZAO(MXLINE,MXLOAD),
     ISNORM(MXL(NE,MXLOAD),SIGMAU(MXL(NE,MXLOAD),SIGMAW(MXL(NE,MXLOAD)
      DATA IRT/0/
C
C     CALCULATE BOUNDARY STRESSES
C
      IRT=IRT+1
      TX=C.0
      NUM=LOADNP(1)
      DO 10 I=1,NUM
      SIGMAU(1,I)=0.0
   10 SIGMAW(1,I)=0.0
      IF(IRT-2)100,200,300
C
  100 READ(5,1000)IPULSE,IDIREC,N1,N2,STAMP
 1000 FORMAT(4I5,E10.2)
C
C     IPULSE = 0 DELTA FUNCTION STRESS
C            = 1 STEP FUNCTION STRESS
C     IDIREC = 0 STRESS IN TRANSVERSE DIRECTION
C            = 1 STRESS IN NORMAL DIRECTION
C     N1          FIRST LOADED NODE POINT
C     N2          LAST LOADED NODE POINT
C     STAMP       STRESS AMPLITUDE
C
      IF(N1.EQ.0) N1=1
      IF(N2.EQ.0) N2=LOADNP(1)
      IF(STAMP.EQ.0.0) STAMP=1.0
      WRITE(6,2000)IPULSE,IDIREC,N1,N2,STAMP
 2000 FORMAT(9H1IPULSE= ,I5,10H, IDIREC= ,I5,6H, N1= ,I5,6H, N2= ,I5,
     I9H, STAMP= ,F10.2)
      RETURN
C
  200 IF(T.NE.0.0.AND.IPULSE.EQ.0) RETURN
  205 DO 210 I=N1,N2
      CS=DCOS(SNORM(I,1))
      SN=DSIN(SNORM(I,1))
      IF(IDIREC.NE.0) GO TO 206
      SIGMAU(1,I)=-STAMP*SN
      SIGMAW(1,I)=STAMP*CS
      GO TO 210
  206 SIGMAU(1,I)=STAMP*CS
      SIGMAW(1,I)=STAMP*SN
  210 CONTINUE
      RETURN
C
  300 IF(IPULSE.EQ.1) GO TO 205
      RETURN
```

```
      END
```

```
      SUBROUTINE LINK3
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1        ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(I,50),
     2        RAD(1,50),ZAD(I,50),SNORM(1,50),TITLE(10),MAXBD,
     3        MBLOCK,NREADS,MAXMBK,TREAL,PDAMP,BETA,KRUN,
     4        NUMOUT,TMAX,TSTRSS,IOUT,JOUT
      DIMENSION UDDN(1000),WDDN(1000),UDN(1000),WDN(1000),UN(1000),
     1WN(1000),PRESSU(1,100),PRESSW(1,100),FILLA(200)
      DIMENSION STNPU(4,100),STNPW(4,100),STADU(4,100,8),STADW(4,100,8)
     1,NADJEL(100),NADJNP(100),NPADJ(100,8),NPOUT(100),NPTP(100)
      DIMENSION BLOCKA(3200),BLOCKB(3200)
      EQUIVALENCE (UDDN(1),BLOCKA(1)),(WDDN(1),BLOCKA(1001)),(UDN(1),
     1BLOCKA(2001)),(PRESSU(I,1),BLOCKA(3001)),(PRESSW(1,1),BLOCKA(3101)
     2),(WDN(1),BLOCKB(1)),(UN(1),BLOCKB(1001)),(WN(1),BLOCKB(2001)),
     3(FILLA(1),BLOCKB(3001))
      DIMENSION COM(198)
      EQUIVALENCE(MAXNP,COM(1)).
C
C     THIS LINK PROCESSES THE 8TH TAPE AND CALCULATES STRESSES,
C     PRINTS AND WRITES BOTH ON THE 8TH TAPE
C
      REWIND 8
C
C     POSITION 8TH TAPE PAST TITLE AND COMMON BLOCKS
C
      READ(8)TITLE
      READ(8)COM
C
      CALL LINK3A(BLOCKA,BLOCKB,STNPU,STNPW,STADU,STADW,NADJEL,NADJNP,
     1NPADJ,NPOUT,NPTP)
      CALL LINK3B(UDDN,WDDN,UDN,WDN,UN,WN,PRESSU,PRESSW,STNPU,STNPW,
     1STADU,STADW,NADJEL,NADJNP,NPADJ,NPOUT,NPTP)
C
      RETURN
      END
```

```
      SUBROUTINE LINK3A(TTADU ,TTADW ,STNPU,STNPW,STADU,STADW,NADJEL,
     1NADJNP,NPADJ,NPOUT,NPTP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1        ISTRES,LINES,NUMPEL,LOADNP(1),PERIOD,DT,NPLOAD(I,50),
     2        RAD(1,50),ZAD(1,50),SNORM(1,50),TITLE(10),MAXBD,
     3        MBLOCK,NREADS,MAXMBK,TREAL,PDAMP,BETA,KRUN,
     4        NUMOUT,TMAX,TSTRSS,IOUT,JOUT
      DIMENSION STNPU(4,100),STNPW(4,100),STADU(4,100,8),STADW(4,100,8)
     1,NADJEL(100),NADJNP(100),NPADJ(100,8),NPOUT(100),NPTP(100)
      DIMENSION NPTN(1000),ANAME( 6),MADJNP(100),MADJEL(100),
     1MPADJ(100,8),TTNPU(4,100),TTNPW(4,100),TTADU(4,100,8),
     2TTADW(4,100,8)
C
C     READ CARD INPUT AND STRESS TABLES OFF BLOCKED STIFF. & STRESS TAPE
C
C     POSITION STRESS TAPE PAST TITLE RECORD
C
      REWIND 10
      READ(10)
C
C     POSITION STRESS TAPE PAST STIFF. TABLES AND NPTN RECORD
C
    1 READ(10) N1,N2
      IF(N2.LT.NUMNP) GO TO 1
      READ(10) (NPTN(I),I=I,NUMNP)
C
C     READ AND PRINT CARD INPUT
C
      READ(5,2)TSTRSS,TMAX,NUMOUT,IOUT,JOUT
      IF(TMAX.EQ.0.0) TMAX=1.0E+20
      IF(IOUT.EQ.0) IOUT=1
      IF(JOUT.EQ.0) JOUT=1
      READ(5,3) (NPOUT(I),I=1,NUMOUT)
    2 FORMAT(2E10.0,3I5)
    3 FORMAT(14I5)
C
C     TSTRSS=BEGINNING TIME FOR RECOMPUTING OUTPUT (SEC.)
C     TMAX  =MAXIMUM OUTPUT TIME (SEC.)
C     NUMOUT=NO. OF OUTPUT NODES (LE.100)
C     IOUT  =NO. OF INTEGRATION POINTS PER PRINTED OUTPUT POINT
C     JOUT  =NO. OF INTEGRATION POINTS PER TAPE OUTPUT POINT
C     NPOUT =OUTPUT NODE NUMBERS
C
      WRITE(6,4)TSTRSS,TMAX,NUMOUT,IOUT,JOUT
    4 FORMAT(19H1OUTPUT START TIME=,1PE15.5/
     1        18H OUTPUT STOP TIME=,1PE15.5/
     2        21H NO. OF OUTPUT NODES=,I5/
     3        25H PRINTED OUTPUT INTERVAL=,I5/
     4        22H TAPE OUTPUT INTERVAL=,I5///
     5        20H OUTPUT NODE NUMBERS)
      WRITE(6,5) (NPOUT(I),I=1,NUMOUT)
    5 FORMAT(1DX,5I5)
C
C     CONVERT TO NEW NUMBERING SYSTEM AND CHECK FOR ERRORS
C
      DO 6 I=1,NUMOUT
      NPOLD=NPOUT(I)
      DO 7 J=1,NUMNP
      JJ=J
      IF(NPTN(J).EQ.NPOLD) GO TO 8
    7 CONTINUE
      WRITE(6,9) NPOLD,(NPTN(J),J=I,NUMNP)
    9 FORMAT(1H1,14HERROR IN SETUP//1DX,(5I//(10X,15I5)))
      CALL EXIT
    8 NPTP(I)=JJ
    6 CONTINUE
C
C     SKIP FOR SUMMARY RERUNS
C
      IF((KRUN.EQ.3).OR.(KRUN.EQ.4)) GO TO 100
```

```
C
C     ZERO OUT STRESS TABLES
C
      DO 11 I=1,100
      NADJNP(I)=0
      MADJNP(I)=0
      NADJEL(I)=0
      MADJEL(I)=0
      DO 12 J=1,8
      NPADJ(I,J)=0
   12 MPADJ(I,J)=0
      DO 11 K=1,4
      STNPU(K,I)=0.0
      TTNPU(K,I)=0.0
      STNPW(K,I)=0.0
      TTNPW(K,I)=0.0
      DO 11 J=1,8
      STADU(K,I,J)=0.0
      TTADU(K,I,J)=0.0
      STADW(K,I,J)=0.0
      TTADW(K,I,J)=0.0
   11 CONTINUE
C
C     READ STRESS TABLES FOR OUTPUT NODES
C
      NOUT=0
   16 READ(10) NPLOW,NPHIGH,              NUMNPR,(NADJNP(I),MADJEL(I),
     1((NPADJ(I,J),J=1,MXADJP),(TTNPU(K,I),TTNPW(K,I),K=1,4),
     2((TTADU(K,I,J),TTADW(K,I,J),K=1,4),J=1,MXADJP),I=1,NUMNPR)
      NPOFF=NPLOW-1
C
      DO 13  I=1 ,NUMOUT
      NPOLD=NPOUT(I)
      NPNEW=NPTP(I)
      IF((NPNEW.LT.NPLOW).OR.(NPNEW.GT.NPHIGH)) GO TO 13
      NOUT=NOUT+1
      NP=NPNEW-NPOFF
      NADJNP(I)=NADJNP(NP)
      NADJEL(I)=NADJEL(NP)
      DO 14 J=1,MXADJP
   14 NPADJ(I,J)=MPADJ(NP,J)
      DO 15 K=1,4
      STNPU(K,I)=TTNPU(K,NP)
      STNPW(K,I)=TTNPW(K,NP)
      DO 15 J=1,MXADJP
      STADU(K,I,J)=TTADU(K,NP,J)
      STADW(K,I,J)=TTADW(K,NP,J)
   15 CONTINUE
   13 CONTINUE
C
      IF(NOUT.GE.NUMOUT) GO TO 100
      IF(NPHIGH.LT.NUMNP) GO TO 16
      WRITE(6,17) NUMOUT,NOUT
   17 FORMAT(1HI,14HERROR IN LNK3A//10X,7HNUMOUT=,I5/10X,7HNOUT  =,I5)
      CALL EXIT
  100 REWIND 10
      RETURN
      END
```



```
      SUBROUTINE LINK3B(UDDN,WDDN,UDN,WDN,UN,WN,PRESSU,PRESSW,STNPU,
     1STNPW,STADU,STADW,NADJEL,NADJNP,NPADJ,NPOUT,NPTP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON MAXNP,MXADJP,MXLINE,MXLOAD,MXNPB,MAXELB,NUMNP,NUMEL,
     1        ISTRES,LINES,NUMPEL,LOADNP(I),PERIOD,DT,NPLOAD(1,50),      3/4/72
     2        RAD(1,50),ZAD(1,50),SNORM(I,50),TITLE(10),MAXBO,           3/4/72
     3        MBLOCK,NREADS,MAXMBK,TREAL,PDAMP,BETA,KRUN,
     4        NUMOUT,TMAX,TSTRSS,IOUT,JOUT
      DIMENSION UDDN(1000),WDDN(1000),UDN(1000),WDN(1000),UN(1000),
     1WN(1000),SIGMAU(1,50),SIGMAW(1,50)
      DIMENSION STNPU(4,100),STNPW(4,100),STADU(4,100,8),STADW(4,100,8)
     1,NADJEL(100),NADJNP(100),NPADJ(100,8),NPOUT(100),NPTP(100)
      DIMENSION SIG(4),NDOFEL(26),NP(26,4),EPSP(26,4),C(26,4,4),
     1SIGPL(100,4),RUFF(1600)
C
C     CALCULATE STRESSES AND WRITE THE 8TH TAPE
C
C     WRITE TITLE RECORD ON 8TH TAPE
C
      REWIND 8
      WRITE(8) TITLE
C
C     INITIALIZE AND WRITE FIRST RECORD ON 8TH TAPE
C
      ICOUNT=1
      NOWON=1
      JC=JOUT
      IC=IOUT
      ET=DT*FLOAT(JOUT)
      NWDS=16+NUMOUT
      WRITE(8      ) TMAX,ET,TREAL,NUMOUT,(NPOUT(I),I=1,NUMOUT)
      DO 2 I=1,4
    2 SIG(I)=0.0
      SIGMX=0.0
      SIGMN=0.0
      THETA=0.0
C
C     POSITION 8TH TAPE AND PRINT START TIME
C
      IF(TSTRSS.GT.(0.5*DT)) GO TO 3
      T=-DT
      GO TO 7
    3 CONTINUE
      IF(NREADS.EQ.1) GO TO 4
      IDUM=NREADS-1
      DO 5 I=1,IDUM
    5 READ(8)NUMELB
    4 READ(8)T
      IF((TSTRSS-T).GT.(1.5*DT)) GO TO 3
    7 T=T+TREAL+DT
      TSTART=T+DT
      WRITE(6,6)TSTART,TM
```

```
    6 FORMAT(///35H OUTPUT HISTORY TAPE STARTS AT TIME ,
     (1PE15.5,5H SEC./36H WHICH CORRESPONDS TO A REAL TIME OF,
     21PE15.5,5H SEC.)
C
C     MAIN LOOP FOR EACH TIME STEP ON OTH TAPE STARTS HERE
C
  400 CONTINUE
C
C     INCREASE COUNTERS, CHECK, AND SKIP A TIME STEP ON OTH TAPE
C     IF NECESSARY
C
      IC=IC+1
      JC=JC+1
      IF((IC.GT.IOUT).OR.(JC.GT.JOUT)) GO TO 401
      DO 402 I=1,NREADS
  402 READ(8,END=999,ERR=999)DUMMY
      GO TO 400
C
C     SKIP FOR SUMMARY OUTPUT
C
  401 IF((KRUN.EQ.3).OR.(KRUN.EQ.4)) GO TO 100
C
C     ZERO OUT PLASTIC STRESS ARRAY
C
      DO 8 I=1,NUMOUT
      DO 8 J=1,4
    8 S(GPL(I,J)=0.0
C
C     SKIP FOR NO PLASTIC ELEMENTS
C
      IF(NUMPEL.EQ.0) GO TO 300
C
C     CALCULATE PLASTIC STRESS TABLES
C
      NUMCEL=0
  209 READ(8,END=999,ERR=999)NUMELR,(NODELI(I),(NP(I,J),EPSP(I,J),
     1IC(I,J,K),K=1,4),J=1,4),I=1,NUMELR)
C
      DO 200 I=1,NUMELR
      II=0
      JJ=0
      KK=0
      LL=0
C
      DO 201 J=1,NUMOUT
      IF(NPTP(J).EQ.NP(I,1)) II=J
      IF(NPTP(J).EQ.NP(I,2)) JJ=J
      IF(NPTP(J).EQ.NP(I,3)) KK=J
      IF(NPTP(J).EQ.NP(I,4)) LL=J
  201 CONTINUE
C
      MM=MAXO(II,JJ,KK,LL)
      IF(MM.EQ.0) GO TO 200
C
      DO 202 J=1,4
      S(G(J)=0.0
      DO 202 K=1,4
  202 SIG(J)=S(G(J)+C(I,J,K)*EPSP(I,K)
C
      ISWTCH=1
      N=II
  205 IF(N.EQ.0) GO TO 204
      DO 203 J=1,4
  203 S(GPL(N,J)=S(GPL(N,J)+SIG(J)
  204 GO TO (206,207,208,200),ISWTCH
C
  206 ISWTCH=2
      N=JJ
      GO TO 205
  207 ISWTCH=3
      N=KK
      GO TO 205
  208 ISWTCH=4
      N=LL
      GO TO 205
C
  200 CONTINUE
C
      NUMCEL=NUMCEL+NUMELR
      IF(NUMCEL.LT.NUMPEL) GO TO 209
C
      DO 210 I=1,NUMOUT
      DUM=NADJEL(I)
      DO 210 J=1,4
  210 SIGPL(I,J)=SIGPL(I,J)/DUM
      GO TO 300
C
C     SKIP PAST PLASTIC DATA ON OTH TAPE
C
  100 IF(NREADS.EQ.1) GO TO 300
      IDUM=NREADS-1
      DO 101 I=1,IDUM
  101 READ(8,END=999,ERR=999)NUMELR
C
C     READ A TIME STEP OFF OTH TAPE
C
  300 IDUM=LOADNP(I)
      READ(   8,END=999,ERR=999)T,KOUNT,TM,(UN(I),UDN(I),UDDN(I),WN(I),
     1WDN(I),WDDN(I),(I=1,NUMNP),(SIGMAU(I,J),SIGMAW(I,J),I=1,IDUM)
C
C     SKIP FOR NO PRINTED OUTPUT
C
      IF(IC.LE.IOUT) GO TO 351
C
C     PRINT CORRECT HEADING (COMPLETE OR SUMMARY OUTPUT)
C
      WRITE(6,350) T,KOUNT,TM
  350 FORMAT(1H1,5HTIME=,1PE15.5,6H SEC.,,5X,6HKOUNT=,I5,5X,
     110HREAL TIME=,1PE15.5,5H SEC.//)
      IF((KRUN.EQ.3).OR.(KRUN.EQ.4)) GO TO 352
      WRITE(6,353)
  353 FORMAT(2X,4HNODE,4X,12HDISP. U (IN),2X,12HVEL.UD (IPS),2X,
```

```
      PW=SIGMAW(J,K)
      GO TO 310
  309 CONTINUE
      PU=0.0
      PW=0.0
C
C     CONVERT ACCELERATIONS TO G'S
C
  310 ACCELU=UDDN(NPNEW)/386.4
      ACCELW=WDDN(NPNEW)/386.4
C
C     SKIP FOR NO PRINTED OUTPUT
C
      IF(IC.LE.IOUT) GO TO 311
C
C     PRINT OUTPUT (COMPLETE OR SUMMARY)
C
      IF((KRUN.EQ.3).OR.(KRUN.EQ.4)) GO TO 355
      WRITE(6,356) NPOUT(I),UN(NPNEW),UDN(NPNEW),ACCELU,PU,SIG(1),
     1SIG(3),SIGMX,THETA,WN(NPNEW),WDN(NPNEW),ACCELW,PW,SIG(2),SIG(4)
     2SIGMN
  356 FORMAT(I5,3X,1P8E14.4/8X,1P7E14.4//)
      GO TO 311
  355 WRITE(6,357) NPOUT(I),UN(NPNEW),UDN(NPNEW),ACCELU,PU,WN(NPNEW),
     1WDN(NPNEW),ACCELW,PW
  357 FORMAT(I5,3X,1P8E14.4)
C
C     LOAD BUFFER FOR TAPE OUTPUT
C
  311 IF(JC.LE.JOUT) GO TO 301
      BUFF(IK   )=NPOUT(I)
      BUFF(IK+1 )=UN(NPNEW)
      BUFF(IK+2 )=UDN(NPNEW)
      BUFF(IK+3 )=ACCELU
      BUFF(IK+4 )=PU
      BUFF(IK+5 )=SIG(1)
      BUFF(IK+6 )=SIG(3)
      BUFF(IK+7 )=SIGMX
      BUFF(IK+8 )=THETA
      BUFF(IK+9 )=WN(NPNEW)
      BUFF(IK+10)=WDN(NPNEW)
      BUFF(IK+11)=ACCELW
      BUFF(IK+12)=PW
      BUFF(IK+13)=SIG(2)
      BUFF(IK+14)=SIG(4)
      BUFF(IK+15)=SIGMN
      IK=IK+16
C
C     WRITE TAPE IF TAPE OUTPUT THIS STEP
C
  301 CONTINUE
      IF(JC.LE.JOUT) GO TO 403
      WRITE(3      ) T,(BUFF(K),K=1,NWDS)
C
C     SET SWITCHES, LOOP BACK, OR WRAP UP
C
```

```
     112HACCL.UDD (G),2X,12HSIGMAU (PSI),2X,12HSIGMAR (PSI),2X,
     212HSIGMAZ (PSI),2X,12HSIGMX  (PSI),3X,11HTHETA (DEG)/10X,
     312HDISP. W (IN),2X,12HVEL.WD (IPS),2X,12HACCL.WDD (G),2X,
     412HSIGMAW (PSI),2X,12HSIGMAT (PSI),2X,12H      (PSI),2X,
     512HSIGMN  (PSI)//)
      GO TO 351
  352 WRITE(6,354)
  354 FORMAT(2X,4HNODE,4X,12HDISP. U (IN),2X,12HVEL.UD (IPS),2X,
     112HACCL.UDD (G),2X,12HSIGMAU (PSI),2X,12HDISP. W (IN),2X,
     212HVEL.WD (IPS),2X,12HACCL.WDD (G),2X,12HSIGMAW (PSI)//)
C
C     SET-UP OUTPUT FOR EACH OUTPUT NODE
C
  351 IK=1
      DO 301 I=1,NUMOUT
      NPNEW=NPTP(I)
C
C     SKIP STRESS CALCULATION FOR SUMMARY OUTPUT
C
      IF((KRUN.EQ.3).OR.(KRUN.EQ.4)) GO TO 320
C
C     CALCULATE STRESSES
C
      NUM=NADJNP(I)
      DO 302 J=1,4
      SIG(J)=STNPU(J,I)*UN(NPNEW)+STNPW(J,I)*WN(NPNEW)
      DO 303 K=1,NUM
      NODE=NPADJ(I,K)
  303 SIG(J)=SIG(J)+STADU(J,I,K)*UN(NODE)+STADW(J,I,K)*WN(NODE)
      SIG(J)=SIG(J)-SIGPL(I,J)
  302 CONTINUE
C
C     CALCULATE PRINCIPAL STRESSES AND ANGLE
C
      DUM1=(SIG(1)-SIG(3))/2.
      DUM2=DUM1*DUM1+SIG(4)*SIG(4)
      IF(DUM2.GT.0.0) GO TO 321
      RADIUS=0.0
      GO TO 322
  321 RADIUS=DSQRT(DUM2)
  322 DUM3=(SIG(1)+SIG(3))/2.
      FIG1=DUM3+RADIUS
      FIG2=DUM3-RADIUS
      SIGMX=FIG2
      IF(DABS(FIG1).GT.DABS(FIG2)) SIGMX=FIG1
      SIGMN=FIG1+FIG2-SIGMX
      CALL ARCTAN(SIGMX-SIG(3),SIG(4),THETA)
C
C     SET-UP BOUNDARY STRESSES
C
  320 DO 309 J=1,LINES
      NUM=LOADNP(J)
      DO 309 K=1,NUM
      NODE=NPLOAD(J,K)
      IF(NODE.NE.NPTP(I)) GO TO 309
      PU=SIGMAU(J,K)
```

```
  403 IF(IC.GT.IOUT) IC=1
      IF(JC.GT.JOUT) JC=1
      GO TO 400
  999 CONTINUE
C
      END FILE 3
      REWIND 3
      RETURN
      END


      SUBROUTINE ARCTAN(X,Y,THETA)                                    00011150
      IMPLICIT REAL*8(A-H,O-Z)                                        00011200
      DATA PI/3.14159265D0/
C                                                                     00011250
C        GET ARCTANGENT IN CORRECT QUADRANT BETWEEN                   00011300
C        +PI AND -PI                                                  00011350
C                                                                     00011400
      IF(X)50,10,50                                                   00011450
10    IF(Y)40,30,20                                                   00011500
20    THETA=PI/2.0
      RETURN                                                          00011600
30    THETA=0.0D0                                                     00011650
      RETURN                                                          00011700
40    THETA=-PI/2.0
      RETURN                                                          00011800
50    THETA=DATAN(DABS(Y/X))                                          00011850
      IF(X)90,60,60                                                   00011900
60    IF(Y)70,80,80                                                   00011950
70    THETA=-THETA                                                    00012000
80    RETURN                                                          00012050
90    IF(Y)110,100,100                                                00012100
100   THETA=PI-THETA
      RETURN                                                          00012200
110   THETA=-PI+THETA
      RETURN                                                          00012300
      END                                                             00012350
```

APPENDIX A1. (MINBND) SOURCE LISTING.

```
      SUBROUTINE MINBND(IROW,ICOL,NUMN,IS,JS,MBW)
      DIMENSION IROW(1),ICOL(1),NAP(8),IS(1),JS(1)
C
C  MATRIX BANDWIDTH MINIMIZATION  - -  KUSEN (1968)
C
      NCP=0
      REWIND 8
      DO 20 I=1,NUMN
      READ (8) IS(I),NAI,(NAP(K),K=1,NAI)
      DO 20 J=1,NAI
      IF(I-NAP(J))10,20,20
   10 NCP=NCP+1
      ICOL(NCP)=I
      IROW(NCP)=NAP(J)
   20 CONTINUE
      REWIND 8
      CALL NUMBME(IROW,ICOL,IS,NCP,NUMN,MBW)
      DO 30 I=1,NUMN
      K=IS(I)
   30 JS(K)=I
      RETURN
      END
      SUBROUTINE NUMBME(IROW,ICOL,IS,NCP,NSIZE,MAXBND)
      DIMENSION IROW(1),ICOL(1),IS(1),INTPH2(2002)
      MBDPRV=NSIZE
      MAXPH2=NSIZE
    1 MAXBND=0
      DO 2 I=1,NCP
      JDIFF=IROW(I)-ICOL(I)
      IF(JDIFF-MAXBND)2,2,1000
 1000 MAXBND=JDIFF
      IRMXBD=IROW(I)
      ICMXBD=ICOL(I)
    2 CONTINUE
      PRINT 6000,  MAXBND
 6000 FORMAT(16H MAXBANDWIDTH = ,I5)
      IF(MAXBND-1)1001,1004,1001
 1001 ISTART=XMAXOF(1,(2*ICMXBD-IRMXBD+1))
      ISTOP=IRMXBD-1
      JJ=IRMXBD
      DO 3 I=ISTART,ISTOP
      IF(I-ICMXBD)1002,3,1002
 1002 II=I
      CALL SEARCH(IROW,ICOL,IRMXBD,I,NCP,NFOUND,MAXBND)
      IF(NFOUND-1)3,6,3
    3 CONTINUE
      ISTART=ICMXBD+1
      ISTOP=XMINOF(NSIZE,(2*IRMXBD-ICMXBD-1))
      JJ=ICMXBD
      DO 4 I=ISTART,ISTOP
      IF(I-IRMXBD)1003,4,1003
 1003 II=I
      CALL SEARCH(IROW,ICOL,ICMXBD,I,NCP,NFOUND,MAXBND)
      IF(NFOUND-1)4,6,4
    4 CONTINUE
 1022 IF(MAXBND-MBDPRV)1010,7,1010
 1010 IEND=2*MAXPH2
      DO 8 I=1,IEND
    8 INTPH2(I)=0
      NTPHS2=0
      MBDPRV=MAXBND
      GO TO 9
    7 IF (NTPHS2-MAXPH2)9,1004,9
 1004 RETURN
    9 CONTINUE
      N2INT=2*NTPHS2
      NTPHS2=NTPHS2+1
      ISTART=XMAXOF(1,(2*ICMXBD-IRMXBD))
      ISTOP=IRMXBD-1
      JJ=IRMXBD
      DO 31 I=ISTART,ISTOP
      IF(I-ICMXBD)1005,31,1005
 1005 II=I
      CALL P2SRCH(IROW,ICOL,IRMXBD,I,NCP,NFOUND,MAXBND,INTPH2,N2INT
      IF(NFOUND-1)31,6,31
   31 CONTINUE
      ISTART=ICMXBD+1
      ISTOP=XMINOF(NSIZE,(2*IRMXBD-ICMXBD))
      JJ=ICMXBD
      DO 44 I=ISTART,ISTOP
      IF(I-IRMXBD)1006,44,1006
 1006 II=I
      CALL P2SRCH(IROW,ICOL,ICMXBD,I,NCP,NFOUND,MAXBND,INTPH2,N2INT
      IF(NFOUND-1)44,6,44
   44 CONTINUE
      RETURN
    6 CALL INTRCH(IROW,ICOL,JJ,II,NCP)
      M1=JJ
      M2=II
      DO 37 I=1,NSIZE
      IF(IS(I)-M1)1008,1007,1008
 1007 J1=I
 1008 IF(IS(I)-M2)37,1009,37
 1009 J2=I
   37 CONTINUE
      IS(J1)=M2
      IS(J2)=M1
      GO TO 1
      END
      SUBROUTINE INTRCH(IROW,ICOL,IIS,IWAS,NCP)
      DIMENSION IROW(1),ICOL(1)
      DO 1 I=1,NCP
      IF(IROW(I)-IIS)2,5000,2
 5000 CALL SWITCH(IROW(I),ICOL(I),IWAS,JH,JL)
      IROW(I)=JH
      ICOL(I)=JL
      GO TO 1
    2 IF(IROW(I)-IWAS)3,5001,3
 5001 CALL SWITCH(IROW(I),ICOL(I),IIS,JH,JL)
      IROW(I)=JH
      ICOL(I)=JL
      GO TO 1
    3 IF(ICOL(I)-IIS)4,5002,4
 5002 CALL SWITCH(ICOL(I),IROW(I),IWAS,JH,JL)
      IROW(I)=JH
      ICOL(I)=JL
      GO TO 1
    4 IF(ICOL(I)-IWAS)1,5003,1
 5003 CALL SWITCH(ICOL(I),IROW(I),IIS,JH,JL)
      IROW(I)=JH
```

```
      ICOL(I)=JL
    1 CONTINUE
      RETURN
      END
      SUBROUTINE P2SRCH(IROW,ICOL,IC,JC,NCP,NF,MAXBND,INTPH2,N2INT)
      DIMENSION IROW(1),ICOL(1),INTPH2(1)
      NF=0
      DO 1 I=1,NCP
      IF(IROW(I)-IC)2,4000,2
 4000 ICAN2=XABSF(JC-ICOL(I))
      IF(ICAN2-MAXBND)1,1,4001
 4001 RETURN
    2 IF(ICOL(I)-JC)3,4002,3
 4002 NOCN2=XABSF(IC-IROW(I))
      IF(NOCN2-MAXBND)1,1,4001
    3 IF(IROW(I)-JC)4,4003,4
 4003 IDO2=XABSF(IC-ICOL(I))
      IF(IDO2-MAXBND)1,1,4001
    4 IF(ICOL(I)-IC)1,4004,1
 4004 NODO2=XABSF(JC-IROW(I))
      IF(NODO2-MAXBND)1,1,4001
    1 CONTINUE
      IF(N2INT)4005,5,4005
 4005 MTI=0
      MTJ=0
      DO 6 I=1,N2INT
      IF(INTPH2(I)-IC)61,4006,61
 4006 MTI=1
   61 IF(INTPH2(I)-JC)6,4007,6
 4007 MTJ=1
    6 CONTINUE
 4008 IF(MTI+MTJ-2)5,4001,5
    5 CONTINUE
      NF=1
      INTPH2(N2INT+1)=IC
      INTPH2(N2INT+2)=JC
      RETURN
      END
      SUBROUTINE SEARCH(IROW,ICOL,IC,JC,NCP,NF,MAXBND)
      DIMENSION IROW(1),ICOL(1)
      NF=0
      DO 1 I=1,NCP
      IF(IROW(I)-IC)2,3000,2
 3000 ICAN=XABSF(JC-ICOL(I))
      IF(ICAN-MAXBND)1,3001,3001
 3001 RETURN
    2 IF(ICOL(I)-JC)3,3002,3
 3002 NOCAN=XABSF(IC-IROW(I))
      IF(NOCAN-MAXBND)1,3001,3001
    3 IF(IROW(I)-JC)4,3003,4
 3003 IDO=XABSF(IC-ICOL(I))
      IF(IDO-MAXBND)1,3001,3001
    4 IF(ICOL(I)-IC)1,3004,1
 3004 NODO=XABSF(JC-IROW(I))
      IF(NODO-MAXBND)1,3001,3001
    1 CONTINUE
      NF=1
      RETURN
      END
      SUBROUTINE SWITCH(ISW,ICH,JST,JMIGH,JLOW)
      JSW=JST
      IF(ICH-JSW)2001,2000,2001
 2000 JSW=ISW
 2001 JMIGH=XMAXOF(ICH,JSW)
      JLOW=XMINOF(ICH,JSW)
      RETURN
      END
```

APPENDIX B. ØS/360 SLAMCODE LINK EDITOR CONTROL CARDS.

```
INSERT MAIN
OVERLAY ALPHA
INSERT LINK1
OVERLAY BETA
INSERT LINK1A,LOADIT,LNK1A1,ORDER,ADJNP,VADJNP,PATH,MINI,GSORTS,SORTI
OVERLAY BETA
INSERT LINK1B,ELAST,STIFF,INS,DISTR,ADJUST,MASS,PRAK
OVERLAY BETA
INSERT LINK1C,STRESS,MODS,AVG
OVERLAY BETA
INSERT LINK1D,SIZE,CLSTER
OVERLAY BETA
INSERT LINK1F,ELUST
OVERLAY ALPHA
INSERT LINK2,MMT,ERASE
OVERLAY GAMMA
INSERT LINK2A,DISKER,ESC
OVERLAY GAMMA
INSERT LINK2B,MSTRES
OVERLAY ALPHA
INSERT LINK3
OVERLAY RHO
INSERT LINK3A
OVERLAY RHO
INSERT LINK3B,ARCTAN
ENTRY MAIN
```

APPENDIX C. ØS/360 FORTRAN H SLAMCODE JOB CONTROL LANGUAGE CARDS.

a.) Requires two 9-track tape drives.

```
/*LOG.
//* EXEC FMLG,PARM.OBJECT='LIST,MAP,NOLY'
//OBJECT.INPUT DD UNIT=2400,VOL=SER=GPIZHM,DSN=SLAMIN
//DATA.FT01F001 DD UNIT=SYSDA,SPACE=(364,125),                          X
//                 DCB=(RECFM=VBS,LRECL=32,BLKSIZE=364)
//DATA.FT03F001 DD UNIT=(SYSDA,SEP=FT01F001),SPACE=(CYL,10),            X
//                 DISP=(NEW,PASS),DSN=&&OUTTAP,                        X
//                 DCB=(RECFM=VS,LRECL=3204,BLKSIZE=3208)
//DATA.FT04F001 DD UNIT=(SYSDA,SEP=(FT01F001,FT03F001)),                X
//                 SPACE=(364,100),                                     X
//                 DCB=(RECFM=VBS,LRECL=36,BLKSIZE=364)
//DATA.FT08F001 DD UNIT=(SYSDA,SEP=(FT01F001,FT03F001)),                X
//                 SPACE=(484,100),DCB=(RECFM=VBS,LRECL=48,BLKSIZE=484)
//DATA.FT10F001 DD UNIT=2400,VOL=SER=GPIZHM,DSN=STIFF1,LABEL=1,         X
//                 DCB=(RECFM=VS,LRECL=2052,BLKSIZE=2056)
//DATA.FT12F001 DD UNIT=(SYSDA,SEP=(FT03F001,FT04F001)),                X
//                 SPACE=(1804,125),DCB=(RECFM=VBS,LRECL=180,BLKSIZE=1804)
//DATA.FT14F001 DD DUMMY
//DATA.FT20F001 DD UNIT=SYSDA,SPACE=(CYL,20),DSN=&&OUTTAPE,             X
//                 DCB=(RECFM=VS,LRECL=2052,BLKSIZE=2056)
//DATA.FT22F001 DD UNIT=SYSDA,SPACE=(CYL,5)
//DATA.INPUT DD *
```

A-1.4

APPENDIX C. ØS/360 FORTRAN H SLAMCODE JOB CONTROL LANGUAGE CARDS.

b.) Requires four 9-track tape drives.

```
/*LOG
// EXEC FHLG,PARM.OBJECT='LIST,MAP,NOLET'
//OBJECT.INPUT DD UNIT=2400,VOL=SER=[illegible],DSN=SLAMCODE
//DATA.FT01F001 DD UNIT=SYSDA,SPACE=(364,120),                          X
//                 DCB=(RECFM=VBS,LRECL=36,BLKSIZE=364)
//DATA.FT03F001 DD UNIT=2400,VOL=SER=[illegible],DSN=OUTTAPE,           X
//                 DCB=(RECFM=VS,LRECL=2052,BLKSIZE=2056)
//DATA.FT04F001 DD UNIT=(SYSDA,SEP=(FT01F001,FT03F001)),                X
//                 SPACE=(364,100),                                     X
//                 DCB=(RECFM=VBS,LRECL=36,BLKSIZE=364)
//DATA.FT08F001 DD UNIT=2400,VOL=SER=GP12[illegible],DSN=OUTTAPE,       X
//                 DCB=(RECFM=VS,LRECL=2052,BLKSIZE=2056)
//DATA.FT10F001 DD UNIT=2400,VOL=SER=GP12[illegible],DSN=STIFF1,LABEL=1,X
//                 DCB=(RECFM=VS,LRECL=2052,BLKSIZE=2056)
//DATA.FT12F001 DD UNIT=(SYSDA,SEP=(FT03F001,FT04F001)),                X
//                 SPACE=(1804,120),DCB=(RECFM=VBS,LRECL=100,BLKSIZE=1804)
//DATA.FT14F001 DD DUMMY
//DATA.FT20F001 DD UNIT=SYSDA,SPACE=(CYL,20),DSN=&&OUTTAPE,             X
//                 DCB=(RECFM=VS,LRECL=2052,BLKSIZE=2056)
//DATA.FT22F001 DD UNIT=SYSDA,SPACE=(CYL,5)
//DATA.INPUT DD *
```

A-1