AD-A043 582 CIVIL ENGINEERING LAB (NAVY) PORT HUENEME CALIF F/G 6/2
A FINITE ELEMENT HEAD INJURY MODEL. VOLUME II. COMPUTER PROGRAM--ETC(U)
JUL 77 T A SHUGAR

UNCLASSIFIED CEL-R-854-2 NL

1 OF 2
AD
A043582

AD A043582

12
NA

Technical Report

R 854-11

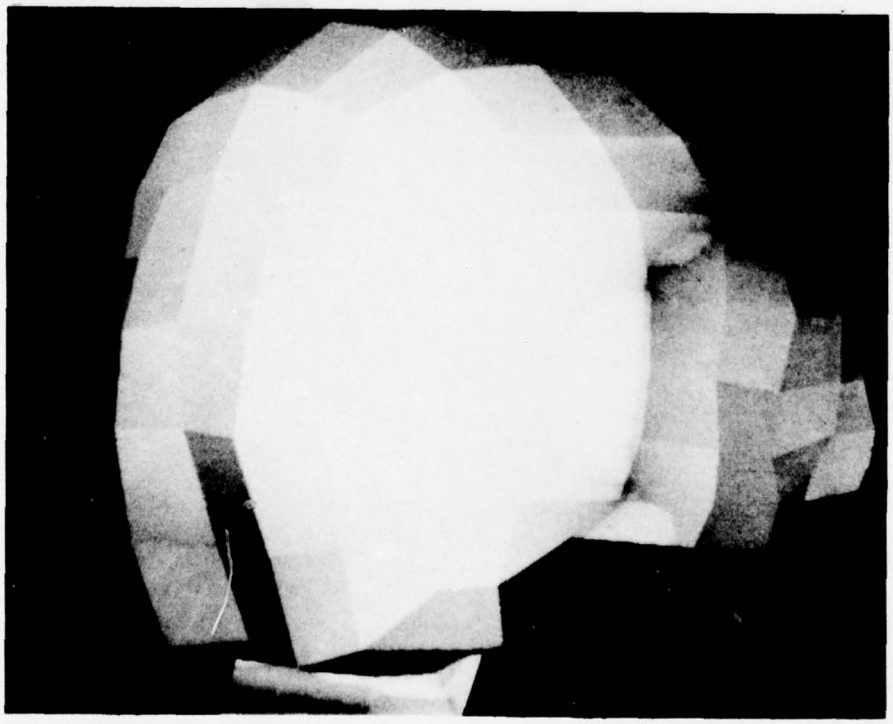
COPY AVAILABLE TO DDC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

Sponsored by

DEPARTMENT OF TRANSPORTATION,
NAVAL AEROSPACE MEDICAL RESEARCH LABORATORY,
and OFFICE OF NAVAL RESEARCH

July 1977

CIVIL ENGINEERING LABORATORY
Naval Construction Battalion Center
Port Hueneme, California 93043



A FINITE ELEMENT HEAD INJURY MODEL – Volume II: Computer Program Documentation

by T. A. Shugar

Approved for public release; distribution unlimited.

AD No.
DDC FILE COPY

DDC
RECEIVED
AUG 31 1977
C

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CEL-R-854-II-2	2. GOVT ACCESSION NO.: DN 244152
3. RECIPIENT'S CATALOG NUMBER: Dept.	
4. TITLE (and Subtitle): A FINITE ELEMENT HEAD INJURY MODEL • Volume II. Computer Program Documentation.	5. TYPE OF REPORT & PERIOD COVERED: Final June 1973 — June 1977
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): T. A. Shugar	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: CIVIL ENGINEERING LABORATORY, Naval Construction Battalion Center, Port Hueneme, California 93043	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: DOT-HS-289-3-550-IA; ONR-61153N, RR023-03-01; NAMRL N00203-75-WR-00142
11. CONTROLLING OFFICE NAME AND ADDRESS: Department of Transportation, Naval Aerospace Medical Research Laboratory, and Office of Naval Research	12. REPORT DATE: July 1977
	13. NUMBER OF PAGES: 158
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 153 p.	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

RR023 03

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

RR023 03 01

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Biomechanics, structural models, head injury prediction, impact, dynamics, transients, finite element.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Volume II contains necessary information and documentation for executing the HIM computer program. Documentation includes a user's manual, a flow chart, CDC 6600 control cards, sample input data, and a FORTRAN IV source code listing of the HIM program. In addition, listings are provided for a preprocessor (skull mesh generator), a bandwidth minimizer, and a subroutine for an improved finite element for simulating the load-deformation response of the skull.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DDC RECEIVED AUG 31 1977

391 111

Library Card

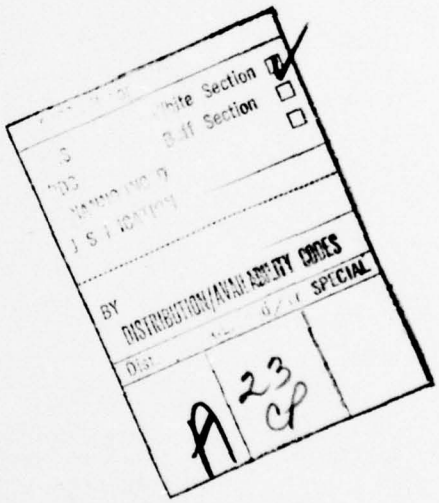
Civil Engineering Laboratory
A FINITE ELEMENT HEAD INJURY MODEL – VOLUME II:
Computer Program Documentation (Final) by T. A. Shugar
TR-854 – II 158 pp illus July 1977 Unclassified

1. Biomechanics 2. Head injury prediction I. DOT-HS-289-3-550

Volume II contains necessary information and documentation for executing the *HIM* computer program. Documentation includes a user's manual, a flow chart, CDC 6600 control cards, sample input data, and a FORTRAN IV source code listing of the HIM program. In addition, listings are provided for a preprocessor (skull mesh generator), a bandwidth minimizer, and a subroutine for an improved finite element for simulating the load-deformation response of the skull.

Volume II of this report contains Appendixes B, C, D, E, F, and G HIM Code documentation. Appendix A can be found in Volume I.

Appendix	Title	Page
B	Summary of FEAP Modifications for Improved Efficiency	1
C	User Instructions for the HIM Code	2
D	Flow Diagram for Main Subroutines	23
E	Sample Job Control Cards	25
F	HIM Code Listing	27
G	Sample Input Data	99
H	Other Associated Code Listings	127



Appendix B

SUMMARY OF FEAP MODIFICATIONS FOR IMPROVED EFFICIENCY

The major changes made to the FEAP program are listed below:

1. The entire program was recompiled using FTN (OPT=2). Several FORTRAN errors were found and corrected.

2. The PUNCH, PUNCHB, TAPE8 and TAPE12 files were removed. Buffer sizes were increased to 3075 for TAPE7 and TAPE10 and to 2050 for TAPE13 and TAPE14. TAPE11 becomes a direct access (QUICKIO) file with a buffer size of 65 words. QUICKIO is a FORTRAN callable subroutine which decreases I/O time.

3. All files were binary blocked using FTNBIN.

4. The global stiffness matrix is saved on a direct-access file in subroutine MAKEA. During forward reduction in USOL, no additional files are used for spill-over processing. All intermediate storage required is written in-place as an appropriate record on the QUICKIO file. This reduces the appropriate record on the QUICKIO file, which in turn reduces the amount of scratch storage required by more than *half*.

5. USOL was completely rewritten to allow direct-access equation processing. A COMPASS routine was included to increase the solution efficiency.

6. RESOLVE was rewritten to accommodate direct-access processing during the back-substitution phase.

7. Subroutines PRTMAT, COMPACT, CKQUAD, PACKD, SLD04, and QSHP8 were removed.

8. Subroutines FELMT, MESHCK, TSOLVE, and RESET were modified to accommodate the node resequencing (bandwidth minimizing) changes. Primarily, this involved adding named COMMON/MINBW/NEW(1500) and internally altering nodal references. Note that fixing NEW at 1,500 words inherently limits the program to 1500 nodes maximum. This COMMON would have to be extended if larger problems are anticipated.

9. Two input changes were made, neither of which affect the execution of existing decks; i.e., no actual changes were made to input processing unless the new features are desired. An optional first card may be included with the entry ISZDT. Blank COMMON will be set to this value allowing execution at reduced core size for smaller problems. The entry is read using I5 format. The default value is 58000. The second option is activated by entering "1" in cc 79 of the FEAP card. If activated, the program expects to encounter the PREFEAP punched output immediately following the FEAP card. PREFEAP punched output contains the optionally reordered nodal point numbering system.

Appendix C

USER INSTRUCTIONS FOR THE HIM CODE

1 CONTROL CARDS

Card 1. (I5)

Columns	Variable	Description	Note
1-5	ISZDT	Blank COMMON will be set to this value. Default value is 58000	(1)

Card 2. (6X, 12A6)

Columns	Variable	Description	Note
7-10	—	Must contain the word FEAP 73	
13-78	—	Output page header	(2)
79	—	Enter 1 if optimized nodal point numbering is employed. Otherwise leave blank	(3)

Card Group 2a. If Card 2 contains a 1 in cc 79, the program expects to encounter a group of cards specifying an optimally reordered nodal point numbering system prior to encountering Card 3.

Card 3. (I5,1X,3A6)

Columns	Variable	Description	Note
1-5	NOIM	Spatial dimension of problem. Use 3	
7-12	—	X coordinate designation	
13-18	—	Y coordinate designation	
19-24	—	Z coordinate designation	

Card 4. (I5,1X,3A6)

Columns	Variable	Description	Note
1-5	NDF	Number of unknown displacements per node. Use 3.	
7-12	—	X displacement designation	
13-18	—	Y displacement designation	
19-24	—	Z displacement designation	

Card 5. (I5,10X,I5,30X,3F10.0)

Columns	Variable	Description	Note
1-5	NEN	Maximum number of nodes connected to any one element. Use 8.	
16-20	MBAN	Maximum expected half-bandwidth	(4)
51-60	CON3 (1)	User defined X scale factor	(5)
61-70	CON3 (2)	User defined Y scale factor	
71-80	CON3 (3)	User defined Z scale factor	

Notes

(1) For typical HIM code simulations this card can be left blank thereby specifying 58000 as the blank COMMON allocation.

(2) Any description that the user wishes to have printed at the top of each output page can be entered.

(3) Large, three-dimensional, dynamic problems can be expensive. A bandwidth minimizer has been employed to resequence the nodal point numbering scheme and optimally reorder the system of equations to provide economy in computational costs. A set of cards has been provided for this purpose which map the old and new numbering schemes. The mapping is transparent to the user; he refers always to the old system. Use of this option is highly recommended.

(4) Typically this value is 150. The default value is 100 and is not sufficient. Therefore 150 should be specified.

(5) These scale factors can be used to alter the size and shape of the HIM skull configuration as provided. Their default values are unity. Also these scale factors can be employed to change the provided geometrical configurations from inches to any appropriate unit of measure; *i.e., centimeters.*

II MATERIAL PROPERTY SPECIFICATION

Card 1. (I5,1X,12A6)

Columns	Variable	Description	Note
1-5	NUMMAT	Number of different material characterizations to follow.	(1)
7-12	–	Must contain the word MATERI	

One group of cards follows for each material. There must be NUMMAT card groups. There are at least two cards for each group; card (a) and card (b). For viscoelastic characterization, additional cards are required within the group as described below.

Card (a). (I5,1X,I5,11A6)

Columns	Variable	Description	Note
1-5	–	Material number (1 to NUMMAT)	
7-11	–	Element type	(2)
12-77	–	Alphanumeric description of each material	

Card (b). (2I5,2F10.0)

Columns	Variable	Description	Note
1-5	NG	Number of viscoelastic shear terms. If not viscoelastic, leave blank.	(3)
6-10	NK	Number of viscoelastic bulk terms. If not viscoelastic, leave blank.	
11-20	–	The infinite shear modulus or elastic shear modulus, G_O.	
21-30	–	The infinite bulk modulus or elastic bulk modulus, K_O.	
31-40	–	Mass density	

Card (c). (8F10.) Omit this and the next card if material is not viscoelastic.

Columns	Variable	Description	Note
1-10	–	First shear modulus term G_1	
11-20	–	First relaxation time term, A_1	
21-30	–	Second shear modulus term, G_2	

etc.,until NG terms are identified.

Card (d). (8F10.)

Columns	Variable	Description	Note
1-10	–	First bulk modulus term, K_1	
11-20	–	First bulk relaxation time term, B_1	
21-30	–	Second bulk modulus term, K_2	

etc., until NK terms are identified.

Notes

(1) NUMMAT is usually assigned a value of 5. This allows the option for a different material specification for each of the following skull-brain regions.

Material Number	Region
1	Outer table bone
2	Diploe
3	Inner table bone
4	Subarachnoid space
5	Brain

(2) For the elastic skull materials (1, 2, and 3) the word ELM21 should be entered. For the subarachnoid space material (4) and brain material (5), an element possessing reduced integration is recommended, and the word ELM22 should be entered. Otherwise ELM21 may be entered for all materials.

(3) Viscoelastic material characterization is specified by the following equations:

$$G(t) = G_0 + \sum_{i=1}^{NG} G_i e^{-t/A_i}$$

$$K(t) = K_0 + \sum_{J=1}^{NK} K_i e^{-t/B_i}$$

Recommended units are the inch-pound-second system, although any consistent system can be used. (See note I(5)).

III SKULL-BRAIN GEOMETRICAL INPUT

Card 1. (I5,1X,A6)

Columns	Variable	Description	Note
1-5	NUMNP	Number of nodal points in the skull-brain system	(1)
7-12	—	Must contain the word NODAL	

There must follow NUMNP nodal point cards in numerical order.

Typical Nodal Point Card (I5,I15,3F10.0)

Columns	Variable	Description	Note
1-5	—	Nodal point number	
15	—	Insert 0 for free node or 1 for restraint in X direction	(2)
16	—	Insert 0 for free node or 1 for restraint in Y direction	
17	—	Insert 0 for free node or 1 for restraint in Z direction	
21-30	—	X coordinate value	
31-40	—	Y coordinate value	
41-50	—	Z coordinate value	

Notes

(1) For the six-cubed, half-skull model configuration provided, this value is 716. This is the model which has been most useful in the development of the HIM code. A complete set of nodal point cards whose coordinates are entered in inches has been provided for this configuration.

(2) The boundary conditions to be used for the HIM simulation are activated in cc 15, 16, and 17. Entering a 0 allows the node to move in any one of three possible directions. A zero in cc 15, 16, and 17 means the node is completely free. A 1 means that the node's displacement is constrained to move in a manner to be specified in subsequent input cards (see Section VI). The sets of nodal point cards which have been provided do not have any boundary condition preset with the following exceptions. The nodal point cards contain a 1 in cc 16 for all nodal points located in the midsagittal plane. Thus Y component displacements for half skull simulation are automatically constrained to zero in the midsagittal plane for the user's convenience.

IV ELEMENT CARDS

Card 1. (I5,1X,A6)

Columns	Variable	Description	Note
1-5	NUMEL	Number of elements in the skull-brain system	(1)
7-12	–	Must contain the word ELEMEN	

There must follow 2 cards for each element, taking the elements in numerical order.

Typical Element Card Set

Card (a) (2I5)

Columns	Variable	Description	Note
1-5	–	Element number	
6-10	–	Material number	(2)

Card (b) (8I4)

Columns	Variable	Description	Note
1-4	–	Node 1	
5-8	–	Node 2	
9-12	–	Node 3	
.	.		
.	.		
.	.		
29-32	–	Node 8	

Notes

(1) For the six-cubed, half-skull model configuration, the value is 555. A complete set of element cards has been provided for this configuration.

(2) A material number for which properties have been identified in Section II is entered. The corresponding element is then assigned those properties.

V INITIAL CONDITIONS

These cards can be omitted if the initial displacement velocity and acceleration of the skull-brain system are zero.

Card 1. (I5,1X,A6)

Columns	Variable	Description	Note
1-5	NICD	Number of initial condition vectors. Specify a "3".	(1)
7-12	—	Must contain the word INITIA	

Card 2. (6X,2A6) Repeat NICD times

Columns	Variable	Description	Note
7-18		Descriptive title for initial conditions	(2)

Card 3. (2I5,7F10.0) Repeat for each degree of freedom

Columns	Variable	Description	Note
1-5	—	Degree of freedom number for X; 3*NODE -2 = entry for Y; 3*NODE -1 = entry for Z; 3*NODE -0 = entry	(3)
6-10	—	Generator increment	(4)
11-20	—	Initial displacement value	
21-30	—	Initial velocity value	
31-40	—	Initial acceleration value	

Card 4. (I5)

Columns	Variable	Description	Note
1-5		Largest node times 3 plus 1 (NUMNP*3+1)	(5)

Notes

(1) When initial conditions of the skull-brain system are specified, the initial displacement, velocity, and acceleration (three vectors) are supplied as input.

(2) Three cards follow Card 1 and merely describe each of the three initial condition vectors. These descriptions will appear above each vector in the subsequent printout.

(3) Any or all nodes may be assigned an initial condition in the X, Y, and Z directions. But the initial condition must correspond to a degree of freedom number associated with the node and not the node number itself. The indicated equation is employed to obtain the degree of freedom number.

(4) The generator increment is used when the initial condition is uniformly applied over the entire skull-brain system, and this is generally the case. It avoids the necessity for inputting all but a few cards. For example, if the skull-brain system is initially moving at 240 in./sec in the +X direction, two cards are required to assign the initial velocity to all nodes. The first would be for node 1 and the second for node NUMNP (largest node number in the system). The first card would contain a degree of freedom number of 1(3* 1-2), a generator increment of 2 (assigns the initial condition to only the degree of freedom numbers associated with the X-direction), and a 240 in cc 21-30. The second card would contain the value of 3* NUMNP-2 in cc 1-5 and 240 in cc 21-30.

(5) This card ends the initial condition data.

VI PROPORTIONAL LOAD OR DISPLACEMENT SPECIFICATION USING ANALYTICAL FUNCTIONS

This section should be skipped if using point-by-point input loading. Such loading is covered in Section VIII. But this section is applicable to simulated dynamic loads or displacements which rise and subside according to a single time function (see Section VIII for the time function). They may all be assigned different magnitudes, but otherwise at any given instant the loads are proportional to one another.

Card 1. (I5,1X,A6)

Columns	Variable	Description	Note
1-5	—	Last node to which a force or displacement is to be specified. Must contain the word FORCE.	(1)

The following group of cards is composed of one card for each node to which a specified force or displacement is assigned.

Typical Card (I5,5X,6F10.0)

Columns	Variable	Description	Note
1-5		Node to which a force or displacement is specified	
11-20		X component of force or displacement	(2)
21-30		Y component of force or displacement	
31-40		Z component of force or displacement	

Notes

(1) Merely means the largest node number to which a force or displacement is applied. Forces and displacements must be applied at nodal points.

(2) This value is the ratio between the simulated peak load component at this node and the largest peak simulated load considering all nodes. It may also be a direction cosine. Only ratios from 0 to 1 are assigned here. The magnitude of the resultant for the largest peak load is prescribed in Section VIII (see Note (9)).

VII PLOT CARD

This card is to be used only when diverting computed response data to an output tape. This action does not suppress the printing of output data.

Card 1. (I5,1X,A6)

Columns	Variable	Description	Note
1-5	—	Number of time steps plus one for which data is output	(1)
7-12		Must contain the word PLOT3D	

Note

(1) Data will be placed on an output tape for subsequent postprocessing for each time step up to, but not including the value indicated. Usually data for all time steps is desired. Therefore this entry should be one more than the number of time steps prescribed in the dynamic analysis. See the variable NTS in the next section.

VIII IMPLICIT TIME-INTEGRATION SPECIFICATION

This card is required on all static or dynamic runs.

Card 1. (I5,1X,A6)

Columns	Variable	Description	Note
1-5	NSEQ	Number of time sequences, usually 1	(1)
7-12		Must contain the word IMPLIC for dynamic problems. Must contain the word VISCOE for static problems.	

NSEQ sets of the following cards are required

Card (a) (F10.0,8I5,2F10.0)

Columns	Variable	Description	Note
1-10	DT	Time step size	(2)
11-15	NTS	Number of time steps in the sequence(s) plus 1	(3)
16-20	INT	Print interval	(4)
21-25	NNI	First node printed, generally 1	(5)
26-30	NNE	Last node printed, generally NUMNP	
31-35	NEI	First element stress printed, generally 1	
36-40	NEF	Last element stress printed, generally NUMEL	
41-45	NPROP	Number of proportional loads	(6)
46-50	NFORC	Largest node to which a force or displacement has been specified in a point-by-point specification of input loading	(7)
51-60	BETA	Newmark integration parameter, usually specify 0.25	(8)
61-70	DEL	Newmark integration parameter, usually specify 0	

Card (b). (215,7F10.0) One for each NPROP. Use only when specifying analytical function input. If using point-by-point input skip to card (c).

Columns	Variable	Description	Note
1-5		Proportional load type, either 1 or 2	(9)
6-10	K	Exponent for type 2	
11-20	TMIN	Start time for loading	
21-30	TMAX	End time for loading	
31-40		AO	
41-50		A1	
51-60		A2	
61-70		A3	
71-80		A4	

Card group (c) (I5,5X,6F10.0). This group of cards is comprised of one card for each node to which an input value is assigned. Node NFORC is the last node from which a card is provided. The group represents a loading vector, and generally there should be one group for each time step in the dynamic analysis. Omit this card group if NPROP is non-zero.

Columns	Variable	Description	Note
1-5		Node to which force or displacement is specified	
11-20		X component value of force or displacement	
21-30		Y component value of force or displacement	
31-40		Z component value of force	

Card (3) (A5) last data card

Columns	Variable	Description	Note
1-5		Must contain the word STOP	

Notes

(1) Little experience has been gained with values other than 1. Potentially, NSEQ can be used to specify a broader range of different loading functions. (Card (a) describes how different functions can be specified with NSEQ > 1). For example, one analysis could consist of two sequences where the first is the onset of loading requiring a small time step and the second is the loading decay which may be slower and during which a larger, more economical, time step can be employed.

(2) This value should be chosen with caution. It generally depends on the rise time and should be no greater than one-fifth of that value. A value of one-tenth would be sufficiently small.

(3) This is the number of time steps plus 1 for which the dynamic analysis is to proceed. Generally, it is chosen so that the entire rise and decay of the loading function can be simulated with the prescribed time step.

(4) Several inches of paper may be required, depending upon NTS, for the printing of output information. A blank in cc 16-20 will cause the printing of data for each time step; issuing a 1 will cause the printing of data for every other time step, etc.

(5) The next four entries also provide for reductions in the amount of output. If a few nodes and elements can be identified as key check points in the analysis it is only necessary to monitor their output. However, data elsewhere should at least be saved on tape (see Section VII).

(6) This value is employed to distinguish among the specification of different input load functions in the dynamic analysis. For example if it is desired that the prescribed load vary as a haversine function and subsequently change to a linear function, then NPROP=2. This is also true of a triangular function. If a single function, such as a ramp function or haversine function, is desired then NPROP=1. If a point-by-point input loading is desired then NPROP is left blank.

(7) This entry is for point-by-point load specification only. If proportioned load functions are employed, NFORC is left blank.

(8) BETA and DEL are the Newmark integration parameters (implicit). Investigators have determined that BETA = 0.25 and DEL = 0 are good choices. If DEL > 0 artificial positive damping is introduced and if DEL < 0 artificial negative damping is introduced. The numerical integration will be unconditionally stable if

$$\text{BETA} \geq \frac{4\,(\text{DEL} + 1)^2}{16}$$

The higher and least important nodes may be damped out by choosing BETA = 0.276 and DEL = 0.05.

(9) Proportional load type 1 is a polynomial function,

$$f(t) = AO + A1*t + A2*t^2 + A3*t^3 + A4*t^4$$

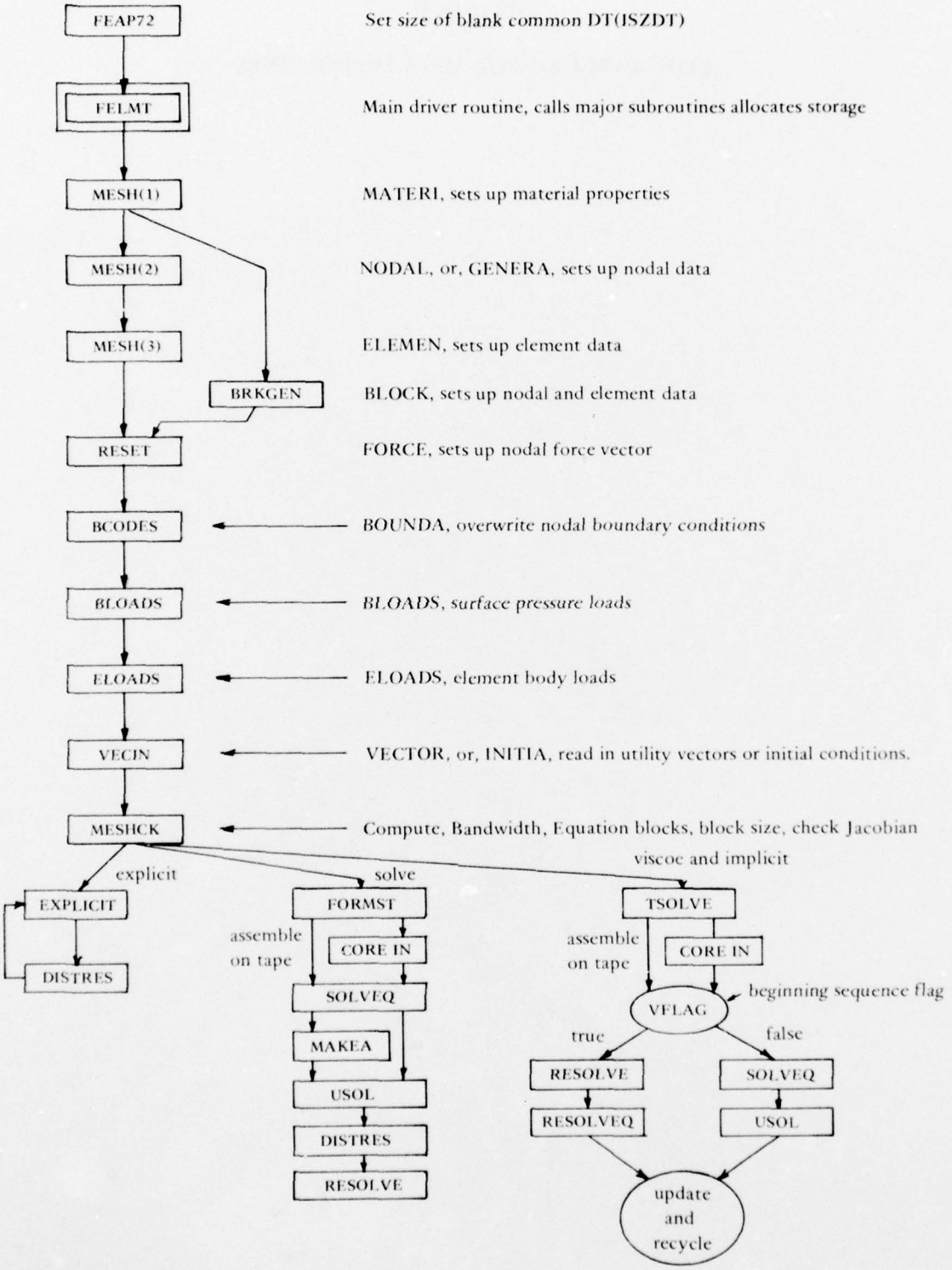
Proportional load type 2 is a transcendental function,

$$f(t) = AO*Sin^K(A1*t) + A2*cos^K(A3*t) + A4$$

A haversine function has been found to be particularly useful in the specification of impact forces, $F(t) = AO*Sin^K(A1*t)$, where AO is the peak force magnitude, A1 is pi divided by the load duration in seconds, and K = 2. Force magnitude must be halved when employing a half-skull model.

Appendix D

FLOW DIAGRAM FOR MAIN SUBROUTINES


FEAP72
Set size of blank common DT(ISZDT)
FELMT
Main driver routine, calls major subroutines allocates storage
MESH(1)
MATERI, sets up material properties
MESH(2)
NODAL, or, GENERA, sets up nodal data
MESH(3)
ELEMEN, sets up element data
BRKGEN
BLOCK, sets up nodal and element data
RESET
FORCE, sets up nodal force vector
BCODES
BOUNDA, overwrite nodal boundary conditions
BLOADS
BLOADS, surface pressure loads
ELOADS
ELOADS, element body loads
VECIN
VECTOR, or, INITIA, read in utility vectors or initial conditions.
MESHCK
Compute, Bandwidth, Equation blocks, block size, check Jacobian
explicit
solve
viscoc and implicit
EXPLICIT
DISTRES
FORMST
assemble on tape
CORE IN
SOLVEQ
MAKEA
USOL
DISTRES
RESOLVE
TSOLVE
assemble on tape
CORE IN
beginning sequence flag
VFLAG
true
false
RESOLVE
RESOLVEQ
SOLVEQ
USOL
update and recycle

Appendix E

SAMPLE JOB CONTROL CARDS

```
$CHARGE,L2317*X-00A.
MONKEY,CM150000,CL262000,T4000,IO4000,TM1,P6.
REQUEST FEAP,MI. (L2313)
SKIPF(FEAP,1,17,H)
COPYHF(FEAP,LGO)
REWIND(FEAP)
UNLOAD(FEAP)
REQUEST,TAPE30,MI.(SAVE)
ATTACH(IO,QUICKIO)
MFL,150000.
SET(0)
LOAD(LGO)
LOAD(IO)
EXECUTE.
EXIT.
EXIT.
```

Appendix F

HIM CODE LISTING

```
$CHARGE,L2317*X=00R.
MONKEY,CM150,CL262000,T4000,I04000,TP1,P6.
REQUEST FEAP,MI. (L2313)
SKIPF(FEAP,1,17,M)
COPYMF(FEAP,LGO)
REWIND(FEAP)
UNLOAD(FEAP)
REQUEST,TAPE30,MI,(SAVE)
ATTACH(IO,GOICKIO)
MFL,150000.
SET(0)
LOAD(LGO)
LOAD(IO)
EXECUTE.
EXIT.
EXIT.
```

```
      PROGRAM FEAP72 (INPUT=65,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     *  TAPE7=3075,TAPE9,TAPE10=3075,TAPE11=65,TAPE13=2050,
     *  TAPE14=2050,TAPE30)
C
CXXXX    NOTE          NOTE          NOTE          NOTE
C
C  TWO ADDL. INPUT FEATURES ARE INCLUDED IN THIS VERSION
C
C  1. AN OPTIONAL CARD MAY BE INCLUDED IMMEDIATELY PRECEDING
C     THE FIRST  FEAP  CARD.  THIS CARD HAS THE ENTRY  ISZDT  (I5).
C     ADMISSABLE RANGE IS  500.LE. ISZDT .LE.65000.
C     THE PROGRAM WILL ASSIGN THIS VALUE TO THE SIZE OF BLANK
C     COMMON AND REL UP TO THIS LIMIT.  THIS WILL ALLOW THE
C     PROGRAM TO LOAD AND EXECUTE IN ONLY THE AMOUNT OF CORE REQD.
C     IF OMITTED, A DEFAULT VALUE OF 58000 IS ASSIGNED TO ISZDT.
C
C  2. A  1  ENTERED IN CC 79 OF THE  FEAP  CARD INDICATES THAT
C     THE NODE RESEQUENCING CARDS PUNCHED BY THE  FEAP PREPROCESSOR
C     ARE TO BE INPUT IMMEDIATELY FOLLOWING THE  FEAP  CARD.
C
CXXXX    NOTE          NOTE          NOTE          NOTE
C                                                                    FEA  4C
C**** FINITE ELEMENT ASSEMBLY PROGRAM * STORAGE ALLOCATION           FEA  5C
C                                                                    FEA  6C
      COMMON /TIMING/ TRICP,NTRI,BACKCP,NBACK
      COMMON DT(1)
      DIMENSION IBIN(6)
      DATA IBIN/7,9,10,13,14,30/
      DATA JFEAP/4HFEAP/
      CALL PINBIN (1,6,IBIN)
      CALL BLKDAT
      TRICP=0.
      BACKCP=0.
      NTRI=0
      NBACK=0
      T1=0.
      T2=0.
C                                                                    FEA  8C
C**** ISZDT MUST COINCIDE WITH SIZE OF ARRAY DT                      FEA  9C
C                                                                    FEA 10C
CXXXX
C  CHECK FOR OPTIONAL FIRST CARD
CXXXX
      ISZDT=58000
      READ (5,100) IFEAP
  100 FORMAT (A4)
      BACKSPACE 5
      IF(IFEAP.EQ.JFEAP) GO TO 50
      READ (5,200) ISZDT
  200 FORMAT (I5)
      IF(ISZDT.LT.500) ISZDT=500
      IF(ISZDT.GT.65000) ISZDT=65000
   50 CALL AREL (LOCF(DT(ISZDT)))
C                                                                    FEA 12C
C**** REASONABLE MAXIMUM BAND WIDTH                                  FEA 13C
C                                                                    FEA 14C
      MBAND=1200
      CALL PELMT(DT,ISZDT,MBAND)                                     FEA 16C
CXXXX
C  PRINT PROGRAM TIMINGS
C  NOTE THAT THE TIMINGS ARE NOT MEANINGFUL
C  IF STACKED JOBS WERE PROCESSED
CXXXX
      IF(NTRI.GT.0) T1=TRICP/NTRI
      IF(NBACK.GT.0) T2=BACKCP/NBACK
      WRITE (6,500) NTRI,TRICP,T1,NBACK,BACKCP,T2
  500 FORMAT ( //
     *46H       PROGRAM  TIMINGS                                  //
     *46H       NUMBER OF      TOTAL        TIME FOR EACH /
     *46H     TRIANGULAR       TIME        FACTORIZATION /
     *46H FACTORIZATIONS         (CP)                (CP) //
     *  I15,F10.3,F20.3//
```

```
     *&AM       NUMBER OF       TOTAL        TIME FOR EACH /
     *&AM        BACK&ARD        TIME           REDUCTION /
     *&AM      REDUCTIONS        (CP)                (CP) //
     *  I15,F10.3,F20.3// )
      STOP                                                                FEA 17C
      END                                                                 FEA 18C
*DECK MANUAL
      SUBROUTINE MANUAL                                                   MAN  1C
C                                                                         MAN  2C
C**** USER INSTRUCTIONS AND INPUT FORMATS FOR FEAP-73 ******************  MAN  3C
C                                                                         MAN  4C
C                                                                         MAN  5C
C     FINITE ELEMENT ANALYSIS PROGRAM                                     MAN  6C
C                                                                         MAN  7C
C                                                                         MAN  8C
C        FEAP73 IS A GENERAL (F)INITE (E)LEMENT (A)NALYSIS (P)ROGRAM      MAN  9C
C        WHICH FURNISHES TO THE USER MESH INPUT/OUTPUT, ELEMENT ASSEMBLY  MAN 10C
C        AND SOLUTION OF EQUATIONS (LINEAR, IMPLICIT AND EXPLICIT TIME    MAN 11C
C        DEPENDENT, NONLINEAR), PRESCRIBED GENERALIZED NODAL FORCES,      MAN 12C
C        PRESCRIBED NODAL AND ELEMENT DATA, AND OUTPUT OF THE             MAN 13C
C        GENERALIZED DISPLACEMENTS AND FORCES.  ELEMENT MATRICES FOR TWO  MAN 14C
C        AND THREE DIMENSIONAL LINEAR ELASTICITY, SHELLS, PLATES, AND     MAN 15C
C        FIELD (LAPLACE EQUATION) PROBLEMS ARE AVAILABLE.  ALTERNATIVELY  MAN 16C
C        USERS MAY SUPPLY THEIR OWN ELEMENT LIBRARY BY PROVIDING A        MAN 17C
C        SUBROUTINE CALLED ELMTNN,  WHERE NN IS A TWO DIGIT NUMBER        MAN 18C
C        IDENTIFYING THE ELEMENT SUBROUTINE. EACH ELEMENT SUBROUTINE      MAN 19C
C        HAS AT LEAST FOUR BASIC FUNCTIONS WHICH ARE DELINEATED BY A      MAN 20C
C        SWITCHING PARAMETER, ISW, IN THE SUBROUTINE.                     MAN 21C
C                                                                         MAN 22C
C        ELMTNN(N,MA,NDIM,NDF,NEL,NELI,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,      MAN 23C
C        IX,F,FORCE,ESTIF,U,VECT,ISW)                                     MAN 24C
C                                                                         MAN 25C
C                        N IS ELEMENT NUMBER.                             MAN 26C
C                        MA IS THE MATERIAL NUMBER.                       MAN 27C
C                        NDIM IS SPATIAL DIMENSION, 1,2,OR 3.             MAN 28C
C                        NDF IS NUMBER OF DEGREES OF FREEDOM PER NODE.    MAN 29C
C                        NEL IS THE NUMBER OF EXTERNAL NODES PER ELEMENT  MAN 30C
C                        NELI IS DIMENSION OF ELEMENT PROPERTY ARRAY.     MAN 31C
C                        NSTF IS THE SIZE OF THE ELEMENT STIFFNESS.       MAN 32C
C                        NSIZV IS THE SIZE OF UTILITY VECTORS.            MAN 33C
C                        NVEC IS THE NUMBER OF UTILITY VECTORS.           MAN 34C
C                        MCT IS A PRINTER LINE COUNTER.                   MAN 35C
C                        DM IS A PARAMETER FOR MATERIAL IDENTIFICATION.   MAN 36C
C                        D(I,J) IS MATERIAL PROPERTY MATRIX (63 CELLS).   MAN 37C
C                        XYZ(NDIM,I) ARE NODAL COORDINATES.               MAN 38C
C                        IX(NELI,I) ARE ELEMENT  PROPERTIES, NODES, ETC.  MAN 39C
C                        F(NDF,I) ARE NODAL GENERALIZED FORCES.           MAN 40C
C                        FORCE(NSTF,2) IS ELEMENT FORCE VECTOR TO BE      MAN 41C
C                        COMPUTED. COLUMN 2 IS LUMPED MASS                MAN 42C
C                        ESTIF(NSTF,NSTF) IS ELEMENT MATRIX TO BE         MAN 43C
C                        COMPUTED.                                        MAN 44C
C                        VECT(NSIZV,I) ARE PRESCRIBED NODAL OR ELEMENT    MAN 45C
C                        QUANTITIES. TEMPERATURES ETC.                    MAN 46C
C                        U(NDF,I) IS SOLUTION VECTOR.                     MAN 47C
C                        ISW IS SWITCHING PARAMETER.                      MAN 48C
C                                                                         MAN 49C
C                            ISW=1, ** MATERIAL CHARACTERIZATION**        MAN 50C
C                            ISW=2, ** CHECK ELEMENT FOR POSITIVE AREA *  MAN 51C
C                            ISW=3, ** ELEMENT STIFFNESS COMPUTATION**    MAN 52C
C                            ISW=4, ** ELEMENT STRESSES AND PRINTOUT**    MAN 53C
C                            ISW=5, ** ELEMENT LOAD COMPUTATION           MAN 54C
C                            ISW=6, ** NONLINEAR GENERALIZED FORCES       MAN 55C
C                        OTHER ISW MAY BE USED FOR SPECIAL PURPOSES.      MAN 56C
C                                                                         MAN 57C
C        USERS CAN GENERATE SURFACE LOADINGS BY PROVIDING SLDNN           MAN 58C
C        SUBROUTINES (WHERE NN IS A TWO DIGIT NUMBER BETWEEN 01 AND 05)   MAN 59C
C        THAT SPECIFY THE LOAD ROUTINE.  THE SUBROUTINE IS ACCESSED BY    MAN 60C
C        THE CALL TO                                                      MAN 61C
C                                                                         MAN 62C
C        SLDNN(NDIM,NDF,NUP,NPRES,IPRES,PR,XYZ,FS)                        MAN 63C
C                                                                         MAN 64C
C                        WHERE IN ADDITION TO QUANTITIES DEFINED ABOVE    MAN 65C
```

```
C                       FOR ELMTNN.                                          MAN 66C
C                       NDP IS THE DIMENSION OF LOADED SURFACE               MAN 67C
C                       NPRES IS NUMBER OF LOADED NODES (MAX 8)              MAN 68C
C                       IPRES(8) ARE NODE NUMBERS OF LOADED NODES.           MAN 69C
C                       PR(8) ARE LOAD VALUES AT CORRESPONDING IPRES         MAN 70C
C                           NODES.                                           MAN 71C
C                       FS(6,8) ARE THE COMPUTED GENERALIZED (NODAL)         MAN 72C
C                           FORCES FOR EACH DEGREE OF FREEDOM AT EACH        MAN 73C
C                           IPRES NODE.                                      MAN 74C
C                                                                            MAN 75C
C     SEE SECTION 7.1 FOR DATA INPUT DETAILS.                                MAN 76C
C                                                                            MAN 77C
C                                                                            MAN 78C
C     INTEGRATION TABLE IS ACCESSED BY THE CALL                              MAN 79C
C                                                                            MAN 80C
C       CALL INTEGL(LIM,NCI,NDIM,LINT,STUW)                                  MAN 81C
C                                                                            MAN 82C
C                       STUW(4,M) INTEGRATION POINTS AND WEIGHTS.            MAN 83C
C                       **NOTE** M MUST BE SET EXPLICITLY AND BE LARGER      MAN 84C
C                       THAN OR EQUAL TO LINT.                               MAN 85C
C                       LINT = RETURNS WITH NUMBER INTEGRATION POINTS.       MAN 86C
C                       NCI = 0  RETURNS GAUSS POINTS AND WEIGHTS IN         MAN 87C
C                           STUW.                                            MAN 88C
C                       LIM = 1 TO 5 IS NUMBER OF GAUSS POINTS/DIR-          MAN 89C
C                       ECTION.                                              MAN 90C
C                       NCI = 1  RETURNS A SPECIAL 3-D GAUSS FORMULA.        MAN 91C
C                           SET LIM = 1 FOR 6 PT. CUBIC ACCURACY             MAN 92C
C                           SET LIM = 2 FOR 14 PT. QUINTIC ACCURACY.         MAN 93C
C                       NCI = 2  RETURNS TRIANGULAR INTEGRATION FORMULA      MAN 94C
C                           SET LIM = 1 FOR 1 PT. LINEAR ACCURACY.           MAN 95C
C                           SET LIM = 2 FOR 3 PT. QUADRATIC ACCURACY.        MAN 96C
C                           SET LIM = 3 FOR 7 PT. QUARTIC ACCURACY.          MAN 97C
C                                                                            MAN 98C
C                                                                            MAN 99C
C  1.) DATA TYPE IDENTIFICATION CARDS (I5,1X,12A6).                          MAN100C
C                                                                            MAN101C
C     EACH DATA SEGMENT IS PREDEDED BY A CARD WHICH IDENTIFIES THE           MAN102C
C     TYPE OF DATA AND LIMITS ON THE AMOUNT OF DATA WHICH IMMEDIATELY        MAN103C
C     FOLLOWS THE CARD.  EXCEPT AS NOTED THE DATA SEGMENTS MAY APPEAR        MAN104C
C     IN ANY ORDER.  THE IDENTITY CARDS MAY ALSO AID THE USER IN             MAN105C
C     INTERPRETING THE INPUT DATA CARDS.  AS SUPPLIED THERE ARE              MAN106C
C     TWENTY-FIVE DIFFERENT DATA IDENTIFICATION CARDS.  THESE ARE            MAN107C
C                                                                            MAN108C
C     COL  7 TO 12      IDENTITY(RESTRICTIONS)                               MAN109C
C                                                                            MAN110C
C     FEAP73            START OF EACH PROBLEM (MUST PRECEDE ALL OTHER        MAN111C
C                       DATA).                                               MAN112C
C     TITLE             CHANGE OUTPUT PAGE HEADINGS                          MAN113C
C     REMARK            COMMENTS ON OUTPUT                                   MAN114C
C     MATERI            MATERIAL CHARACTERIZATION.                           MAN115C
C     NODAL             NODAL CARDS                                          MAN116C
C     POLAR             POLAR CONVERSION. (PRECEDE BY NODAL,GENERA, OR       MAN117C
C                       BLOCK )                                              MAN118C
C     ELEMEN            ELEMENT CONNECTION CARDS.                            MAN119C
C     GENERA            GENERATE NODES IN A LINEAR PATH BY ANY INCREMNT       MAN120C
C     BLOCK             GENERATE ALL MESH DATA (BOTH NODAL AND ELEMENT)      MAN121C
C                       FOR A 2,(OR 3)DIMENSIONAL REGION WHOSE BOUNDARY      MAN122C
C                       MAY BE DEFINED BY 4(8)OR 8(20)COLLOCATED POINTS      MAN123C
C     BOUNDA            BOUNDARY CODE PRESCRIPTION (PRECEDE BY NODAL OR      MAN124C
C                       GENERATE OR BLOCK)                                   MAN125C
C     VECTOR            PRESCRIBED NODAL OR ELEMENT DATA (PRECEDE BY         MAN126C
C                       NODAL OR POLAR AND ELEMEN)                           MAN127C
C     FORCE             NODAL GENERALIZED FORCES (PRECEDE BY NODAL OR        MAN128C
C                       GENERA OR BLOCK).                                    MAN129C
C     BLOADS            SURFACE LOADINGS (SAME AS FORCE).                    MAN130C
C     ELOADS            ELEMENT LOADINGS (SAME AS FORCE).                    MAN131C
C     MESH              CHECK CONSISTENCY OF MESH ONLY (SAME AS SOLVE)       MAN132C
C     PLOT              PLOT MESH, RESULTS (SAME AS SOLVE)
C     INITIA            INITIAL CONDITION PRESCRIPTION FOR DYNAMIC           MAN134C
C                       ANALYSIS (PRECEDE BY NODAL, GENERA OR BLOCK)         MAN135C
C     SOLVE             COMPLETE FORMULATION AND SOLUTION FROM ELEMENTS       MAN136C
C                       (PRECEDE BY MATERI,NODAL OR GENERA, AND ELEMEN       MAN137C
```

```
C                       OR PRECEDE BY MATERI AND BLOCK )                    MAN138C
C       RESOLV          USE PREVIOUS PROBLEM DESCRIPTION WITH NEW LOAD      MAN139C
C                       ONLY (PRECEDE BY SOLVE AND NEW LOADING CARDS).      MAN140C
C       EXPLIC          DYNAMIC SOLUTION BY EXPLICIT INTEGRATION. (SAME     MAN141C
C                       AS SOLVE)                                           MAN142C
C       IMPLIC          IMPLICIT INTEGRATION OF DYNAMIC PROBLEMS            MAN143C
C                       (PRECEDE BY SAME DATA AS FOR SOLVE)                 MAN144C
C       VISCOE          QUASI-STATIC LINEAR VISCOELASTIC INTEGRATION        MAN145C
C                       (PRECEDE BY SAME DATA AS FOR SOLVE)                 MAN146C
C       FOURIE          FOURIER COMPOSITION (SAME AS SOLVE)                 MAN147C
C       ADDUP           ACCUMULATE FOURIER SOLUTION (AFTER FOURIE)          MAN148C
C       STOP            NORMAL EXIT (MUST FOLLOW ALL DATA)                  MAN149C
C                                                                           MAN150C
C       ***NOTE*** EACH IDENTIFIER IS PUNCHED STARTING IN COL 7 (LEFT       MAN151C
C           JUSTIFIED).
C                                                                           MAN153C
C     EXCESS CARDS MAY EXIST BETWEEN EACH SECTION OF DATA, HOWEVER,         MAN154C
C     THE DATA TO BE USED MUST IMMEDIATELY FOLLOW THE TYPE CARD AND         MAN155C
C     MUST BE IN PROPER ORDER.  NO PARTICULAR ORDER OF THE TYPE             MAN156C
C     CARDS IS NECESSARY EXCEPT THAT THE FEAP73 CARD MUST ALWAYS BE         MAN157C
C     THE FIRST CARD IN EACH SET OF DATA, AND RESTRICTIONS MUST BE          MAN158C
C     OBSERVED.                                                             MAN159C
C                                                                           MAN160C
C                                                                           MAN161C
C  2.) PROBLEM INITIATION AND CONTROL CARDS                                 MAN162C
C                                                                           MAN163C
C     CARD 1. (6A,12A6)                                                     MAN164C
C                                                                           MAN165C
C     COL  7 TO 12      MUST CONTAIN WORD FEAP73                            MAN166C
C     COL 13 TO 78      OUTPUT PAGE HEADER                                  MAN167C
C                                                                           MAN168C
C     CARD 2. (I5,1X,3A6)                                                   MAN169C
C                                                                           MAN170C
C     COL  1 TO  5      NDIM - SPATIAL DIMENSION OF PROBLEM (1 TO 3)        MAN171C
C     COL  7 TO 12      NAMES TO BE PRINTED AS OUTPUT HEADERS TO            MAN172C
C     COL 13 TO 18      COORDINATES - IF BLANK SET TO 1,2,3 AS NEEDED.      MAN173C
C     COL 19 TO 24                                                          MAN174C
C                                                                           MAN175C
C     CARD 3. (I5,1X,6A6)                                                   MAN176C
C                                                                           MAN177C
C     COL  1 TO  5      NDF - NUMBER OF UNKNOWNS PER NODE (1 TO 6)          MAN178C
C     COL  7 TO 12      NAMES TO BE PRINTED AS OUTPUT HEADERS OF THE        MAN179C
C     COL 13 TO 18      GENERALIZED DISPLACEMENTS AND FORCES - IF            MAN180C
C          ........     BLANK SET TO 1,2,3,4,5,6 AS NECESSARY               MAN181C
C     COL 37 TO 42                                                          MAN182C
C                                                                           MAN183C
C     CARD 4. (6I5,5F10.0)                                                  MAN184C
C                                                                           MAN185C
C     COL  1 TO  5      NEN - MAXIMUM NUMBER OF NODES CONNECTED TO ANY      MAN186C
C                       ELEMENT (1 TO 20).                                  MAN187C
C     COL  6 TO 10      NEXTRA - INCREASES ELEMENT MATRIX SIZE FROM          MAN188C
C                       NDF*NEN TO NDF*NEN + NEXTRA                         MAN189C
C     COL 11 TO 15      IREC - COMPUTE GENERALIZED FORCE CHECK IF            MAN190C
C                       NONZERO (FOR TIME INVARIANT ANALYSIS ONLY)          MAN191C
C     COL 16 TO 20      MBAN - MAXIMUM EXPECTED BANDWIDTH, DEFAULT IS       MAN192C
C                       SET TO 100. USED AS AN ERROR CHECK TO PREVENT       MAN193C
C                       RUNNING WITH AN OBVIOUS ERROR.                      MAN194C
C     COL 21 TO 25      IBUF - BUFFER SIZE FOR STORAGE OF HISTORY            MAN195C
C                       EFFECTS IN TIME DEPENDENT ANALYSIS. DEFAULT IS      MAN196C
C                       IBUF = ISZDT/20                                     MAN197C
C     COL 26 TO 30      NC1  - USER INTEGER CONSTANT                        MAN198C
C     COL 31 TO 40      CON1 - USER DEFINED CONSTANT                        MAN199C
C     COL 41 TO 50      CON2 - USER DEFINED CONSTANT                        MAN200C
C     COL 51 TO 60      CON3(1), 1-COORD. MULTIPLIER, DEFAULT = 1.0
C     COL 61 TO 70      CON3(2), 2-COORD. MULTIPLIER, DEFAULT = 1.0
C     COL 71 TO 80      CON3(3), 3-COORD. MULTIPLIER, DEFAULT = 1.0
C                                                                           MAN204C
C   2.1)  *REMARK* USER COMMENTS ON OUTPUT. (6A,12A6)                       MAN205C
C                                                                           MAN206C
C     SUBSEQUENT CARDS                                                      MAN207C
C                                                                           MAN208C
C     COL  7 TO 12      MUST CONTAIN REMARK                                 MAN209C
```

```
C        COL 13 TO 78     STATEMENTS TO BE OUTPUT . USE AS MANY REMARK     MAN210C
C                         CARDS AS DESIRED. INSERT BEFORE ANY TYPE CARD.   MAN211C
C                                                                          MAN212C
C      2.2) TITLE CHANGE ON OUTPUT (6X,12A6)                               MAN213C
C                                                                          MAN214C
C        COL  7 TO 12     MUST CONTAIN TITLE                               MAN215C
C        COL 13 TO 78     NEW TITLE DESCRIPTOR                             MAN216C
C                                                                          MAN217C
C      2.3) EXECUTION TERMINATION (6X,A4)                                  MAN218C
C                                                                          MAN219C
C        COL  7 TO 10     MUST CONTAIN STOP, INSERT AFTER LAST PROBLEM.     MAN220C
C                                                                          MAN221C
C                                                                          MAN222C
C      3.) MATERIAL CHARACTERIZATION (I5,1X,12A6)                          MAN223C
C                                                                          MAN224C
C        COL  1 TO  5     NUMMAT - NUMBER OF DIFFERENT MATERIAL CHARACT-   MAN225C
C                         ERIZATIONS TO FOLLOW.                            MAN226C
C        COL  7 TO 12     MUST CONTAIN WORD MATERI                         MAN227C
C                                                                          MAN228C
C      THE FOLLOWING CARDS ARE SUPPLIED FOR EACH MATERIAL TO BE CHARAC     MAN229C
C      TERIZED (MUST BE EXACTLY NUMMAT SETS OF CARDS)                      MAN230C
C                                                                          MAN231C
C      CARD 1.) ELEMENT SELECTOR CARD (I5,1X,A5,11A6)                      MAN232C
C                                                                          MAN233C
C        COL  1 TO  5     MATERIAL NUMBER (1 TO NUMMAT)                    MAN234C
C        COL  7 TO 11     ELMNN - WHERE NN IS NUMBER OF ELEMENT CLASS (01  MAN235C
C                         TO 30) TO WHICH THE CHARACTERIZATION BELONGS.    MAN236C
C        COL 12 TO 77     ALPHANUMERIC INFORMATION TO BE OUTPUT.           MAN237C
C                                                                          MAN238C
C      CARD 2.), ETC. ** USER DEFINED FOR EACH ELEMENT TYPE PROVIDED.      MAN239C
C                                                                          MAN240C
C      4.) NODAL CARDS (I5,1X,A6)                                          MAN241C
C                                                                          MAN242C
C        COL  1 TO  5     NUMNP - NUMBER OF NODAL POINTS                   MAN243C
C        COL  7 TO 12     MUST CONTAIN NODAL                               MAN244C
C                                                                          MAN245C
C      SUBSEQUENT CARDS  LAST NODAL CARD MUST NOT BE GENERATED.            MAN246C
C      (I5,I15,3F10.0)                                                     MAN247C
C                                                                          MAN248C
C        COL  1 TO  5     NODE NUMBER                                      MAN249C
C        COL 15           1 IF 1 DISPLACEMENT IS SPECIFIED                 MAN250C
C        COL 16           1 IF 2 DISPLACEMENT IS SPECIFIED                 MAN251C
C        COL 17           1 IF 3 DISPLACEMENT IS SPECIFIED                 MAN252C
C        COL 18           1 IF 4 DISPLACEMENT IS SPECIFIED                 MAN253C
C        COL 19           1 IF 5 DISPLACEMENT IS SPECIFIED                 MAN254C
C        COL 20           1 IF 6 DISPLACEMENT IS SPECIFIED                 MAN255C
C        COL 21 TO 30     1 COORDINATE VALUE                               MAN256C
C        COL 31 TO 40     2 COORDINATE VALUE . AS REQUIRED                 MAN257C
C        COL 41 TO 50     3 COORDINATE VALUE                               MAN258C
C                                                                          MAN259C
C      NODAL CARDS MUST BE IN ORDER. MISSING NODES ARE INTERPOLATED        MAN260C
C      LINEARLY FROM INPUT NODES. IF SUCCEEDING CARDS HAVE IDENTICAL       MAN261C
C      BOUNDARY CODES, THIS BOUNDARY CODE WILL BE ASSIGNED TO THE INTER-   MAN262C
C      VENING NODES. IN ALL OTHER CASES THE BOUNDARY CODE IS SET TO ZERO   MAN263C
C      .TERMINATE ON NODE NUMNP OR A BLANK CARD.                           MAN264C
C                                                                          MAN265C
C      4.1) NON SEQUENTIAL NODAL GENERATOR OPTION. (I5,1X,A6)              MAN266C
C                                                                          MAN267C
C        COL  1 TO  5     NUMBER OF NODAL POINTS                           MAN268C
C        COL  7 TO 12     MUST CONTAIN GENERA                              MAN269C
C                                                                          MAN270C
C      SUBSEQUENT CARDS (2I5,I10,3F10.0)                                   MAN271C
C                                                                          MAN272C
C        COL  1 TO  5     NODE-NUMBER                                      MAN273C
C        COL  6 TO 10     NODE-NUMBER-INCREMENT WHICH WILL BE SUCESSIVELY   MAN274C
C                         ADDED TO NODE-NUMBER UNTIL SUM IS GREATER THAN   MAN275C
C                         NODE-NUMBER ON FOLLOWING CARD (ALGEBRAIC).       MAN276C
C        COL 15 TO 20     BOUNDARY CODE. SAME AS INPUT FOR NODAL.          MAN277C
C                         IF SUCCEEDING CARDS HAVE IDENTICAL BOUNDARY      MAN278C
C                         CODES, THIS BOUNDARY CODE WILL BE ASSIGNED TO    MAN279C
C                         THE INTERVENING NODES. IN ALL OTHER CASES THE    MAN280C
C                         BOUNDARY CODE IS SET TO ZERO.                    MAN281C
```

```
C       COL 21 TO 30     1 COORDINATE VALUE  *                                  MAN282C
C       COL 31 TO 40     2 COORDINATE VALUE  * AS REQUIRED *                    MAN283C
C       COL 41 TO 50     3 COORDINATE VALUE  *                                  MAN284C
C                                                                               MAN285C
C       *TERMINATE WITH BLANK CARD *                                            MAN286C
C                                                                               MAN287C
C     4.2) BOUNDARY CODE PATCH-UP OPTION. (6X,A6)                               MAN288C
C                                                                               MAN289C
C       COL  6 TO 12     MUST CONTAIN BOUNDA                                    MAN290C
C                                                                               MAN291C
C       SUBSEQUENT CARDS. (2I5)                                                 MAN292C
C                                                                               MAN293C
C       COL  1 TO  5     N, NODE NUMBER TO HAVE REDEFINED BOUNDARY CODE.        MAN294C
C       COL  6 TO 10     NX, GENERATOR INCREMENT TO BE ADDED ALGEBRAI-          MAN295C
C                        CALLY TO N, UNTIL SUM EXCEEDS (MAX OR MIN) THE         MAN296C
C                        N OF THE FOLLOWING CARD.                               MAN297C
C       COL 11 TO 15     IBC(I), (I=1,2...NDF) CODE FOR SPECIFYING FORCE        MAN298C
C       COL 16 TO 20     OR DISPLACEMENT BOUNDARY CONDITIONS.                   MAN299C
C       COL  ...                                                                MAN300C
C                              IBC(I) .EQ. 0, FORCE SPECIFIED.                  MAN301C
C                              IBC(I) .GT. 0, DISPLACEMENT SPECIFIED, NO        MAN302C
C                              INTERVENING GENERATION.                          MAN303C
C                              IBC(I) .LT. 0, DISPLACEMENT SPECIFIED,           MAN304C
C                              GENERATE BETWEEN MISSING NODES IN ALGEBRAIC      MAN305C
C                              INCREMENTS OF NX.                                MAN306C
C                                                                               MAN307C
C       * TERMINATE WITH A BLANK CARD. *                                        MAN308C
C                                                                               MAN309C
C     4.3) POLAR OR CYLINDRICAL COORDINATE CONVERSION TO CARTESIAN              MAN310C
C          COORDINATES (6X,A6)                                                  MAN311C
C                                                                               MAN312C
C       COL  7 TO 12     MUST CONTAIN POLAR     (LEFT JUSTIFIED)               MAN313C
C                                                                               MAN314C
C     CARD 1.  (3I5,5X,2F10.0)                                                  MAN315C
C                                                                               MAN316C
C       COL 11 TO 15     N3, INCREMENT ADDED (ALGEBRAICALLY), N1 TO N2          MAN317C
C       COL  1 TO  5     N1, FIRST NODE TO BE CONVERTED                         MAN318C
C       COL  6 TO 10     N2, LAST  NODE TO BE CONVERTED                         MAN319C
C       COL 21 TO 30     X0, ORIGIN OF POLAR X-COORDINATE                       MAN320C
C       COL 31 TO 40     Y0, ORIGIN OF POLAR Y-COORDINATE                       MAN321C
C                                                                               MAN322C
C     5.) ELEMENT CARDS (I5,1X,A6)                                              MAN323C
C                                                                               MAN324C
C       COL  1 TO  5     NUMEL = NUMBER OF ELEMENTS                             MAN325C
C       COL  7 TO 12     MUST CONTAIN ELEMEN                                    MAN326C
C                                                                               MAN327C
C       SUBSEQUENT CARDS (4I5,20I3/20I4)                                        MAN328C
C                                                                               MAN329C
C     CARD 1.                                                                   MAN330C
C                                                                               MAN331C
C       COL  1 TO  5     ELEMENT NUMBER                                         MAN332C
C       COL  6 TO 10     MATERIAL NUMBER                                        MAN333C
C       COL 11 TO 15     NUMBER OF SUBSEQUENT ELEMENTS USING SAME               MAN334C
C                        STIFFNESS MATRIX * SAVES RECOMPUTATION OF              MAN335C
C                        SIMILAR MATRICES.  ELEMENT MUST ALSO HAVE              MAN336C
C                        SAME ELEMENT FORCE VECTOR * IF THESE ARE               MAN337C
C                        IN THE STIFFNESS SUBROUTINE *                          MAN338C
C       COL 16 TO 20     PRINT ELEMENT MATRIX IF NONZERO.                       MAN339C
C       COL 21 TO 23     IXD(1) ELEMENT INCREMENT ARRAY ON NODE 1.              MAN340C
C       COL 24 TO 26     IXD(2)  * IF NOT INPUT IS SET AUTOMATICALLY            MAN341C
C                        UP TO                                                  MAN342C
C       COL 78 TO 80     IXD(20)   FOR SERENDIPITY ELEMENTS * SEE REPORT        MAN343C
C                                                                               MAN344C
C     CARD 2.                                                                   MAN345C
C                                                                               MAN346C
C       COL  1 TO  4     NODE 1                                                 MAN347C
C       COL  5 TO  8     NODE 2                                                 MAN348C
C       COL  9 TO 12     NODE 3          IX(NELI+1) ARRAY                       MAN349C
C                            CONTINUE IN I4 FORMAT TO A MAXIMUM                 MAN350C
C       COL 77 TO 80     NODE 20                                                MAN351C
C                                                                               MAN352C
C     NODES MUST BE NUMBERED AROUND ELEMENT BY RIGHT HAND SCREW RULE.           MAN353C
```

```
C      ELEMENT CARDS MUST BE IN ORDER. MISSING ELEMENTS ARE GENERATED BY  MAN354C
C      INCREMENTING NODES.                                               MAN355C
C        LAST ELEMENT CARD MUST NOT BE GENERATED.                        MAN356C
C      * TERMINATE ON ELEMENT NUMEL OR A BLANK CARD *                    MAN357C
C                                                                        MAN358C
C      5.1) BLOCK GENERATOR. GENERATES ALL MESH DATA.(6A,A6)             MAN359C
C                                                                        MAN360C
C        COL  1 TO  5     NUMBER OF NODAL POINTS TO BE GENERATED.        MAN361C
C        COL  7 TO 12     MUST CONTAIN BLOCK                             MAN362C
C                                                                        MAN363C
C        SUBSEQUENT CARDS (10I5/6I5/(10X,3F10.0))                        MAN364C
C                                                                        MAN365C
C      CARD 1.                                                           MAN366C
C                                                                        MAN367C
C        COL  1 TO  5     NN, NUMBER OF POINTS REQUIRED TO DEFINE       MAN368C
C                         BOUNDARY OF REGION. FOR 2-DIM., NN=4 OR 8.     MAN369C
C                         FOR 3-DIM., NN= 8 OR 20.                       MAN370C
C        COL  6 TO 10     NUMBER OF ELEMENTS IN R-DIRECTION.             MAN371C
C        COL 11 TO 15     NUMBER OF ELEMENTS IN S-DIRECTION.             MAN372C
C        COL 16 TO 20     NUMBER OF ELEMENTS IN T-DIRECTION.             MAN373C
C        COL 21 TO 25     INITIAL NODE NUMBER, DEFAULT = 1.              MAN374C
C        COL 26 TO 30     INITIAL ELEMENT NUMBER, DEFAULT = 1.           MAN375C
C        COL 31 TO 35     MATERIAL NUMBER OVER REGION. DEFAULT = 1       MAN376C
C        COL 36 TO 40     BOUNDARY CODE SKIP. A NON-ZERO ENTRY WILL OMIT MAN377C
C                         SETTING ALL INTERIOR BOUNDARY CODES TO ZERO.    MAN378C
C        COL 41 TO 45     IREUSE - REUSE ELEMENT STIFFNESS OPTION - USES MAN379C
C                         EACH ELEMENT STIFFNESS IREUSE TIMES BEFORE      MAN380C
C                         GENERATING A NEW ELEMENT STIFFNESS MATRIX.      MAN381C
C        COL 46 TO 50     ELEMENT STIFFNESS-PRINT. A NON-ZERO ENTRY  WILL MAN382C
C                         CAUSE PRINT-OUT OF FIRST ELEMENT.               MAN383C
C        COL 51 TO 55     IMSH - IF NONZERO SUPPRESSES PRINT OF NODES.    MAN384C
C        COL 56 TO 60     IELM - IF NONZERO SUPPRESSES PRINT OF ELEMENTS. MAN385C
C                                                                        MAN386C
C      CARD 2.  (BOUNDARY CODE AS DEFINED IN NODAL CARD.)                MAN387C
C                                                                        MAN388C
C        COL  1 TO 10     BOUNDARY CODE OVER FACE -R.                    MAN389C
C        COL 11 TO 20     BOUNDARY CODE OVER FACE +R.                    MAN390C
C        COL 21 TO 30     BOUNDARY CODE OVER FACE -S.                    MAN391C
C        COL 41 TO 50     BOUNDARY CODE OVER FACE +S.                    MAN392C
C        COL 41 TO 50     BOUNDARY CODE OVER FACE -T.                    MAN393C
C        COL 51 TO 60     BOUNDARY CODE OVER FACE +T.                    MAN394C
C                                                                        MAN395C
C      CARD 3.  (REPEAT NN TIMES.)                                       MAN396C
C                                                                        MAN397C
C        COL 11 TO 20     1-COORDINATE OF BOUNDARY-DEFINING-POINT.       MAN398C
C        COL 21 TO 30     2-COORDINATE OF BOUNDARY-DEFINING-POINT.       MAN399C
C        COL 31 TO 40     3-COORDINATE OF BOUNDARY-DEFINING-POINT.       MAN400C
C                                                                        MAN401C
C      NOTE, BLOCK GENERATES ONLY 4 PT. QUADRILATERALS OR 8 PT. BRICKS.  MAN402C
C      INPUT OF CARDS 3.) FOLLOW ORDER RULES FOR ELEMENT INPUT (SEE 5.)  MAN403C
C        R-S-T ARE LOCAL COORDINATES. I.E. (-1 .LE. R,S,T .LE. 1. )      MAN404C
C      WHERE R IS DIRECTED FROM NODE 1 TO 2, S IS IN PLANE OF FIRST      MAN405C
C      THREE NODES, AND T IS NORMAL TO R-S PLANE.                        MAN406C
C                                                                        MAN407C
C      6.)  VECTOR CARDS, I.E. USER DEFINED INPUT (IS,1X,A6)             MAN408C
C                                                                        MAN409C
C        COL  1 TO  5     NVEC, NUMBER OF DIFFERENT VECTORS (7 MAX)      MAN410C
C        COL  7 TO 12     MUST CONTAIN VECTOR                            MAN411C
C                                                                        MAN412C
C        SUBSEQUENT CARDS                                                MAN413C
C                                                                        MAN414C
C      CARD 1.  (2IS)                                                    MAN415C
C                                                                        MAN416C
C        COL  1 TO  5     NSIZV, VECTOR LENGTH,COMMON TO ALL NVEC VECTORS MAN417C
C        COL  6 TO 10     IPICK, CODED PARAMETER.                        MAN418C
C                                                                        MAN419C
C                         IPICK = 0, VECTORS ASSOCIATED WITH NODES       MAN420C
C                         IPICK = 1, VECTORS ASSOCIATED WITH DEG.FREEDOM MAN421C
C                         IPICK = 2, VECTORS ASSOCIATED WITH ELEMENTS    MAN422C
C                                                                        MAN423C
C      CARD 2.  (6X, 2A6) REPEAT NVEC TIMES                              MAN424C
C                                                                        MAN425C
```

```
C        COL  7 TO 18      DESCRIPTIVE TITLE FOR VECTOR                          MAN426C
C                                                                                MAN427C
C      CARD 3.  (2I5,7F10.0)                                                     MAN428C
C                                                                                MAN429C
C        COL  1 TO  5      POSITION NUMBER OF VECTOR ELEMENT, 1 TO NSIZV         MAN430C
C        COL 11 TO 20      VECTOR ELEMENT VALUE OF VECTOR 1                      MAN431C
C        COL  6 TO 10      GENERATOR INCREMENT                                   MAN432C
C        COL 21 TO 30      VECTOR ELEMENT VALUE OF VECTOR 2                      MAN433C
C        COL ........      AS REQUIRED FOR NVEC VECTORS                          MAN434C
C                                                                                MAN435C
C      LINEAR INTERPOLATION IS PERFORMED ON ALL VECTORS BETWEEN                  MAN436C
C      NON-CONSECUTIVE POSITION NUMBERS SPECIFIED IN COL 1 TO 5 IF               MAN437C
C      INCREMENT IS NONZERO.                                                     MAN438C
C      IF DESCRIPTIVE TITLES OF ALL VECTORS ARE BLANK CARDS,PRINTING             MAN439C
C      OF THE VECTOR VALUES IS SUPRESSED.                                        MAN440C
C                                                                                MAN441C
C      * TERMINATE ON BLANK CARD *                                               MAN442C
C                                                                                MAN443C
C      6.1)  INITIAL CONDITIONS FOR TIME DEPENDENT ANALYSIS.                     MAN444C
C                                                                                MAN445C
C        COL  1 TO  5      NICD, NUMBER OF INITIAL CONDITION VECTORS             MAN446C
C        COL  7 TO 12      MUST CONTAIN INITIA                                   MAN447C
C                                                                                MAN448C
C        SUBSEQUENT CARDS                                                        MAN449C
C                                                                                MAN450C
C      CARD 1. (6X,2A6) REPEAT NICD TIMES                                        MAN451C
C                                                                                MAN452C
C        COL  7 TO 18      DESCRIPTIVE TITLE FOR INITIAL CONDITIONS              MAN453C
C                                                                                MAN454C
C      CARD 2.  (2I5,7F10.0)                                                     MAN455C
C                                                                                MAN456C
C        COL  1 TO  5      POSITION NUMBER, AS IN VECTOR CARDS FOR IPICK=1       MAN457C
C        COL  6 TO 10      GENERATOR INCREMENT                                   MAN458C
C        COL 11 TO 20      INITIAL CONDITION 1                                   MAN459C
C        COL 21 TO 30      INITIAL CONDITION 2                                   MAN460C
C        COL ........      AS REQUIRED FOR NICD INITIAL CONDITIONS                MAN461C
C                                                                                MAN462C
C      INTERPOLATION BETWEEN INPUT VALUES AS DESCRIBED IN VECTOR INPUT.          MAN463C
C      **** NOTE **** IF MISSING THE INITIAL CONDITIONS ARE SET ZERO             MAN464C
C
C      6.2)  TWO AND THREE DIMENSIONAL PLOTTING PORTHOLE. (TAPE30)
C
C        COL  1 TO  5      NSTEP, NUMBER OF TIME STEPS TO BE PLOTTED
C        COL  7 TO 12      MUST CONTAIN PLOT
C
C      THIS PLOT CONTROL CARD WRITES ALL INFORMATION TO TAPE30
C      FOR USE IN PROGRAM PLOT2D IF NDIM = 2. IF NDIM = 3, A THREE
C      DIMESIONAL  PORTHOLE RECORD IS WRITTEN IN SUBROUTINE PLOT2D.
C
C                                                                                MAN465C
C      7.) FORCE CARDS (I5,1X,A6)                                                MAN466C
C                                                                                MAN467C
C        COL  1 TO  5      LAST NODE TO WHICH A FORCE IS TO BE SPECIFIED          MAN468C
C        COL  7 TO 12      MUST CONTAIN FORCE                                    MAN469C
C                                                                                MAN470C
C        SUBSEQUENT CARDS (I5,5X,6F10.0)                                         MAN471C
C                                                                                MAN472C
C        THE FOLLOWING VALUES ARE EACH INTERPRETTED AS FORCES IF THE             MAN473C
C        CORRESPONDING BOUNDARY CODE IS A 0 *ZERO* AND AS A DISPLACEMENT         MAN474C
C        IF THE CORRESPONDING BOUNDARY CODE IS 1 *ONE*.                          MAN475C
C                                                                                MAN476C
C        COL  1 TO  5      NODE TO WHICH FORCE OR DISPLACEMENT IS APPLIED         MAN477C
C        COL 11 TO 20      VALUE OF 1 FORCE/DISPLACEMENT                         MAN478C
C        COL 21 TO 30      VALUE OF 2 FORCE/DISPLACEMENT     * AS *              MAN479C
C        COL 31 TO 40      VALUE OF 3 FORCE/DISPLACEMENT * REQUIRED *            MAN480C
C        COL 41 TO 50      VALUE OF 4 FORCE/DISPLACEMENT                         MAN481C
C        COL 51 TO 60      VALUE OF 5 FORCE/DISPLACEMENT                         MAN482C
C        COL 61 TO 70      VALUE OF 6 FORCE/DISPLACEMENT                         MAN483C
C                                                                                MAN484C
C                                                                                MAN485C
C      7.1) SURFACE LOAD CARDS (I5,1X,A6)                                        MAN486C
C                                                                                MAN487C
```

```
C       COL  1 TO  5    NUMBER OF LOADED FACE CARDS                          MAN488C
C       COL  7 TO 12    MUST CONTAIN BLOADS                                  MAN489C
C                                                                            MAN490C
C     CARD 1. (I5,1X,A5,I4,8I5,8I3)                                          MAN491C
C                                                                            MAN492C
C       COL  1 TO  5    DIMENSION OF LOADING SURFACE, (1 OR 2).              MAN493C
C       COL  7 TO 11    SLD(NN), ALPHA-NUMERIC NAME OF SURFACE LOADING       MAN494C
C                       SUBROUTINE (NN IS BETWEEN 1 AND 5)                   MAN495C
C       COL 12 TO 15    NRT,NUMBER OF ADDITIONAL ELEMENT LOAD                MAN496C
C                       SURFACES TO BE GENERATED FROM CURRENT MODEL.         MAN497C
C       COL 16 TO 20    IPRES(N), NODE NUMBERS DEFINING LOADING SURFACE      MAN498C
C       COL 21 TO 25    OF CURRENT ELEMENT.                                  MAN499C
C       COL ...         (IDENTIFY FROM 2 TO 8 AS REQUIRED)                   MAN500C
C       COL 51 TO 55                                                         MAN501C
C       COL 56 TO 58    INC(N), INCREMENT VALUE ADDED TO IPRES(N) TO         MAN502C
C       COL 59 TO 61    IDENTIFY NODE NUMBERS OF A GENERATED SEQUENCE.       MAN503C
C       COL ...         (IDENTIFY FROM 2 TO 8 AS REQUIRED)                   MAN504C
C                                                                            MAN505C
C     CARD 2. (8F10.0)                                                       MAN506C
C                                                                            MAN507C
C       COL  1 TO 80    LOAD AT NODES GIVEN ON PREVIOUS CARD.                MAN508C
C                       MUST CORRESPOND IN SEQUENCE TO THE NODE NUMBERS      MAN509C
C                                                                            MAN510C
C                                                                            MAN511C
C                                                                            MAN512C
C       7.2) ELEMENT LOAD CARDS (I5,1X,A6 )                                  MAN513C
C                                                                            MAN514C
C       COL  1 TO  5    NLD, NUMBER OF ELEMENT LOAD CARDS.                   MAN515C
C       COL  7 TO 11    MUST CONTAIN ELOADS                                  MAN516C
C                                                                            MAN517C
C       SUBSEQUENT CARDS (I5,1X,A5,I4,I5,6F10.0)                             MAN518C
C                                                                            MAN519C
C       COL  1 TO  5    IEL, INITIAL ELEMENT OF A GENERATED SEQUENCE.        MAN520C
C       COL  7 TO 11    ELM(NN), ALPHA-NUMERIC NAME OF ELEMENT               MAN521C
C                       SUBROUTINE WHERE ELEMENT LOADS ARE COMPUTED.         MAN522C
C                       USED AS CHECK TO INSURE IEL, ET.2 ARE PROPER         MAN523C
C                       ELEMENTS.                                            MAN524C
C       COL 12 TO 15    INC, INCREMENT NUMBER IN A GENERATED SEQUENCE.       MAN525C
C                       (DEFAULT = 1).                                       MAN526C
C       COL 16 TO 20    JEL, TERMINAL ELEMENT NUMBER IN A GENERATED          MAN527C
C                       SEQUENCE, IF JEL = 0, ONLY IEL IS COUNTED.           MAN528C
C       COL 21 TO 80    USER DEFINED VALUES FOR DETERMINING BODY LOADS       MAN529C
C                       IN THE ISW=5 PORTION OF ELM(NN).                     MAN530C
C                                                                            MAN531C
C       NOTE, USER MUST PROVIDE COMPUTATION OF LOADS IN ELMNN.               MAN532C
C       PR IS TRANSFERED TO SUBROUTINE ELMNN IN THE U VECTOR,                MAN533C
C       WHEN ISW =5, ONLY.                                                   MAN534C
C                                                                            MAN535C
C     7.3) PROPORTIONAL LOADS FOR TIME DEPENDENT ANALYSIS                    MAN536C
C                                                                            MAN537C
C       TRANSFER TO THIS OPTION OCCURS ONLY FOR TIME ANALYSES.               MAN538C
C                                                                            MAN539C
C       ONE CARD FOR EACH PROPORTIONAL LOAD REQUIRED                         MAN540C
C                                                                            MAN541C
C       COL  1 TO  5    PROPORTIONAL LOAD TYPE, 1,2 OR 3                     MAN542C
C       COL  6 TO 10    K, TABLE CONSTANT                                    MAN543C
C       COL 11 TO 20    TMIN, SMALLEST TIME LOADING IS VALID                 MAN544C
C       COL 21 TO 30    TMAX, LARGEST TIME LOADING IS VALID                  MAN545C
C       COL 31 TO 40    A0                                                   MAN546C
C       COL 41 TO 50    A1                                                   MAN547C
C       COL 51 TO 60    A2                                                   MAN548C
C       COL 61 TO 70    A3                                                   MAN549C
C       COL 71 TO 80    A4                                                   MAN550C
C                                                                            MAN551C
C       LOAD TYPE 1.  T = TIME                                               MAN552C
C                                                                            MAN553C
C           PROP = A0 + A1*T + A2*T*T + A3*T*T*T + A4*T*T*T*T                MAN554C
C                                                                            MAN555C
C       LOAD TYPE 2.                                                         MAN556C
C                                                                            MAN557C
C           PROP = A0*(SIN(A1*T))**K + A2*(COS(A3*T))**K + A5                MAN558C
C                                                                            MAN559C
```

```
C        LOAD TYPE 3.                                                          MANSA0C
C                                                                              MANSA1C
C            PROP = USER DEFINED FUNCTION FROM SUBROUTINE EXPRLD(PROP,T)       MANSA2C
C                                                                              MANSA3C
C        **NOTE** PROPORTIONAL LOADS CAN BE ACCUMULATED FROM DIFFERENT         MANSA4C
C                 TYPES AT THE SAME TIME.                                      MANSA5C
C                                                                              MANSA6C
C     8.) INITIATION OF TIME INDEPENDENT SOLUTION (I5,1X,A6)                   MANSA7C
C                                                                              MANSA8C
C        COL  1 TO  4     IOUT, OUTPUT CONTROL CODE.                           MANSA9C
C                                                                              MANSB0C
C                         IOUT .EQ. 0, ALL STRESSES AND DISP. PRINTED          MANSB1C
C                         IOUT .NE. 0, SELECTED PRINTOUT, MORE DATA INPUT      MANSB2C
C                         SEE SECTION 9 FOR DATA PREPARATION.                  MANSB3C
C                                                                              MANSB4C
C        COL  7 TO 12     MUST CONTAIN  SOLVE  *INDICATES ALL DATA INPUT*      MANSB5C
C                         COMPLETE FORMULATION AND SOLUTION OF EQUATIONS.      MANSB6C
C        COL  7 TO 12     MUST CONTAIN  RESOLV TO OBTAIN SUBSEQUENT            MANSB7C
C                         SOLUTIONS WHERE BOUNDARY CODES DO NOT CHANGE         MANSB8C
C                         AND ALL PRESCRIBED DISPLACEMENTS ARE ZERO.           MANSB9C
C                                                                              MANSC0C
C     8.1) INITIATION OF DYNAMIC SOLUTION BY EXPLICIT INTEGRATION.             MANSC1C
C                                                                              MANSC2C
C        COL  1 TO  4     IOUT,  OUTPUT CONTROL FOR DISPLACEMENT AND           MANSC3C
C                         STRESS PRINTOUT. SEE SECT. 9 FOR DATA INPUT.         MANSC4C
C                         IF IOUT .NE. 0, THE SPATIAL CONTROL DATA             MANSC5C
C                         COMES AT THE END OF THE DYNAMIC SEGMENT.             MANSC6C
C        COL  7 TO 12     MUST CONTAIN EXPLIC                                  MANSC7C
C                                                                              MANSC8C
C        SUBSEQUENT CARDS (2I5,2F10.0,2I5)                                     MANSC9C
C                                                                              MANSD0C
C        COL  1 TO  5     NUMBER OF TIME STEPS                                 MANSD1C
C        COL  6 TO 10     PRINT INTERVAL                                       MANSD2C
C        COL 11 TO 20     TIME INCREMENT                                       MANSD3C
C        COL 21 TO 30     NEWMARK BETA-DAMPING TERM (GAMMA - .5)               MANSD4C
C        COL 31 TO 35     NUMBER OF TIME EVOLUTION STRESS PLOTS.               MANSD5C
C        COL 36 TO 40     NPROP, NUMBER OF PROPORTIONAL LOADS TO BE INPUT      MANSD6C
C        COL 41 TO 45     NFORC, LAST NODE ON WHICH A FORCE IS CHANGED          MANSD7C
C                         DURING EACH TIME STEP.                               MANSD8C
C        COL 46 TO 50     KKK, STABILITY CHECK OVERRIDE ** CAUTION USE         MANSD9C
C                         ONLY WHEN A BETTER ESTIMATE OF THE STABLE TIME       MANSE0C
C                         STEP IS AVAILABLE THAN CAN BE PERFORMED BY CODE      MANSE1C
C                         KKK ZERO, USES INTERNAL STABILITY CHECK.             MANSE2C
C                         KKK NONZERO, DISREGARDS STABILITY CHECK.             MANSE3C
C                                                                              MANSE4C
C        SUBSEQUENT CARDS (3I5) ONE FOR EACH STRESS PLOT.                      MANSE5C
C                                                                              MANSE6C
C        COL  1 TO  5     ELEMENT NUMBER CONTAINING STRESS TO BE PLOTTED.      MANSE7C
C        COL  6 TO 10     LOCAL COORDINATE POINT CODE, 1 TO 7, AS              MANSE8C
C                         PATTERNED AFTER, COL 11 TO COL 17, IN SOLVE.         MANSE9C
C        COL 11 TO 15     STRESS COMPONENT CODE, 1 TO 6 FOR SIGMA(I,J),        MANSF0C
C                         I.E., SIGMA(1,1)=1, SIGMA(1,2)=2, SIGMA(1,3)=3,      MANSF1C
C                         SIGMA(2,2) = 4, SIGMA(2,3) = 5, SIGMA(3,3) = 6.      MANSF2C
C                                                                              MANSF3C
C        IF(NPROP.NE.0) READ PROPORTIONAL LOAD CARDS, SEE SECT. 7.3            MANSF4C
C                                                                              MANSF5C
C        IF(NFORC.NE.0) READ FORCE CARDS AT EACH TIME STEP. IF OUTPUT IS       MANSF6C
C        LIMITED BY IOUT NONZERO, THE FIRST FORCE CARD SET PRECEDES            MANSF7C
C        OUTPUT CARDS AND THE REMAINDER FOLLOW THE OUTPUT CARDS NO BLANK       MANSF8C
C        CARDS MAY BE USED BETWEEN SETS OF CARDS OTHER THAN THE USUAL          MANSF9C
C        BLANK TERMINATOR CARD FOR FORCE INPUT CARDS.                          MANSG0C
C                                                                              MANSG1C
C        IF(IOUT.NE.0)   DATA FOR SPATIAL PRINTOUT CONTROL, SEE SECT.9.        MANSG2C
C                                                                              MANSG3C
C                                                                              MANSG4C
C     SPECIAL COMMENTS FOR DYNAMIC OPTION                                      MANSG5C
C                                                                              MANSG6C
C     (1) ONLY COLUMNS 1 TO 60 ARE AVAILABLE FOR PAGE HEADING.                 MANSG7C
C     (2) MAXIMUM ADVANTAGE OF ELEMENT REUSE OPTION SHOULD BE TAKEN.           MANSG8C
C     (3) INITIAL CONDITIONS FOR DISPLACEMENT AND VELOCITY VECTORS,            MANSG9C
C         AS WELL AS STORAGE FOR ACCELERATION VECTOR, MAY BE MADE              MANSH0C
C         THROUGH INPUT OF AN INITIAL CONDITION CARD SET, WITHOUT              MANSH1C
```

```
C        SPECIFIED INITIAL CONDITIONS THEY ARE AUTOMATICALLY SET ZERO.   MANA32C
C    (4) SPATIAL LOADING IS INPUT THROUGH FORCE OR BOUNDARY               MANA33C
C        PRESSURE CARDS. ALL LOADS VARY PROPORTIONALY WITH TIME           MANA34C
C    (5) EXTREME CAUTION ON ORDER OF DATA CARDS MUST BE OBSERVED.  NO     MANA35C
C        EXTRA CARDS ARE PERMITTED AND STRICT COUNTS ARE OBSERVED         MANA36C
C        EXCEPT FOR THE NUMBER OF FORCE CARDS USED IN EACH TIME STEP.     MANA37C
C                                                                         MANA38C
C    8.2) INITIATION OF IMPLICIT TIME INTEGRATIONS (I5,1X,A6)             MANA39C
C                                                                         MANA40C
C      COL  1 TO  5    NSEQ, NUMBER OF TIME SEQUENCES                     MANA41C
C      COL  7 TO 12    MUST CONTAIN VISCOE FOR LINEAR VISCOELASTIC        MANA42C
C                      QUASI-STATIC PROBLEMS (ONE INITIAL CONDITION       MANA43C
C                      ONLY MUST BE USED)                                 MANA44C
C      COL  7 TO 12    MUST CONTAIN IMPLIC FOR DYNAMIC IMPLICIT           MANA45C
C                      INTEGRATIONS (THREE INITIAL CONDITIONS ARE         MANA46C
C                      REQUIRED, MORE CAN BE SPECIFIED WITHOUT ERROR)     MANA47C
C                                                                         MANA48C
C      SUBSEQUENT CARDS, ONE SET FOR EACH TIME SEQUENCE                   MANA49C
C                                                                         MANA50C
C    CARD 1. (F10.0,8I5,2F10.0)                                           MANA51C
C                                                                         MANA52C
C      COL  1 TO 10    DT, TIME INCREMENT (NONZERO FOR IMPLIC)            MANA53C
C      COL 11 TO 15    NTS, NUMBER OF TIME STEPS IN SEQUENCE              MANA54C
C      COL 16 TO 20    INT, PRINT INTERVAL (DEFAULT 1)                    MANA55C
C      COL 21 TO 25    NNI, FIRST NODE PRINTED                            MANA56C
C      COL 26 TO 30    NNE, LAST  NODE PRINTED                            MANA57C
C      COL 31 TO 35    NEI, FIRST ELEMENT STRESS TO BE PRINTED            MANA58C
C      COL 36 TO 40    NEF, LAST  ELEMENT STRESS TO BE PRINTED            MANA59C
C      COL 41 TO 45    NPROP, NUMBER OF PROPORTIONAL LOADS IN SEQUENCE    MANA60C
C      COL 46 TO 50    NFORC, LAST NODE FOR GENERALIZED FORCES TO BE      MANA61C
C                      INPUT FOR EACH TIME IN SEQUENCE (SEE SECT.7.       MANA62C
C                      FOR DATA PREPARATION FORMATS)                      MANA63C
C      COL 51 TO 60    BETA, NEWMARK INTEGRATION PARAMETER (IMPLIC)       MANA64C
C      COL 61 TO 70    DEL = GAMMA - 0.5, NEWMARK INTEGRATION             MANA65C
C                      PARAMETER (IMPLIC)                                 MANA66C
C                                                                         MANA67C
C    CARD 2. (2I5,7F10.0)                                                 MANA68C
C                                                                         MANA69C
C      ONE FOR EACH NPROP. SEE SECT.7.3 FOR DATA PREPARATION              MANA70C
C                                                                         MANA71C
C                                                                         MANA72C
C      SUBSEQUENT CARDS FOR EACH TIME STEP IN THE SEQUENCE                MANA73C
C                                                                         MANA74C
C      FORCE CARDS.  SEE SECTION 7. FOR DATA PREPARATION FORMATS.         MANA75C
C                                                                         MANA76C
C
C    *** SPECIAL USE OF SECTION 8.2 FOR STATIC RUNS WITH REPEATED
C        LOAD CASES ...
C
C            NSEQ = 1
C            CONTROL WORD = VISCOE
C            DT = -1.  (MINUS)
C            NTS = NUMBER OF LOAD CASES PLUS 1
C            INT = 1
C            NPROP = 0
C            NNI,NNE,NEI,NEF AS DEFINED ABOVE
C            NFORC = HIGHEST NODE OF ALL LOAD CASES WITH NON-ZERO LOAD
C
C    *** USE FORCE DATA INPUT, SECTION 7., FOR EACH LOAD CASE
C        BACK TO BACK, WHERE LAST NODE = NFORC. NOTE LAST NODE
C        MAY OR MAY NOT HAVE NON-ZERO FORCES. THE LAST LOAD CASE
C        IS A DUMMY CASE.
C
C                                                                         MANA77C
C    9.) OUTPUT CONTROL FOR LIMITED PRINTS                                MANA78C
C                                                                         MANA79C
C    DISPLACEMENT OUTPUT CONTROL, IF IOUT .NE. 0                          MANA80C
C                                                                         MANA81C
C    CARD 1.  (I5)                                                        MANA82C
C                                                                         MANA83C
C      COL  1 TO  5    NUMDIS - NUMBER OF DISPLACEMENT PRINT CARDS        MANA84C
C                                                                         MANA85C
```

```
C         SUBSEQUENT CARDS (2I5) SKIP IF NUMDIS = 0                          MANA86C
C                                                                            MANA87C
C         COL  1 TO  5     NODAL NUMBER TO BE OUTPUT.                        MANA88C
C         COL  6 TO 10     HIGHER NODE NUMBER OF A GENERATED SEQUENCE.       MANA89C
C                          IF ZERO JUST FIRST NODE IS COUNTED.               MANA90C
C         COL 11 TO 15     INCREMENT TO GENERATOR, DEFAULT = 1               MANA91C
C                          *** REPEAT UNTIL,NUMDIS CARDS HAVE BEEN READ      MANA92C
C                                                                            MANA93C
C       STRESS OUTPUT CONTROL, IF IOUT .NE. 0.                               MANA94C
C                                                                            MANA95C
C       CARD 1.  (I5,5X,711)                                                 MANA96C
C                                                                            MANA97C
C         COL  1 TO  5     NUMSTR - NUMBER OF STRESS OUTPUT CARDS            MANA98C
C         COL 11 TO 17     NSIG(7) - PRINT PATTERN WITHIN AN ELEMENT.        MANA99C
C                          LOCAL POINTS OF EACH ELEMENT CAN BE               MAN700C
C                          SUPRESSED BY NON-ZERO ENTRIES AS FOLLOWS,         MAN701C
C                                                              E.G.          MAN702C
C               COL  11    SUPRESS PRINT AT LOCAL POINT 1, ( 0, 0, 0)        MAN703C
C               COL  12    SUPRESS PRINT AT LOCAL POINT 2, (-1, 0, 0)        MAN704C
C               COL  13    SUPRESS PRINT AT LOCAL POINT 3, ( 1, 0, 0)        MAN705C
C               COL  14    SUPRESS PRINT AT LOCAL POINT 4, ( 0,-1, 0)        MAN706C
C               COL  15    SUPRESS PRINT AT LOCAL POINT 5, ( 0, 1, 0)        MAN707C
C               COL  16    SUPRESS PRINT AT LOCAL POINT 6, ( 0, 0,-1)        MAN708C
C               COL  17    SUPRESS PRINT AT LOCAL POINT 7, ( 0, 0, 1)        MAN709C
C                                                                            MAN710C
C         SUBSEQUENT CARDS (2I5) SKIP IF NUMSTR = 0                          MAN711C
C                                                                            MAN712C
C         COL  1 TO  5     ELEMENT NUMBER TO BE PRINTED.                     MAN713C
C         COL  6 TO 10     HIGHER ELEMENT NUMBER OF A GENERATED SEQUENCE.    MAN714C
C                          IF ZERO ONLY FIRST ELEMENT IS COUNTED.            MAN715C
C         COL 11 TO 15     INCREMENT TO GENERATOR, DEFAULT = 1               MAN716C
C                          *** REPEAT UNTIL,NUMSTR CARDS HAVE BEEN READ      MAN717C
C                                                                            MAN718C
C                                                                            MAN719C
C*************************************************************************** MAN720C
C                                                                            MAN721C
      RETURN                                                                 MAN722C
      END                                                                    MAN723C
*DECK FELMT
      SUBROUTINE FELMT(IUT,ISZDT,MHAND)                                      FEL   1C
C                                                                            FEL   2C
C**** MAIN ASSEMBLY PROGRAM AND STORAGE ALLOCATION FOR VARIABLES             FEL   3C
C                                                                            FEL   4C
      REAL LABL                                                              FEL   5C
      LOGICAL CHECK,FLAG,NPL                                                 FEL   6C
      DIMENSION  CONWD(24),FLAG(7),TITLE(12),TYPE(7),DT(ISZDT),HE(2)         FEL   7C
      DIMENSION NEWBUF (15)
      COMMON /TAPES/ IIP5,IIP6                                               FEL   8C
      COMMON/FORTER/NF(50),XN,F1,F2,F3,F4                                    FEL   9C
      COMMON/GAUS/ LIM,SGAUSS(5,5),WGAUSS(5,5)                               FEL  10C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UM,RHED(6),RM     FEL  11C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)          FEL  12C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SH(3,3),X(3,20),LD(120)          FEL  13C
      COMMON/TITLES/ XTIL(3),UTTL(6)                                         FEL  14C
      COMMON/VALUES/ NC1,CON1,CON2,CON3(3)                                   FEL  15C
      COMMON/DYNAMO/TIME,NSIG(7),NI,NSTEP,DS,NUMPLT,NEDATA(20,3),NPR,NPL     FEL  16C
      COMMON/PLOTS/IPLOT
CXXXXXX
C  PUT IN COMMON TO HOLD NODE RENUMBERING ARRAY
CXXXXXX
      COMMON /MINHW/ NEW(1500)
      DATA CONWD/6HMATERI,6HNODAL ,6HPOLAR ,6HELEMEN,6HSOLVE ,6HRESOLV,      FEL  17C
     X 6HROUNDA,6HFORCE ,6HLOADS,6HELOADS,6HVECTOR,6HMESH  ,6HPLOT  ,
     X 6HFOURIE,6HREMARK,6HINITIA,6HGENERA,6HTITLE ,6HBLOCK ,6HEXPLIC,      FEL  19C
     X 6HADDUP ,6HVISCOE,6HIMPLIC,6HNEWTON/                                  FEL  20C
      DATA HE/4HNOT ,4H    /, BLANK/6H      /                                FEL  21C
      DATA NFLAG,NLIST/7,24/,M1,M2,M3,M4,M5,M6,MB/7*1/,ITA/9/                FEL  22C
      DATA MHW1/1H1/
      REWIND 30
C                                                                            FEL  23C
C**** INITIATE SEARCH FOR START OF PROBLEM                                   FEL  24C
  100 READ (IIP5,1000) CC,HEAD,MHW
```

```
      CC=HEAD(1)                                                        FEL 26C
      IF(CC.EQ.START) GO TO 240                                         FEL 27C
      IF(CC.EQ.6HFEAP73) GO TO 240
      IF(CC.EQ.CEASE) RETURN                                            FEL 28C
      GO TO 100                                                         FEL 29C
120   CONTINUE                                                          FEL 30C
      DO 122 I=1,12                                                     FEL 31C
122   HEAD(I)=TITLE(I)                                                  FEL 32C
CXXXX
C  IF CC 79 OF FEAP73 CARD IS NON-ZERO THEN READ NODE
C  CORRESPONDENCE ARRAY PUNCHED IN PREPROCESSOR
CXXXX
  240 DO 600 I=1,1500
  600 NEW(I)=I
      IF(MBAN.NE.MBW1) GO TO 650
      J=0
      K=0
  605 READ (5,601) I,NEWBUF
  601 FORMAT (16I5)
      J=J+1
      IF(I.EQ.9999) GO TO 650
      IF(I.LT.J) GO TO 610
      WRITE (6,602)
      STOP 1
  602 FORMAT (50H0** ERROR **  NODE RESEQUENCING CARD OUT OF ORDER.     )
  610 DO 615 I=1,15
      K=K+1
      IF(NEWBUF(I).NE.0) NEW(K)=NEWBUF(I)
  615 CONTINUE
      GO TO 605
C
  650 READ (ITPS,1000) CC,TITLE
      NDIM = CC                                                         FEL 34C
      DO 151 I = 1,NDIM                                                 FEL 35C
      CC = TITLE(I)                                                     FEL 36C
      XHED(I) = CC                                                      FEL 37C
      IF(CC.EQ.BLANK) XHED(I) = XTIL(I)                                 FEL 38C
150   CONTINUE                                                          FEL 39C
      READ(ITPS,1000) CC,TITLE                                          FEL 40C
      NDF = CC                                                          FEL 41C
      DO 152 I = 1,NDF                                                  FEL 42C
      CC = TITLE(I)                                                     FEL 43C
      IF(CC.EQ.BLANK) CC = UTIL(I)                                      FEL 44C
      FHED(I) = CC                                                      FEL 45C
      UHED(I) = CC                                                      FEL 46C
152   PHED(I) = CC                                                      FEL 47C
      NUMEL = 0                                                         FEL 48C
      READ(ITPS,1001) NEN,NEXTRA,IREC,MBAN,IBUF,NC1,CON1,CON2,CON3      FEL 49C
      DO 153 I=1,NDIM
153   IF(CON3(I).EQ.0.0) CON3(I) = 1.0
      IF(MBAN.LE.0) MBAN = MBAND                                        FEL 50C
      ISIZ = IS/DI                                                      FEL 51C
      IDI = IS/DI/2                                                     FEL 52C
      NICO = 3                                                          FEL 53C
      NVEC = 1                                                          FEL 54C
      NPL = .TRUE.                                                      FEL 55C
      NUMPLT = 3                                                        FEL 56C
      NT = 1                                                            FEL 57C
      IPLOT = 0
C.... INITIALISE TIME AND CLOCK                                         FEL 58C
      CALL TIC(TIME,0)                                                  FEL 59C
      DO 121 I=1,7                                                      FEL 60C
  121 TIMF(I) = 0.0                                                     FEL 61C
      NFLT = NEN + 2                                                    FEL 62C
      NTERM = 0                                                         FEL 64C
      DO 10 I=1,NFLAG                                                   FEL 65C
10    FLAG(I)=.FALSE.                                                   FEL 66C
C                                                                       FEL 67C
C.... STORAGE IS SET FOR A NSTF X NSTF ELEMENT MATRIX AND FORCE VECTOR  FEL 68C
C.... MAXIMUM SIZE FOR NSTF IS 120. THIS IS CONTROLED BY DIMENSION ON LD FEL 69C
C                                                                       FEL 70C
      NSTF = NDF*NEN + NEXTRA                                           FEL 71C
```

```
      MS = 6*NSIF + 1                                                   FEL 72C
      MO = MSIF*(NSIF+2) + 1                                            FEL 73C
      IF(MS .GT. IDT) GO TO 710                                         FEL 74C
      IPG=1                                                             FEL 75C
      IF(IBUF.LE.0) IBUF = ISZDT/20                                     FEL 76C
      IF(IREC.NE.0) IREC = 1                                            FEL 77C
      WRITE(IPA,2000) HEAD,IPG,NDIM,NDF,NEN,NSIF,ME(IMEC+1),MBAN,IBUF,  FEL 78C
     X        NCI,CON1,CON2,CON3                                         FEL 79C
      IPG = IPG + 1                                                     FEL 80C
      IF(NDIM.GT.3.OR.NDF.GT.6.OR.NEN.GT.20) GO TO 720                  FEL 81C
C                                                                       FEL 82C
C**** SEARCH THE LIST OF NAMES FOR A TRANSFER ADDRESS                   FEL 83C
C                                                                       FEL 84C
  125 READ (IIPS,1000) CC,TITLE,MBW
      I = CC                                                            FEL 86C
      CC=TITLE(1)                                                       FEL 87C
      DO 126 J = 1,NLIST                                                FEL 88C
      IF(CC.EQ.CONWD(J)) GO TO 127                                      FEL 89C
  126 CONTINUE                                                          FEL 90C
      IF(CC.EQ.START) GO TO 120                                         FEL 91C
      IF(CC.EQ.BMFEAP73) GO TO 120
      IF(CC.EQ.CEASE) RETURN                                            FEL 92C
      GO TO 125                                                         FEL 93C
C                                                                       FEL 94C
C**** NAMES(MAT,NOD,POL,ELE,SOL,RES,BCS,FOR,BLD,ELD,VEC,MES,PLO,FOU,REM, FEL 95C
  127 GO TO(210,220,226,230,200,270,228,260,250,255,245,200,350,200,131,
     X 345,225,133,220,240,200,200,200,200) ,J                          FEL 97C
C****  ICO,GEN,TIT,BLO,EXP,ADD,VIS,IMP,NEW)                            FEL 98C
C                                                                       FEL 99C
  131 WRITE(IIPA,2003) (TITLE(I),I=2,12)                                FEL100C
      GO TO 125                                                         FEL101C
  133 DO 134 I = 2,12                                                   FEL102C
  134 HEAD(I) = TITLE(I)                                                FEL103C
      GO TO 125                                                         FEL104C
C                                                                       FEL105C
C**** INPUT THE MATERIAL CHARACTERIZATIONS.                             FEL106C
C                                                                       FEL107C
  210 J = 1                                                             FEL108C
      IF(FLAG(1)) GO TO 215                                             FEL109C
      FLAG(1)=.TRUE.                                                    FEL110C
      IF(I.LE.0) GO TO 700                                              FEL111C
      NUMMAT = I                                                        FEL112C
      M1 = M0                                                           FEL113C
      M2 = M0     + NUMMAT                                              FEL114C
      M0 = M2 + 64*NUMMAT                                               FEL115C
  215 IF(M0.GT.ISZDT) GO TO 710                                         FEL116C
      CALL MESH(J,NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEN,NELI,                 FEL117C
     X DT,DT(M1),DT(M2),DT(M3),DT(M4),DT(M5),DT(M6))                    FEL118C
      IF(IPG.LE.0) GO TO 100                                            FEL119C
      GO TO 125                                                         FEL120C
C                                                                       FEL121C
C**** INPUT THE NODAL LOCATIONS AND BOUNDARY CONDITION CODES.           FEL122C
C                                                                       FEL123C
  220 IF(FLAG(2)) GO TO 225                                             FEL124C
      FLAG(2)=.TRUE.                                                    FEL125C
      IF(I.LE.0) GO TO 700                                              FEL126C
      NUMNP = I                                                         FEL127C
      M3 = M0                                                           FEL128C
      M4 = M0 + NUMNP                                                   FEL129C
      M5 = M4 + NDIM*NUMNP                                              FEL130C
      M0 = M5 + NDF*NUMNP                                               FEL131C
      IF(M0.GT.ISZDT) GO TO 710                                         FEL132C
      IF(.NOT.FLAG(3)) M6 = M0                                          FEL133C
  225 CONTINUE                                                          FEL134C
      IF(CC.EQ.CONWD(19)) GO TO 227                                     FEL135C
      J = 2                                                             FEL136C
      IF(CC.EQ.CONWD(17)) J = 4                                         FEL137C
      GO TO 215                                                         FEL138C
  226 IF(.NOT.FLAG(2)) GO TO 730                                        FEL139C
      CALL CONVERT(NUMNP,NDIM,DT(M4))                                   FEL140C
      GO TO 125                                                         FEL141C
  227 CONTINUE                                                          FEL142C
```

```
      CALL HRKGEN(NUMNP,NUMEL,NDIM,NDF,NEN,NEL1,DT(M3),DT(M4),DT(M5),     FEL143C
     X   DT(M6))                                                          FEL144C
      IF(IPG.LE.0) GO TO 705                                             FEL145C
      IF(.NOT.FLAG(3))  M0 = M0 + NEL1*NUMEL                             FEL146C
      FLAG(3) = .TRUE.                                                   FEL147C
      IF(M0.GT.ISZDT) GO TO 710                                          FEL148C
      GO TO 125                                                          FEL149C
C                                                                        FEL150C
C.... BOUNDARY CODE OVERWRITE                                            FEL151C
C                                                                        FEL152C
228   IF(.NOT.FLAG(2)) GO TO 730                                         FEL153C
      CALL BCODES(NUMNP,DT(M3))                                          FEL154C
      GO TO 125                                                          FEL155C
C                                                                        FEL156C
C.... INPUT THE ELEMENT CONNECTION ARRAY.                                FEL157C
C                                                                        FEL158C
230   J = 3                                                              FEL159C
      IF(FLAG(3)) GO TO 215                                              FEL160C
      FLAG(3)=.TRUE.                                                     FEL161C
      IF(I.LE.0) GO TO 700                                               FEL162C
      NUMEL = I                                                          FEL163C
      M6 = M0                                                            FEL164C
      M0 = M0 + NEL1*NUMEL                                               FEL165C
      GO TO 215                                                          FEL166C
C                                                                        FEL167C
C.... INPUT THE VECTORS OF PRESCRIBED VALUE.                             FEL168C
C                                                                        FEL169C
245   IF(.NOT.FLAG(2)) GO TO 730                                         FEL170C
      IF(I.LE.0) GO TO 700                                               FEL171C
      IF(FLAG(4)) GO TO 247                                              FEL172C
      FLAG(4) = .TRUE.                                                   FEL173C
      READ(ITPS,1001) NSIZV,IPICK                                        FEL174C
      NVEC = I                                                           FEL175C
      MB = M0                                                            FEL176C
      MA = MB + NSIZV*I                                                  FEL177C
      M0 = MA                                                            FEL178C
      IF(M0+2*I.GT.ISZDT) GO TO 710                                      FEL179C
247   CALL VECIN (NUMNP,NUMEL,NDF,NSIZV,IPICK,NVEC,DT(MA),DT(MB) )       FEL180C
      IF(IPG.LE.0) GO TO 100                                             FEL181C
      GO TO 125                                                          FEL182C
250   CONTINUE                                                           FEL183C
      IF(I.LE.0) GO TO 125                                               FEL184C
C                                                                        FEL185C
C.... INPUT BOUNDARY LOADS                                               FEL186C
C                                                                        FEL187C
      CALL BLOADS(I,NUMNP,NDIM,NDF,DT(M3),DT(M4),DT(M5))                 FEL188C
      IF(IPG .LE. 0) GO TO 705                                           FEL189C
      GO TO 125                                                          FEL190C
C                                                                        FEL191C
C.... INPUT ELEMENT LOADS                                                FEL192C
C                                                                        FEL193C
255   IF(I.LE. 0) GO TO 125                                              FEL194C
      CALL ELOADS(I,NDIM,NDF,NEN,NEL1,NSTF,NSIZV,NVEC,DT(M1),DT(M2),     FEL195C
     ) DT(M3),DT(M4),DT(M6),DT(M5),DT,DT(M9),DT(MB))                     FEL196C
      IF(IPG .LE. 0) GO TO 705                                           FEL197C
      GO TO 125                                                          FEL198C
C                                                                        FEL199C
C.... INPUT THE INITIAL CONDITIONS                                       FEL200C
C                                                                        FEL201C
345   IF(FLAG(4)) GO TO 347                                              FEL202C
      FLAG(4) = .TRUE.                                                   FEL203C
      NICD = I                                                           FEL204C
      NSICD = NDF*NUMNP                                                  FEL205C
      MG = M0                                                            FEL206C
      M0 = MG + NICD*NSICD                                               FEL207C
      MC = M0                                                            FEL208C
      IF(M0+2*I.GT.ISZDT) GO TO 710                                      FEL209C
347   CALL VECIN (NUMNP,NUMEL,NDF,NSICD,1,NICD,DT(MC),DT(MG))           FEL210C
      IF(IPG.LE.0) GO TO 100                                             FEL211C
      GO TO 125                                                          FEL212C
C
C...  TWO AND THREE DIMESIONAL PLOTTING PORTHOLE CONTROL.
```

```
C
350   CONTINUE
      IF(NDIM.EQ.2 .AND.NEN.NE.4) GO TO 707
      CHECK = .TRUE.
      WRITE(IIP6,2040)
      DO 351 K=1,3
      IF(.NOT.FLAG(K)) CHECK = .FALSE.
351   CONTINUE
      IF(.NOT.CHECK) GO TO 707
      KSTEPS = 1
      NVEC = 8
      NSIZV = NUMEL
      IF(NDIM.EQ.3) NVEC = 2
      IF(NDIM.EQ.3) NSIZV = 6
      IF(FLAG(4)) GO TO 352
      FLAG(4) = .TRUE.
      MB = M0
      M0 = MB + NVEC*NSIZV
      IF(M0 .GT. ISZDT) GO TO 710
352   IPLOT = NDIM
      IDUM = NUMMAT
      GO TO 125
C                                                                       FEL213C
C*** START THE SOLUTION FOR FIRST LOADING CASE                          FEL214C
C                                                                       FEL215C
200   IF(FLAG(5)) GO TO 401                                             FEL216C
      CHECK=.TRUE.                                                      FEL217C
      CALL TIC(TYME,1)                                                  FEL218C
      DO 123 J=1,3                                                      FEL219C
      IF(.NOT.FLAG(J)) CHECK=.FALSE.                                    FEL220C
123   CONTINUE                                                          FEL221C
      IF(.NOT.CHECK) GO TO 707                                          FEL222C
      NTM = 0                                                           FEL223C
      IF (CC.NE.CONWD(22).AND.CC.NE.CONWD(23)) GO TO 515                FEL224C
C*** SET UP VISCOELASTIC/NONLINEAR SOLUTION CALL                        FEL225C
      NTM = 1                                                           FEL226C
      NSEQ = 1                                                          FEL227C
      I=0                                                               FEL228C
      ISIZ = ISZDT - IBUF                                               FEL229C
      IDT = ISIZ/2                                                      FEL230C
      IF(NICD.NE.0) GO TO 515                                           FEL231C
      WRITE(IIP6,2038)                                                  FEL232C
      NICD = 1                                                          FEL233C
      IF(CC.EQ.CONWD(23)) NICD=3                                        FEL234C
      NSICD = NDF*NUMNP                                                 FEL235C
      MG = M0                                                           FEL236C
      M0 = M0 + NICD*NSICD                                              FEL237C
      IF(M0.GT.ISIZ) GO TO 710                                          FEL238C
      DO 510 J = MG,M0                                                  FEL239C
510   DT(J) = 0.                                                        FEL240C
515   M7 = M0                                                           FEL241C
      M7A = M0                                                          FEL242C
      IF(IDT.GE.M7) M7 = IDT + 1                                        FEL243C
      MU = M7                                                           FEL244C
      ID1 = IDT + 1                                                     FEL245C
      M0 = M7 + NDF*NUMNP                                               FEL246C
      IF(M0.GT.ISIZ) GO TO 710                                          FEL247C
      MB = M0-1                                                         FEL248C
C
C *** SAVE MESH DATA ON TAPE FOR PLOT2D/3D
C
      IF(IPLOT.GT.0) CALL PLOT2D(0,KSTEPS,HEAD,NUMNP,NUMEL,DT(M3),
     X DT(M4),DT(M6),DT(MB),NVEC,NSIZV,NEL1,NICD,IDUM,IDUM,NDF)
C                                                                       FEL249C
C*** CHECK THE MESH FOR CONSISTENCY OF INPUT DATA                       FEL250C
C                                                                       FEL251C
      CALL MESHCK(NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEN,NEL1,IDT,ID1,MB,MAXHAN FEL252C
     X ,DT,DT(M1),DT(M2),DT(M3),DT(M4),DT(M5),DT(M6),DT(M7),ISZA,NEQB,
     X IBLK,NDEG)
      MX = ID1 - NEQB                                                   FEL254C
CXXXX
C  OPEN TAPE11 QUICKIO FILE IF OUT-OF-CORE SOLN
```

```
CXXXX
      IF(IBLK.GT.0) CALL QUICKIO (6LTAPE11,0,ISZA)
401   IOUT = I                                                          FEL258C
      IF(I .NE. 0) WRITE(IIP6,2039)                                     FEL259C
      CALL TICTOC(TYME,2)                                               FEL262C
      IF(CC.EQ.CONWD(24)) NSEQ=I                                        FEL263C
      IF(CC.EQ.CONWD(24)) GO TO 520                                     FEL264C
      IF(CC .EQ. CONWD(20)) GO TO 415                                   FEL265C
      IF(MAXBAN.GT.MBAN) GO TO 740                                      FEL266C
      IF(CC .EQ. CONWD(12) .OR. CC .EQ. CONWD(13)) GO TO 125            FEL267C
      IF(IPG .LE. 0) GO TO 705                                          FEL268C
      IF(CC.NE.CONWD(14)) GO TO 405                                     FEL269C
C                                                                       FEL270C
C.... FOURIER COMPOSITION                                               FEL271C
C                                                                       FEL272C
      READ(IIP5,1002) J,F1,F2,F3,F4                                     FEL273C
      WRITE(IIP6,2002) HEAD,IPG,J,F1,F2,F3,F4                           FEL274C
      IPG = IPG + 1                                                     FEL275C
      NTERM = NTERM +1                                                  FEL276C
      NF(NTERM) = J                                                     FEL277C
      XN = J                                                            FEL278C
C                                                                       FEL279C
C.... FORM THE STIFFNESS FOR THE ELEMENTS                               FEL280C
C                                                                       FEL281C
405   IF(FLAG(5)) GO TO 410                                             FEL282C
      LA = 1                                                            FEL283C
      J = (MAXBAN+1)*NDEG + NDF*NUMNP*(1 + NTM)                         FEL284C
      IF(ISIZ-J.LT.M7A) GO TO 410                                       FEL285C
C.... ONCORE SOLUTION IS POSSIBLE                                       FEL286C
      WRITE(IIP6,2031)                                                  FEL287C
      NEQB = NDEG                                                       FEL288C
      IBLK = 0                                                          FEL289C
C.... CHECK TO SEE IF IDEST IS TO BE MOVED UP                           FEL290C
      J = NUMNP*NDF-1                                                   FEL291C
C.... SAVE FORCE VECTOR FOR NONLINEAR SOLUTION                          FEL292C
      MX=M5+J                                                           FEL293C
      REWIND 9                                                          FEL294C
      WRITE(9)(DT(I),I=M5,MX)                                           FEL295C
      IF(M7.EQ.M7A) GO TO 409                                           FEL296C
      DO 408 K = 1,J                                                    FEL297C
408   DT(M7A+K) = DT(M7+K)                                              FEL298C
      DT(M7A) = DT(M7)                                                  FEL299C
      M7 = M7A                                                          FEL300C
409   LA = M7 + J + 1                                                   FEL301C
      MU = LA + NDEG*MAXBAN                                             FEL302C
      MX = MU + NUMNP*NDF                                               FEL303C
      M0 = MX + NDEG                                                    FEL304C
      ID1 = NDEG*MAXBAN + 1                                             FEL305C
410   FLAG(5) = .TRUE.                                                  FEL306C
      IF(NTM.EQ.1) GO TO 520                                            FEL307C
      CALL FORMST(NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEN,NEL1,MB,DT,DT(M1),   FEL308C
     X DT(M2),DT(M3),DT(M4),DT(M5),DT(M6),DT,DT(M9),DT(M7),DT(MB),      FEL309C
     X DT(MU),DT(LA),NSIF,IBLK,NVEC,NSIZV,MAXBAN,IREC,NDEG)             FEL310C
      CALL TICTOC(TYME,3)                                               FEL311C
      IF(IPG .LE. 0) GO TO 705                                          FEL312C
C                                                                       FEL313C
C.... FORM THE GLOBAL STIFFNESS AND SOLVE USING GAUSS ELIMINATION       FEL314C
C                                                                       FEL315C
      CALL SOLVEG(NUMNP,NUMEL,NDF,ID1,MB,MAXBAN,ITA,NSTF,ISZA,NEQB,IBLK, FEL316C
     XDT(LA),DT(M5),DT(MU),DT(M7),DT,DT(M9),DT(MX),NDEG)                FEL317C
      CALL TICTOC(TYME,4)                                               FEL318C
      GO TO 420                                                         FEL319C
C                                                                       FEL320C
C.... DYNAMIC SOLUTION BY EXPLICIT INTEGRATION                          FEL321C
C                                                                       FEL322C
415   IF(NICD .GE. 3) GO TO 418                                         FEL323C
      WRITE(IIP6,2038)                                                  FEL324C
      NICD = 3                                                          FEL325C
      NSICD = NDF*NUMNP                                                 FEL326C
      MG = M0                                                           FEL327C
      M0 = M0 + NICD*NSICD                                              FEL328C
      IF(M0 .GE. ISZDT) GO TO 710                                       FEL329C
```

```
      DO 417 J=MG,M0                                                    FEL330C
417   DT(J) = 0.0                                                       FEL331C
418   MC = MG + NSICD                                                   FEL332C
      MD = MC + NSICD                                                   FEL333C
      ME = M0                                                           FEL334C
      MF = ME + NSTF                                                    FEL335C
      M0 = 2*NSTF + MF                                                  FEL336C
      IF(M0 .GT. ISZDT) GO TO 710                                       FEL337C
      CALL EXPLCT(NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEN,NEL1,NSTF,NVEC,NSIZV,  FEL338C
     X DT(M1),DT(M2),DT(M3),DT(M4),DT(M5),DT(M6),DT(M7),DT(1),DT(M9),    FEL339C
     X DT(M8),DT(MG),DT(MC),DT(MD),DT(ME),DT(MF),TYME,IOUT,NSICD)       FEL340C
      GO TO 125                                                         FEL341C
270   IOUT = I                                                          FEL342C
      CALL TICTOC(TYME,1)                                               FEL343C
C                                                                       FEL344C
C.... FORM THE NEW LOAD AND DO A RESOLUTION ONLY                        FEL345C
C                                                                       FEL346C
      CALL RESVEC(NUMNP,NDF,M7-1,MAXBAN,ISZA,NEQB,IBLK,DT(LA),DT(M5),   FEL347C
     X DT(MU),DT(M5),DT(M7),DT(MX),NDF*NUMNP)                           FEL348C
      CALL TICTOC(TYME,5)                                               FEL349C
  420 CONTINUE                                                          FEL350C
C                                                                       FEL351C
C.... COMPUTE AND OUTPUT THE NODAL DISPLACEMENTS AND ELEMENT STRESSES   FEL352C
C                                                                       FEL353C
      IF(IREC.GT.0.AND.IBLK.EQ.0) REWIND 7                              FEL354C
      CALL DISTRS(NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEN,NEL1,NSTF,NVEC,NSIZV,  FEL355C
     X DT(M1),DT(M2),DT(M3),DT(M4),DT(M5),DT(M6),DT(MU),DT,DT(M9),DT(M8)  FEL356C
     X,IOUT,IREC,IBLK)                                                  FEL357C
      CALL TICTOC(TYME,6)                                               FEL358C
      IF(IBLK.GT.0) REWIND 7                                            FEL359C
      IF(IBLK.GT.0) READ(7)(DT(I),I=1,M8)                               FEL360C
      I = 0                                                             FEL361C
      WRITE(IIP6,2030) TYME                                             FEL362C
      DO 271 N =1, 7                                                    FEL363C
271   TYME(N) = 0.0                                                     FEL364C
C                                                                       FEL365C
C.... INPUT THE FORCE VECTORS ON NODES (RESET TO ZERO AUTOMATICALLY     FEL366C
C.... AFTER EACH PROBLEM SOLUTION OR RESOLUTION).                       FEL367C
C                                                                       FEL368C
260   CONTINUE                                                          FEL369C
      IF(.NOT.FLAG(2)) GO TO 730                                        FEL370C
      CALL RESET(I,NUMNP,NDF,DT(M5))                                    FEL371C
CXXXX
C  CLOSE FILE 11 IF OUT-OF-CORE SOLN WAS DONE
CXXXX
      IF(IBLK.GT.0) CALL QUICKIO (6LTAPE11*4)
      GO TO 125                                                         FEL372C
C                                                                       FEL373C
C.... TIME INTEGRATION SUBROUTINE FOR DYNAMIC AND VISCOELASTIC SOLUTIONS FEL374C
C                                                                       FEL375C
520   IF(FLAG(7)) GO TO 540                                             FEL376C
      FLAG(7) = .TRUE.                                                  FEL377C
      I = M0 + NSICD - 1                                                FEL378C
      IF(I.GT.ISIZ) GO TO 710                                           FEL379C
      DO 530 J = M0,I                                                   FEL380C
530   DT(J) = 0.                                                        FEL381C
      IF(IBLK.EQ.0) GO TO 540                                           FEL382C
      MU = M0 + NSICD                                                   FEL383C
      IF(MU+NSICD.GT.ISIZ) GO TO 710                                    FEL384C
  540 ML=1
      IF((CC.EQ.CONWD(22)).OR.(CC.EQ.CONWD(23)))
     X   CALL TSOLVE(NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEN,NEL1,NSTF,NVEC,    FEL386C
     1   NSIZV,NICD,NSICD,IBLK,ISZA,NEQB,MAXBAN,NDEG,IDI,M8,IBUF,DT,     FEL387C
     2   DT(M1),DT(M2),DT(M3),DT(M4),DT(M5),DT(M6),DT(M7),DT(M8),        FEL388C
     3   DT,DT(M9),DT(ML),DT(LA),DT(M0),DT(MX),DT(ISIZ+1),DT(MG),DT(MU),  FEL389C
     4   NSEQ,TYME)                                                     FEL390C
      IF(CC.EQ.CONWD(24))                                               FEL391C
     XCALL NEWTON (NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEN,NEL1,NSTF,NVEC,NSIZV,  FEL392C
     1IBLK,ISZA,NEQB,MAXBAN,NDEG,IDI,M8,IBUF,DT,DT(M1),DT(M2),DT(M3),     FEL393C
     2DT(M4),DT(M5),DT(M6),DT(M7),DT(M8),DT,DT(M9),DT(ML),DT(LA),DT(M0),  FEL394C
     3DT(MX),DT(ISIZ+1),DT(MU),NSEQ,TYME)                               FEL395C
      I = 0                                                             FEL396C
```

```
      GO TO 260                                                          FEL397C
C**** PROGRAM EXECUTON TYME ARRAY,REFERENCE SUBROUTINE CLOCK             FEL398C
  705 WRITE(IIP6,2030) TYME                                              FEL399C
      GO TO 100                                                          FEL400C
  707 WRITE(IIP6,2033)(FLAG(J),J=1,3)                                    FEL401C
      GO TO 100                                                          FEL402C
  700 WRITE(IIP6,2034) CC                                                FEL403C
      GO TO 100                                                          FEL404C
  710 WRITE(IIP6,2035) MO,ISIZ,CC                                        FEL405C
      GO TO 100                                                          FEL406C
  720 WRITE(IIP6,2036) NDIM,NDF,NEN                                      FEL407C
      GO TO 100                                                          FEL408C
  730 WRITE(IIP6,2037) CC                                                FEL409C
      GO TO 100                                                          FEL410C
  735 WRITE(IIP6,2038)                                                   FEL411C
      GO TO 100                                                          FEL412C
  740 WRITE(IIP6,2032) MAXBAN                                            FEL413C
      GO TO 100                                                          FEL414C
C****                                                                    FEL415C
C**** FORMATS                                                            FEL416C
C****                                                                    FEL417C
 1000 FORMAT (F5.0,1X,12A6,A1)
 1001 FORMAT(615,5F10.0)                                                 FEL419C
 1002 FORMAT(15,5X,4F10.0)                                               FEL420C
 2000 FORMAT(1H1,12A6,30X,4HPAGE,I4////                                  FEL421C
     X/19X,32H FINITE ELEMENT ANALYSIS PROGRAM///13X, 22(2H *) //        FEL422C
     X  13X,2H *,18X,34H DIMENSIONAL PROBLEM   * * * * * *//             FEL423C
     X  13X,2H *,18X,34H DEGREES OF FREEDOM PER NODE   * *//             FEL424C
     X 13X,2H *,18X,34H NODES CONNECTED TO EACH ELEMENT *//              FEL425C
     X 13X,2H *,18X,34H ELEMENT STIFFNESS SIZE  * * * * *//13X,22(2H *)// FEL426C
     X  13X,2H *,8X,34H A GENERALIZED FORCE CHECK HAS    * //            FEL427C
     X  13X,2H *,9X,A4,14HBEEN REQUESTED,14X,1H*//                       FEL428C
     X  13X,2H *,18X,34H IS MAXIMUM PERMISSIBLE HALFBAND * //            FEL429C
     X 13X,2H *,18X,34H WORD BUFFER AREA RESERVED FOR H *//13X,22(2H *)// FEL430C
     X  13X,2H *,8X,18H CONSTANT NO1--- =,18,7X,1H* //                   FEL431C
     X 13X,2H *,8X,18H CONSTANT 1----- =,E12.4,4H    *//                 FEL432C
     X 13X,2H *,8X,18H CONSTANT 2----- =,E12.4,4H    *//                 FEL433C
     X 13X,2H *,8X,18H MULTIPLIER 1--- =,E12.4,4H    *//
     X 13X,2H *,8X,18H MULTIPLIER 2--- =,E12.4,4H    *//
     X 13X,2H *,8X,18H MULTIPLIER 3--- =,E12.4,4H    *//13X,22(2H *)//)  FEL436C
 2002 FORMAT(1H1,12A6,30X,4HPAGE,I4//                                    FEL437C
     X   5X,37HFOURIER COEFFICIENTS FOR HARMONIC NO.  , I4//             FEL438C
     X    10X, 25HRADIAL      COEFFICIENT = E12.4//                      FEL439C
     X    10X, 25HTANGENTIAL  COEFFICIENT = E12.4//                      FEL440C
     X    10X, 25HAXIAL       COEFFICIENT = E12.4//                      FEL441C
     X    10X, 25HTHERMAL     COEFFICIENT = E12.4//  )                   FEL442C
 2003 FORMAT(/20X,11A6)                                                  FEL443C
 2030 FORMAT (1H1,21X,12HELAPSED TIME///10X,25HINPUT PROPERTIES AND MESH FEL444C
     ,F10.3/10X,25HCHECK AND PLOT INPUT DATA,F10.3/10X,14HFORM STIFFNESS FEL445C
     ,F21.3/10X,21HSOLUTION OF EQUATIONS,F14.3/10X,23HRESOLUTION OF EQUA FEL446C
     ,TIONS,F12.3/10X,14HOUTPUT ANSWERS,F21.3/10X,10HTOTAL TIME,F25.3 )  FEL447C
 2031 FORMAT(// 5X,16HIN-CORE SOLUTION  )                                FEL448C
 2032 FORMAT (27HOBANDWIDTH EXCEEDED, MBAND=I5)                          FEL449C
 2033 FORMAT(55HOALL INFORMATION FOR THIS PROBLEM HAS NOT BEEN PROVIDED/ FEL450C
     X 15HOMATERIAL CARDS  , L2/                                         FEL451C
     X 15HONODAL CARDS     , L2/                                         FEL452C
     X 15HOELEMENT CARDS   , L2/ )                                       FEL453C
 2034 FORMAT(5X,A6,16H TYPE CARD ERROR/)                                 FEL454C
 2035 FORMAT(20HOREQUIRED  STORAGE = I7/20HOAVAILABLE STORAGE = I7/      FEL455C
     X 25HOSTORAGE EXCEEDED DURING ,A6/)                                 FEL456C
 2036 FORMAT(20HOINPUT SIZE ERROR,  ,8H NDIM = ,I5,8H, NDF = ,I5,       FEL457C
     C  8H, NEL = ,I5)                                                   FEL458C
 2037 FORMAT(1HOATTEMPT TO INPUT ,A6,20H BEFORE NODAL POINTS )           FEL459C
 2038 FORMAT(5X, 9HW**NON-FATAL ERROR** INITIAL DYNAMIC VECTORS HAVE NOT FEL460C
     X BEEN INPUT **INITIAL CONDITION ASSUMED ZERO      )                FEL461C
 2039 FORMAT(43HOREQUEST MADE FOR LIMITED AMOUNT OF OUTPUT,              FEL462C
     X ,25H ADDITIONAL DATA REQUIRED/1X)                                 FEL463C
 2040 FORMAT(////33H REQUEST FOR PLOTS  IS INITIATED. )
      END                                                                FEL464C
*DECK BLKDATA
      SUBROUTINE BLKDAT
      REAL LABL                                                          2C
```

```
      LOGICAL PRT,NPL                                                    3C
      COMMON/DYNAMO/TIME,NSIG(7),NT,NSTEP,DT,NUMPLT,NEDATA(20,3),NPR,NPL 4C
      COMMON/GAUS/ LIM,SGAUSS(5,5),WGAUSS(5,5)                           5C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UM,RHED(6),RM 6C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)      7C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)      8C
      COMMON/TITLES/ XTTL(3),UTTL(6)                                     9C
      COMMON /TAPES/ ITP5,ITP6                                           10C
      COMMON/XTAPE/ ITP13,ITP14,ITPRD,ITPWR                              11C
      DATA SHAPE/80*0.0/,START,CEASE/6HPREAP72,6HSTOP  /                 12C
      DATA WORD/SHELM01,SHELM02,SHELM03,SHELM04,SHELM05,SHELM06,SHELM07, 13C
     X SHELM08,SHELM09,SHELM10,SHELM11,SHELM12,SHELM13,SHELM14,SHELM15,  14C
     X SHELM16,SHELM17,SHELM18,SHELM19,SHELM20,SHELM21,SHELM22,SHELM23,  15C
     X SHELM24,SHELM25,SHELM26,SHELM27,SHELM28,SHELM29,SHELM30/          16C
      DATA SGAUSS/5*0.,-.57735027,.57735027,3*0.,-.77459667,0.,.77459667 17C
     C ,2*0.,-.86113631,-.33998104,.33998104,.86113631,0.,-.90617985,    18C
     C -.53846931,.0,.53846931,.90617985/                                19C
      DATA WGAUSS/5*2.,5*1.,.55555556,.88888889,.55555556,2*0.,.34785485 20C
     C ,2*0.,.52145151,.34785485,0.,.23692689,.47862867,.56888889,.47862867 21C
     C ,.23692689/                                                       22C
      DATA LABL(1)/6H (112,  /                                           23C
      DATA XTTL/6H      1,6H      2,6H      3/, XM/6H ORD. /            24C
      DATA UTTL/6H      1,6H      2,6H      3,6H      4,6H      5,6H      6/ 25C
      DATA FM,RM,UM/6H FORCE,6H FORCE,6H DISPL/                          26C
      DATA AWORD1,AWORD2/8H  F13.4,,8H 6E13.4)/,AWORD3/8H  F12.2,/       27C
      DATA ITP5,ITP6 /5,6/  , NSIG/7*0/, TIME/0.0/,NUMPLT,NPR,NPL/0,0,1/ 28C
      DATA ITP13,ITP14/13,14/                                            29C
      RETURN
      END                                                                30C
*DECK TICTOC
      SUBROUTINE TICTOC(TYME, I)                                         TIC  1C
C                                                                        TIC  2C
C    SUBROUTINE TO TIME PROGRAM SEGMENTS AS MEASURE OF EFFICIENCY        TIC  3C
C                                                                        TIC  4C
      DIMENSION TYME(7)                                                  TIC  5C
      IF (I .LE. 0) GO TO 100                                            TIC  6C
      CALL SECOND(T1)                                                    TIC  7C
      TYME (I) = T1 - T0                                                 TIC  8C
      TYME(7) = TYME(7) + TYME(I)                                        TIC  9C
      T0 = T1                                                            TIC 10C
      RETURN                                                             TIC 11C
  100 CALL SECOND(T0)                                                    TIC 12C
      RETURN                                                             TIC 13C
      END                                                                TIC 14C
*DECK MESH
      SUBROUTINE MESH(MC,NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEL,NEL1,           MES  1C
     X DT,ITPE,D,ICOD,XYZ,F,IX)                                          MES  2C
C**** INPUT OF MESH AND MATERIAL PROPERTIES .                            MES  3C
      COMMON /TAPES/ ITP5,ITP6                                           MES  4C
      REAL LABL                                                          MES  5C
      DIMENSION ITPE(1),D(3,21,1),ICOD(1),XYZ(NDIM,1),F(NDF,1),          MES  6C
     X IX(NEL1,1),IXD(20),DX(3),IXP(20),XWORD(1)                         MES  7C
      COMMON/GAUS/ LIM,SGAUSS(5,5),WGAUSS(5,5)                           MES  8C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UM,RHED(6),RM MES  9C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)      MES 10C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)      MES 11C
      COMMON/VALUES/ NCI,CON1,CON2,CON3(3)                               MES 12C
      DATA BLANK/6HBLANK /                                               MES 13C
      GO TO(1,2,3,2),MC                                                  MES 14C
1     CONTINUE                                                           MES 15C
      MCT = 0                                                            MES 16C
      DO 300 I = 1,NUMMAT                                                MES 17C
      MCI = MCT - 1                                                      MES 18C
      IF(MCI.GT.0) GO TO 10                                              MES 19C
      WRITE(ITP6,2001) HEAD,IPG,NUMMAT                                   MES 20C
      IPG = IPG + 1                                                      MES 21C
      MCT = 4                                                            MES 22C
10    CONTINUE                                                           MES 23C
      READ(ITP5,1002) M,DM,XWORD                                         MES 24C
      IF(M.GT.NUMMAT.OR.M.LE.0) GO TO 301                                MES 25C
      TYPE(M) = DM                                                       MES 26C
      WRITE(ITP6,2002) DM,M,XWORD                                        MES 27C
```

```
      DO 280 J =1,3                                                    MES 280
      DO 280 K = 1,21                                                  MES 290
280   P(J,K,M)=0.0                                                     MES 300
C                                                                      MES 310
      K = 0                                                            MES 320
      DO 290 J = 1,30                                                  MES 330
      IF(DM.EQ.*ORD(J)) GO TO 400                                      MES 340
290   CONTINUE                                                         MES 350
C**** ERROR IF EXIT ON LOOP 290                                        MES 360
  301 WRITE(IIPG,2032)M,DM                                             MES 370
      IPG = 0                                                          MES 380
      RETURN                                                           MES 390
400   CALL ELMLIB(N,M ,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  MES 400
     X IX,F,FORCE,ESTIF,U,VECT,1)                                      MES 410
300   CONTINUE                                                         MES 420
      RETURN                                                           MES 430
2     CONTINUE                                                         MES 440
C                                                                      MES 450
C**** SET XYZ(I,N) TO BLANKS                                           MES 460
C                                                                      MES 470
      DO 20 N = 1,NUMNP                                                MES 480
      XYZ(I,N) = BLANK                                                 MES 490
      DO 20 I = 1,NDF                                                  MES 500
20    F(I,N) = 0.0                                                     MES 510
C                                                                      MES 520
C**** START FILLING DATA                                               MES 530
C                                                                      MES 540
      N = 0                                                            MES 550
      NXP = 0                                                          MES 560
21    NP = N                                                           MES 570
      IF(MC.EQ.2.AND.N.GE.NUMNP) GO TO 30                              MES 580
      NX = NXP                                                         MES 590
      READ(IIP5,1006) N,NXP,I,DX                                       MES 600
      IF(N.LE.0.OR.N.GT.NUMNP) GO TO 30                                MES 610
      IF(MC.EQ.2) NXP = 1                                              MES 620
      ICOD(N) = I                                                      MES 630
      DO 26 I=1,NDIM                                                   MES 640
26    XYZ(I,N) = DX(I)*CON3(I)                                         MES 650
      IF(NX.EQ.0) GO TO 21                                             MES 660
      IF((N-NP)*NX.GE.0) GO TO 25                                      MES 670
      NX = - NX                                                        MES 680
      WRITE(IIP6,2033) NP,N                                            MES 690
25    CONTINUE                                                         MES 700
C                                                                      MES 710
C**** GENERATE THE NODES BETWEEN NP AND N IN INTERVALS OF NX           MES 720
C                                                                      MES 730
      LX = (IABS(N-NP) + IABS(NX) - 1)/IABS(NX)                        MES 740
      DO 22 I = 1,NDIM                                                 MES 750
22    DX(I) = (XYZ(I,N) - XYZ(I,NP))/LX                                MES 760
23    NP = NP + NX                                                     MES 770
      IF(NX.GT.0.AND.NP.GE.N) GO TO 21                                 MES 780
      IF(NX.LT.0.AND.NP.LE.N) GO TO 21                                 MES 790
      DO 24 I = 1,NDIM                                                 MES 800
24    XYZ(I,NP) = XYZ(I,NP-NX) + DX(I)                                 MES 810
      ICOD(NP) = 0                                                     MES 820
      IF(ICOD(NP-NX).EQ.ICOD(N)) ICOD(NP) = ICOD(N)                    MES 830
      GO TO 23                                                         MES 840
C                                                                      MES 850
C**** CHECK IF ALL DATA HAS BEEN INPUT OR GENERATED AND PRINT OUTPUT    MES 860
C                                                                      MES 870
30    CALL PRTMSH(1,NUMNP,NDIM,ICOD,XYZ)                               MES 880
      RETURN                                                           MES 890
C**** INPUT ELEMENT CONNECTION ARRAY                                   MES 900
3     MCT = 0                                                          MES 910
      NSIDE =(NEL+1)/NDIM**2                                           MES 920
      N=0                                                              MES 930
      NAP = 0                                                          MES 940
C**** SET UP INCREMENT ARRAY IF NOT INPUT OFF CARDS                    MES 950
      GO TO 132                                                        MES 960
130   DO 151 L = 1,NEL                                                MES 970
151   IX(L) = IXP(L)                                                   MES 980
      IF(IX(1).NE.0) GO TO 157                                         MES 990
```

```
      DO 15  L = 1,NEL                                                  MES100C
150   IX0(L) = NSIDE                                                    MES101C
      N1 = 4*NSIDE + 1                                                  MES102C
      N2 = 8*NSIDE - 4                                                  MES103C
      IF(N2.GT.NEL) GO TO 154                                           MES104C
      DO 152 L = N1,N2                                                  MES105C
152   IX0(L) = 1                                                        MES106C
154   N1 = 2*NSIDE                                                      MES107C
      N3 = N1 + 2 - NSIDE                                               MES108C
      N4 = N1 + N2                                                      MES109C
      DO 156 L = N3,N1                                                  MES110C
      IX0(L) = 1                                                        MES111C
      IX0(L + N1) = 1                                                   MES112C
      IF(N2.GT.NEL) GO TO 156                                           MES113C
      IX0(L + N2) = 1                                                   MES114C
      IX0(L + N4) = 1                                                   MES115C
156   CONTINUE                                                          MES116C
157   CONTINUE                                                          MES117C
132   READ(IIPS,1003) M,MA,IH,IP,IAP                                    MES118C
      IF(M.LE.0) RETURN                                                 MES119C
      READ(IIPS,1004) (IX(I,M),I=1,NEL)                                 MES120C
      IF(M.LE.0) M = M - 1                                              MES121C
      IX(NEL1,M) = MA + 10*IP + 100*IH                                  MES122C
      K1 = 1                                                            MES123C
      DO 133 I1 = 1,NEL                                                 MES124C
      IF(IX(I1,M).EQ.0) GO TO 134                                       MES125C
      K1 = K1 + 1                                                       MES126C
133   CONTINUE                                                          MES127C
134   IF(K1.GT.NEL) GO TO 140                                           MES128C
      DO 138 I1 = K1,NEL                                                MES129C
138   IX0(I1) = 0                                                       MES130C
140   M=M+1                                                             MES131C
      MCT = MCT - 1                                                     MES132C
      IF(MCT.GT.0) GO TO 142                                            MES133C
      READ(IIPS,2011) MEAD, IPG,NUMEL,(I,I=1,NEL)                       MES134C
      IPG = IPG + 1                                                     MES135C
      MCT = 50                                                          MES136C
142   CONTINUE                                                          MES137C
      IF(M.LE.0) GO TO 170                                              MES138C
      DO 158 L = 1,NEL                                                  MES139C
158   IX(L,M) = IX(L,M-1) + IX0(L)                                      MES140C
      IX(NEL1,M) = MAP                                                  MES141C
170   CONTINUE                                                          MES142C
      K = 0                                                             MES143C
      DO 34  I1=1,NEL                                                   MES144C
      IF(IX(I1,M).EQ.0) GO TO 141                                       MES145C
      DO 325 J1=I1,NEL                                                  MES146C
      IF(IX(J1,M).EQ.0) GO TO 340                                       MES147C
      KK=IABS(IX(I1,M)-IX(J1,M)) + 1                                    MES148C
      IF(K.LE.KK) K=KK                                                  MES149C
325   CONTINUE                                                          MES150C
340   CONTINUE                                                          MES151C
141   MAP= MOD(IX(NEL1,M),10)                                           MES152C
      IH = IX(NEL1,M)/100                                               MES153C
      IP = MOD(IX(NEL1,M),100)/10                                       MES154C
      WRITE(IIPR,2005) M,MAP,IH,IP,(IX(I,M),I=1,NEL),K                  MES155C
      IF(M-N) 185,180,140                                               MES156C
185   WRITE(IIPR,2031) N                                                MES157C
      IPG = 0                                                           MES158C
      RETURN                                                            MES159C
180   MAP = MA                                                          MES160C
      IF(NUMEL.GT.N) GO TO 130                                          MES161C
      RETURN                                                            MES162C
C**** FORMATS                                                           MES163C
1002  FORMAT(15,1X,A5,11A6)                                             MES164C
1003  FORMAT(4I5,2013)                                                  MES165C
1004  FORMAT(2014)                                                      MES166C
1005  FORMAT(2I5,11X,6F10.0)                                            MES167C
 001  6ORMAT(1H1,12A6,3X,4HPAGE,I4//15,10H MATERIALS//                  MES168
     X 20H MATERIAL PROPERTIES //1X)                                    MES169C
2002  FORMAT(5X,A5,1X, 8HMATERIAL,I5,5X,11A6/1X)                        MES170C
2005  FORMAT(2515)                                                      MES171C
```

```
2011  FORMAT(1H1,12A6,3X,4HPAGE,I4//I5,9H ELEMENTS//                    MES172C
     X 20H ELMT MATL  RE  PRNT    ,3X,26HNODES CONNECTED TO ELEMENT/    MES173C
     X 20H  NO.  NO. USES         ,1X,20(2H *,I2,1H*))                  MES174C
C****                                                                  MES175C
C**** ERROR MESSAGES *                                                 MES176C
C****                                                                  MES177C
2031  FORMAT (24H0 ELEMENT CARD ERROR, N=I5)                           MES178C
2032  FORMAT(26H0MATERIAL CARD ERROR, M = ,I3,2X,8H,TYPE = ,A5/)       MES179C
2033  FORMAT(                                                          MES180C
     X5X,58H**NON-FATAL ERROR** INCREMENT FOR GENERATION BETWEEN NODES, MES181C
     X   I5,4H AND,I5,37H IS OF INCORRECT SIGN ** SIGN CHANGED/1X)     MES182C
      END                                                              MES183C
*DECK MESHCK
      SUBROUTINE MESHCK(NUMNP,NUMEL,NUMMAT,NDIM,NDF,NEL,NEL1,IDT,IDI,  MES  1C
     X  M7,NB,DT,ITYPE,D,ICOD,XYZ,IF,IX,IDEST,IS2A,NEQB,IBLK,NDEG)
      INTEGER XYZ,X,BLANK
      REAL LAML                                                        MES  3C
      DIMENSION ITYPE(1),D(63,1),XYZ(NDIM,1),IX(NEL1,1),IDEST(NDF,1)   MES  4C
     X  ,ICOD(1),IF(NDF,1)
      COMMON/LABELS/ LABL(6),XMED(3),XM,FMED(6),FM,UMED(6),UM,RMED(6),RM MES  6C
     X ,AMOD1,AMOD2,AMOD3,HEAD(12),START,CEASE,IPG,NSTR,NORD(30)       MES  7C
      COMMON/SHARE/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)   MES  8C
      COMMON /IAPES/ IIPS,IIPB                                         MES  9C
      COMMON /MINBW/ NEW(1500)
      DATA BLANK/4HBLANK /                                             MES 10C
CXXXX
C  REVISE ALL ARRAYS BASED ON MINIMUM BANDWIDTH
CXXXX
      DO 10 N=1,NUMNP
      NN=NEW(N)
   10 IDEST(I,NN)=ICOD(N)
      DO 15 N=1,NUMNP
   15 ICOD(N)=IDEST(I,N)
      DO 30 I=1,NDIM
      DO 20 N=1,NUMNP
      NN=NEW(N)
   20 IDEST(I,NN)=XYZ(I,N)
      DO 25 N=1,NUMNP
   25 XYZ(I,N)=IDEST(I,N)
   30 CONTINUE
      DO 35 N=1,NUMEL
      DO 35 I=1,NEL
      K=IX(I,N)
      IF(K.EQ.0) GO TO 35
      IX(I,N)=NEW(K)
   35 CONTINUE
   36 CONTINUE
      IDI = IDT + 1                                                    MES 11C
C**** SET UP THE DESTINATION VECTOR                                    MES 12C
      NDEG = 0                                                         MES 13C
      DO 14  N = 1,NUMNP                                               MES 14C
      IC = ICOD(N)                                                     MES 15C
      IL = IDEG0                                                       MES 16C
      DO 12  I = 1,NDF                                                 MES 17C
      IDEST(I,N) = 0                                                   MES 18C
      IF(IC.LT.IL) GO TO 150                                           MES 19C
  100 IC = IC - IL                                                     MES 20C
      IF(IC.GT.IL) GO TO 100                                           MES 21C
      GO TO 120                                                        MES 22C
  150 NDEG = NDEG + 1                                                  MES 23C
      IDEST(I,N) = NDEG                                                MES 24C
  120 IL = IL/10                                                       MES 25C
  140 CONTINUE                                                         MES 26C
      MCT = 0                                                          MES 27C
      DO 16  IJ=1,NUMNP
      N=NEW(IJ)
      MCT = MCT + 1                                                    MES 29C
      IF(MCT.GT.0) GO TO 150                                           MES 30C
      MCT = 50                                                         MES 31C
      WRITE(IIPB,2033)  HEAD,IPG,(XMED(I),I=1,NDF)                     MES 32C
      IPG = IPG + 1                                                    MES 33C
  160 WRITE (IIPB,2000) IJ,(IDEST(I,N),I=1,NDF)
```

```
C.... COMPUTE THE BANDWIDTH OF THE MESH                                  MES 350
      MB =                                                               MES 360
      DO 19 N = 1,NUMEL                                                  MES 370
      MM = 0                                                             MES 380
      MA = 0                                                             MES 390
      DO 17 I = 1,NEL                                                    MES 400
      K = IX(I,N)                                                        MES 410
      IF(K.EQ.0) GO TO 180                                               MES 420
      DO 17 J = 1,NDF                                                    MES 430
      IC = IDEST(J,K)                                                    MES 440
      IF(IC.GT.MM) MM = IC                                               MES 450
      IF(IC .EQ. 0) GO TO 170                                            MES 460
      IF(IC.LT.NN.OR.NN.EQ.0) NN = IC                                    MES 470
  170 CONTINUE                                                           MES 480
  180 IF(MM-NN.GT.MB) MB = MM-NN                                         MES 490
  190 CONTINUE                                                           MES 500
      MB = MB + 1                                                        MES 510
      MB1 = MB + 2                                                       MES 520
      NEQB = IDI/MB1                                                     MES 530
      IF( NEQB.GT.NDEG) NEQB = NDEG                                      MES 540
      IBLK = (NDEG+NEQB - 1)/NEQB                                        MES 550
      WRITE(IOPS,2031) HEAD,(PG,NDEG,MM,NEQB,IBLK                        MES 560
      IPG = IPG + 1                                                      MES 570
      ISZA = NEQB*(MB+1)                                                 MES 580
C.... CHECK MESH FOR CONSISTENCY OF DATA                                 MES 590
      DO 50 N = 1,NUMEL                                                  MES 600
      MA = MOD(IX(NEL1,N),10)                                            MES 610
      IF(MA.GT.NUMMAT.OR.MA.LE.0) GO TO 40                               MES 620
      DM = TYPE(MA)                                                      MES 630
      FLAG = 1.0                                                         MES 640
      DO 205 I=1,NEL                                                     MES 650
      K = IX(I,N)                                                        MES 660
      IF(K.GT.NUMAP.OR.K.LT.0) GO TO 40                                  MES 670
      IF(K.EQ.0) GO TO 205                                               MES 680
      NEN = I                                                            MES 690
      IF( XYZ(1,K) .NE. BLANK) GO TO 195                                 MES 700
      FLAG = -1.                                                         MES 710
      WRITE(IOPE,2034)K,N                                                MES 720
      IPG = 0                                                            MES 730
      GO TO 205                                                          MES 740
  195 CONTINUE                                                           MES 750
      DO 200 J = 1,NDIM                                                  MES 760
  200 X(J,I) = XYZ(J,K)                                                  MES 770
  205 CONTINUE                                                           MES 780
      IF(FLAG .GT. 0.0)                                                  MES 790
     X CALL ELMLIB(N,MA,NDIM,NDF,NEN,NEL1,NSIF,NSIZV,NVEC,MCT,DM,D,XYZ,  MES 800
     X IX,F,FORCE,ESTIF,U,VEC1,2)                                        MES 810
   60 IF(DM.GT.0.0) GO TO 50                                             MES 820
   40 WRITE(IOPS,2030) N,MA,DM,(IX(J,N),J=1,NEL)                         MES 830
      IPG = 0                                                            MES 840
   50 CONTINUE                                                           MES 850
      IF(IPG .GT. 0) WRITE(IOPS,2032)                                    MES 860
      RETURN                                                             MES 870
 2000 FORMAT(1I0,5X,6I6)                                                 MES 880
 2030 FORMAT(8HOELEMENT,I5,10H, MATERIAL,I5,10H, JACOBIAN,E12.4/         MES 890
     X          9H IX ARRAY,20I5)                                       MES 900
 2031 FORMAT(1H1,12A6,3UX,4HPAGE,I4//17H EQUATION SUMMARY//              MES 910
     X 5X,21HNUMBER OF EQUATIONS =,I6//5X,21HMAXIMUM HALFBAND    =,I6//  MES 920
     X 5X,21HEQUATIONS PER BLOCK =,I6//                                  MES 930
     X 5X,21HNUMBER OF BLOCKS    =,I6//4X,28(1H*)//////1X)               MES 940
 2032 FORMAT(5X,46HNO ERRORS DETECTED DURING A CHECK OF THE MESH )       MES 950
 2033 FORMAT(1H1,12A6,3UX,4HPAGE,I4//19H DESTINATION VECTOR//            MES 960
     X 8X,4HNODE,2X,3HDOF,6A6)                                           MES 970
 2034 FORMAT(21H **FATAL ERROR**NODE, I6, 34H NOT INPUT BUT IS USED IN E MES 980
     XLEMENT, I6 )                                                       MES 990
      END                                                                MES1000
*DECK PRTELM
      SUBROUTINE PRTELM(NE,ME,NEL,NEL1,IX)                               PRT  10
      REAL LABL                                                          PRT  20
      DIMENSION IX(NEL1,1)                                               PRT  30
      COMMON /IAPES/ IIPS,IIPE                                           PRT  40
      COMMON/LABELS/ LABL(6),XHED(3),XH,FHED(6),FH,UHED(6),UH,RHED(6),RH PRT  50
```

```
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)        PRT  6C
      MCT = 0                                                              PRT  7C
      DO 300 N = NE,ME                                                     PRT  8C
      MCT = MCT - 1                                                        PRT  9C
      IF(MCT.GT.0) GO TO 100                                               PRT 10C
      WRITE(IIP6,2011) HEAD,IPG,ME,(I,I=1,NEL)                             PRT 11C
      IPG = IPG + 1                                                        PRT 12C
      MCT = 58                                                             PRT 13C
  100 CONTINUE                                                             PRT 14C
      K = 0                                                                PRT 15C
      MA = MOD(IX(NELI,N),10)                                              PRT 16C
      IR = IX(NELI,N)/100                                                  PRT 17C
      IP = MOD(IX(NELI,N),100)/10                                          PRT 18C
      DO 140 II = 1,NEL                                                    PRT 19C
      IF(IX(II,N).EQ.0) GO TO 150                                          PRT 20C
      DO 135 J1 = II,NEL                                                   PRT 21C
      IF(IX(J1,N).EQ.0) GO TO 140                                          PRT 22C
      KK = IABS(IX(II,N)-IX(J1,N)) + 1                                     PRT 23C
      IF(K.LT.KK) K = KK                                                   PRT 24C
  135 CONTINUE                                                             PRT 25C
  140 CONTINUE                                                             PRT 26C
  150 WRITE(IIP6,2005) N,MA,IR,IP,(IX(I,N),I=1,NEL),K                      PRT 27C
  300 CONTINUE                                                             PRT 28C
      RETURN                                                               PRT 29C
 2005 FORMAT(2515)                                                         PRT 30C
 2011 FORMAT(1H1,12A6,3(1X,4HPAGE,I4)//15,9H ELEMENTS//                    PRT 31C
     X 20H ELMT MATL  RE  PRNT    ,3X,26HNODES CONNECTED TO ELEMENT/       PRT 32C
     X 20H  NO.  NO. USES         ,1X,20(2H *,I2,1H*))                     PRT 33C
      END                                                                  PRT 34C
*DECK PRTMSH
      SUBROUTINE PRTMSH(N1,NUMNP,NDIM,ICOD,XYZ)                            PRT  1C
      REAL LABL                                                            PRT  2C
      DIMENSION ICOD(1),XYZ(NDIM,1)                                        PRT  3C
      COMMON /TAPES/ ITP5,ITP6                                             PRT  4C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UM,RHED(6),RM   PRT  5C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)        PRT  6C
      DATA BLANK/8HBLANK /                                                 PRT  7C
      MCT = 0                                                              PRT  8C
      DO 33 N = N1,NUMNP                                                   PRT  9C
      MCT = MCT - 1                                                        PRT 10C
      IF(MCT.GT.0) GO TO 31                                                PRT 11C
      WRITE(IIP6,2010) HEAD,IPG,NUMNP,(XHED(I),XM,I=1,NDIM)                PRT 12C
      IPG = IPG + 1                                                        PRT 13C
      MCT = 58                                                             PRT 14C
   31 IF(XYZ(1,N).EQ.BLANK) GO TO 32                                       PRT 15C
      WRITE(IIP6,2006) N,ICOD(N),(XYZ(I,N),I=1,NDIM)                       PRT 16C
      GO TO 33                                                             PRT 17C
   32 WRITE(IIP6,2007) N                                                   PRT 18C
   33 CONTINUE                                                             PRT 19C
      RETURN                                                               PRT 20C
 2006 FORMAT(2I12,7F13.4)                                                  PRT 21C
 2007 FORMAT(I12,5X,31HHAS NOT BEEN INPUT OR GENERATED )                   PRT 22C
 2010 FORMAT(1H1,12A6,3(1X,4HPAGE,I4)//15,13H NODAL POINTS//               PRT 23C
     X 12H NODAL POINT,6X,6H*H,C,** 7(1X,2A6))                            PRT 24C
      END                                                                  PRT 25C
*DECK PRTMAT
      SUBROUTINE PRTMAT(HEAD,IPG,NELM,NSTF,ESTIF,FORCE,LD,NT)              PRT  1C
      COMMON /TAPES/ ITP5,ITP6                                             PRT  2C
      DIMENSION ESTIF(NT,NT),FORCE(NT),LD(NT),HEAD(12)                     PRT  3C
      MBLK = (NSTF+23)/24                                                  PRT  4C
      MBLK = (NSTF + 8)/9                                                  PRT  5C
      N1 = 1                                                               PRT  6C
      DO 200 NN = 1,MBLK                                                   PRT  7C
      N2 = N1 + 8                                                          PRT  8C
      IF(N2.GT.NSTF) N2 = NSTF                                             PRT  9C
      M1 = 1                                                               PRT 10C
      DO 100 MM = 1,MBLK                                                   PRT 11C
      M2 = M1 + 23                                                         PRT 12C
      IF(M2.GT.NSTF) M2 = NSTF                                             PRT 13C
      WRITE(IIP6,2000) HEAD,IPG,NELM,(J,J=N1,N2)                           PRT 14C
      IPG = IPG + 1                                                        PRT 15C
      DO 50 I = M1,M2                                                      PRT 16C
```

```
   50 WRITE(IPR,2002)  I,LD(I),(ESTIF(I,J),J=N1,N2)                    PRT 17C
100   M1 = M2 + 1                                                      PRT 18C
      WRITE(IPR,2001)  (FORCE(J),J=N1,N2)                              PRT 19C
200   N1 = N2 + 1                                                      PRT 20C
      RETURN                                                           PRT 21C
2000  FORMAT(1H1,12A6,3(X,6HPAGE,I4//5X,7HELEMENT,I5,7H MATRIX//       PRT 22C
     X 2X,7HROW/COL,3X,9I12)                                           PRT 23C
2001  FORMAT(/3X,5HFORCE,4X,1P9E12.3)                                  PRT 24C
2002  FORMAT(/I4,3H LD,I5,1P9E12.3)                                    PRT 25C
      END                                                              PRT 26C
*DECK BLOADS
      SUBROUTINE BLOADS(NLD,NUMNP,NDIM,NDF,ICOD,XYZ,F)                 BLO  1C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(5),FF,UHED(5),UM,HHED(6),HH BLO  2C
     X ,ANOHD1,ANORD2,ANOHD3,HEAD(12),START,CEASE,IPG,NSTR,NOHD(30)    BLO  3C
      COMMON/SHARE/ XJAC,SHAPE(4,20),SG(3,3),SM(3,3),X(3,20),LD(120)   BLO  4C
      COMMON/TAPES/ ITPS,IPR                                           BLO  5C
      DIMENSION ICOD(1),XYZ(NDIM,1),F(NDF,1),IPRES(8),PR(8),SHD(6),    BLO  6C
     X FS(6,8),INC(8)                                                  BLO  7C
      DATA SHD/SMSLD01,SMSLD02,SMSLD03,SMSLD04,SMSLD05/                BLO  8C
      MCT = 0                                                          BLO  9C
      DO 900 N = 1,NLD                                                 BLO 10C
      READ(ITPS,1000) NDP,SLD,NRT,IPRES,INC,PR                         BLO 11C
      DO 200 I = 1,8                                                   BLO 12C
      IF(IPRES(I).EQ.0) GO TO 300                                      BLO 13C
      JM = IPRES(I)                                                    BLO 14C
      IF(JM.GT.NUMNP.OR.JM.LT.1) GO TO 910                             BLO 15C
      JM = JM + NRT*INC(I)                                             BLO 16C
      IF(JM.GT.NUMNP.OR.JM.LE.0) GO TO 920                             BLO 17C
200   NPRES = I                                                        BLO 18C
C.... CHECK FOR MATCH OF LOAD TYPE                                     BLO 19C
300   DO 400 JM = 1,5                                                  BLO 20C
      JMM = JM                                                         BLO 21C
      IF(SLD.EQ.SHD(JM)) GO TO 500                                     BLO 22C
400   CONTINUE                                                         BLO 23C
      WRITE(IPR,2030) SLD                                              BLO 24C
      IPG = 0                                                          BLO 25C
      RETURN                                                           BLO 26C
500   DO 52 I = 1,NPRES                                                BLO 27C
      J = IPRES(I)                                                     BLO 28C
      DO 51 K = 1,NDIM                                                 BLO 29C
510   X(K,I) = XYZ(K,J)                                                BLO 30C
      DO 52 K = 1,NDF                                                  BLO 31C
520   FS(K,I) = 0.0                                                    BLO 32C
      GO TO (1,2,3,4,5),JMM                                            BLO 33C
1     CALL SLOAD1(ND,NDIM,NDF,NDP,NPRES,IPRES,PR,XYZ,FS)               BLO 34C
      GO TO 600                                                        BLO 35C
2     CALL SLOAD2(ND,NDIM,NDF,NDP,NPRES,IPRES,PR,XYZ,FS)               BLO 36C
      GO TO 600                                                        BLO 37C
3     CALL SLOAD3(ND,NDIM,NDF,NDP,NPRES,IPRES,PR,XYZ,FS)               BLO 38C
      GO TO 600                                                        BLO 39C
4     CALL SLOAD4(ND,NDIM,NDF,NDP,NPRES,IPRES,PR,XYZ,FS)               BLO 40C
      GO TO 600                                                        BLO 41C
5     CALL SLOAD5(ND,NDIM,NDF,NDP,NPRES,IPRES,PR,XYZ,FS)               BLO 42C
600   MCT = MCT + NPRES + 2                                            BLO 43C
      IF(MCT.GT.50) GO TO 610                                          BLO 44C
      WRITE(IPR,2000) HEAD,IPG,(FHED(I),FF,I=1,NDF)                    BLO 45C
      IPG = IPG + 1                                                    BLO 46C
      MCT = 5                                                          BLO 47C
610   WRITE(IPR,2001) NDP,SLD                                          BLO 48C
      DO 62 I = 1,NPRES                                                BLO 49C
620   WRITE(IPR,2002) IPRES(I),PR(I),(FS(J,I),J=1,NDF)                 BLO 50C
C.... ADD TO THE GENERALIZED FORCES                                    BLO 51C
      DO 74 I = 1,NPRES                                                BLO 52C
      K = IPRES(I)                                                     BLO 53C
      IC = ICOD(K)                                                     BLO 54C
      IL = 10000                                                       BLO 55C
      DO 73 J = 1,NDF                                                  BLO 56C
      IF(IC.LT.IL) GO TO 720                                           BLO 57C
710   IC = IC - IL                                                     BLO 58C
      IF(IC.GE.IL) GO TO 710                                           BLO 59C
      GO TO 730                                                        BLO 60C
720   F(J,K) = F(J,K) + FS(J,I)                                        BLO 61C
```

```
730   IL = IL/10                                                        BLO 620
740   CONTINUE                                                          BLO 630
      IF(NRT.LE.0) GO TO 900                                            BLO 640
      NRT = NRT - 1                                                     BLO 650
      DO 800 I = 1,8                                                    BLO 660
800   IPRES(I) = IPRES(I) + INC(I)                                      BLO 670
      GO TO 500                                                         BLO 680
900   CONTINUE                                                          BLO 690
      RETURN                                                            BLO 700
910   WRITE(IPR6,2031) NDP,SLD,IPRES                                    BLO 710
      GO TO 930                                                         BLO 720
920   WRITE(IPR6,2031) NDP,SLD,IPRES,NRT,INC                            BLO 730
930   IPG = 0                                                           BLO 740
      RETURN                                                            BLO 750
C**** FORMAT STATEMENTS                                                 BLO 760
1000  FORMAT(I5,1X,A5,I4,8I5,8I3/8F10.0)                                BLO 770
2000  FORMAT(1H1,12A6,3X,4HPAGE,I4//5X,26HDISTRIBUTED BOUNDARY LOADS//  BLO 780
     1  6X,4HNODE, 9X,4HLOAD,6(1X,2A6))                                 BLO 790
2001  FORMAT(//1H ,13H DIMENSIONAL ,A5,5H LOAD)                         BLO 800
2002  FORMAT(1H0,1P7E13.4)                                              BLO 810
2030  FORMAT(17H **FATAL ERROR** ,A5,16H LOAD CARD ERROR/1X)            BLO 820
2031  FORMAT(21H **FATAL ERROR** NDP=,I4,3X,A5,3X,5HNODES,8I5/15X,I5,   BLO 830
     1  21H INCREMENTS WITH INC=,8I5,24H WILL CAUSE NODAL ERROR*)        BLO 840
      END                                                               BLO 850
*DECK SLD01
      SUBROUTINE SLD01(NDIM,NDF,NDP,NPRES,IPRES,PR,XZ,FS)               SLD  10
      DIMENSION IPRES(8),PR(8),FS(6,8)                                  SLD  20
      COMMON/GAUS/ LIM,SGAUSS(5,5),WGAUSS(5,5)                          SLD  30
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SH(3,3),X(3,20),LD(120)     SLD  40
      NSIDE = NPRES - 1                                                 SLD  50
      IF(NDP.EQ.2) NSIDE = NPRES/4                                      SLD  60
      LIM = NSIDE + 1                                                   SLD  70
C**** COMPUTE NORMAL PRESSURE GENERALIZED FORCES                        SLD  80
      LIM3 = LIM                                                        SLD  90
      IF(NDP.EQ.1) LIM3 = 1                                             SLD 100
      NDP = NDP + 1                                                     SLD 110
      DO 404 II = 1,LIM3                                                SLD 120
      SS = SGAUSS(II,LIM)                                               SLD 130
      WS = WGAUSS(II,LIM)                                               SLD 140
      IF(LIM3.NE.1) GO TO 401                                           SLD 150
      SS =-1.0                                                          SLD 160
      WS = 1.0                                                          SLD 170
401   CONTINUE                                                          SLD 180
      DO 404 JJ = 1,LIM                                                 SLD 190
      TT = SGAUSS(JJ,LIM)                                               SLD 200
      WT = WGAUSS(JJ,LIM)*WS                                            SLD 210
      CALL SHICK2(TT,SS,-1.,NDIM,NPRES,NSIDE)                           SLD 220
      PN = 0.                                                           SLD 230
      DO 402 I = 1,NPRES                                                SLD 240
402   PN = PN + PR(I)*SHAPE(4,I)                                        SLD 250
      PN = PN*WT                                                        SLD 260
      DO 406 I = 1,NPRES                                                SLD 270
      DO 405 J=1,NDP                                                    SLD 280
406   FS(J,I) = FS(J,I) + SG(NDP,J)*SHAPE(4,I)*PN                       SLD 290
404   CONTINUE                                                          SLD 300
      NDP = NDP - 1                                                     SLD 310
      RETURN                                                            SLD 320
      END                                                               SLD 330
*DECK SLD02
      SUBROUTINE SLD02(NDIM,NDF,NDP,NPRES,IPRES,PR,X,FS)                SLD  10
      DIMENSION AR(3,4),IPRES(8),PR(8),X(3,20),FS(6,8)                  SLD  20
      DATA AR/-3.,-4.,1.,4.,0.,-4.,-1.,4.,3./                           SLD  30
C**** ADD HORIZONTAL AND VERTICAL COMPONENT STRESS LOADS                 SLD  40
      DO 600 K=1,3                                                      SLD  50
      DO 600 L=1,3                                                      SLD  60
      FS(1,K) = FS(1,K) + AR(K,L)*X(2,L)*PR(1)/6.0                      SLD  70
600   FS(2,K) = FS(2,K) - AR(K,L)*X(1,L)*PR(2)/6.0                      SLD  80
      RETURN                                                            SLD  90
      END                                                               SLD 100
*DECK SLD03
      SUBROUTINE SLD03(NDIM,NDF,NDP,NPRES,IPRES,PR,XZ,FS)               SLD  10
      DIMENSION IPRES(8),PR(8),FS(6,8)                                  SLD  20
```

```
      COMMON/GAUS/ LIM,SGAUSS(5,5),WGAUSS(5,5)                          SLD  3C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)     SLD  4C
      NSIDE = NPRES - 1                                                 SLD  5C
      LIM = NSIDE + 1                                                   SLD  6C
C**** COMPUTE NORMAL PRESSURE GENERALIZED FORCES                        SLD  7C
      DO 404 JJ = 1,LIM                                                 SLD  8C
      TT = SGAUSS(JJ,LIM)                                               SLD  9C
      WT = WGAUSS(JJ,LIM)                                               SLD 10C
      CALL BRICK2(TT,-1.,-1.,NDIM,NPRES,NSIDE)                          SLD 11C
      PR = 0.                                                           SLD 12C
      RR = 0.                                                           SLD 13C
      DO 402 I = 1,NPRES                                                SLD 14C
      RR = RR + SHAPE(4,I)*X(1,I)                                       SLD 15C
402   PR = PR + PR(I)*SHAPE(4,I)                                        SLD 16C
      PR = PR*WT*RR                                                     SLD 17C
      DO 406 I = 1,NPRES                                                SLD 18C
      RR = SHAPE(4,I)*PR                                                SLD 19C
      DO 406 J=1,2                                                      SLD 20C
406   FS(J,I) = FS(J,I) + SG(2,J)*RR                                    SLD 21C
404   CONTINUE                                                          SLD 22C
      RETURN                                                            SLD 23C
      END                                                               SLD 24C
*DECK BRKGEN
      SUBROUTINE BRKGEN(NUMAP,NUMEL,NDIM,NDF,NEL,NEL1,ICOD,XYZ,F,IX)   BRK  1C
      REAL LABL                                                         BRK  2C
      DIMENSION ICOD(1),XYZ(NDIM,1),IX(NEL1,1),F(NDF,1)                 BRK  3C
      COMMON /TAPES/ ITP5,ITP6                                          BRK  4C
      COMMON /LABELS/ LABL(6),XHED(3),XH,FHED(6),FH,UHED(6),UH,RHED(6),RH BRK  5C
     X ,AXOMEN,ARCRD2,ARCRD3,MEAD(12),START,CEASE,IPG,NSTR,WORD(30)     BRK  6C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)     BRK  7C
C                                                                       BRK  8C
C**** GENERATION OF NODAL COORDINATES AND BOUNDARY CODES FOR BRICK SHAPE BRK  9C
C**** REGIONS USING ELEMENT SHAPE FUNCTIONS * * *                       BRK 10C
C                                                                       BRK 11C
C**** INPUT THE REGION PROPERTIES                                       BRK 12C
C                                                                       BRK 13C
      READ(ITP5,1006) NN,NR,NS,NT,NI,NE,MA,NBC,IREUSE,IPRINT,IMSH,IELM  BRK 14C
      READ(ITP5,1001) I1,I2,I3,I4,I5,I6                                 BRK 15C
      READ(ITP5,1002) ((X(I,J),I=1,3),J=1,NN)                           BRK 16C
      IF(NS.LE.0) NS=1                                                  BRK 17C
      IF(NR.LE.0) NR=1                                                  BRK 18C
      IF(NT.LE.0) NT=1                                                  BRK 19C
      IF(NI.LE.0) NI = 1                                                BRK 20C
      IF(NE.LE.0) NE = 1                                                BRK 21C
      IF(MA.LE.0) MA = 1                                                BRK 22C
      WRITE(ITP6,2000) HEAD,IPG,NR,NS,NT,NI,NE,MA                       BRK 23C
      WRITE(ITP6,2100) I1,I2,I3,I4,I5,I6,(XHED(I),XH,I=1,NDIM)          BRK 24C
      DO 10 N = 1,NN                                                    BRK 25C
10    WRITE(ITP6,2001) (X(I,N),I=1,NDIM)                                BRK 26C
C                                                                       BRK 27C
C**** SET THE CONTROL CONSTANTS                                         BRK 28C
C                                                                       BRK 29C
      NSIDE = (NN-1)/NDIM**2                                            BRK 30C
      DR = 2./FLOAT(NR)                                                 BRK 31C
      DS = 2./FLOAT(NS)                                                 BRK 32C
      DT = 2./FLOAT(NT)                                                 BRK 33C
      NR = NR + 1                                                       BRK 34C
      NS = NS + 1                                                       BRK 35C
      NT = NT + 1                                                       BRK 36C
      IF(NDIM.EQ.2) NT = 1                                              BRK 37C
      NRS = NR*NS                                                       BRK 38C
      NF = NRS*NT + NI - 1                                              BRK 39C
      IF(NF.GT.NUMNP) GO TO 400                                         BRK 40C
      IF(NBC.NE.0) GO TO 30                                             BRK 41C
C                                                                       BRK 42C
C**** SET THE BOUNDARY CONDITION CODES TO ZERO                          BRK 43C
C                                                                       BRK 44C
      DO 20 I = NI,NF                                                   BRK 45C
      DO 15 J = 1,NDF                                                   BRK 46C
15    F(J,I) = 0.0                                                      BRK 47C
20    ICOD(I) = 0                                                       BRK 48C
30    CONTINUE                                                          BRK 49C
```

```
      N = N1                                                            BRK 50C
      ME = NE - 1                                                       BRK 51C
      T = -1.0                                                          BRK 52C
      DO 300 K = 1,NT                                                   BRK 53C
      S = -1.0                                                          BRK 54C
      DO 200 J = 1,NS                                                   BRK 55C
      R = -1.0                                                          BRK 56C
      DO 100 I = 1,NR                                                   BRK 57C
      CALL BRICK2(R,S,T,NDIM,NN,NSIDE)                                  BRK 58C
C                                                                       BRK 59C
C.... COMPUTE THE NODAL COORDINATES OF THE N -TH NODE                   BRK 60C
C                                                                       BRK 61C
      DO 50 L = 1,NDIM                                                  BRK 62C
      CC = 0.                                                           BRK 63C
      DO 40 M = 1,NN                                                    BRK 64C
40    CC = CC +  SHAPE(4,M)*X( L,M)                                     BRK 65C
50    XYZ(L,N) = CC                                                     BRK 66C
C                                                                       BRK 67C
C.... SET THE SURFACE BOUNDARY CONDITIONS TO PRESCRIBED CONDITIONS      BRK 68C
C                                                                       BRK 69C
      IF(I.EQ.1 )  ICOD(N) = ICOD(N) + I1                               BRK 70C
      IF(I.EQ.NR)  ICOD(N) = ICOD(N) + I2                               BRK 71C
      IF(J.EQ.1 )  ICOD(N) = ICOD(N) + I3                               BRK 72C
      IF(J.EQ.NS)  ICOD(N) = ICOD(N) + I4                               BRK 73C
      IF(K.EQ.1 )  ICOD(N) = ICOD(N) + I5                               BRK 74C
      IF(K.EQ.NT)  ICOD(N) = ICOD(N) + I6                               BRK 75C
      N = N + 1                                                         BRK 76C
      IF(K .EQ.NT .AND. K.NE.1) GO TO 100                               BRK 77C
      IF(I.EQ.NR .OR. J.EQ. NS) GO TO 100                               BRK 78C
      ME = ME + 1                                                       BRK 79C
      IX(NEL1 ,ME) = MA                                                 BRK 80C
      IX(1,ME) = N - 1                                                  BRK 81C
      IX(2,ME) = N                                                      BRK 82C
      IX(3,ME) = N + NR                                                 BRK 83C
      IX(4,ME) = N + NR - 1                                             BRK 84C
      IF(NDIM.EQ.2) GO TO 100                                           BRK 85C
      IX(5,ME) = N + NRS -1                                             BRK 86C
      IX(6,ME) = N + NRS                                                BRK 87C
      IX(7,ME) = N + NRS + NR                                           BRK 88C
      IX(8,ME) = N + NRS + NR -1                                        BRK 89C
100   R = R + DR                                                        BRK 90C
200   S = S + DS                                                        BRK 91C
300   T = T + DT                                                        BRK 92C
      IF(IREUSE .EQ. 0) GO TO 360                                       BRK 93C
      J = (IREUSE - 1)*100 + MA                                         BRK 94C
      DO 320 I = NE,ME,IREUSE                                           BRK 95C
320   IX(NEL1,I) = J                                                    BRK 96C
360   IF(IPRINT .NE. 0)IX(NEL1,NE) = IX(NEL1,NE) + 10                   BRK 97C
      IF(IMSH.EQ.0) CALL PRTMSH(NI,NF,NDIM,ICOD,XYZ)                    BRK 98C
      IF(IELM.EQ.0) CALL PRTELM(NE,ME,NEL,NEL1,IX)                      BRK 99C
      IF(ME.GT.NUMEL) NUMEL = ME                                        BRK100C
      RETURN                                                            BRK101C
400   WRITE(IPR,2030)                                                   BRK102C
      IPR = 0                                                           BRK103C
      RETURN                                                            BRK104C
1000  FORMAT(1A1S)                                                      BRK105C
1001  FORMAT(6I10)                                                      BRK106C
1002  FORMAT(10X,3F10.0)                                                BRK107C
2000  FORMAT(1H1,12A6,3(X,4HPAGE,I4//17H NODE GENERATIONS//             BRK108C
     X    10X,25HNUMBER OF R-INCREMENTS     ,I5/                       BRK109C
     X    10X,25HNUMBER OF S-INCREMENTS     ,I5/                       BRK110C
     X    10X,25HNUMBER OF T-INCREMENTS     ,I5/                       BRK111C
     X    10X,25HFIRST NODES NUMBER         ,I5/                       BRK112C
     X    10X,25HFIRST ELEMENT NUMBER       ,I5/                       BRK113C
     X    10X,25HELEMENT MATERIAL NUMBER    ,I5/ 1X)                   BRK114C
2100  FORMAT(                                                           BRK115C
     X    10X,25H1-FACE BOUNDARY CODE       ,I10/                      BRK116C
     X    10X,25H2-FACE BOUNDARY CODE       ,I10/                      BRK117C
     X    10X,25H3-FACE BOUNDARY CODE       ,I10/                      BRK118C
     X    10X,25H4-FACE BOUNDARY CODE       ,I10/                      BRK119C
     X    10X,25H5-FACE BOUNDARY CODE       ,I10/                      BRK120C
     X    10X,25H6-FACE BOUNDARY CODE       ,I10//5X,3(2A6))           BRK121C
```

```
2001  FORMAT(5X,,F3F12.3)                                                    HRK122C
2030  FORMAT(5X,47H **FATAL ERROR** INSUFFICIENT STORAGE FOR NODES)          HRK123C
      END                                                                    HRK124C
*DECK ELOADS
      SUBROUTINE ELOADS(NLD,NDIM,NDF,NEN,NEL1,NSTF,NSIZV,NVEC,ITYPE,D,       ELO  1C
     1  ICOD,XYZ,IX,F,FORCE,ESTIF,VECT)                                      ELO  2C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UM,RHED(6),RM     ELO  3C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)          ELO  4C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SR(3,3),X(3,20),LD(120)          ELO  5C
      COMMON /IAPLS/ ITP5,ITP6                                               ELO  6C
      DIMENSION ITYPE(1),ICOD(1),XYZ(NDIM,1),IX(NEL1,1),F(NDF,1),            ELO  7C
     1   FORCE(NDF,1),PR(6)                                                  ELO  8C
C.... LOADINGS ASSOCIATED WITH ELEMENTS                                      ELO  9C
      DO 60 M = 1,NLD                                                        ELO 10C
      READ(ITP5,1000) IEL,ELD,INC,JEL,PR                                     ELO 11C
      INC = IABS(INC)                                                        ELO 12C
      IF(INC.EQ.0) INC = 1                                                   ELO 13C
      IF(JEL.EQ.0) JEL = IEL                                                 ELO 14C
      IF(JEL.GT.IEL) GO TO 120                                               ELO 15C
      IC = IEL                                                               ELO 16C
      IEL = JEL                                                              ELO 17C
      JEL = IC                                                               ELO 18C
120   N = IEL                                                                ELO 19C
      DO 100 JM = 1,30                                                       ELO 20C
      IF(ELD.EQ.WORD(JM)) GO TO 200                                          ELO 21C
100   CONTINUE                                                               ELO 22C
      DM = BM                                                                ELO 23C
110   WRITE(ITP6,2030) ELD,BM,N                                              ELO 24C
      IPG = 0                                                                ELO 25C
      RETURN                                                                 ELO 26C
200   DO 500 N = IEL,JEL,INC                                                 ELO 27C
      MA = MOD(IX(NEL1,N),10)                                                ELO 28C
      DM = ITYPE(MA)                                                         ELO 29C
      IF(DM.NE.ELD) GO TO 110                                                ELO 30C
      DO 310 I = 1,NSTF                                                      ELO 31C
310   FORCE(I,1) = 0.                                                        ELO 32C
      DO 330 I = 1,NEN                                                       ELO 33C
      K = IX(I,N)                                                            ELO 34C
      IF(K.EQ.0) GO TO 330                                                   ELO 35C
      LD(I) = K                                                              ELO 36C
      NEL = I                                                                ELO 37C
      DO 320 J = 1,NDIM                                                      ELO 38C
320   X(J,I) = XYZ(J,K)                                                      ELO 39C
330   CONTINUE                                                               ELO 40C
      CALL ELMLIB(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,       ELO 41C
     1 IX,F,FORCE,ESTIF,PR,VECT,5)                                           ELO 42C
C.... ADD FORCES TO THE GENERALIZED FORCES                                   ELO 43C
      MCT = MCT - NEL - 2                                                    ELO 44C
      IF(MCT.GT.0) GO TO 350                                                 ELO 45C
      WRITE(ITP6,2000) HEAD,IPG,(FHED,FM,I=1,NDF)                            ELO 46C
      IPG = IPG + 1                                                          ELO 47C
      MCT = 50                                                               ELO 48C
350   WRITE(ITP6,2001) N,(I,PR(I),I=1,6)                                     ELO 49C
      DO 400 I = 1,NEL                                                       ELO 50C
      K = LD(I)                                                              ELO 51C
      IC = ICOD(K)                                                           ELO 52C
      IL = 100000                                                            ELO 53C
      DO 430 J = 1,NDF                                                       ELO 54C
      IF(IC.LT.IL) GO TO 420                                                 ELO 55C
440   IC = IC - IL                                                           ELO 56C
      IF(IC.GE.IL) GO TO 440                                                 ELO 57C
      GO TO 430                                                              ELO 58C
420   F(J,K) = F(J,K) + FORCE(J,I)                                           ELO 59C
430   IL = IL/10                                                             ELO 60C
410   CONTINUE                                                               ELO 61C
420   WRITE(ITP6,2002) K,(FORCE(J,1),J=1,NDF)                                ELO 62C
500   CONTINUE                                                               ELO 63C
600   CONTINUE                                                               ELO 64C
      RETURN                                                                 ELO 65C
1000  FORMAT(I5,1X,A5,I4,I5,6F10.0)                                          ELO 66C
2000  FORMAT(1H1,12A6,3X,4HPAGE,I4//5X,13HELEMENT LOADS//6X,4HNODE,          ELO 67C
     1 6(1X,2A4))                                                            ELO 68C
```

```
2001  FORMAT(/5X,7HELEMENT,I5,6H LOADS,6(I3,1PE13.4,2H ,))                    FLO 69C
2002  FORMAT(I1),1P6E13.4)                                                    FLO 70C
2030  FORMAT(17H **FATAL ERROR** ,A5,25H LOAD CARD ERROR, TYPE = , A5,        FLO 71C
     1  12H FOR ELEMENT, I5)                                                 FLO 72C
      END                                                                     FLO 73C
*DECK BCODES
      SUBROUTINE BCODES(NUMNP,ICOD)                                           BCO   1C
C                                                                             BCO   2C
C.... BOUNDARY CODE GENERATOR                                                 BCO   3C
C                                                                             BCO   4C
C.... ASSUME ICOD IS FILLED AND ONLY CHANGES ARE TO OCCUR                     BCO   5C
C.... IF  NOT USER SHOULD INITIALIZE BY PLACING TWO CARDS WITH N=1 AND        BCO   6C
C.... NX =1 ON FIRST CARD AND N = NUMNP AND NX = 0 ON SECOND                  BCO   7C
C                                                                             BCO   8C
      COMMON/LABELS/ LABL(6),XHED(3),XH,FHED(6),FH,UHED(6),UH,RHED(6),RH      BCO   9C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)           BCO  10C
      COMMON/IAPES/ ITP5,ITP6                                                 BCO  11C
      DIMENSION ICOD(1),IBC(6),NBC(6)                                         BCO  12C
      N = 0                                                                   BCO  13C
      NX = 1                                                                  BCO  14C
100   NP = N                                                                  BCO  15C
      NXP = NX                                                                BCO  16C
      READ(ITP5,1000) N,NX,IBC                                                BCO  17C
      IF(N.LE.0.OR.N.GT.NUMNP) GO TO 500                                      BCO  18C
      IT = IABS(IBC(1))                                                       BCO  19C
      DO 200 I = 2,6                                                          BCO  20C
200   IT = 10*IT + IABS(IBC(I))                                               BCO  21C
      ICOD(N) = IT                                                            BCO  22C
      IF(NXP.EQ.0) GO TO 100                                                  BCO  23C
      IF((N-NP)*NXP.GE.0) GO TO 300                                           BCO  24C
      NXP = - NXP                                                             BCO  25C
      WRITE(ITP6,2030) NP,N                                                   BCO  26C
300   NP = NP + NXP                                                           BCO  27C
      IF(NXP.GT.0.AND.NP.GE.N) GO TO 100                                      BCO  28C
      IF(NXP.LT.0.AND.NP.LE.N) GO TO 100                                      BCO  29C
      IT = 0                                                                  BCO  30C
      IF(IBC(1).LT.0) IT = 1                                                  BCO  31C
      DO 400 I = 2,6                                                          BCO  32C
      IT = 10*IT                                                              BCO  33C
      IF(IBC(I).LT.0) IT = IT + 1                                             BCO  34C
400   CONTINUE                                                                BCO  35C
      ICOD(NP) = IT                                                           BCO  36C
      GO TO 300                                                               BCO  37C
C                                                                             BCO  38C
C.... OUTPUT THE NONZERO CODES                                                BCO  39C
C                                                                             BCO  40C
500   MCT = 0                                                                 BCO  41C
      J = 0                                                                   BCO  42C
      DO 600 N = 1,NUMNP                                                      BCO  43C
      IF(ICOD(N).EQ.0) GO TO 600                                              BCO  44C
      J = J + 1                                                               BCO  45C
      NBC(J) = N                                                              BCO  46C
      IBC(J) = ICOD(N)                                                        BCO  47C
      IF(J.LT.6.AND.N.LT.NUMNP) GO TO 600                                     BCO  48C
      MCT = MCT - 1                                                           BCO  49C
      IF(MCT.GT.0) GO TO 610                                                  BCO  50C
      WRITE(ITP6,2000) HEAD,IPG                                               BCO  51C
      IPG = IPG + 1                                                           BCO  52C
      MCT = 56                                                                BCO  53C
610   WRITE(ITP6,2001) (NBC(I),IBC(I),I=1,J)                                  BCO  54C
      J = 0                                                                   BCO  55C
600   CONTINUE                                                                BCO  56C
      RETURN                                                                  BCO  57C
C                                                                             BCO  58C
C.... FORMATS                                                                 BCO  59C
C                                                                             BCO  60C
1000  FORMAT(I5I5)                                                            BCO  61C
2000  FORMAT(1H1,12A6,3X,4HPAGE,I4//23H NONZERO BOUNDARY CODES//              BCO  62C
     1  6(5X,12HNODE/NO. CODE,3X))                                           BCO  63C
 2001 FORMAT (6(I9,1H,,I7,3X))
2030  FORMAT( 49H **NON FATAL ERROR** NODE INCREMENT BETWEEN NODES,I5,         BCO  65C
     1   4H AND,I5,34H OF INCORRECT SIGN. SIGN CHANGED**)                     BCO  66C
```

```
      END                                                               HCO 67C
*DECK VECIN
      SUBROUTINE VECIN(NUMNP,NUMEL,NDF,NSIZV,IPICK,N    ,PRTV,VECT)     VEC  1C
C                                                                       VEC  2C
C.... VECIN ALLOWS INPUT OF USER DEFINED VECTORS.                       VEC  3C
C                                                                       VEC  4C
C.... IPICK IS A CODED PARAMETER AS FOLLOWS.                            VEC  5C
C....     IPICK = 0, IMPLIES VECTORS ARE ASSOCIATED WITH NODES          VEC  6C
C....     IPICK = 1, IMPLIES VECTORS ARE ASSOCIATED WITH DEGREES FREEDOM VEC  7C
C....     IPICK = 2, IMPLIES VECTORS ARE ASSOCIATED WITH ELEMENTS       VEC  8C
C                                                                       VEC  9C
      DIMENSION DX(7),PRTV(2,N),VECT(NSIZV,N),VWORD(6)                  VEC 10C
      REAL LAML                                                         VEC 11C
      COMMON /TAPES/ ITPS,ITP6                                          VEC 12C
      COMMON/LABELS/ LABL(6),XMED(3),XM,FMED(6),FM,UMED(6),UM,RMED(6),RM VEC 13C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)     VEC 14C
      DATA VWORD/6HNODAL ,6HPOINTS,6HDEG. F,6HREEDOM,6HNO. EL,6HEMENTS / VEC 15C
      DATA BLANK,F0,PRINT / 6H      , 6H    NO,6H PRINT /               VEC 16C
C                                                                       VEC 17C
C.... INPUT HEADER DESCRIPTIONS                                         VEC 18C
      READ(ITPS,1000) ((PRTV(I,J),I=1,2),J=1,N)                         VEC 19C
      IPRINT = 0                                                        VEC 20C
      DO 10 I=1,2                                                       VEC 21C
      DO 10 J=1,N                                                       VEC 22C
      IF(PRTV(I,J) .NE. BLANK) IPRINT = 1                               VEC 23C
10    CONTINUE                                                          VEC 24C
      IF(IPRINT .EQ. 1) GO TO 11                                        VEC 25C
      PRTV(1,1) = F0                                                    VEC 26C
      PRTV(2,1) = PRINT                                                 VEC 27C
11    CONTINUE                                                          VEC 28C
      DO 12 I = 1,NSIZV                                                 VEC 29C
      DO 12 J = 1,N                                                     VEC 30C
12    VECT(I,J) = 0.0                                                   VEC 31C
      MCT = 0                                                           VEC 32C
      L = 0                                                             VEC 33C
      IF(IPICK .LT. 0 .OR. IPICK .GT. 2)  GO TO 450                     VEC 34C
      IP1 = 2*IPICK + 1                                                 VEC 35C
      IP2 = 2*IPICK + 2                                                 VEC 36C
      NSKIP = 1                                                         VEC 37C
      IF(IPICK .EQ. 1) NSKIP = NDF                                      VEC 38C
      I = 0                                                             VEC 39C
      NXP = 0                                                           VEC 40C
200   IP = 1                                                            VEC 41C
      NX = NXP                                                          VEC 42C
      READ(ITPS,1001) I,NXP,(DX(J),J=1,N)                               VEC 43C
      IF(I.LE.0.OR.I.GT.NSIZV) GO TO 290                                VEC 44C
      DO 21  J = 1,N                                                    VEC 45C
210   VECT(I,J) = DX(J)                                                 VEC 46C
      IF(NX.EQ.0) GO TO 200                                             VEC 47C
      IF((I-IP)*NX.GE.0) GO TO 220                                      VEC 48C
      NX = - NX                                                         VEC 49C
220   LX = (IABS(I-IP)+IABS(NX)-1)/IABS(NX)                             VEC 50C
      DO 23  J = 1,N                                                    VEC 51C
230   DX(J) = (VECT(I,J)-VECT(IP,J))/LX                                 VEC 52C
240   IP = IP + NX                                                      VEC 53C
      IF(NX.GT.0.AND.IP.GE.I) GO TO 200                                 VEC 54C
      IF(NX.LT.0.AND.IP.LE.I) GO TO 200                                 VEC 55C
      DO 25  J = 1,N                                                    VEC 56C
250   VECT(IP,J) = VECT(IP-NX,J) + DX(J)                                VEC 57C
      GO TO 240                                                         VEC 58C
290      IF(IPRINT.EQ.0) GO TO 500                                      VEC 59C
C.... OUTPUT VECTORS                                                    VEC 60C
      IK = 0                                                            VEC 61C
      DO 300 K = 1,NSIZV                                                VEC 62C
      KK = (K-1)/NSKIP                                                  VEC 63C
      KK = KK*NSKIP + 1                                                 VEC 64C
       IF(KK .EQ. K) IK = IK + 1                                        VEC 65C
      MCT = MCT + 1                                                     VEC 66C
      IF(MCT.GT.0) GO TO 300                                            VEC 67C
      WRITE(ITP6,2000) HEAD,IPG,VWORD(IP1),VWORD(IP2),NSIZV,PRTV        VEC 68C
      IPG = IPG + 1                                                     VEC 69C
      MCT = 50                                                          VEC 70C
```

```
  300 WRITE(ITP6,2001)IK,(VECT(K,J),J=1,N)                              VEC 71C
      GO TO 500                                                         VEC 72C
  400 WRITE(ITP6,3000) I                                                VEC 73C
      IPG = 0                                                           VEC 74C
      GO TO 500                                                         VEC 75C
  450 WRITE(ITP6,3001) IPICK                                            VEC 76C
      IPG = 0                                                           VEC 77C
  500 CONTINUE                                                          VEC 78C
      IF(IPRINT .EQ. 0)WRITE(ITP6,2000)HEAD,IPG,VWORD(IP1),VWORD(IP2),  VEC 79C
     X NSIZV,PRTV                                                       VEC 80C
      RETURN                                                            VEC 81C
C.... FORMATS                                                           VEC 82C
 1000 FORMAT(6X,2A6)                                                    VEC 83C
 1001 FORMAT(2I5,7F10.0)                                                VEC 84C
 2000 FORMAT(1H1,12A6,3(6X,4HPAGE,I4//5X,                              VEC 85C
     X 31H INPUT VECTORS ASSOCIATED WITH ,2A6,5X,16H VECTOR LENGTH =I5// VEC 86C
     X 4X,6HNUMBER, 7(3X,2A6)/1X)                                       VEC 87C
 2001 FORMAT(I10,7E15.5)                                                VEC 88C
 3000 FORMAT(25H0VECTOR CARD ERROR, NODE=,I5)                           VEC 89C
 3001 FORMAT(32H VECTOR PARAMETER ERROR, IPICK = I5)                    VEC 90C
      END                                                               VEC 91C
*DECK PLOT2D
      SUBROUTINE PLOT2D(ISW,KSTEPS,HED,NUMNP,NUMEL,ICOD,XYZ,IX,VECT,
     X NVEC,NSIZV,NELI,NICD,TIME,V,NDF)
      COMMON/MINHW/NEW(1500)
      COMMON/PLOTS/IPLOT
      DIMENSION NSTEPS(2),PRSCAN(2),IPSCAN(2),IDUM(10),HED(12),ICOD(1),
     X XYZ(NDF,1),IX(NELI,1),VECT(1),V(NDF,1)
      DATA INPUTCK,RA,NSTEPS,PRSCAN,IPSCAN,IDUM,DUM
     X/0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,10*0.0,0.0/
C
      IF(IPLOT.EQ.3) GO TO 100
C
C     PLOTING FOR TWO DIMENSIONAL PROBLEMS WITH FOUR NODE QUADS
C
      IF(ISW .GE. 1) GO TO 50
C
C     WRITE MESH DATA TO TAPE 30
C
      NUMMAT = TIME
      NSTEPS(1)=KSTEPS
      NSTEPS(2)=KSTEPS
      DO 20 N=1,NUMNP
      VECT(N)=0.0
      IC=ICOD(N)
      IL=10000
      K=0
      DO 10 J=1,2
    5 IF(IC.LT.IL) GO TO 10
      IC=IC-IL
      IF(IC.GT.IL) GO TO 5
      K=K+1
      VECT(N)=J
      IF(K.EQ.2) VECT(N)=3
   10 IL=IL/10
   20 CONTINUE
C
      NE=NELI-1
      DO 30 N=1,NUMEL
      RA=MOD(IX(NELI,N),10)
   30 IX(NE,N)=RA
C
      WRITE(30) (HED(I),I=1,8),NUMNP,NUMEL,NUMMAT,INPUTCK,RA,NSTEPS,
     XPRSCAN,(IPSCAN,IDUM,DUM,DUM
      WRITE(30) (VECT(N),XYZ(1,N),XYZ(2,N),N=1,NUMNP),(((IX(I,N),I=1,5),
     XN=1,NUMEL),DUM,DUM
C
      DO 40 N=1,NUMEL
   40 IX(NE,N)=0
C
      MMM=0
      RETURN
```

```
C
C       WRITE SYSTEM RESPONSE RECORDS FOR EACH TIME STEP
C
50      IF(ISW .NE. 1) GO TO 75
        NMM = MMN + 1
        NMMM = MMN
        IF(NICD.EQ.3) GO TO 70
        NMMM = -MMM
        WRITE(30) NMMM,TIME,(V(1,N),V(2,N),N=1,NUMNP)
        GO TO 90
70      WRITE(30) MMM,TIME,(V(1,N),V(2,N),N=1,NUMNP),(V(1,NUMNP+N),
       XV(2,NUMNP+N),N=1,NUMNP),(V(1,2*NUMNP+N),V(2,2*NUMNP+N),N=1,NUMNP)
        GO TO 90
75      CONTINUE
        WRITE(30) NMMM,TIME,(VECT(N),VECT(NUMEL+N),VECT(2*NUMEL+N),
       XVECT(3*NUMEL+N),N=1,NUMEL)
        WRITE(30) NMMM,TIME,(VECT(4*NUMEL+N),VECT(5*NUMEL+N),
       XVECT(6*NUMEL+N),VECT(7*NUMEL+N),N=1,NUMEL)
90      CONTINUE
        RETURN
C
100     CONTINUE
        NDIM = IPLOT
        NEL = NELI-2
C
C       THREE DIMENSIONAL PLOTTING SUBROUTINE....
C
C          HED(12) ...TITLE OF PROBLEM
C          NUMNP ... NUMBER OF NODAL POINTS
C          NUMEL ... NUMBER OF ELEMENTS
C          KSTEPS ... NUMBER OF TIME STEPS
C          NICD ... NUMBER OF INITIAL CONDITIONS. 1=STATIC, 3=DYNAMIC.
C          NEL ... NUMBER OF NODES PER ELEMENT
C          NELI ... NEL + 2 ( MATERIAL LOCATION IN CONNECTIVITY ARRAY)
C          NDIM ... SPATIAL DIMENSION OF PROBLEM (2 OR 3)
C          NSIZV ... NUMBER OF STRESS/STRAIN COMPONENTS, 6.
C          ICOD(NUMNP) ... BOUNDARY CODE , SIX DIGIT BOOLIAN NUMBER.
C          XYZ(NDIM,NUMNP) ... CARTESIAN COORDINATES
C          IX(NELI,NUMEL) ... CONNECTIVITY ARRAY AND MATERIAL PROPERTY NO.
C          V(NDIM,NUMNP*NICD) ... DISPLACEMENT VECTOR IF STATIC, ALSO
C                                 VELOCITY AND ACCELERATION IF DYNAMIC.
C          VECT(2,9) ... STRESS AND STRAIN  AT ELEMENT CENTER.
C                        ORDER IS ... (XX),(XY),(XZ),(YY),(YZ),(ZZ)
C
        IF(ISW.GE.1) GO TO 200
C
C       MESH DATA RECORDS FIRST.
C
        NUMMAT = TIME
        WRITE(30) (HED(I),I=1,12),NUMNP,NUMEL,NUMMAT,KSTEPS,NICD,NDIM,NELI
        WRITE(30) (ICOD(N),(XYZ(K,N),K=1,NDIM),N=1,NUMNP),
       X  ((IX(I,N),I=1,NEL),IX(NELI,N),N=1,NUMEL),(NEW(I),I=1,1500)
        RETURN
C
200     IF(ISW.GT.1) GO TO 250
C
C       RESPONSE DATA ... DISPLACEMENTS,AND (VELOCITIES AND ACCELERATIONS
C                         IF DYNAMIC).
C
        WRITE(30)TIME,(((V(I,N+(K-1)*NUMNP),I=1,NDIM),N=1,NUMNP),K=1,NICD)
        RETURN
C
C       STRESS AND STRAIN AT ELEMENT CENTER FOR EACH ELEMENT RECORD
C
250     CONTINUE
        WRITE(30) (VECT(J),J=1,NSIZV),(VECT(J+NSIZV),J=1,NSIZV)
        RETURN
        END
*DECK SOLVEQ
        SUBROUTINE SOLVEQ(NUMNP,NUMEL,NDF,ID1,M7,MB,IIA,NSTF,ISZA,NEQB,     SOL  1C
       X IHLF,DT,F,U,IDEST,FORCE,ESTIF,FD,NDEG)                             SOL  2C
        DIMENSION DT(M7),U(1),IDEST(1),F(1),FD(NEQB)                        SOL  3C
```

```
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)        SOL  4C
C**** ADD ELEMENT STIFFNESS TO GLOBAL STIFFNESS                            SOL  5C
      IB = ID1 + ISZA - NEQB                                               SOL  6C
      IF(IBLK.GT.0)                                                        SOL  7C
     XCALL MAKEA(DT(ID1),DT(IB),ISZA,NEQB,ITA,NUMEL,NSTF,NSTF,LD,FORCE,    SOL  8C
     X ESTIF,7,IBLK)                                                       SOL  9C
C**** SOLVE EQUATIONS                                                      SOL 10C
      CALL USOL(DT,DT(ID1),FD,NEQB,MB,1,IBLK,ISZA,ITA,A,10,11,10)          SOL 11C
C**** PUT SOLUTION VECTOR INTO U(I)                                        SOL 12C
      L = NDEG                                                             SOL 13C
      N = NDEG                                                             SOL 14C
      M = NDF*NUMAP                                                        SOL 15C
      IF(IBLK.EQ.0) GO TO 200                                              SOL 16C
      N = NEQB - IBLK*NEQB + NDEG                                          SOL 17C
      REWIND 7                                                             SOL 18C
      REWIND 10                                                            SOL 19C
      READ (7) DT                                                          SOL 20C
C**** PUT THE SOLUTION BACK IN ORDER                                       SOL 21C
100   READ(10) (FD(I),I=1,N)                                               SOL 22C
200   IF(IDESP(M),NE.L) GO TO 250                                          SOL 23C
      IF(IBLK.GT.0) U(M) = FD(N)                                           SOL 24C
      IF(IBLK.EQ.0) U(M) = U(N)                                            SOL 25C
      L = L - 1                                                            SOL 26C
      IF(L.EQ.0) L = -1                                                    SOL 27C
      N = N - 1                                                            SOL 28C
      GO TO 300                                                            SOL 29C
250   U(M) = F(M)                                                          SOL 30C
300   M = M - 1                                                            SOL 31C
      IF(M.LE.0) GO TO 400                                                 SOL 32C
      IF(N.GT.0) GO TO 200                                                 SOL 33C
      IF(IBLK.EQ.0) GO TO 250                                              SOL 34C
      N = NEQB                                                             SOL 35C
      GO TO 100                                                            SOL 36C
400   CONTINUE                                                             SOL 37C
      IF(IBLK.GT.0) REWIND 7
      IF(IBLK.GT.0) READ(7) DT
      RETURN                                                               SOL 38C
      END                                                                  SOL 39C
*DECK COREIN
      SUBROUTINE COREIN(A,U,NEQ,ESTIF,FORCE,LD,NSTF)                       COR  1C
      DIMENSION A(NEQ,1),U(1),ESTIF(NSTF,NSTF),FORCE(NSTF,2),LD(1)         COR  2C
      DO 200 I = 1,NSTF                                                    COR  3C
      K = LD(I)                                                            COR  4C
      IF(K.EQ.0) GO TO 200                                                 COR  5C
      U(K) = U(K) + FORCE(I,1)                                             COR  6C
      DO 190 J = 1,NSTF                                                    COR  7C
      L = LD(J)-K+1                                                        COR  8C
      IF(L.GT.0) A(K,L) = A(K,L) + ESTIF(I,J)                              COR  9C
190   CONTINUE                                                             COR 10C
200   CONTINUE                                                             COR 11C
      RETURN                                                               COR 12C
      END                                                                  COR 13C
*DECK MAKEA
      SUBROUTINE MAKEA(A,B,ISZA,NEQB,ITA, NELM,IELM,JELM,LD,FORCE,ESTIF,   MAK  1C
     X  ITEL,IBLK)                                                         MAK  2C
C**** ASSEMBLY ROUTINE FOR ELEMENT STIFFNESS MATRICES * MODIFIED *         MAK  3C
      DIMENSION A(ISZA),LD(IELM),FORCE(IELM,2),ESTIF(IELM,IELM),B(1)       MAK  4C
      REWIND ITA                                                           MAK  5C
      MB = SQRT(FLOAT(IBLK))                                               MAK  6C
      NEMB = NEQB*MB                                                       MAK  7C
      MB = MB - 1                                                          MAK  8C
      MM =                                                                 MAK  9C
      N10 = 0                                                              MAK 10C
      NTI = 0                                                              MAK 11C
      ILOW = 1                                                             MAK 12C
      DO 400 N = 1,IBLK                                                    MAK 13C
C**** INITIALIZE RECORD                                                    MAK 14C
      DO 20 I = 1,ISZA                                                     MAK 15C
20    A(I) = 0.0                                                           MAK 16C
      IUPR = ILOW + NEQB - 1                                               MAK 17C
      ITAP = 10                                                            MAK 18C
      NEL = NTI                                                            MAK 19C
```

```
      IF(MM.GT.0) GO TO 30                                                  MAK 20C
      ITAP = ITEL                                                           MAK 21C
      NEL = NELM                                                            MAK 22C
      NIO = 0                                                               MAK 23C
       REWIND IO                                                            MAK 24C
30    CONTINUE                                                              MAK 25C
      REWIND ITAP                                                           MAK 26C
      IF(ITAP.EQ.ITEL) READ(ITAP) ZIP                                       MAK 27C
      DO 300 K = 1,NEL                                                      MAK 28C
      READ(ITAP) ESTIF,FORCE,LD                                             MAK 29C
      DO 200 I=1,JELM                                                       MAK 30C
      NR = LD(I)                                                            MAK 31C
      IF(NR.LT.ILOW.OR.NR.GT.IUPR) GO TO 200                                MAK 32C
      NR = NR - NTI                                                         MAK 33C
      B(NR) = B(NR) + FORCE(I,1)                                            MAK 34C
      DO 100 J=1,JELM                                                       MAK 35C
      NC = LD(J)                                                            MAK 36C
      IF(NC.LT.NR) GO TO 100                                                MAK 37C
      ICELL = (NC - NR)*NEQB + MR                                           MAK 38C
      A(ICELL) = A(ICELL) + ESTIF(I,J)                                      MAK 39C
100   CONTINUE                                                              MAK 40C
200   CONTINUE                                                              MAK 41C
      IF(MM.GT.0) GO TO 260                                                 MAK 42C
      IF(MB.LE.0) GO TO 260                                                 MAK 43C
      DO 220 I = 1,JELM                                                     MAK 44C
      NR = LD(I) - NTI                                                      MAK 45C
      IF(NR.GT.NEQB.AND.NR.LE.NEMB) GO TO 240                               MAK 46C
220   CONTINUE                                                              MAK 47C
      GO TO 260                                                             MAK 48C
240   WRITE(11) ESTIF,FORCE,LD                                              MAK 49C
      NIO = NIO + 1                                                         MAK 50C
260   CONTINUE                                                              MAK 51C
300   CONTINUE                                                              MAK 52C
CXXXX
C  WRITE QUICKIO FILE
CXXXX
      CALL QUICKIO (6LTAPE11,1,A)
      ILOW = IUPR + 1                                                       MAK 54C
      MM = MM + 1                                                           MAK 55C
      IF(MM.GT.MB) MM = 0                                                   MAK 56C
400   NTI = NTI + NEQB                                                      MAK 57C
      RETURN                                                                MAK 58C
      END                                                                   MAK 59C
*DECK USOL
      SUBROUTINE USOL (A,B,,MAXB,NEQB,MB,LL,NBLOCK,NSB,NORG,NBKS,NTI,       USO  1C
     X   NTC,NPST)                                                          USO  2C
      DIMENSION A(NSB),B(NSB),MAXB(NEQB)                                    USO  3C
      COMMON /TIMING/ TRICP,NTRI,BACKCP,NBACK
      NTRI=NTRI+1
      CALL SECOND (TTT)
C                                                                           USO  4C
      NMB=NEQB*MB                                                           USO  5C
      NEM = 0                                                               USO  6C
      IF(NBLOCK.EQ.0) GO TO 110                                             USO  7C
      NC=MB*LL                                                              USO  8C
      NBR=(NMB-1)/NEQB+1                                                    USO  9C
      INC=NEQB-1                                                            USO 10C
      NEQB2 = NEQB*NEQB                                                     USO 11C
      REWIND NORG
      IXNORG=0
CXXXX
C  MAXB IS SAVED ON FILE 9 (NORG)
C  EQN BLOCKS ARE RANDOMLY READ/WRITTEN IN-PLACE ON FILE 11
CXXXX
      N=1
  100 IXNORG=IXNORG+1
      IF((N.GT.1).AND.(NBR.EQ.1)) GO TO 110
      CALL QUICKIO (6LTAPE11,2,A,IXNORG)
  110 DO 300 I=1,NEQB                                                       USO 27C
      D=A(I)                                                                USO 28C
      IF(D.EQ.0.0) GO TO 300                                                USO 29C
      DO 130 J=I,NMB,NEQB                                                   USO 30C
```

```
      IF (A(J).NE.0.) MAX = J                                          USO 31C
  130 CONTINUE                                                         USO 32C
      MAXB(I) = MAX                                                    USO 33C
      JL=I+1                                                           USO 34C
      IF (JL.GT.NEQB) GO TO 300                                        USO 35C
      II=I                                                             USO 36C
      DI=I./D
      DO 200 J=JL,NEQB                                                 USO 37C
      II=II+NEQB                                                       USO 38C
      IF(II.GT.NMB) GO TO 200                                          USO 39C
      C=A(II)*DI
      IF (C.EQ.0.0) GO TO 200                                          USO 41C
      KK = J                                                           USO 42C
      CALL REDVCB ((MAX-II)/NEQB+1,NEQB,NEQB,C,A(II),A(KK))
      KK=J +NMB                                                        USO 46C
      JJ=I+NMB                                                         USO 47C
      A(KK)=A(KK)-C*A(JJ)
  200 CONTINUE                                                         USO 52C
  300 CONTINUE                                                         USO 53C
      IF(NBLOCK.EQ.0) GO TO 930                                        USO 54C
C
      WRITE (NORG) MAXB
      CALL QUICKIO (6LTAPE11,3,A,IANORG)
      IP=IANORG
C                                                                      USO 56C
C     SUBSTITUTE INTO REMAINING EQUATIONS                              USO 57C
C                                                                      USO 58C
      DO 800 NN=1,NBB                                                  USO 59C
      IF(N+NN.GT.NBLOCK) GO TO 800                                     USO 60C
      IP=IP+1
      CALL QUICKIO (6LTAPE11,2,B,IP)
      IL = NN*NEQB2 + 1                                                USO 65C
      DO 700 I=1,NEQB                                                  USO 66C
      D = A(I)                                                         USO 67C
      IF(D.EQ.0.0) GO TO 700                                           USO 68C
      II=IL                                                            USO 69C
      MAX = MAXB(I)                                                    USO 70C
      DI=1./D
      DO 690 K=1,NEQB                                                  USO 71C
      IF(II.GT.NMB) GO TO 700                                          USO 72C
      C=A(II)*DI
      IF (C.EQ.0.0) GO TO 690                                          USO 74C
      KK = K                                                           USO 75C
      CALL REDVCB ((MAX-II)/NEQB+1,NEQB,NEQB,C,A(II),B(KK))
      KK = K + NMB                                                     USO 79C
      JJ = I + NMB                                                     USO 80C
      B(KK)=B(KK)-C*A(JJ)
  690 II = II + NEQB                                                   USO 85C
  700 IL = IL - INC                                                    USO 86C
      IF(NBB.NE.1) GO TO 750                                           USO 87C
      DO 740 I=1,NSB                                                   USO 88C
  740 A(I)=B(I)                                                        USO 89C
      GO TO 800                                                        USO 90C
  750 CALL QUICKIO (6LTAPE11,3,B,IP)
  800 CONTINUE                                                         USO 92C
      N = N + 1                                                        USO 96C
      IF(N.LE.NBLOCK) GO TO 100                                        USO 97C
      CALL SECOND (TTTT)
      TRICP=TRICP+(TTTT-TTT)
      NBACK=NBACK+1
C                                                                      USO 98C
C     BACKSUBSTITUTION - RESULTS ON TAPE NRST                          USO 99C
C                                                                      USO100C
      LS=NEQB
      NEB=NEQB*(NBB+1)                                                 USO102C
      NUM=NBB*NEQB                                                     USO103C
      MAX=NSB
      DO 905 I=1,MAX                                                   USO105C
  905 B(I)= .                                                          USO106C
      REWIND NRST                                                      USO107C
      N = 1                                                            USO108C
      NN=NBLOCK+1
```

```
      937 NN=NN-1
          BACKSPACE NORG
          READ (NORG) MAXH
          BACKSPACE NORG
          CALL QUICKIO (BLIMPE11,2,A,NN)
          K=NEM
          DO 91  J=1,NUM                                                    USO114C
          I=K-NEQN                                                          USO115C
          H(K)=A(I)                                                         USO116C
      910 K=K-1                                                             USO117C
          I=NMH                                                             USO118C
          K=0
          DO 92  J=1,NEQN                                                   USO121C
          I=I+1                                                             USO122C
          K=K+1                                                             USO123C
      920 H(K)=H(I)                                                         USO124C
      930 DO 955 I=1,NEQN                                                   USO125C
          J=NEQN+I-1                                                        USO126C
          MAX = MAXH(J)                                                     USO127C
          IF (A(J).EQ.0.) GO TO 955                                         USO128C
          KK=J
          JJ=KK+1                                                           USO131C
          IL=J*NEQN                                                         USO132C
          C=H(KK)                                                           USO133C
          IF(MAX.LT.IL) GO TO 950                                           USO134C
          DO 94  II=IL,MAX,NEQN                                             USO135C
          C=C-A(II)*H(JJ)                                                   USO136C
      940 JJ=JJ+1                                                           USO137C
      950 H(KK)=C/A(J)                                                      USO138C
      955 CONTINUE                                                          USO139C
          IF(NBLOCK.EQ.1) RETURN                                            USO140C
          I=0                                                               USO141C
          K=0
          DO 96  J=1,NEQN                                                   USO144C
          K=K+1                                                             USO145C
          I=I+1                                                             USO146C
      960 A(I)=H(K)                                                         USO147C
          WRITE (NRSI) (A(I),I=1,LS)                                        USO148C
          N = N + 1                                                         USO149C
          IF(N.LE.NBLOCK) GO TO 937                                         USO150C
          CALL SECOND (TIT)
          BACKCP=BACKCP+(TIT-TIT)
          RETURN                                                            USO151C
          END                                                               USO152C
     *DECK ISOLVE
          SUBROUTINE ISOLVE(NUMEP,NUMEL,NUMMAT,NDIM,NDF,NEN,NEL,NSTF,NVEC,  TSO   1C
         1   NSIZV,NICD,NSICD,IBLK,ISZA,NEQN,MAXBAN,NDEG,ID1,MB,IBUF,DS,    TSO   2C
         2   TYPE,D,ICOD,XYZ,F,IX,IDEST,VEC1,FORCE,ESTIF,LP,A,DU,MAXH,H,U,UF,  TSO   3C
         3   NSEG,TIME)                                                      TSO   4C
          REAL LAML                                                         TSO   5C
          LOGICAL NPR,NPL                                                   TSO   6C
          DIMENSION TYPE(1),ICOD(1),XYZ(NDIM,1),F(NDF,1),IX(NEL+1),         TSO   7C
         1   IDEST(NDF,1),VEC1(NSIZV,1),FORCE(NSTF,2),ESTIF(NSTF,NSTF),     TSO   8C
         2   A(NEQN,1),DU(NDF,1),H(IBUF),U(NDF,1),DF(1),DS(MB)              TSO   9C
          COMMON/DYNAMU/TIME,NSIG(7),NI,NSTEP,DT,NUMPLT,NEDATA(20,3),NPR,NPL TSO  10C
          COMMON/LABELS/ LABL(6),XHED(3),XH,FHED(5),FH,UHED(6),UM,RHED(6),RH TSO  11C
         X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)      TSO  12C
          COMMON/SHAPE/ XJAC,SHAPE(4,20),SG(3,3),SX(3,3),X(3,20),LD(20)     TSO  13C
          COMMON/TAPES/ ITP5,ITP6                                           TSO  14C
          COMMON/VIDAT/DTP,NH,ISZH,DUL(6,20), UL(6,20),UDL(6,20),C6         TSO  15C
          COMMON/VTAPE/ ITP13,ITP14,ITRD,ITWR                               TSO  16C
          COMMON/PLOTS/IPLOT
          COMMON /MINRAY/ NEW(1500)
          LOGICAL VFLAG                                                     TSO  17C
     C**** MAIN VISCOELASTIC SUBROUTINE FOR THE GENERAL PROGRAM             TSO  18C
          DO 2 N=1,NEN                                                      TSO  19C
          DO 2 J=1,NDF                                                      TSO  20C
        2 UDL(J,N)=0.                                                       TSO  21C
          DO 3 I=2,7                                                        TSO  22C
        3 NSIG(I)=1                                                         TSO  23C
          IF(IBLK.EQ.0) GO TO 4                                             TSO  24C
          KU = NUMEP*NICD                                                   TSO  25C
```

```
      REWIND 7                                                          TSO 260
      WRITE(7) DS                                                       TSO 270
C.... SET THE OUTPUT LABELS FOR DISPLACEMENT PRINTS                     TSO 280
4     HEAD(12) = 6H TIME=                                               TSO 290
      IFLG =-NSICO                                                      TSO 300
      I = 2                                                             TSO 310
      DO 5 N = 1,NDIM                                                   TSO 320
      LABL(I) = AWORD1                                                  TSO 330
5     I = I + 1                                                         TSO 340
      LABL(I) = AWORD2                                                  TSO 350
      DTP = 0.                                                          TSO 360
      TIME = 0.                                                         TSO 370
      MH = MAXHAN + 1                                                   TSO 380
      NEI = NEM + 1                                                     TSO 390
      DO 10 N = 1,NUMEL                                                 TSO 400
10    IX(NEI,N) = 0                                                     TSO 410
C.... PUT INITIAL DATA ON TAPE                                          TSO 420
      ISZH = IHLF                                                       TSO 430
      ITRD = ITP13                                                      TSO 440
      ITWR = ITP14                                                      TSO 450
      REWIND ITRD                                                       TSO 460
      IF(IBLK.GT.0) WRITE(ITRD) ((V(I,J),I=1,NDF),J=1,KU)               TSO 470
      DO 11 N = 1,ISZH                                                  TSO 480
11    H(N) = 0.                                                         TSO 490
      NTH= (NM+ISZH-1)/ISZH                                             TSO 500
      IF(NTH.LE.1) GO TO 13                                             TSO 510
      NT = NTH*2                                                        TSO 520
      DO 12 N = 1,NT                                                    TSO 530
12    WRITE(ITRD) H                                                     TSO 540
13      REWIND ITRD                                                     TSO 550
      IF(IBLK.GT.0) READ(ITRD) DM                                       TSO 560
      NEP = NUMNP + NUMNP                                               TSO 570
      CF = 0.                                                           TSO 580
      DO 90 N = 1,NSEQ                                                  TSO 590
      READ(ITP5,1000) DT,NTS,INT,NNI,NNF,NEI,NEF,NPROP,NFORC,BETA,DEL   TSO 600
      IF(INT.LE.0) INT = 1                                              TSO 610
      GAM=0.5*DEL                                                       TSO 620
      WRITE(ITP6,2000) HEAD,TIME,IPG,DT,NTS,INT,NNI,NNF,NEI,NEF         TSO 630
      IF(NICO.NE.1) WRITE(ITP6,2002) BETA,GAM                           TSO 640
      IPG = IPG + 1                                                     TSO 650
      IF(NICO.EQ.1) GO TO 1                                             TSO 660
      IF(DT.EQ.DTP) GO TO 901                                           TSO 670
      C1=1.-GAM/BETA                                                    TSO 680
      C2=(1.-GAM/BETA/2.)*DT                                            TSO 690
      C3=GAM/BETA/DT                                                    TSO 700
      C4=1.-0.5/BETA                                                    TSO 710
      C5=1./BETA/DT                                                     TSO 720
      CA=C3/DT                                                          TSO 730
1     IF(NPROP.GT.0) PROP = PROPLD(TIME,NPROP)                          TSO 740
      IF(NFORC.GT.0.AND.NPROP.GT.0) WRITE(ITP6,4001)                    TSO 750
      VFLAG = .FALSE.                                                   TSO 760
      PROP = 1.0                                                        TSO 770
      DO 80 NT = 1,NTS                                                  TSO 780
      IF(NPROP.GT.0) PROP = PROPLD(TIME,DT,0)                           TSO 790
      IF(NFORC.GT.0) CALL RESET(-NFORC,NUMNP,NDF,F)                     TSO 800
      IF(IBLK.EQ.0) GO TO 15                                            TSO 820
      IF(.NOT.VFLAG) GO TO 20                                           TSO 840
      DO 14 I = 1,NDEG                                                  TSO 850
14    DF(I) = 0.0                                                       TSO 860
      GO TO 2?                                                          TSO 870
15    DO 19 N = 1,NUMNP                                                 TSO 880
      DO 18 K = 1,NDF                                                   TSO 890
      J = IDEST(K,N)                                                    TSO 900
      IF(J.EQ.0) GO TO 18                                               TSO 910
      IF(VFLAG) GO TO 17                                                TSO 920
      DO 16 I = 1,MAXHAN                                                TSO 930
16    A(J,I) = 0.0                                                      TSO 940
17    DF(J) = F(K,N)*PROP                                               TSO 950
18    CONTINUE                                                          TSO 960
19    CONTINUE                                                          TSO 970
20    NH = 1                                                            TSO 980
C.... OUTPUT THE SOLUTION VECTOR FOR THE CURRENT TIME                   TSO100C
```

```
      IF(IPLOT .GT. 0)CALL PLOT2D(1,NTS,HEAD,NUMNP,NUMEL,ICOD,XYZ,IX,
     X VECT,NVEC,NSIZV,NEL1,NICD,TIME,U,NDF)
      NSIG(1) = 1
      IF(NN1.EQ.1.OR.(N1/INT)*INT.NE.N1) GO TO 40                        TS0101C
      IF(N.EQ.1 .AND. N1.EQ.1) GO TO 40
      NSIG(1) = 0
      MCT = 0                                                            TS0102C
      DO 30 IJ=N1,NNP
      N=NEW(IJ)
      MCT = MCT - 1                                                      TS0104C
      IF(MCT.GT.0) GO TO 30                                              TS0105C
      WRITE(IPR,2001) HEAD,TIME,IPG,PROP,(XHED(I),XH,I=1,NDIM),
     X                (UHED(I),UH,I=1,NDF)                              TS0107C
      IPG = IPG + 1                                                      TS0108C
      MCT = 50                                                           TS0109C
   30 WRITE (IPR,LABL) IJ,(XYZ(I,N),I=1,NDIM),(U(I,N),I=1,NDF)
40    IF(NICD.EQ.1) GO TO 44                                             TS0111C
C**** UPDATE FOR DYNAMIC PROBLEMS                                        TS0112C
      DO 42 N = 1,NUMNP                                                  TS0113C
      DO 42 I = 1,NDF                                                    TS0114C
      TEMP = U(I,N+NUMNP)                                                TS0115C
      DM   = U(I,N+NEP)                                                  TS0116C
      U(I,N+NUMNP) = C1*TEMP + C2*DM                                     TS0117C
42    U(I,N+NEP  ) = C4*DM - C5*TEMP                                     TS0118C
44    MR = 0                                                             TS0119C
      REWIND ITMP
      IF(IREST.GT.0) WRITE(ITMP) ((U(I,J),I=1,NDF),J=1,KU)
      IF(NTS.GT.1) READ(ITMP) M
      TEMP = 0.                                                          TS0120C
      MCT = 0
      DO 49 N = 1,NUMEL                                                  TS0121C
      NPR = .TRUE.                                                       TS0122C
      IF(N.GE.NEL1.AND.N.LE.NEP) NPR = .FALSE.                           TS0123C
      IF(M.EQ.1 .AND. N1.EQ.1) NPR=.TRUE.
      MA = MOD(IX(NEL1,N),10)                                            TS0124C
      IF(MA.LE.0) MRR= IX(NEL1,N)/100                                    TS0125C
      IF(MR.LE.0) MR = MRR                                               TS0126C
      DO 60 I = 1,NSTF                                                   TS0127C
      FORCE(I,1) = 0.                                                    TS0128C
      FORCE(I,2)=0.                                                      TS0129C
      LD(I) = 0                                                          TS0130C
      IF(MR.NE.MRR.OR.VFLAG) GO TO 60                                    TS0131C
      DO 50 J = 1,NSTF                                                   TS0132C
50    ESTIF(I,J) = 0.0                                                   TS0133C
60    CONTINUE                                                           TS0134C
      L = 0                                                              TS0135C
      DO 110 I = 1,NEN                                                   TS0136C
      K = IX(I,N)                                                        TS0137C
      IF(K.LE.0)  GO TO 120                                              TS0138C
      NEL = I                                                            TS0139C
      DO 100 J = 1,NDIM                                                  TS0140C
100   X(J,I) = XYZ(J,K)                                                  TS0141C
      DO 110 J = 1,NDF                                                   TS0142C
      IF (NICD.NE.1) UDL(J,I)=U(J,K+NEP)                                 TS0143C
      DUL(J,I) = U(J,K)                                                  TS0144C
      UL(J,I) = U(J,K)                                                   TS0145C
      L = L + 1                                                          TS0146C
110   LD(L) = IDEST(J,K)                                                 TS0147C
120   DM = TYPE(MA)                                                      TS0148C
C     (REMOVED PAGE CHANGER)
      TEMP = DM                                                          TS0150C
C**** COMPUTE ELEMENT STRESSES AND UPDATE FORCES                         TS0151C
      CALL ELMLIB(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,   TS0152C
     X IX,M,FORCE,ESTIF,U,VECT,6)                                        TS0153C
      IF(IPLOT .EQ. 3)CALL PLOT2D(2,NTS,HEAD,NUMNP,NUMEL,ICOD,XYZ,IX,
     X VECT,NVEC,NSIZV,NEL1,NICD,TIME,U,NDF)
C**** FORM STIFFNESSIF NEEDED FOR THE NEXT TIME STEP                     TS0154C
      IF((.NOT.VFLAG.AND.MR.EQ.MRR).OR.(VFLAG.AND.IX(NEL1,N).EQ.1))      TS0155C
     XCALL ELMLIB(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,   TS0156C
     X IX,M,FORCE,ESTIF,U,VECT,3)                                        TS0157C
C**** MODIFY FOR THE DISPLACEMENT B.C.                                   TS0158C
      DO 130 L = 1,NEL                                                   TS0159C
```

```
      J = IX(L,N)                                                          TS0160C
      IC = ICOD(J)                                                         TS0161C
      IL = 100030                                                          TS0162C
      NE = NDF*(L-1)                                                       TS0163C
      DO 125 K = 1,NDF                                                     TS0164C
      NE = NE + 1
      IF(DT.LT.0.0) FORCE(NE,1) = 0.0
      Q = P(K,J)*PROP                                                      TS0165C
      IF(Q.EQ.0.0) GO TO 122                                               TS0166C
      IF(IC.LT.IL.AND.IBLK.GT.0) P(K,J) = 0.0                              TS0167C
      IF(IC.LT.IL) GO TO 123                                               TS0169C
      IX(NE,N) = 1                                                         TS0170C
      DO 121 I = 1,NSTF                                                    TS0171C
121   FORCE(I,1) = FORCE(I,1) - ESTIF(I,NE)*Q                              TS0172C
122   IC = IC - IL                                                         TS0173C
      IF(IC.GT.IL) GO TO 122                                               TS0174C
123   IF(IBLK.GT.0) FORCE(NE,1) = FORCE(NE,1) + Q                          TS0175C
125   IL = IL/10                                                           TS0176C
130   CONTINUE                                                             TS0177C
      IF(VFLAG) GO TO 300                                                  TS0178C
      IF(IBLK.EQ.0) CALL COREIN(A,DF,NEQB,ESTIF,FORCE,LD,NSTF)             TS0179C
      IF(IBLK.GT.0) WRITE(7) ESTIF,FORCE,LD                                TS0180C
      GO TO 400                                                            TS0181C
300   CONTINUE                                                             TS0182C
C**** ADD THE FORCE TO THE SOLUTION FOR A RESOLVE                          TS0183C
      DO 310 K = 1,NSTF                                                    TS0184C
      J = LD(K)                                                            TS0185C
      IF(J.GT.0) DF(J) = DF(J) + FORCE(K,1)                                TS0186C
310   CONTINUE                                                             TS0187C
400   CONTINUE                                                             TS0188C
      IF(IPLOT.EQ.2)CALL PLOT2D(2,NTS,HEAD,NUMNP,NUMEL,ICOD,AYZ,IX,
     X VECT,NVEC,NSIZV,NEL1,NICD,TIME,U,NDF)
      IF(NTS.GT.1) WRITE(ITWR) H                                           TS0189C
      IF(IBLK.GT.0) WRITE(ITWR) (IX(NE1,N),N=1,NUMEL)                      TS0190C
      IF(.NOT.VFLAG) CALL SOLVEQ(NUMNP,NUMEL,NDF,ID1,MB,MAXBAN,9,NSTF,     TS0191C
     1   ISZA,NEQB,IBLK,A,DU,DF,IDEST,FORCE,ESTIF,MAXB,NDEG)               TS0192C
      IF(VFLAG) CALL RESVEQ(NUMNP,NDF,MB,MAXBAN,ISZA,NEQB,IBLK,A,DF,DF,    TS0193C
     1   IDEST,IDEST,MAXB,IFLG)                                            TS0194C
C**** UPDATE THE SOLUTION                                                  TS0195C
      I = ITRD                                                             TS0197C
      ITRD = ITWR                                                          TS0198C
      ITWR = I                                                             TS0199C
      IF(IBLK.GT.0) BACKSPACE ITRD                                         TS0200C
      IF(IBLK.GT.0) READ(ITRD) (IX(NE1,N),N=1,NUMEL)                       TS0201C
      REWIND ITRD                                                          TS0202C
      IF(IBLK.GT.0) READ(ITRD) ((U(I,J),I=1,NDF),J=1,KU)                   TS0203C
      K = 0                                                                TS0204C
      DO 700 N = 1,NUMNP                                                   TS0205C
      DO 700 I = 1,NDF                                                     TS0206C
      K = K + 1                                                            TS0207C
      DU(I,N) = DF(K)                                                      TS0208C
      IF(IDEST(I,N).EQ.0) DU(I,N) = P(I,N)*PROP - U(I,N)                   TS0209C
      TEMP=DU(I,N)                                                         TS0210C
      IF(DT.LT.0.0) U(I,N) = 0.0
      IF(NILD.EQ.1) GO TO 700                                              TS0211C
      U(I,N+NUMNP)=U(I,N+NUMNP)+C3*TEMP                                    TS0212C
      U(I,N+NEP)=U(I,N+NEP)+C6*TEMP                                        TS0213C
700   U(I,N)=U(I,N)+TEMP                                                   TS0214C
      VFLAG = .TRUE.                                                       TS0215C
      DTP = DT                                                             TS0216C
      MR = MR - 1                                                          TS0217C
800   TIME = TIME + DT                                                     TS0218C
900   CONTINUE                                                             TS0219C
      RETURN                                                               TS0220C
901   WRITE(IIPS,2030)                                                     TS0221C
      IPG = 0                                                              TS0222C
      RETURN                                                               TS0223C
C**** FORMATS                                                              TS0224C
1000  FORMAT(8I10,8,8I5,2E10.3)                                            TS0225C
2000  FORMAT(1H1,12A6,E13.5,17X,4HPAGE,I4//5X,23HTIME DEPENDENT SOLUTION   TS0226C
     X//10X,17HTIME INCREMENT  = 1PE15.4/                                  TS0227C
     X  10X,17HNUMBER OF STEPS = I5/                                       TS0228C
```

```
     X  10X,17HPRINT INTERVAL  = I5/                                          TS0229C
     X  10X,14HPRINT NODES    ,I5,3H TO,I5/                                   TS0230C
     X  10X,14HPRINT ELEMENTS,I5,3H TO,I5)                                    TS0231C
2001 FORMAT(1H1,12A6,E13.5,17X,4HPAGE,I4//5X,12HNODAL VALUES,
     X 50X,4H ERR=,E13.5//12H NODAL POINT,9(1X,2A6))
2002 FORMAT(1X,30HNEWMARK INTEGRATION PARAMETERS/                             TS0234C
     C10X17HBETA VALUE      = F6.3/                                          TS0235C
     C10X17HGAMMA VALUE     = F6.3)                                          TS0236C
2136 FORMAT(5X,53H**FATAL ERROR 33** TIME STEP ZERO FOR DYNAMIC PROBLEM      TS0237C
     X/1X)                                                                   TS0238C
 4001 FORMAT(1HO,**WARNING** BOTH THE PROPORTIONAL LOADING AND FORCE ARE
     X BEING RESET ON EACH TIME STEP)                                        TS0240C
      END                                                                    TS0241C
*DECK RESVEC
      SUBROUTINE RESVEC(NUMNP,NDF,M7,MB,ISZA,NEQB,IBLK,DT,F,U,IDES,IDEST RES  1C
     X ,MAXB,NES)                                                            RES  2C
      DIMENSION DT(M7),F(NDF,1),U(1),IDEST(1),IDES(1)                        RES  3C
      NF = IABS(NES)                                                         RES  4C
      IF(IBLK.GT.0) REWIND 7                                                 RES  5C
      IF(NES.LE.0) GO TO 125                                                 RES  6C
      REWIND 10                                                              RES  7C
C.... SAVE THE DESTINATION VECTOR FOR UNPACKING THE SOLUTION                 RES  8C
      WRITE(10)(IDEST(I),I=1,NE)                                             RES  9C
C.... COMPACT THE FORCE VECTOR, F, INTO U.                                   RES 10C
      DO 100 I = 1,NE                                                        RES 11C
      J = IDEST(I)                                                           RES 12C
      IF(J.EQ.0) GO TO 100                                                   RES 13C
      U(J) = F(I)                                                            RES 14C
100   CONTINUE                                                               RES 15C
C.... USE RESOLV TO GET THE NEW SOLUTION                                     RES 16C
125   CALL RESOLV (DT,U,ISZA,NEQB,MB,IBLK,B,MAXB)                            RES 17C
C.... REEXPAND THE SOLUTION VECTOR                                           RES 18C
      IF(IBLK.EQ.0.OR.NES.LT.0) GO TO 150                                    RES 19C
      REWIND 10                                                              RES 20C
      READ(10)(IDES(I),I=1,NE)                                               RES 21C
150   N = NE                                                                 RES 22C
      DO 200 I = 1,NE                                                        RES 23C
      IF(IBLK.GT.0) J = IDES(N)                                              RES 24C
      IF(IBLK.EQ.0) J = IDEST(N)                                             RES 25C
      IF(J.EQ.0) U(N) = 0.                                                   RES 26C
      IF(J.NE.0) U(N) = U(J)                                                 RES 27C
200   N = N - 1                                                              RES 28C
      IF(IBLK.GT.0) READ(7) DT                                               RES 29C
      RETURN                                                                 RES 30C
      END                                                                    RES 31C
*DECK RESOLV
      SUBROUTINE RESOLV (A,U,NSH,NEQB,MB,NBLOCK,NBKS,MAXB)                   RES  1C
      DIMENSION A(NSH),U(1),MAXB(NEQB)                                       RES  2C
      COMMON /TIMING/ TMICP,NTRI,BACKCP,NBACK
      NBACK=NBACK+1
      CALL SECOND (TTT)
C                                                                            RES  3C
C**** RESOLVE USING DECOMPOSED MATRIX                                        RES  4C
C                                                                            RES  5C
CXXXX
C  NOTE  MAXB WAS STORED ON FILE NORG (9) IN SBRT USOL
C  NORG IS SET TO 9 IN THIS ROUTINE
CXXXX
      DATA NORG/9/
      REWIND NORG
      JJ =                                                                   RES  7C
      DO 300 N = 1,NBLOCK                                                    RES  8C
C                                                                            RES  9C
      IF(NBLOCK.EQ.0) GO TO 50
      READ (NORG) MAXB
      CALL QUICKIO (QLTAPE(1),2,A,N)
C                                                                            RES 11C
   50 DO 200 I=1,NEQB
      JJ=JJ+1
      D=A(I)
      IF(D.EQ.0.0) GO TO 200
      UJ=U(JJ)
```

```
      IF(UJ.EQ.0.) GO TO 200
      NMB=MAXB(I)
      II=NEQB+1
      CALL REDVC6 ((NMB-II)/NEQB+1,NEQB,1,UJ/D,A(II),U(JJ+1))
  200 CONTINUE
300   CONTINUE                                                    RES 28C
C                                                                 RES 29C
C**** BACKSUBSTITUTION                                            RES 30C
C                                                                 RES 31C
      NN=NBLOCK+1
      JJ = NEQB*NBLOCK                                            RES 32C
      IF(NBLOCK.EQ.0) JJ = NEQB                                   RES 33C
      DO 600 N = 1,NBLOCK                                         RES 34C
C                                                                 RES 35C
      IF(NBLOCK.EQ.0) GO TO 350                                   RES 36C
      NN=NN-1
      BACKSPACE NORG
      READ (NORG) MAXB
      BACKSPACE NORG
      CALL QUICKIO (6LTAPE11,2,A,NN)
C                                                                 RES 40C
  350 IL=NEQB+1
      DO 500 I=1,NEQB
      IL=IL-1
      D=A(IL)
      IF(D.EQ.0.) GO TO 500
      UJ=U(JJ)
      II=IL+NEQB
      KK=JJ
      NMB=MAXB(IL)
      DO 400 J=II,NMB,NEQB
      KK=KK+1
      UJ=UJ-A(J)*U(KK)
  400 CONTINUE
      U(JJ)=UJ/D
  500 JJ=JJ-1
600   CONTINUE                                                    RES 58C
      CALL SECOND (TTTT)
      BACKCP=BACKCP+(TTTT-TTT)
      RETURN                                                      RES 59C
      END                                                         RES 60C
*DECK PROPLD
      FUNCTION PROPLD(T,N)                                        ROPL  1C
      DIMENSION TABLE(9,5),TLAB(3)                                ROPL  2C
      COMMON /TAPES/ ITP5,ITP6                                    ROPL  3C
      DATA TLAB/10HPOLYNOMIAL,10HPERIODIC  ,10HUSER INPUT/        ROPL  4C
      IF(N.EQ.0) GO TO 200                                        ROPL  5C
C**** INPUT TABLE                                                 ROPL  6C
      NTERMS = N                                                  ROPL  7C
      READ(ITP5,1000) ((TABLE(I,J),I=1,9),J=1,N)                  ROPL  8C
      WRITE(ITP6,2001)                                            ROPL  9C
      DO 100 J = 1,N                                              ROPL 10C
      K = TABLE(1,J)                                              ROPL 11C
      IF(K.LT.1.OR.K.GT.3) GO TO 700                              ROPL 12C
      L = TABLE(2,J)                                              ROPL 13C
100   WRITE(ITP6,2000) TLAB(K),L,(TABLE(I,J),I=3,9)               ROPL 14C
      RETURN                                                      ROPL 15C
200   PROPLD = 0.                                                 ROPL 16C
C**** INTERPOLATE THE TABLE                                       ROPL 17C
      DO 300 J = 1,NTERMS                                         ROPL 18C
      TMIN = TABLE(3,J)                                           ROPL 19C
      TMAX = TABLE(4,J)                                           ROPL 20C
      IF(T.LT.TMIN.OR.T.GT.TMAX) GO TO 300                        ROPL 21C
      K = TABLE(1,J)                                              ROPL 22C
      GO TO (201,202,203),K                                       ROPL 23C
201   TT = 1.                                                     ROPL 24C
      DO 211 I = 5,9                                              ROPL 25C
      PROPLD = PROPLD + TABLE(I,J)*TT                             ROPL 26C
211   TT = TT*T                                                   ROPL 27C
      RETURN
C82 LPOH                                                  003 OT UG
202   K = TABLE(2,J)                                              ROPL 29C
```

```
      IF(K.EQ.0) K = 1                                                  HOPL 30C
      PROPLD = PROPLD + TABLE(9,J) +TABLE(5,J)*(SIN(TABLE(6,J)*T))**K   HOPL 31C
     X    + TABLE(7,J)*(COS(TABLE(8,J)*T))**K                          HOPL 32C
      RETURN                                                            HOPL 33C
C33 LPOP                                                    003 07 06
203   CALL EXPRLD(PROP,T,TABLE(2,N))                                    HOPL 34C
      PROPLD = PROP                                                     HOPL 35C
300   CONTINUE                                                          HOPL 36C
      RETURN                                                            HOPL 37C
700   WRITE(IIP6,2030) K                                                HOPL 38C
      IPG = 0                                                           HOPL 39C
      RETURN                                                            HOPL 40C
1000  FORMAT(2F5.0,7F10.0)                                              HOPL 41C
2000  FORMAT(2X,A10,I5,7E14.5)                                          HOPL 42C
2001  FORMAT(//5X,23HPROPORTIONAL LOAD TABLE//                          HOPL 43C
     X 3X,9HTYPE LOAD,5H  EXP,5X,9HMIN. TIME,5X,9HMAX. TIME,5X,        HOPL 44C
     X 2HA0,12X,2HA1,12X,2HA2,12X,2HA3,12X,2HA4/1X)                     HOPL 45C
2030  FORMAT(5(F PROPORTIONAL LOAD INPUT TABLE ERROR. INPUT TYPE =,I5/)  HOPL 46C
      END                                                               HOPL 47C
*DECK RESET
      SUBROUTINE RESET(NN,NUMNP,NDF,F)
      COMMON /TAPES/ ITPS,ITP6                                          RES  2C
      REAL LABL                                                         RES  3C
      DIMENSION F(NDF,1)                                                RES  4C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UH,RHED(6),RM RES  5C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)      RES  6C
      COMMON /MINOW/ NEW(1500)
      DIMENSION FT(7)
      MCT = 0                                                           RES  7C
      N=NN
      IF(N.GT.0) GO TO 200                                              RES  8C
      DO 100 I = 1,NUMNP                                                RES  9C
      DO 100 J = 1,NDF                                                  RES 10C
  100 F(J,I)=0.0                                                        RES 11C
      IF(N.EQ.0) RETURN                                                 RES 12C
      N = -N                                                            RES 13C
  200 CONTINUE                                                          RES 14C
      READ (ITP5,1000) J,(FT(I),I=1,NDF)
      K=NEW(J)
      DO 210 I=1,NDF
  210 F(I,K)=FT(I)
      MCT = MCT - 1                                                     RES 16C
      IF(MCT.GT.0) GO TO 250                                            RES 17C
      WRITE(ITP6,2000) HEAD,IPG,NUMNP,(FHED(I),FM,I= 1,NDF)             RES 18C
      IPG = IPG + 1                                                     RES 19C
      MCT = 50                                                          RES 20C
  250 WRITE (ITP6,2001) J,(FT(I),I=1,NDF)
      IF(J.GE.NUMNP) RETURN                                             RES 22C
      IF(J.GE.N) RETURN                                                 RES 23C
      GO TO 200                                                         RES 24C
1000  FORMAT(I5,5X,7F10.0)                                              RES 25C
2000  FORMAT(1H1,12A6,30X,4HPAGE,I4//I5,13H NODAL FORCES//              RES 26C
     X 12H NODAL POINT,7(1X,2A6))                                       RES 27C
2001  FORMAT(I12,6E13.4)                                                RES 28C
      END                                                               RES 29C
*DECK ELMLIB
      SUBROUTINE ELMLIB(N,MA,NDIM,NDF,NEL,NELI,NSTF,NSIZV,NVEC,MCT,DM,D, ELM  1C
     X XYZ,IX,F,FORCE,ESTIF,U,VECT,ISW)                                 ELM  2C
      REAL LABL                                                         ELM  3C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UH,RHED(6),RM ELM  4C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)     ELM  5C
C                                                                       ELM  6C
C**** ELEMENT LIBRARY FOR FEAP72                                        ELM  7C
C                                                                       ELM  8C
C****    ISW = 1 FOR MATERIAL CHARACTERIZATION                          ELM  9C
C****    ISW = 2 FOR CHECK ON MESH                                      ELM 10C
C****    ISW = 3 FOR ELEMENT STIFFNESS FORMULATION                      ELM 11C
C****    ISW = 4 FOR ELEMENT OUTPUTS                                    ELM 12C
C****    ISW = 5 FOR ELEMENT LOAD/VECTOR COMPUTATIONS                   ELM 13C
C****    ISW = 6 FOR NON-LINEAR LOAD VECTOR COMPUTATIONS                ELM 14C
C                                                                       ELM 15C
      DO 55 J = 1,30                                                    ELM 16C
```

```
      IF(DM.EQ.KWORD(J)) GO TO 57                                       ELM 17C
56    CONTINUE                                                          ELM 18C
57    GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,  ELM 19C
     X 23,24,25,26,27,28,29,30) , J                                     ELM 20C
1     CALL ELMT01(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 21C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 22C
      GO TO 99                                                          ELM 23C
2     CALL ELMT02(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 24C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 25C
      GO TO 99                                                          ELM 26C
3     CALL ELMT03(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 27C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 28C
      GO TO 99                                                          ELM 29C
4     CALL ELMT04(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 30C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 31C
      GO TO 99                                                          ELM 32C
5     CALL ELMT05(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 33C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 34C
      GO TO 99                                                          ELM 35C
6     CALL ELMT06(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 36C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 37C
      GO TO 99                                                          ELM 38C
7     CALL ELMT07(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 39C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 40C
      GO TO 99                                                          ELM 41C
8     CALL ELMT08(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 42C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 43C
      GO TO 99                                                          ELM 44C
9     CALL ELMT09(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 45C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 46C
      GO TO 99                                                          ELM 47C
10    CALL ELMT10(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 48C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 49C
      GO TO 99                                                          ELM 50C
11    CALL ELMT11(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 51C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 52C
      GO TO 99                                                          ELM 53C
12    CALL ELMT12(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 54C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 55C
      GO TO 99                                                          ELM 56C
13    CALL ELMT13(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 57C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 58C
      GO TO 99                                                          ELM 59C
14    CALL ELMT14(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 60C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 61C
      GO TO 99                                                          ELM 62C
15    CALL ELMT15(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 63C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 64C
      GO TO 99                                                          ELM 65C
16    CALL ELMT16(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 66C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 67C
      GO TO 99                                                          ELM 68C
17    CALL ELMT17(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 69C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 70C
      GO TO 99                                                          ELM 71C
18    CALL ELMT18(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 72C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 73C
      GO TO 99                                                          ELM 74C
19    CALL ELMT19(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 75C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 76C
      GO TO 99                                                          ELM 77C
20    CALL ELMT20(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 78C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 79C
      GO TO 99                                                          ELM 80C
21    CALL ELMT21(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 81C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 82C
      GO TO 99                                                          ELM 83C
22    CALL ELMT22(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 84C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 85C
      GO TO 99                                                          ELM 86C
23    CALL ELMT23(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 87C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 88C
```

```
      GO TO 99                                                          ELM 89C
24    CALL ELMT24(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 90C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 91C
      GO TO 99                                                          ELM 92C
25    CALL ELMT25(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 93C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 94C
      GO TO 99                                                          ELM 95C
26    CALL ELMT26(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 96C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM 97C
      GO TO 99                                                          ELM 98C
27    CALL ELMT27(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM 99C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM100C
      GO TO 99                                                          ELM101C
28    CALL ELMT28(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM102C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM103C
      GO TO 99                                                          ELM104C
29    CALL ELMT29(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM105C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM106C
      GO TO 99                                                          ELM107C
30    CALL ELMT30(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,XYZ,  ELM108C
     X IX,F,FORCE,ESTIF,U,VECT,ISW)                                     ELM109C
99    RETURN                                                            ELM110C
      END                                                               ELM111C
*DECK INTEGL
      SUBROUTINE INTEGL(LIM,NC1,NDIM,LINT,STUW)                         INT  1C
      REAL LABL                                                         INT  2C
      DIMENSION SIRI(7,3),TIRI(7,3),UIRI(7,3),WIRI(7,3),STUW(4,1)       INT  3C
      DIMENSION SIR(7),TIR(7),UIR(7),WB(2),SB(2),WC(2),SC(2)            INT  4C
      COMMON/GAUSS/ LZZ,SGAUSS(5,5),WGAUSS(5,5)                         INT  5C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UM,RHED(6),RM INT  6C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)     INT  7C
      DATA SIRI/0.33333333,6*0.0,0.5,0.0,0.5,4*0.0,0.33333333,0.05971587, INT  8C
     C          2*0.47014206,0.79742699,2*0.10128651/                  INT  9C
      DATA TIRI/0.33333333,6*0.0,2*0.5,0.0,4*0.0,0.33333333,0.47014206, INT 10C
     C          0.05971587,0.47014206,0.10128651,0.79742699,0.10128651/ INT 11C
      DATA UIRI/0.33333333,6*0.0,0.0,2*0.5,4*0.0,0.33333333,2*0.47014206 INT 12C
     C          ,0.05971587,2*0.10128651,0.79742699/                   INT 13C
      DATA WIRI/0.1,0.7*0.33333333,0.225,3*0.13239415,3*0.12593918/     INT 14C
      DATA SIR,TIR,UIR/1.0,2*0.0,4*1.0,0.,1.,0.,2*1.,-1.,-1.,0.,0.,1.,1. INT 15C
     C               ,  -1.,-1.,1./                                     INT 16C
      DATA WB,SB,WC,SC/1.3333333,0.886426593,1.0,0.795822426,0.0,       INT 17C
     C                 0.335180055,0.0,0.758786911/                     INT 18C
C**** INTEGRATION TABLE CONSTRUCTION                                    INT 19C
      IF(NC1.LT.0) NC1 = 0                                              INT 20C
      IF(NC1.GT.2) NC1 = 2                                              INT 21C
      NC = NC1 + 1                                                      INT 22C
      GO TO (241,244,249),NC                                            INT 23C
C**** NC1 = 0, GAUSS INTEGRATION                                        INT 24C
241   II = 0                                                            INT 25C
      IF(LIM.LT.1) LIM = 2                                              INT 26C
      IF(LIM.GT.5) LIM = 5                                              INT 27C
      LIM3 = LIM                                                        INT 28C
      IF(NDIM.EQ.2) LIM3 = 1                                            INT 29C
      DO 243 I = 1,LIM3                                                 INT 30C
      UU = SGAUSS(I,LIM)                                                INT 31C
      WU = WGAUSS(I,LIM)                                                INT 32C
      IF(NDIM.NE.2) GO TO 242                                           INT 33C
      UU = -1.0                                                         INT 34C
      WU = 1.0                                                          INT 35C
242   CONTINUE                                                          INT 36C
      DO 243 J = 1,LIM                                                  INT 37C
      TT = SGAUSS(J,LIM)                                                INT 38C
      WT = WGAUSS(J,LIM)*WU                                             INT 39C
      DO 243 K = 1,LIM                                                  INT 40C
      II = II + 1                                                       INT 41C
      STUW(1,II) = SGAUSS(K,LIM)                                        INT 42C
      STUW(2,II) = TT                                                   INT 43C
      STUW(3,II) = UU                                                   INT 44C
      STUW(4,II) = WGAUSS(K,LIM)*WT                                     INT 45C
243   CONTINUE                                                          INT 46C
      LINT = II                                                         INT 47C
      GO TO 248                                                         INT 48C
```

```
C**** NCI = 1, TRONS INTEGRATION                                        INT 49C
244   IF(NDIM.NE.3.OR.LIM.EQ.1) GO TO 241                               INT 50C
      LINT = 6                                                          INT 51C
      IF(LIM.GE.2) LINT = 14                                            INT 52C
      RS = SR(LIM)                                                      INT 53C
      RW = WR(LIM)                                                      INT 54C
      DO 240 I = 1,7                                                    INT 55C
      I2 = 2*I                                                          INT 56C
      I1 = I2 - 1                                                       INT 57C
      IF(I.NE.4) GO TO 245                                              INT 58C
      RS = SC(LIM)                                                      INT 59C
      RW = WC(LIM)                                                      INT 60C
245   CONTINUE                                                          INT 61C
      STUW(1,I1) = SIR(I)*RS                                            INT 62C
      STUW(1,I2) = -STUW(1,I1)                                          INT 63C
      STUW(2,I1) = TIR(I)*RS                                            INT 64C
      STUW(2,I2) = -STUW(2,I1)                                          INT 65C
      STUW(3,I1) = UIR(I)*RS                                            INT 66C
      STUW(3,I2) = -STUW(3,I1)                                          INT 67C
      STUW(4,I1) = RW                                                   INT 68C
240   STUW(4,I2) = RW                                                   INT 69C
      GO TO 248                                                         INT 70C
C**** NCI = 2, TRIANGULAR COORDINATE INTEGRATION                        INT 71C
249   LINT = 1                                                          INT 72C
      IF(LIM.GT.3) LIM = 3                                              INT 73C
      IF(LIM.EQ.2) LINT = 3                                             INT 74C
      IF(LIM.GE.3) LINT = 7                                             INT 75C
      DO 247 I = 1,LINT                                                 INT 76C
      STUW(1,I) = STRI(I,LIM)                                           INT 77C
      STUW(2,I) = TTRI(I,LIM)                                           INT 78C
      STUW(3,I) = UTRI(I,LIM)                                           INT 79C
247   STUW(4,I) = WTRI(I,LIM)                                           INT 80C
248   CONTINUE                                                          INT 81C
      RETURN                                                            INT 82C
      END                                                               INT 83C
*DECK COMPACT
      SUBROUTINE COMPACT(ESTIF,FORCE,ISIZE,NSIZE,JSIZE,ITAPE)           COM  1C
C                                                                       COM  2C
C**** REDUCE LAST EQUATIONS OF THE NSIZE X NSIZE STIFFNESS MATR   TO A  COM  3C
C**** FINAL JSIZE X JSIZE STIFFNESS MATRIX & STORE ON ITAPE             COM  4C
C                                                                       COM  5C
      DIMENSION ESTIF(ISIZE,ISIZE),FORCE(ISIZE)                         COM  6C
      II=NSIZE-JSIZE                                                    COM  7C
      IF(II.LE.0) RETURN                                                COM  8C
      DO 700 II=1,II                                                    COM  9C
      KK=NSIZE-II                                                       COM 10C
      K1=KK+1                                                           COM 11C
      C = ESTIF(K1,K1)                                                  COM 12C
      IF(C.EQ.0.0) GO TO 700                                            COM 13C
      DO 650 JJ=K1,KK                                                   COM 14C
      ESTIF(K1,JJ)=ESTIF(K1,JJ)/C                                       COM 15C
      CC=ESTIF(K1,JJ)                                                   COM 16C
      FORCE(JJ)=FORCE(JJ)-CC*FORCE(K1)                                  COM 17C
      DO 650 LL=1,KK                                                    COM 18C
  650 ESTIF(LL,JJ)=ESTIF(LL,JJ)-ESTIF(LL,K1)*CC                         COM 19C
      FORCE(K1)=FORCE(K1)/C                                             COM 20C
  700 CONTINUE                                                          COM 21C
      IF(ITAPE.GT.0) WRITE(ITAPE) ESTIF,FORCE                           COM 22C
      RETURN                                                            COM 23C
      END                                                               COM 24C
*DECK CKQUAD
      SUBROUTINE CKQUAD(FLAG,K,NEL)                                     CKQ  1C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)     CKQ  2C
      FLAG = 1.                                                         CKQ  3C
      SS = 1.0                                                          CKQ  4C
      DO 30 I = 1,2                                                     CKQ  5C
      SS = -SS                                                          CKQ  6C
      TT = 1.0                                                          CKQ  7C
      DO 30 J = 1,2                                                     CKQ  8C
      TT = -TT                                                          CKQ  9C
      CALL USHPR(SS,TT,K,XXX,YYY,NEL)                                   CKQ 10C
      IF(K.EQ.1) CALL THERM(SG,SK,1.0)                                  CKQ 11C
```

```
      IF(XJAC.LE.0.0) GO TO 40                                          CKG 12C
30    CONTINUE                                                          CKG 13C
      RETURN                                                            CKG 14C
40    FLAG = XJAC                                                       CKG 15C
      RETURN                                                            CKG 16C
      END                                                               CKG 17C
*DECK CKHRIK
      SUBROUTINE CKHRIK(NDIM,NEN,FLAG)                                  CKH  1C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)     CKH  2C
      FLAG = 1.                                                         CKH  3C
      NSIDE = (NEN-1)/NDIM**2                                           CKH  4C
      L = 1                                                             CKH  5C
      IF(NDIM.EQ.2) L = 2                                               CKH  6C
      SS = 1.0                                                          CKH  7C
      DO 30 I = L,2                                                     CKH  8C
      SS = -SS                                                          CKH  9C
      TT = 1.0                                                          CKH 10C
      DO 30 J = 1,2                                                     CKH 11C
      TT = -TT                                                          CKH 12C
      UU = 1.0                                                          CKH 13C
      DO 30 K = 1,2                                                     CKH 14C
      UU = -UU                                                          CKH 15C
      CALL BRICK2(UU,TT,SS,NDIM,NEN,NSIDE)                             CKH 16C
      IF(XJAC.LE.0.0) GO TO 40                                          CKH 17C
30    CONTINUE                                                          CKH 18C
      RETURN                                                            CKH 19C
40    FLAG = XJAC                                                       CKH 20C
      RETURN                                                            CKH 21C
      END                                                               CKH 22C
*DECK PACKD
      SUBROUTINE PACKD(D,I,C11,C12,C33,M)                               PAC  1C
      DIMENSION D(3,21,1)                                               PAC  2C
      GO TO (291,293,294),I                                             PAC  3C
291   CONTINUE                                                          PAC  4C
      D( 1, 1,M) = C11                                                  PAC  5C
      D( 1, 2,M) = C12                                                  PAC  6C
      D( 2, 2,M) = C11                                                  PAC  7C
      D( 2, 1,M) = C12                                                  PAC  8C
      D( 3, 3,M) = 4.0*C33                                              PAC  9C
      RETURN                                                            PAC 10C
293   CONTINUE                                                          PAC 11C
      D(1, 7,M) = C12                                                   PAC 12C
      D(2,15,M) = C12                                                   PAC 13C
      D(3,19,M) = C11                                                   PAC 14C
294   CONTINUE                                                          PAC 15C
      D(1, 1,M) = C11                                                   PAC 16C
      D(1, 5,M) = C12                                                   PAC 17C
      D(1,13,M) = C33                                                   PAC 18C
      D(1,16,M) = C33                                                   PAC 19C
      D(2, 2,M) = C33                                                   PAC 20C
      D(2, 4,M) = C33                                                   PAC 21C
      D(2,11,M) = C11                                                   PAC 22C
      D(2,17,M) = C33                                                   PAC 23C
      D(3, 3,M) = C33                                                   PAC 24C
      D(3, 7,M) = C33                                                   PAC 25C
      D(3,12,M) = C33                                                   PAC 26C
      D(3,14,M) = C33                                                   PAC 27C
      RETURN                                                            PAC 28C
      END                                                               PAC 29C
*DECK BRICK2
      SUBROUTINE BRICK2(R,S,T,NDIM,NEL,NSIDE)                           BRI  1C
      DIMENSION N1(12),N2(12),N3(12),NODE(12),S2ORD(12),S3ORD(12),SS(3) BRI  2C
      DIMENSION COLM(3)                                                 BRI  3C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)     BRI  4C
      DATA NODE/2,4,6,8,9,10,11,12,14,16,18,20/                         BRI  5C
      DATA  N1/1,2,1,2,3,3,3,3,1,2,1,2/ ,N2/2,3,2,3,1,1,1,1,2,3,2,3/   BRI  6C
      DATA  N3/3,1,3,1,2,2,2,2,3,1,3,1/                                 BRI  7C
      DATA S2ORD/-.5,-.5, .5,-.5,-.5, .5, .5,-.5,-.5, .5, .5,-.5/       BRI  8C
      DATA S3ORD/-.5, .5,-.5,-.5,-.5,-.5, .5, .5, .5, .5, .5,-.5/      BRI  9C
      SS(1) = R                                                         BRI 10C
      SS(2) = S                                                         BRI 11C
      SS(3) = T                                                         BRI 12C
```

```
      K1 = 4                                                                    ARI 130
      L1 = 1                                                                    ARI 140
      IF(NSIDE.EQ.1) GO TO 110                                                  ARI 150
      K1 = 12                                                                   ARI 160
      L1 = 2                                                                    ARI 170
C**** FORM MIDSIDE SHAPE FUNCTIONS                                              ARI 180
      DO 100 L=1,12                                                             ARI 190
      N = NODE(L)                                                               ARI 200
      I = N1(L)                                                                 ARI 210
      J = N2(L)                                                                 ARI 220
      K = N3(L)                                                                 ARI 230
      SJ = S2ORD(L)                                                             ARI 240
      SL = S3ORD(L)                                                             ARI 250
      RP = 1. -SS(I)**2                                                         ARI 260
      SP = 0.5 + SJ*SS(J)                                                       ARI 270
      TP = 0.5 + SL*SS(K)                                                       ARI 280
      SHAPE(I,N) = -2.0*SS(I)*SP*TP                                             ARI 290
      SHAPE(J,N) = SJ*RP*TP                                                     ARI 300
      SHAPE(K,N) = SL*RP*SP                                                     ARI 310
100   SHAPE(4,N) = RP*SP*TP                                                     ARI 320
C**** FORM CORNER SHAPE FUNCTIONS                                               ARI 330
110   SJ = -0.5                                                                 ARI 340
      K = 0                                                                     ARI 350
      DO 250 J = 1,2                                                            ARI 360
      L = 1                                                                     ARI 370
      DO 200 I = 5,8                                                            ARI 380
      RP = .5 + S2ORD(I)*K                                                      ARI 390
      SP = .5 + SJORD(I)*S                                                      ARI 400
      TP = .5 + SJ*T                                                            ARI 410
      SHAPE(I,L + K) = S2ORD(I)*SP*TP                                           ARI 420
      SHAPE(2,L + K) = RP*S3ORD(I)*TP                                           ARI 430
      SHAPE(3,L + K) = RP*SP*SJ                                                 ARI 440
      SHAPE(4,L + K) = RP*SP*TP                                                 ARI 450
200   L = L + L1                                                                ARI 460
      K = K1                                                                    ARI 470
250   SJ = 0.5                                                                  ARI 480
      IF(NSIDE.EQ.1) GO TO 360                                                  ARI 490
C**** CORRECT BASIC CORNER FUNCTIONS BY PROPORTIONS OF MIDSIDE FUNCTIONS        ARI 500
      K = 8                                                                     ARI 510
      L = 9                                                                     ARI 520
      DO 350 I = 1,8,2                                                          ARI 530
      DO 300 J = 1,4                                                            ARI 540
      SHAPE(J,I) = SHAPE(J,I)-0.5*(SHAPE(J,I+1)+SHAPE(J,K)+SHAPE(J,L))          ARI 550
300   SHAPE(J,I+12)=SHAPE(J,I+12)-0.5*(SHAPE(J,I+13)+SHAPE(J,K+12)              ARI 560
     X  + SHAPE(J,L))                                                           ARI 570
      K = I + 1                                                                 ARI 580
350   L = L + 1                                                                 ARI 590
C**** FORM THE JACOBIAN DETERMANANT                                             ARI 600
360   DO 370 I = 1,3                                                           ARI 610
      DO 370 J = 1,3                                                           ARI 620
370   SK(I,J) = 0.0                                                             ARI 630
      DO 400 I = 1,NDIM                                                         ARI 640
      DO 400 J = 1,NDIM                                                         ARI 650
      DO 400 K = 1,NEL                                                          ARI 660
400   SK(I,J) = SK(I,J) + SHAPE(J,K)*X(I,K)                                     ARI 670
      IF(NDIM.EQ.2) SK(3,3) = 1.0                                               ARI 680
      SG(1,1) =  SK(2,2)*SK(3,3) - SK(2,3)*SK(3,2)                              ARI 690
      SG(2,2) =  SK(1,1)*SK(3,3) - SK(1,3)*SK(3,1)                              ARI 700
      SG(3,3) =  SK(1,1)*SK(2,2) - SK(1,2)*SK(2,1)                              ARI 710
      SG(1,2) = -SK(1,2)*SK(3,3) + SK(1,3)*SK(3,2)                              ARI 720
      SG(1,3) =  SK(1,2)*SK(2,3) - SK(1,3)*SK(2,2)                              ARI 730
      SG(2,1) = -SK(2,1)*SK(3,3) + SK(2,3)*SK(3,1)                              ARI 740
      SG(2,3) = -SK(1,1)*SK(2,3) + SK(2,1)*SK(1,3)                              ARI 750
      SG(3,1) =  SK(2,1)*SK(3,2) - SK(2,2)*SK(3,1)                              ARI 760
      SG(3,2) = -SK(1,1)*SK(3,2) + SK(1,2)*SK(3,1)                              ARI 770
      XJAC = 0.0                                                                ARI 780
      DO 600 I = 1,NDIM                                                         ARI 790
600   XJAC = XJAC + SG(I,1)*SK(1,I)                                             ARI 800
      DO 800 J=1,NEL                                                            ARI 810
      DO 710 I = 1,NDIM                                                         ARI 820
      TEMP = 0.                                                                 ARI 830
      DO 700 K = 1,NDIM                                                         ARI 840
```

```
700   TEMP = TEMP + SG(K,I)*SHAPE(K,J)                                  RRI 85C
710   COLM(I) = TEMP                                                    RRI 86C
      DO 800 I=1,NDIM                                                   RRI 87C
800   SHAPE(I,J) = COLM(I)                                              RRI 88C
      RETURN                                                            RRI 89C
      END                                                               RRI 90C
*DECK QSHPB
      SUBROUTINE QSHPB(SS,TT,IW,XXX,YYY,NEL)                            QSH  1C
C**** SHAPE FUNCTION SUBROUTINE FOR 8 POINT ISOPARAMETRIC ELEMENT.      QSH  2C
      DIMENSION SORD(8),TORD(8),ST(2)                                   QSH  3C
      COMMON/SHAPE/ XJAC,SHAPE(4,20),SG(3,3),SA(3,3),X(3,20),LD(120)    QSH  4C
      COMMON /XSTF/ XS,XI,YS,YI,ZS,ZI                                   QSH  5C
      DATA SORD/-1.,0.,1.,1.,1.,0.,-1.,-1./,TORD/-1.,-1.,-1.,0.,1.,1.,1.,0./  QSH  6C
      IF(NEL.GT.4) GO TO 50                                             QSH  7C
C**** FOUR POINT ELEMENT SHAPE FUNCTIONS                                QSH  8C
      J = 1                                                             QSH  9C
      DO 30 I = 1,8,2                                                   QSH 10C
      SI = SORD(I)/2.                                                   QSH 11C
      TI = TORD(I)/2.                                                   QSH 12C
      SP = .5 + SI*SS                                                   QSH 13C
      TP = .5 + TI*TT                                                   QSH 14C
      SHAPE(1,J) = SI*TP                                                QSH 15C
      SHAPE(2,J) = SP*TI                                                QSH 16C
      SHAPE(3,J) = SP*TP                                                QSH 17C
30    J = J + 1                                                         QSH 18C
      GO TO 350                                                         QSH 19C
C**** EIGHT POINT ELEMENT SHAPE FUNCTIONS                               QSH 20C
50    DO 100 I = 1,8,2                                                  QSH 21C
      SI=SORD(I)                                                        QSH 22C
      TI=TORD(I)                                                        QSH 23C
      SP=SI*SS                                                          QSH 24C
      TP=TI*TT                                                          QSH 25C
      SM=1.-SS                                                          QSH 26C
      TM=1.-TT                                                          QSH 27C
      STP=(SI+TI-1.)*0.25                                               QSH 28C
      SHAPE(1,I)=SI*TM*(STP+SP*0.25)                                    QSH 29C
      SHAPE(2,I)=TI*SM*(STP+TP*0.25)                                    QSH 30C
      SHAPE(3,I)=SP*TP*STP                                              QSH 31C
100   CONTINUE                                                          QSH 32C
      DO 200 I=2,8,4                                                    QSH 33C
      TI=TORD(I)                                                        QSH 34C
      TP=1.+TI*TT                                                       QSH 35C
      SSM=1.-SS*SS                                                      QSH 36C
      SHAPE(1,I)=-SSM*TP                                                QSH 37C
      SHAPE(2,I)=0.5*SSM*TI                                             QSH 38C
      SHAPE(3,I)=0.5*SSM*TP                                             QSH 39C
200   CONTINUE                                                          QSH 40C
      DO 300 I=4,8,4                                                    QSH 41C
      SI=SORD(I)                                                        QSH 42C
      SP=1.+SI*SS                                                       QSH 43C
      TT=1.-TT*TT                                                       QSH 44C
      SHAPE(1,I)=0.5*SI*TTM                                             QSH 45C
      SHAPE(2,I)=-TT*SP                                                 QSH 46C
      SHAPE(3,I)=0.5*SP*TTM                                             QSH 47C
300   CONTINUE                                                          QSH 48C
C**** FORM JACOBIAN AND TRANSFORMATION MATRIX FOR PLATE ELEMENT         QSH 49C
350   XXX=0.0                                                           QSH 50C
      YYY=0.                                                            QSH 51C
      XS=0.                                                             QSH 52C
      YS=0.                                                             QSH 53C
      ZS = 0.                                                           QSH 54C
      XT=0.                                                             QSH 55C
      YT=0.                                                             QSH 56C
      ZT = 0.                                                           QSH 57C
      DO 400 I = 1,NEL                                                  QSH 58C
      SP=SHAPE(1,I)                                                     QSH 59C
      TP=SHAPE(2,I)                                                     QSH 60C
      FP=SHAPE(3,I)                                                     QSH 61C
      XI=X(1,I)                                                         QSH 62C
      YI=X(2,I)                                                         QSH 63C
      XXX=XXX+XI*FP                                                     QSH 64C
      YYY=YYY+YI*FP                                                     QSH 65C
```

```
      XS=XS+XI*SP                                                       QSH  66C
      YS=YS+YI*SP                                                       QSH  67C
      XT=XT+XI*TP                                                       QSH  68C
      YT=YT+YI*TP                                                       QSH  69C
      IF(IW.EQ.J) GO TO 400                                             QSH  70C
      ZI = X(3,I)                                                       QSH  71C
      ZS = ZS + ZI*SP                                                   QSH  72C
      ZI = ZT + ZI*TP                                                   QSH  73C
400   CONTINUE                                                          QSH  74C
      IF(IW.EQ.1) RETURN                                                QSH  75C
      XJAC=XS*YT-XT*YS                                                  QSH  76C
      SG(1,1)=YT/XJAC                                                   QSH  77C
      SG(1,2)=-XT/XJAC                                                  QSH  78C
      SG(2,1)=-YS/XJAC                                                  QSH  79C
      SG(2,2)=XS/XJAC                                                   QSH  80C
      DO 52, I = 1,NEL                                                  QSH  81C
      DO 51, J=1,2                                                      QSH  82C
      TEMP=0.                                                           QSH  83C
      DO 500 K=1,2                                                      QSH  84C
500   TEMP = TEMP+SG(K,J)*SHAPE(K,I)                                    QSH  85C
510   ST(J)=TEMP                                                        QSH  86C
      DO 52, J=1,2                                                      QSH  87C
520   SHAPE(J,I) = ST(J)                                                QSH  88C
      RETURN                                                            QSH  89C
      END                                                               QSH  90C
*DECK ELMT21
      SUBROUTINE ELMT21(N,MA,NDIM,NDF,NEL,NELI,NSTF,NSIZV,NVEC,MCT,DM,D, ELM   1C
     X  XYZ,IX,H,FORCE,ESTIF,U,VECT,ISW)                                ELM   2C
      REAL LABL                                                         ELM   3C
      LOGICAL NPR,NPL,NOPRNT                                            ELM   4C
      DIMENSION D(63,1),ESTIF(NSTF,NSTF),FORCE(NSTF,2),STOW(4,27),XX(3),
     X  IX(NEL+1),U(NDF,1),DEPS(4,3),XIG(4,3),TAB1(11),TAB2(11)         ELM   6C
     X  ,SHED(3,3),H(1),IG(6),JG(6)                                     ELM   7C
      DIMENSION VECT(NSIZV,NVEC)
      COMMON/DYNAMD/TIME,NSIG(7),NT,NSTEP,DT,NUMPLT,NEDATA(20,3),NPR,NPL ELM   8C
      COMMON/LABELS/ LABL(6),SHED(3),XH,FHED(6),FH,UHED(6),UH,RHED(6),RH ELM   9C
     X  ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)     ELM  10C
      COMMON/SHAPE/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)    ELM  11C
      COMMON/TAPES/ ITPS,ITP6                                           ELM  12C
      COMMON/VISUAL/DTP,NH,ISZH,DUL(6,20), UL(6,20),UDL(6,20),C6        ELM  13C
      COMMON/VTAPE/ ITP13,ITP14,ITRD,ITWR                               ELM  14C
      COMMON/PLOTS/IPLOT
      DATA TAB1(1),TAB2(1)/8H(1H0,17 ,8H  ( 8X  /,8WORD/1H)/           ELM  15C
      DATA SHED/5H   XX=,6H   XY=,6H   XZ=,6H   XY=,6H   YY=,6H   YZ=,  ELM  16C
     X   6H   XZ=,6H   YZ=,6H   ZZ=/                                    ELM  17C
      DATA FWORD,EWORD,AWORD/8H,0PF12.3,8H,1PE12.3,8H  ,12X  /          ELM  18C
      DATA SH,FH,BLANK/6HSTRESS,6HSTRAIN,6H       /,YWORD/8H,A5,17  /   ELM  19C
      DATA IG/1,1,2,1,2,3/,JG/1,2,2,3,3,3/
      IF(ISW.GT.6) GO TO 4000                                           ELM  21C
      NSIDE = (NEL+1)/NDIM**2                                           ELM  22C
      GO TO (1,2,3,4,5,4), ISW                                          ELM  23C
C**** CHARACTERIZE THE MATERIAL FOR ISOTROPIC LINEAR VISCOELASTICITY     ELM  24C
1     READ(ITPS,1000) NO,NK,D(4,MA),D(5,MA),D(3,MA)                     ELM  25C
      WRITE(ITP6,2000) NO,NK,D(4,MA),D(5,MA),D(3,MA)                    ELM  26C
      LIM = NSIDE+1
      CALL INTEGL(LIM,0,NDIM,LINT,STOW)
      CA = 1.
      CT = 1.
      DTP = 1.0
      NH = 1
C**** SET UP HEADS FOR PRINTOUTS                                        ELM  32C
      I = 2                                                             ELM  33C
      DO 60, J=1,NDIM                                                   ELM  34C
      TAB1(I) = FWORD                                                   ELM  35C
      TAB2(I) = XWORD                                                   ELM  36C
600   I = I + 1                                                         ELM  37C
      DO 61, J = 1,6                                                    ELM  38C
      TAB1(I) = EWORD                                                   ELM  39C
      TAB2(I) = EWORD                                                   ELM  40C
610   I = I + 1                                                         ELM  41C
      TAB1(I) = SWORD                                                   ELM  42C
      TAB2(I) = SWORD                                                   ELM  43C
```

```
      IAR2(2) = IWORD                                                   ELM 44C
      D(1,MA) = NG + .01                                                ELM 45C
      D(2,MA) = NK + .01                                                ELM 46C
      NU = 5 + 2*NG                                                     ELM 47C
      IF(NG.EQ.2) GO TO 100                                             ELM 48C
      READ(IRS,1001) (D(I,MA),I=5,NU)                                   ELM 49C
      WRITE(IPS,2001)(D(I,MA),I=5,NU)                                   ELM 50C
100   IF(NK.EQ.3) GO TO 200                                             ELM 51C
      NL = NU + 1                                                       ELM 52C
      NU = NU + 2*NK                                                    ELM 53C
      READ(IRS,1001)  (D(I,MA),I=NL,NU)                                 ELM 54C
      WRITE(IPS,2002) (D(I,MA),I=NL,NU)                                 ELM 55C
200   IF(NU.LT.64) RETURN                                               ELM 56C
      WRITE(IPS,2030) MA,NG,NK,NU                                       ELM 57C
      IPS = 0                                                           ELM 58C
      RETURN                                                            ELM 59C
C**** CHECK MESH FOR NEGATIVE JACOBIANS AT ELEMENT CORNERS              ELM 60C
2     CALL CHKJAC(NDIM,NEL,IM)                                          ELM 61C
      NG = D(1,MA)                                                      ELM 62C
      NK = D(2,MA)                                                      ELM 63C
      MM = MM + (LINT+1)*(NK+NG*((NDIM+1)*NDIM)/2)                      ELM 64C
      RETURN                                                            ELM 65C
C**** COMPUTE THE VISCOELASTIC STIFFNESS MATRICES FOR SHEAR AND BULK    ELM 66C
C**** SET UP THE MATERIAL PROPERTIES                                    ELM 67C
3     NL = 4                                                            ELM 68C
      NG = D(1,MA)                                                      ELM 69C
      GG = D(4,MA)                                                      ELM 70C
      IF(NG.LE.0) GO TO 150                                             ELM 71C
      NL = 2*NG + NL                                                    ELM 72C
      DO 14  I = 6,NL,2                                                 ELM 73C
140   GG = GG + D(I,MA)*EXP(DT/D(I+1,MA))                               ELM 74C
150   NK = D(2,MA)                                                      ELM 75C
      XK = D(5,MA)                                                      ELM 76C
      IF(NK.LE.0) GO TO 170                                             ELM 77C
      IL = NL + 2                                                       ELM 78C
      NL = NL + 2*NK                                                    ELM 79C
      DO 16  I = IL,NL,2                                                ELM 80C
160   XK = XK + D(I,MA)*EXP(DT/D(I+1,MA))                               ELM 81C
C**** COMPUTE THE INTEGRALS FOR THIS ELEMENT                            ELM 82C
170   CONTINUE                                                          ELM 83C
      CALL VSTIF(NDIM,NDF,NEL,NSIDE,ESTIF(NDF+1,1),FORCE(1,2),STOR,LINT) ELM 84C
      SS=D(3,MA)*CA                                                     ELM 85C
      CALL ASSEMB(NDIM,NDF,NEL,NSIF,ESTIF(NDF+1,1),ESTIF,FORCE(1,2),GG, ELM 86C
     C XK,SS)                                                           ELM 87C
      RETURN                                                            ELM 88C
C**** STRESS COMPUTATION FOR THE VISCOELASTIC ELEMENT                   ELM 89C
C**** VISCOELASTIC LOAD HISTORY COMPUTATION AND ELEMENT ASSEMBLY FROM TP ELM 90C
4     CONTINUE                                                          ELM 91C
C**** COMPUTATION AND OUTPUT OF STRESS AND HISTORY OF VISCOELASTIC SOLN. ELM 92C
C**** DO THE INTEGRATION AND OUTPUT THE STRESS                          ELM 93C
      NG = D(1,MA)                                                      ELM 94C
      NK = D(2,MA)                                                      ELM 95C
      GG = D(4,MA)                                                      ELM 96C
      XK = D(5,MA)                                                      ELM 97C
C**** CHECK FOR M STORAGE                                               ELM 98C
      LL = MM + (LINT+1)*(NK+NG*((NDIM+1)*NDIM)/2)                      ELM 99C
      IF(LL.GT.ISZM) CALL VBUFF(IIRD,ITAB,M,NM,ISZM)                    ELM100C
      NL = LINT + 1                                                     ELM101C
      DO 50  LL = 1,NL                                                  ELM102C
      NOPRNT = NPR                                                      ELM103C
      IF(LL.NE.1) NOPRNT = .TRUE.                                       ELM104C
      NPL = .FALSE.                                                     ELM105C
      IF(NUMPLT.LE.0) GO TO 210                                         ELM106C
      DO 205 NP = 1,NUMPLT                                              ELM107C
205   IF(NEDATA(NP,1).EQ.N.AND.NEDATA(NP,2).EQ.LL) NPL = .TRUE.         ELM108C
210   CONTINUE                                                          ELM109C
      IF(LL.NE.1) GO TO 220                                             ELM110C
      SS = 0.0                                                          ELM111C
      TT = 0.0                                                          ELM112C
      UC = 0.0                                                          ELM113C
      IF(NDIM.EQ.2) UC = -1.0                                           ELM114C
      GO TO 230                                                         ELM115C
```

```
220   SS = STUR(1,LL-1)                                                    FLM116C
      TT = STUR(2,LL-1)                                                    FLM117C
      UU = STUR(3,LL-1)                                                    FLM118C
      RR = STUR(4,LL-1)                                                    FLM119C
230   CALL BRICK2(SS,TT,UU,NDIM,NEL,NSIDE)                                 FLM120C
C**** COMPUTE THE PRESENT STRAIN AND STRAIN INCREMENT                      FLM121C
      DTH = 0.                                                             FLM122C
      THP = 0.                                                             FLM123C
      DO 33. I = 1,NDIM                                                    FLM124C
      CC = 0.0                                                             FLM125C
      DO 300 K = 1,NEL                                                     FLM126C
300   CC = CC + SHAPE(4,K)*X(I,K)                                          FLM127C
      XX(I) = CC                                                           FLM128C
      DO 32. J = 1,NDIM                                                    FLM129C
      CC = 0.                                                              FLM130C
      DD = 0.                                                              FLM131C
      DO 31. K = 1,NEL                                                     FLM132C
      CC = CC + SHAPE(I,K)*DUL(J,K) + SHAPE(J,K)*DUL(I,K)                  FLM133C
310   DD = DD + SHAPE(I,K)*DUL(J,K) + SHAPE(J,K)*DUL(I,K)                  FLM134C
      DEPS(I,J) = DD/XJAC                                                  FLM135C
      DEPS(J+1,I) = CC/XJAC                                                FLM136C
      XIG(I,J) = GG*DEPS(J+1,I)                                            FLM137C
320   XIG(J+1,I) = XIG(I,J)                                                FLM138C
      DTH = DTH + DEPS(I,I)                                                FLM139C
330   THP = THP + DEPS(I+1,I)                                              FLM140C
      DO 34. I = 1,NDIM                                                    FLM141C
      DEPS(I,I) = DEPS(I,I) - DTH/3.0                                      FLM142C
340   DEPS(I+1,I) = DEPS(I+1,I)/2.0                                        FLM143C
      XIK = THP*(XK/2.-GG/3.)                                              FLM144C
      YIK = XIK                                                            FLM145C
      DTH = DTH/2.                                                         FLM146C
C**** UPDATE THE SHEAR HISTORY                                             FLM147C
      IG = 0                                                               FLM148C
      IF(NG.LE.0) GO TO 410                                                FLM149C
      DO 40. I = 1,NG                                                      FLM150C
      IG = IG + 2                                                          FLM151C
      DD = U(IG+1,MA)                                                      FLM152C
      CC = U(IG,MA)*HIST(DTP/DD)                                           FLM153C
      DD = EXP(-DT/DD)                                                     FLM154C
      DO 402 J = 1,NDIM                                                    FLM155C
      DO 402 K = J,NDIM                                                    FLM156C
      EE = H(NH) + CC*DEPS(J,K)                                            FLM157C
      XIG(J,K) = XIG(J,K) + EE                                             FLM158C
      EE = EE*DD                                                           FLM159C
      XIG(K+1,J) = XIG(K+1,J) + EE                                         FLM160C
      H(NH) = EE                                                           FLM161C
402   NH = NH + 1                                                          FLM162C
400   CONTINUE                                                             FLM163C
C**** UPDATE THE BULK HISTORY                                              FLM164C
410   IF(NA.LE.0) GO TO 439                                                FLM165C
      DO 42. I = 1,NA                                                      FLM166C
      IG = IG + 2                                                          FLM167C
      DD = U(IG+1,MA)                                                      FLM168C
      EE = H(NH) + U(IG,MA)*HIST(DTP/DD)*DTH                               FLM169C
      XIK = XIK + EE                                                       FLM170C
      EE = EE*EXP(-DT/DD)                                                  FLM171C
      YIK = YIK + EE                                                       FLM172C
      H(NH) = EE                                                           FLM173C
420   NH = NH + 1                                                          FLM174C
430   CONTINUE                                                             FLM175C
      DO 43. I = 1,NDIM                                                    FLM176C
      XIG(I,I) = XIG(I,I) + XIK                                            FLM177C
435   XIG(I+1,I) = XIG(I+1,I) + YIK                                        FLM178C
C**** OUTPUT THE STRESSES FOR THE CURRENT TIME STEP                        FLM179C
      IF(NOPRNT) GO TO 840                                                 FLM180C
      NCT = NCT - 1                                                        FLM181C
      IF(NCT.GT.0) GO TO 839                                               FLM182C
      WRITE(IIPR,2004) HEAD,TIME,IPG,(XMED(I),XM,I=1,NDIM),                FLM183C
     X ((SMED(I,J),SM,I=J,NDF),J=1,NDF)                                    FLM184C
      WRITE(IIPR,2003) (BLANK,BLANK,I=1,NDIM),                             FLM185C
     X ((SMED(I,J),EM,I=J,NDF),J=1,NDF)                                    FLM186C
      IPG = IPG + 1                                                        FLM187C
```

```
      MCT = 16                                                          FLM189C
630   WRITE(IOPG,IAR1) NP,((XX(I),I=1,NDIM),((XIG(I,J),J=1,NDIM),I=1,NDIM) FLM190C
      WRITE(IOPG,IAR2) LM,MA,((DEPS(J,I,I),J=1,NDIM),I=1,NDIM)          FLM191C
640   IF(NPL) CALL PLDATA(NUMEL,NDIM,NPLL,SHED,XX,XIG,FORCE)            FLM192C
      IF(LL.NE.1 .OR. IPLOT.EQ. 0.0) GO TO 650                          FLM193C
      IF(IPLOT.EQ.3)  GO TO 645                                         FLM194C
      VECT(N,1)= XIG(1,1)                                               FLM195C
      VECT(N,2)= XIG(2,2)                                               FLM196C
      VECT(N,3)= 0.0                                                    FLM197C
      VECT(N,4)= XIG(1,2)                                               FLM198C
      VECT(N,5)= DEPS(2,1)                                              FLM199C
      VECT(N,6)= DEPS(3,2)                                              FLM200C
      VECT(N,7)= 0.0                                                    FLM201C
      VECT(N,8)= DEPS(3,1)                                              FLM202C
      GO TO 650                                                         FLM203C
645   KK = 0                                                            FLM204C
      DO 646 I=1,NDIM                                                   FLM205C
      DO 646 J=1,NDIM                                                   FLM206C
      KK = KK + 1                                                       FLM207C
      VECT(KK,1) = XIG(I,J)                                             FLM208C
646   VECT(KK,2) = DEPS(J,I,I)                                          FLM209C
C                                                                       FLM210C
C     CALCULATE PRINCIPLE STRAINS                                       FLM211C
C                                                                       FLM212C
650   IF(NOPRNT) GO TO 550                                              FLM213C
      IF (MA.NE.1) GO TO 550                                            FLM214C
      IF(NDIM.NE.3) GO TO 550                                           FLM215C
      CALL PRINC(DEPS)                                                  FLM216C
      WRITE(6,3000) DEPS(1,1),DEPS(1,2),DEPS(1,3)                       FLM217C
      NCT = NCT-2                                                       FLM218C
 3000 FORMAT(5X,22H $$$PRINCIPAL STRAINS= 3E15.5 /)                     FLM219C
550   IF(ISW.EQ.4 .OR. LL.EQ.1) GO TO 500                               FLM220C
C**** COMPUTE THE LOADS FOR THE NEXT TIME STEP                          FLM221C
      DO 44 I = 1,NDIM                                                  FLM222C
      DO 44 J = 1,NDIM                                                  FLM223C
      XIG(I,J) = XIG(J,I)*WW                                            FLM224C
440   XIG(J,I) = XIG(I,J)                                               FLM225C
      II = 0                                                            FLM226C
      WW=XJAC*WW*U(3,MA)                                                FLM227C
      DO 47 I = 1,NEL                                                   FLM228C
      EE=SHAPE(4,I)*WW                                                  FLM229C
      DO 46 J = 1,NDF                                                   FLM230C
      CC = 0.0                                                          FLM231C
      DO 45 K = 1,NDIM                                                  FLM232C
450   CC = CC + SHAPE(K,I)*XIG(J,K)                                     FLM233C
460   FORCE(II+J,1)=FORCE(II+J,1)-CC-EE*UDL(J,I)                        FLM234C
470   II = II + NDF                                                     FLM235C
500   CONTINUE                                                          FLM236C
      RETURN                                                            FLM237C
C**** ELEMENT LOAD COMPUTATION FOR BODY TYPE FORCES, ETC.               FLM238C
5     CONTINUE                                                          FLM239C
      RETURN                                                            FLM240C
4000  IPG = 0                                                           FLM241C
      RETURN                                                            FLM242C
C**** FORMAT STATEMENTS                                                 FLM243C
 1000 FORMAT(2I5,3E10.0)                                                FLM244C
1001  FORMAT(8E10.0)                                                    FLM245C
2000  FORMAT(23H  VISCOELASTIC MATERIAL//I10,12H SHEAR TERMS/           FLM246C
     X I10,12H BULK  TERMS//10X,12H G-INFINITY=,1PE12.4/10X,            FLM247C
     X 12H A-INFINITY=,1PE12.4/10X,12H DENSITY   =, 1PE12.4/1X)         FLM248C
2001  FORMAT(10X,14H SHEAR MODULUS,6X,4HTIME/(10X,1P2E12.4))           FLM249C
2002  FORMAT(10X,14H BULK  MODULUS,6X,4HTIME/(10X,1P2E12.4))           FLM250C
2003  FORMAT(6X,9(2A6))                                                 FLM251C
2004  FORMAT(1H1,12A6,E13.5,20X,4HPAGE,I3//5X,16HELEMENT STRESSES//     FLM252C
     1 1X,7HELEMENT,9(2A6))                                             FLM253C
2030  FORMAT(6H MATERIAL,I3,17H EXCEEDS STORAGE.,I5,13H SHEAR TERMS.,I5, FLM254C
     X 16H BULK TERMS NEED,I4,7H WORDS. )                               FLM255C
      END                                                               FLM256C
      SUBROUTINE PRINC(STRAIN)                                          FLM257C
C                                                                       FLM258C
C     CALCULATE PRINCIPLE STRAIN                                        FLM259C
C                                                                       FLM260C
```

```
      DIMENSION STRAIN(4,3)
C
      S11 = STRAIN(2,1)
      S12 = STRAIN(3,1)/2.
      S13 = STRAIN(4,1)/2.
      S22 = STRAIN(3,2)
      S23 = STRAIN(4,2)/2.
      S33 = STRAIN(4,3)
C
      B1 = -(S11+S22+S33)
      B2 = S11*S22+S11*S33+S22*S33 -(S13*S13+S12*S12+S23*S23)
      B3 = S13*S13*S22+S12*S12*S33+S23*S23*S11
     1    -(S11*S22*S33 +2.*S12*S23*S13)
C
      IF(B1.EQ.0.0.AND.B2.EQ.0.0.AND.B3.EQ.0.0) GO TO 10
      CALL CUBIC(B1,B2,B3,ITELL)
   10 STRAIN(1,1) = B1
      STRAIN(1,2) = B2
      STRAIN(1,3) = B3
      RETURN
      END
      SUBROUTINE CUBIC(B1,B2,B3,ITELL)
      DATA ZC,ZU/-0.5,0.866025404/
C
C     THIS ROUTINE DETERMINES THE THREE ROOTS OF A RATIONAL CUBIC
C     EQUATION OF FORM,
C
C              X**3 + B1*X**2 + B2*X + B3 = 0
C
C     THE USER SUPPLYS THE COEFFICIENTS (B1,B2,B3) THROUGH THE
C     SUBROUTINE CALL LIST. THE SUBROUTINE RETURNS THE ROOTS THROUGH
C     THE SAME VARIABLE LIST, ALONG WITH THE INTEGER, ITELL, WHICH TELLS
C     THE FORM OF THE ROOTS.  THERE ARE ONLY TWO POSSIBLE FORMS FOR
C     ROOTS OF A CUBIC EQUATION...
C
C              FORM-1, (ITELL=1) THE ROOTS ARE ALL REAL NUMBERS.
C
C              FORM-2, (ITELL=2) THE FIRST ROOT,B1, IS REAL AND THE
C                                OTHER TWO ROOTS ARE COMPLEX CONJUGATE
C                                PAIRS, WHERE B2 IS THE REAL PART AND B3
C                                IS THE IMAGINARY PART.
C
C
      T3 = 1./3.
      Q = 2.*(B1**T3)**3 - B1*B2*T3 + B3
      Q2 = Q*ZC
      R = B2 - B1*B1*T3
      R = Q2*Q2 + (R*T3)**3
      R2 = SQRT(ABS(R))
      A = Q2 + R2
      B = Q2 - R2
      TERM = -B1*T3
C
      IF(R.GT.0.0) GO TO 50
C
C     PATH FOR REAL ROOT DETERMINATION  (FORM=1)
C
      ITELL = 1
      THETA = ATAN2(R2,Q2)
      RC = (SQRT(Q2*Q2+R2*R2))**T3
      ZA = RC*COS(THETA*T3)
      ZB = RC*SIN(THETA*T3)
      B1 = ZA*ZA + TERM
      B2 = ZA*(ZA*ZC-ZB*ZU) + TERM
      B3 = ZA*(ZA*ZC + ZB*ZU) + TERM
C
      RETURN
C
C     PATH FOR ONE REAL ROOT AND TWO COMPLEX, (FORM=2)
C
   50 ITELL = 2
      AT3 = (ABS(A))**T3
```

```
      RT3 = (GAMS(G))**T3
      AT3 = SIGN(AT3,A)
      RT3 = SIGN(RT3,R)
      B1 = RT3 * RT3 * TERM
      B2 = .5*(RT3+RT3) + TERM
      B3 = .5*(RT3-RT3)
C
      RETURN
      END
*DECK VISCO
      SUBROUTINE VSTIF(NDIM,NDF,NEL,NSIDE,ST,FORCE,STOR,LINT)                    VST   1C
      DIMENSION STOR(4,27),ST(1),FORCE(NDF,1)                                    VST   2C
      COMMON/SHAPE/XJAC,SHAPE(4,20),XJUNK(18),X(3,20),LD(120)                    VST   3C
C**** INITIALIZE THE ARRAYS FOR STIFNESS AND INTEGRALS                           VST   4C
      DO 200 II = 1,LINT                                                         VST   5C
      SS = STOR(1,II)                                                            VST   6C
      TT = STOR(2,II)                                                            VST   7C
      UU = STOR(3,II)                                                            VST   8C
      WU = STOR(4,II)                                                            VST   9C
      CALL RATIO2(SS,TT,UU,NDIM,NEL,NSIDE)                                       VST  10C
      DVOL = WU*XJAC                                                             VST  11C
C**** FORM THE INDEPENDENT INTEGRALS                                             VST  12C
      NS =                                                                       VST  13C
      DO 100 J = 1,NEL                                                           VST  14C
      TEMP=SHAPE(4,J)*XJAC*WU                                                    VST  15C
      DO 100 N = 1,NDIM                                                          VST  16C
      FORCE(N,J)=TEMP*FORCE(N,J)                                                 VST  17C
      CC = SHAPE(N,J)*DVOL                                                       VST  18C
      DO 100 I = 1,J                                                             VST  19C
      L = NDIM                                                                   VST  20C
      IF(I.EQ.J) L = N                                                           VST  21C
      DO 100 M = 1,L                                                             VST  22C
      NS = NS + 1                                                                VST  23C
100   ST(NS) = ST(NS) + CC*SHAPE(M,I)                                            VST  24C
200   CONTINUE                                                                   VST  25C
      RETURN                                                                     VST  26C
      END                                                                        VST  27C
      SUBROUTINE ASSEMB(NDIM,NDF,NEL,NSTF,ST,ESTIF,FORCE,GG,XK,RO)               ASS   1C
      DIMENSION ESTIF(NSTF,NSTF),ST(1),FORCE(1)                                  ASS   2C
C**** ASSEMBLY OF ELEMENT INTO THE STIFNESS                                      ASS   3C
      NU = NEL*NDIM                                                              ASS   4C
      NS = (NU*(NU+1))/2                                                         ASS   5C
      J1 = NDF*(NEL-1)                                                           ASS   6C
      ND1 = NDIM + 1                                                             ASS   7C
      XX = XK - 2.*GG/3.                                                         ASS   8C
      DO 23 J = 1,NEL                                                            ASS   9C
      DO 22 N = 1,NDIM                                                           ASS  10C
      NI = ND1 - N                                                               ASS  11C
      I1 = J1                                                                    ASS  12C
      DO 21 I = J,NEL                                                            ASS  13C
      L = 1                                                                      ASS  14C
      IF(I.EQ.J) L = N                                                           ASS  15C
      DO 20 M = L,NDIM                                                           ASS  16C
      MI = ND1 - M                                                               ASS  17C
      CC = ST(NS)                                                                ASS  18C
      ST(NS) = 0.0                                                               ASS  19C
      ESTIF(I1+MI,J1+NI) = ESTIF(I1+MI,J1+NI) + XX*CC                            ASS  20C
      CC = GG*CC                                                                 ASS  21C
      ESTIF(I1+NI,J1+MI) = ESTIF(I1+NI,J1+MI) + CC                               ASS  22C
      IF(MI.NE.NI) GO TO 200                                                     ASS  23C
      DO 10 K = 1,NDIM                                                           ASS  24C
100   ESTIF(I1+K,J1+K) = ESTIF(I1+K,J1+K) + CC                                   ASS  25C
200   NS = NS - 1                                                                ASS  26C
210   I1 = I1 - NDF                                                              ASS  27C
220   CONTINUE                                                                   ASS  28C
230   J1 = J1 - NDF                                                              ASS  29C
      NU = NEL*NDF                                                               ASS  30C
      DO 300 J = 2,NU                                                            ASS  31C
      K = J - 1                                                                  ASS  32C
      DO 300 I = 1,K                                                             ASS  33C
      ESTIF(I,J) = ESTIF(I,J) + ESTIF(J,I)                                       ASS  34C
300   ESTIF(J,I) = ESTIF(I,J)                                                    ASS  35C
```

```
      DO 400 I=1,ND                                                     ASS 36C
  400 ESTIF(I,I)= ESTIF(I,I)+RU*FORCE(I)                                ASS 37C
      RETURN                                                            ASS 38C
      END                                                               ASS 39C
      FUNCTION HIST(C)                                                  ISTC  1C
C**** HISTORY TERM FOR THE INTEGRATION OF VISCOELASTIC EQUATIONS         ISTC  2C
C**** HIST = (1. - EXP(-C))/C                                           ISTC  3C
      IF(C.GT.0.01) GO TO 200                                           ISTC  4C
      HIST = 1.                                                         ISTC  5C
      DO 100 I = 1,3                                                    ISTC  6C
      J = 5 - I                                                         ISTC  7C
  100 HIST = 1.-C/FLOAT(J)*HIST                                         ISTC  8C
      RETURN                                                            ISTC  9C
  200 HIST = (1. - EXP(-C))/C                                           ISTC 10C
      RETURN                                                            ISTC 11C
      END                                                               ISTC 12C
      SUBROUTINE VBUFF(ITRU,ITWR,H,IH,ISZH)                             VBU   1C
C**** BUFFER THE HISTORY VECTORS AND RESET THE POINTER                  VBU   2C
      DIMENSION H(ISZH)                                                 VBU   3C
      WRITE(ITWR) H                                                     VBU   4C
      READ (ITRU) H                                                     VBU   5C
      IH = 1                                                            VBU   6C
      RETURN                                                            VBU   7C
      END                                                               VBU   8C
*DECK ELM122
      SUBROUTINE ELM122(N,MA,NDIM,NDF,NEL,NELI,NSTF,NSIZV,NVEC,MCT,DM,D, ELM   1C
     X  XYZ,IX,H,FORCE,ESTIF,U,VECT,ISH)                                ELM   2C
      REAL LABL                                                         ELM   3C
      LOGICAL NPR,NPL,NOPRNT                                            ELM   4C
      DIMENSION D(63,1),ESTIF(NSTF,NSTF),FORCE(NSTF,2),STOW(4,8),XX(3), ELM   5C
     X  IX(NEL+1),U(NDF,1),DEPS(4,3),XIG(4,3),TAB1(11),TAB2(11)         ELM   6C
     X  ,SHL(3,3),H(1),IG(6),JG(6)                                      ELM   7C
      DIMENSION VECT(NSIZV,NVEC)                                        ELM   8C
      COMMON/DYNAMO/TIME,NSIG(7),NI,NSTEP,DT,NUMPLT,NEDATA(20,3),NPR,NPL ELM   9C
      COMMON/LABELS/ LABL(8),XHED(3),YHED(6),FHED(6),UHED(6),VHED(6),RHED(6),RH ELM  9C
     X  ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)    ELM  10C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SH(3,3),X(3,20),LD(120)     ELM  11C
      COMMON/TAPES/ ITPS,ITP6                                           ELM  12C
      COMMON/VISDAT/DTP,RH,ISZH,DUL(6,20), UL(6,20),UUL(6,20),C8        ELM  13C
      COMMON/VTAPE/ ITP13,ITP14,ITRU,ITWR                               ELM  14C
      COMMON/FLUTS/IFLUI                                                ELM  15C
      DATA TAB1(1),TAB2(1)/8H(1H0,I7 ,8H  ( 8X  /,5WORD/1H)/           ELM  16C
      DATA SHEC/8H   XX-,8H   XY-,8H   XZ-,8H   YY-,8H   YZ-,           ELM  17C
     X  8H   ZZ-,8H   YZ-,8H   ZZ-/                                     ELM  18C
      DATA FWORD,EWORD,XWORD/8H,8HF12.3,8H,1PE12.3,8H  ,12A  /          ELM  19C
      DATA SH,FH,BLANK/6HSTRESS,6HSTRAIN,6H       /,YWORD/8H,A5,I7  /   ELM  20C
      DATA IG/1,1,2,1,2,3/,JG/1,2,2,3,3,3/                              ELM  21C
C
C     GENERAL 2D,3D, FLUID ELEMENT (ONE POINT QUADRATURE)
C
      IF(ISW.GT.4) GO TO 4000                                           ELM  22C
      NSIDE = (NEL-1)/NDIM**2                                           ELM  23C
      GO TO (1,2,3,4,5,4), ISW                                          ELM  24C
C**** CHARACTERIZE THE MATERIAL FOR ISOTROPIC LINEAR VISCOELASTICITY     ELM  25C
    1 READ(ITPS,1000) NG,NK,D(4,MA),D(5,MA),D(3,MA)                     ELM  26C
      WRITE(ITP6,2000) NG,NK,D(4,MA),D(5,MA),D(3,MA)                    ELM  27C
      CALL INTEGL(1,0,NDIM,LINT,STOW)                                   ELM  28C
      CA = .0                                                           ELM  29C
      DT = .                                                            ELM  30C
      DTP = .0                                                          ELM  31C
      NH = 1                                                            ELM  32C
C**** SET UP HEADS FOR PRINTOUTS                                        ELM  33C
      I = 0                                                             ELM  34C
      DO 801 J=1,NDIM                                                   ELM  35C
      TAB1(I) = FWORD                                                   ELM  36C
      TAB2(I) = XWORD                                                   ELM  37C
  800 I = I + 1                                                         ELM  38C
      DO 81  J = 1,6                                                    ELM  39C
      TAB1(I) = EWORD                                                   ELM  40C
      TAB2(I) = EWORD                                                   ELM  41C
  810 I = I + 1                                                         ELM  42C
      TAB1(I) = SWORD                                                   ELM  43C
```

```
      1AM2(1) = SWORD                                                   ELM 43C
      1AM2(2) = YWORD                                                   ELM 44C
      D(1,MA) = NG + .01                                                ELM 45C
      D(2,MA) = NK + .01                                                ELM 46C
      NL = 3 + 2*NG                                                     ELM 47C
      IF(NU.EQ.0) GO TO 100                                             ELM 48C
      READ(IIPS,1001) (D(I,MA),I=6,NU)                                  ELM 49C
      WRITE(IIPS,2001)(D(I,MA),I=6,NU)                                  ELM 50C
100   IF(NK.EQ.0) GO TO 200                                             ELM 51C
      NL = NU + 1                                                       ELM 52C
      NU = NU + 2*NK                                                    ELM 53C
      READ(IIPS,1001)  (D(I,MA),I=NL,NU)                                ELM 54C
      WRITE(IIPS,2002) (D(I,MA),I=NL,NU)                                ELM 55C
200   IF(NU.LT.64) RETURN                                               ELM 56C
      WRITE(IIPS,2030) MA,NG,NK,NU                                      ELM 57C
      IRG = 0                                                           ELM 58C
      RETURN                                                            ELM 59C
C.... CHECK MESH FOR NEGATIVE JACOBIANS AT ELEMENT CORNERS              ELM 60C
2     CALL CKJBRK(NDIM,NEL,DM)                                          ELM 61C
      NG = D(1,MA)                                                      ELM 62C
      NK = D(2,MA)                                                      ELM 63C
      DM = DM + (LINT+1)*(NK+NG*((NDIM+1)*NDIM)/2)                      ELM 64C
      RETURN                                                            ELM 65C
C.... COMPUTE THE VISCOELASTIC STIFFNESS MATRICES FOR SHEAR AND BULK    ELM 66C
C.... SET UP THE MATERIAL PROPERTIES                                    ELM 67C
3     NL = 4                                                            ELM 68C
      GG = D(1,MA)                                                      ELM 69C
      GG = D(4,MA)                                                      ELM 70C
      IF(NG.LE.0) GO TO 150                                             ELM 71C
      NL = 2*NG + NL                                                    ELM 72C
      DO 14  I = 6,NL,2                                                 ELM 73C
140   GG = GG + D(I,MA)*EXP(DT/D(I+1,MA))                               ELM 74C
150   KK = D(2,MA)                                                      ELM 75C
      XK = D(5,MA)                                                      ELM 76C
      IF(NK.LE.0) GO TO 170                                             ELM 77C
      IL = NL + 2                                                       ELM 78C
      NL = NL + 2*NK                                                    ELM 79C
      DO 16  I = IL,NL,2                                                ELM 80C
160   XK = XK + D(I,MA)*EXP(DT/D(I+1,MA))                               ELM 81C
C.... COMPUTE THE INTEGRALS FOR THIS ELEMENT                            ELM 82C
170   CONTINUE                                                          ELM 83C
      CALL VSTIF(NDIM,NDF,NEL,NSIDE,ESTIF(NDF+1,1),FORCE(1,2),STOW,LINT) ELM 84C
      SS=D(3,MA)*CA                                                     ELM 85C
      CALL ASSEMB(NDIM,NDF,NEL,NSTF,ESTIF(NDF+1,1),ESTIF,FORCE(1,2),GG, ELM 86C
     C XK,SS)                                                           ELM 87C
      RETURN                                                            ELM 88C
C.... STRESS COMPUTATION FOR THE VISCOELASTIC ELEMENT                   ELM 89C
C.... VISCOELASTIC LOAD HISTORY COMPUTATION AND ELEMENT ASSEMBLY FROM TP ELM 90C
4     CONTINUE                                                          ELM 91C
C.... COMPUTATION AND OUTPUT OF STRESS AND HISTORY OF VISCOELASTIC SOLN. ELM 92C
C.... DO THE INTEGRATION AND OUTPUT THE STRESS                          ELM 93C
      NG = D(1,MA)                                                      ELM 94C
      KK = D(2,MA)                                                      ELM 95C
      GG = D(4,MA)                                                      ELM 96C
      XK = D(5,MA)                                                      ELM 97C
C.... CHECK FOR H STORAGE                                               ELM 98C
      LL = DM + (LINT+1)*(NK+NG*((NDIM+1)*NDIM)/2)                      ELM 99C
      IF(LL.GT.ISZM) CALL VBUFF(ITRD,IIAR,M,NM,ISZM)                    ELM100C
      NL = LINT                                                         ELM101C
      DO 50  LL = 1,NL                                                  ELM102C
      NOPRNT = NPR                                                      ELM103C
      IF(LL.NE.1) NOPRNT = .TRUE.
      NPL = .FALSE.                                                     ELM105C
      IF(NUMPLT.LE.0) GO TO 210                                         ELM106C
      DO 205 NP = 1,NUMPLT                                              ELM107C
205   IF(NEDATA(NP,1).EQ.N.AND.NEDATA(NP,2).EQ.LL) NPL = .TRUE.         ELM108C
210   CONTINUE                                                          ELM109C
      IF(LL.NE.1) GO TO 220                                             ELM110C
      SS = 0.0                                                          ELM111C
      TT = 0.0                                                          ELM112C
      UU = 0.0                                                          ELM113C
      WW = STOW(4,1)
```

```
      IF(NDIM.EQ.2) UU = -1.0                                          ELM114C
      GO TO 230                                                        ELM115C
  220 SS = STOR(1,LL-1)                                                ELM116C
      TT = STOR(2,LL-1)                                                ELM117C
      UU = STOR(3,LL-1)                                                ELM118C
      WW = STOR(4,LL-1)                                                ELM119C
  230 CALL BRICK2(SS,TT,UU,NDIM,NEL,NSIDE)                             ELM120C
C**** COMPUTE THE PRESENT STRAIN AND STRAIN INCREMENT                  ELM121C
      DTH = 0.                                                         ELM122C
      THP = 0.                                                         ELM123C
      DO 33 I = 1,NDIM                                                 ELM124C
      CC = 0.0                                                         ELM125C
      DO 30 K = 1,NEL                                                  ELM126C
  300 CC = CC + SHAPE(4,K)*X(I,K)                                      ELM127C
      XX(I) = CC                                                       ELM128C
      DO 32 J = 1,NDIM                                                 ELM129C
      CC = 0.                                                          ELM130C
      DD = 0.                                                          ELM131C
      DO 31 K = 1,NEL                                                  ELM132C
      CC = CC + SHAPE(I,K)*UL(J,K) + SHAPE(J,K)*UL(I,K)                ELM133C
  310 DD = DD + SHAPE(I,K)*DUL(J,K) + SHAPE(J,K)*DUL(I,K)              ELM134C
      DEPS(I,J) = DD/XJAC                                              ELM135C
      DEPS(J,I) = CC/XJAC                                              ELM136C
      XIG(I,J) = GG*DEPS(J,I)                                          ELM137C
  320 XIG(J,I) = XIG(I,J)                                              ELM138C
      DTH = DTH + DEPS(I,I)                                            ELM139C
  330 THP = THP + DEPS(I,I)                                            ELM140C
      DO 34 I = 1,NDIM                                                 ELM141C
      DEPS(I,I) = DEPS(I,I) - DTH/3.0                                  ELM142C
  340 DEPS(I,I) = DEPS(I,I)/2.0                                        ELM143C
      XIK = THP*(XK/2.-GG/3.)                                          ELM144C
      YIK = XIK                                                        ELM145C
      DTH = DTH/2.                                                     ELM146C
C**** UPDATE THE SHEAR HISTORY                                         ELM147C
      IG = 4                                                           ELM148C
      IF(NG.LE.0) GO TO 410                                            ELM149C
      DO 400 I = 1,NG                                                  ELM150C
      IG = IG + 2                                                      ELM151C
      DD = U(IG+1,MA)                                                  ELM152C
      CC = U(IG,MA)*HIST(DTP/DD)                                       ELM153C
      DD = EXP(-DT/DD)                                                 ELM154C
      DO 402 J = 1,NDIM                                                ELM155C
      DO 402 K = J,NDIM                                                ELM156C
      EE = H(NH) + CC*DEPS(J,K)                                        ELM157C
      XIG(J,K) = XIG(J,K) + EE                                         ELM158C
      EE = EE*DD                                                       ELM159C
      XIG(K,J) = XIG(K,J) + EE                                         ELM160C
      H(NH) = EE                                                       ELM161C
  402 NH = NH + 1                                                      ELM162C
  400 CONTINUE                                                         ELM163C
C**** UPDATE THE BULK HISTORY                                          ELM164C
  410 IF(NK.LE.0) GO TO 430                                            ELM165C
      DO 420 I = 1,NK                                                  ELM166C
      IG = IG + 2                                                      ELM167C
      DD = U(IG+1,MA)                                                  ELM168C
      EE = H(NH) + U(IG,MA)*HIST(DTP/DD)*DTH                           ELM169C
      XIK = XIK + EE                                                   ELM170C
      EE = EE*EXP(-DT/DD)                                              ELM171C
      YIK = YIK + EE                                                   ELM172C
      H(NH) = EE                                                       ELM173C
  420 NH = NH + 1                                                      ELM174C
  430 CONTINUE                                                         ELM175C
      DO 435 I = 1,NDIM                                                ELM176C
      XIG(I,I) = XIG(I,I) + XIK                                        ELM177C
  435 XIG(I+1,I) = XIG(I+1,I) + YIK                                    ELM178C
C**** OUTPUT THE STRESSES FOR THE CURRENT TIME STEP                    ELM179C
      IF(NOPRNT) GO TO 640                                             ELM180C
      NCT = NCT - 1                                                    ELM181C
      IF(NCT.GT.0) GO TO 630                                           ELM182C
      WRITE(IOP6,2004) HEAD,TIME,IPG,(XHED(I),XH,I=1,NDIM),            ELM183C
     X ((SHED(I,J),SH,I=J,NDF),J=1,NDF)                                ELM184C
      WRITE(IOP6,2003) (BLANK,BLANK,I=1,NDIM),                         ELM185C
```

```
     X ((SHED(I,J),EM,I=J,NDF),J=1,NDF)                                        ELM186C
       IPG = IPG + 1                                                           ELM187C
       MCT = 15                                                                ELM188C
630    WRITE(IPG,1AH1) NA,(XX(I),I=1,NDIM),((XIG(I,J),J=I,NDIM),I=1,NDIM)      ELM189C
       WRITE(IPG,1AH2) UM,MA,((DEPS(J,I),J=I,NDIM),I=1,NDIM)                   ELM190C
640    IF(NPL) CALL PLDATA(NUMEL,NDIM,N,LL,SHED,XX,XIG,FORCE)                  ELM191C
       IF(LL.NE.1 .OR. IPLOT.EQ. 0.0) GO TO 650
       IF(IPLOT.EQ.3)  GO TO 645
       VECT(N,1)= XIG(1,1)
       VECT(N,2)= XIG(2,2)
       VECT(N,3)= 0.0
       VECT(N,4)= XIG(1,2)
       VECT(N,5)= DEPS(2,1)
       VECT(N,6)= DEPS(3,2)
       VECT(N,7)= 0.0
       VECT(N,8)= DEPS(3,1)
       GO TO 650
645    KK =
       DO 646 I=1,NDIM
       DO 646 J=1,NDIM
       KK = KK + 1
       VECT(KK,1) = XIG(I,J)
646    VECT(KK,2) = DEPS(J,I)
650    IF(ISW.EQ.4) GO TO 500
C**** COMPUTE THE LOADS FOR THE NEXT TIME STEP                                 ELM193C
       DO 440 I = 1,NDIM                                                       ELM194C
       DO 44  J = 1,NDIM                                                       ELM195C
       XIG(I,J) = XIG(J,I)*WW                                                  ELM196C
440    XIG(J,I) = XIG(I,J)                                                     ELM197C
       II =                                                                    ELM198C
       WW=XJAC*WW*D(3,MA)                                                      ELM199C
       DO 47  I = 1,NEL                                                        ELM200C
       EE=SHAPE(4,I)*WW                                                        ELM201C
       DO 46  J = 1,NDF                                                        ELM202C
       CC = 0.0                                                                ELM203C
       DO 45  K = 1,NDIM                                                       ELM204C
450    CC = CC + SHAPE(K,I)*XIG(J,K)                                           ELM205C
460    FORCE(II+J,1)=FORCE(II+J,1)-CC-EE*UDL(J,I)                              ELM206C
470    II = II + NDF                                                           ELM207C
500    CONTINUE                                                                ELM208C
       RETURN                                                                  ELM209C
C**** ELEMENT LOAD COMPUTATION FOR BODY TYPE FORCES, ETC.                      ELM210C
5      CONTINUE                                                                ELM211C
       RETURN                                                                  ELM212C
4000   IPG = 0                                                                 ELM213C
       RETURN                                                                  ELM214C
C**** FORMAT STATEMENTS                                                        ELM215C
 1000  FORMAT(2I5,3F10.0)                                                      ELM216C
1001   FORMAT(8F10.0)                                                          ELM217C
2000   FORMAT(23H  VISCOELASTIC MATERIAL//I10,12H SHEAR TERMS/                 ELM218C
     X I10,12H BULK  TERMS//10X,12H G-INFINITY=,1PE12.4/10X,                   ELM219C
     X 12H K-INFINITY=,1PE12.4/10X,12H DENSITY   =, 1PE12.4/1X)                ELM220C
2001   FORMAT(10X,14H SHEAR MODULUS,6X,4HTIME/(10X,1P2E12.4))                  ELM221C
2002   FORMAT(10X,14H BULK  MODULUS,6X,4HTIME/(10X,1P2E12.4))                  ELM222C
2003   FORMAT(8X,9(2A6))                                                       ELM223C
2004   FORMAT(1H1,12A6,E13.5,21X,4HPAGE,I3//5X,16HELEMENT STRESSES//           ELM224C
     1 1X,7HELEMENT,9(2A6))                                                    ELM225C
2030   FORMAT(9H MATERIAL,I3,17H EXCEEDS STORAGE.,I5,13H SHEAR TERMS.,I5,      ELM226C
     X 16H BULK TERMS NEED,I4,7H MORE. )                                       ELM227C
       END                                                                     ELM228C
*DECK ELMT23
       SUBROUTINE ELMT23(N,MA,NDIM,NDF,NEL,NELI,NSTF,NSIZV,NVEC,MCT,DM,D,      ELM  1C
     X XYZ,IX,MF,FORCE,ESTIF,U,VECT,ISW)                                       ELM  2C
       REAL LABL                                                               ELM  3C
       LOGICAL NPR,NPL,NUPRNT                                                  ELM  4C
       DIMENSION D(43,1),ESTIF(NSTF,NSTF),FORCE(NSTF,2),STOW(4,4),XX(2),       ELM  5C
     X  IX(NEL+1),U(NDF,1),DEPS(3,2),XIG(3,2),H(1),SHED(6)                     ELM  6C
       DIMENSION VECT(NSIZV,NVEC)                                              ELM  7C
       COMMON/DYNAMO/TIME,NSIG(7),NT,NSTEP,DT,NUMPLT,NEDATA(20,3),NPR,NPL      ELM  8C
       COMMON/LABELS/ LABL(6),XHED(3),FHED(6),EM,UHED(6),UM,RHED(6),MA         ELM  9C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSIR,WORD(30)            ELM 10C
       COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SW(3,3),X(3,20),LD(120)           ELM 11C
```

```
      COMMON/TAPES/ ITP5,ITP6                                                  FLM 11C
      COMMON/VISDAT/ DTP,NM,ISZM,DUL(6,20),UL(6,20),UUL(6,20),C6               FLM 12C
      COMMON/VTAPE/ ITP13,ITP14,ITRD,ITWR                                      FLM 13C
      COMMON/PLOTS/ IPL12D
      DATA SHED/3H RR,3H RZ,3H ZZ,3H TT,6HPRINC1,6HPRINC2/                     FLM 14C
      DATA PT,TT/1.333333333333333,.666666666666667/                           FLM 15C
      IF(ISW.GT.6) GO TO 4000                                                  FLM 16C
      NSIDE =(NEL+1)/4                                                         FLM 17C
      GO TO (1,2,3,4,5,4), ISW                                                 FLM 18C
C**** CHARACTERIZE THE MATERIAL FOR ISOTROPIC LINEAR VISCOELASTICITY           FLM 19C
1     READ(ITP5,1000) NG,NK,D(4,MA),D(5,MA),D(3,MA)                            FLM 20C
      WRITE(ITP6,2000) NG,NK,D(4,MA),D(5,MA),D(3,MA)                           FLM 21C
      CALL INTEGL(2,0,NDIM,LINT,STUW)                                          FLM 22C
      CA = 0.0                                                                 FLM 23C
      DT = 1.E                                                                 FLM 24C
      DTP = 0.0                                                                FLM 25C
      NM = 1                                                                   FLM 26C
      D(1,MA) = NG + .01                                                       FLM 27C
      D(2,MA) = NK + .01                                                       FLM 28C
      NU = 5 + 2*NG                                                            FLM 29C
      IF(NG.EQ.0) GO TO 100                                                    FLM 30C
      READ(ITP5,1001) (D(I,MA),I=6,NU)                                         FLM 31C
      WRITE(ITP6,2001)(D(I,MA),I=6,NU)                                         FLM 32C
100   IF(NK.EQ.0) GO TO 200                                                    FLM 33C
      NL = NU + 1                                                              FLM 34C
      NU = NU + 2*NK                                                           FLM 35C
      READ(ITP5,1001)  (D(I,MA),I=NL,NU)                                       FLM 36C
      WRITE(ITP6,2002) (D(I,MA),I=NL,NU)                                       FLM 37C
200   IF(NU.LT.64) RETURN                                                      FLM 38C
      WRITE(ITP5,2030) MA,NG,NK,NU                                             FLM 39C
      IPG = 0                                                                  FLM 40C
      RETURN                                                                   FLM 41C
C**** CHECK MESH FOR NEGATIVE JACOBIANS AT ELEMENT CORNERS                     FLM 42C
2     CALL CKMESH(NDIM,NEL,DM)                                                 FLM 43C
      NG = D(1,MA)                                                             FLM 44C
      NK = D(2,MA)                                                             FLM 45C
      NM = NM + (LINT+1)*(NK+NG*((NDIM+1)*NDIM)/2+NG)                          FLM 46C
      CONV = 45./ATAN(1.)/2.                                                   FLM 47C
      RETURN                                                                   FLM 48C
C**** COMPUTE THE VISCOELASTIC STIFFNESS MATRICES FOR SHEAR AND BULK           FLM 49C
C**** SET UP THE MATERIAL PROPERTIES                                           FLM 50C
3     NL = 4                                                                   FLM 51C
      NG = D(1,MA)                                                             FLM 52C
      GG = D(4,MA)                                                             FLM 53C
      RO = D(3,MA)                                                             FLM 54C
      IF(NG.LE.0) GO TO 150                                                    FLM 55C
      NL = 2*NG + NL                                                           FLM 56C
      DO 140 I = 6,NL,2                                                        FLM 57C
140   GG = GG + D(I,MA)*EXP(DT/D(I+1,MA))                                      FLM 58C
150   NK = D(2,MA)                                                             FLM 59C
      KK = D(5,MA)                                                             FLM 60C
      IF(NK.LE.0) GO TO 170                                                    FLM 61C
      IL = NL + 2                                                              FLM 62C
      NL = NL + 2*NK                                                           FLM 63C
      DO 160 I = IL,NL,2                                                       FLM 64C
160   KK = KK + D(I,MA)*EXP(DT/D(I+1,MA))                                      FLM 65C
170   C11 = KK + PT*GG                                                         FLM 66C
      C12 = KK - TT*GG                                                         FLM 67C
      C13 = C12                                                                FLM 68C
      C14 = 0.0                                                                FLM 69C
      C22 = C11                                                                FLM 70C
      C23 = C12                                                                FLM 71C
      C33 = C11                                                                FLM 72C
      C34 = 0.0                                                                FLM 73C
      C44 = GG                                                                 FLM 74C
      DO 39 KK = 1,LINT                                                        FLM 75C
      S  = STUW(1,KK)                                                          FLM 76C
      T  = STUW(2,KK)                                                          FLM 77C
      WW = STUW(4,KK)                                                          FLM 78C
      CALL BRICK2(S,T,-1.,2,NEL,NSIDE)                                         FLM 79C
      RR = 0.0                                                                 FLM 80C
      DO 38 I = 1,NEL                                                          FLM 81C
```

```
180   RR = RR + X(I,I)*SHAPE(4,I)                                          FLM 82C
      DVOL = WW*RR*XJAC                                                     FLM 83C
      II = I                                                                FLM 84C
      DO 380 I = 1,NEL                                                      FLM 85C
C**** COMPUTE LUMPED MASS                                                   FLM 86C
      TEMP = DVOL*SHAPE(4,I)*RHO                                            FLM 87C
      FORCE(II  ,2) = FORCE(II  ,2) + TEMP                                  FLM 88C
      FORCE(II+1,2) = FORCE(II+1,2) + TEMP                                  FLM 89C
C**** COMPUTE ELEMENT STIFFNESS                                             FLM 90C
      RN = SHAPE(1,I)/XJAC*DVOL                                             FLM 91C
      ZN = SHAPE(2,I)/XJAC*DVOL                                             FLM 92C
      ET = SHAPE(4,I)/RR*DVOL                                               FLM 93C
      A11 = C11*RN + C12*ET + C14*ZN                                        FLM 94C
      A12 = C13*ZN + C14*RN                                                 FLM 95C
      A21 = C12*RN + C22*ET                                                 FLM 96C
      A22 = C23*ZN                                                          FLM 97C
      A31 = C13*RN + C23*ET + C34*ZN                                        FLM 98C
      A32 = C33*ZN + C34*RN                                                 FLM 99C
      A41 = C14*RN + C44*ZN                                                 FLM100C
      A42 = C34*ZN + C44*RN                                                 FLM101C
      JJ = II                                                               FLM102C
      DO 375 J = I,NEL                                                      FLM103C
      RN = SHAPE(1,J)/XJAC                                                  FLM104C
      ZN = SHAPE(2,J)/XJAC                                                  FLM105C
      ET = SHAPE(4,J)/RR                                                    FLM106C
      ESTIF(II  ,JJ  ) = ESTIF(II  ,JJ  ) + RN*A11 + ET*A21 + ZN*A41       FLM107C
      ESTIF(II  ,JJ+1) = ESTIF(II  ,JJ+1) + ZN*A31 + RN*A41                FLM108C
      IF(I.NE.J)                                                            FLM109C
     XESTIF(II+1,JJ  ) = ESTIF(II+1,JJ  ) + RN*A12 + ET*A22 + ZN*A42       FLM110C
      ESTIF(II+1,JJ+1) = ESTIF(II+1,JJ+1) + ZN*A32 + RN*A42                FLM111C
375   JJ = JJ + NDF                                                         FLM112C
380   II = II + NDF                                                         FLM113C
390   CONTINUE                                                              FLM114C
C**** CONSTRUCT SYMMETRIC ESTIF                                             FLM115C
      DO 395 I = 2,NSTF                                                     FLM116C
      K = I-1                                                               FLM117C
      DO 395 J = 1,K                                                        FLM118C
      ESTIF(I,J) = ESTIF(I,J) + ESTIF(J,I)                                  FLM119C
395   ESTIF(J,I) = ESTIF(I,J)                                               FLM120C
      IF(CA.EQ.0.0) RETURN                                                  FLM121C
      DO 19 I=1,NSTF                                                        FLM122C
190   ESTIF(I,1) = ESTIF(I,1) + CA*FORCE(I,2)                               FLM123C
      RETURN                                                                FLM125C
C**** STRESS COMPUTATION FOR THE VISCOELASTIC ELEMENT                       FLM126C
C**** VISCOELASTIC LOAD HISTORY COMPUTATION AND ELEMENT ASSEMBLY FROM TP    FLM127C
4     CONTINUE                                                              FLM128C
C**** COMPUTATION AND OUTPUT OF STRESS AND HISTORY OF VISCOELASTIC SOLN.    FLM129C
C**** DO THE INTEGRATION AND OUTPUT THE STRESS                              FLM130C
      NG = U(1,MA)                                                          FLM131C
      NK = U(2,MA)                                                          FLM132C
      GG = U(4,MA)                                                          FLM133C
      XK = U(5,MA)                                                          FLM134C
C**** CHECK FOR H STORAGE                                                   FLM135C
      LL = KH + (LINT+1)*(NK+NG*((NDIM+1)*NDIM)/2+NG)                       FLM136C
      IF(LL.GT.ISZH) CALL VBUFF(ITRD,IIWR,H,NH,ISZH)                        FLM137C
      NL = LINT + 1                                                         FLM138C
      DO 60 LL = 1,NL                                                       FLM139C
      NOPRNT = NPR                                                          FLM140C
      IF(NSIG(LL).NE.0) NOPRNT = .TRUE.                                     FLM141C
      NPL = .FALSE.                                                         FLM142C
      IF(NUMPLT.LE.0) GO TO 210                                             FLM143C
      DO 205 NP = 1,NUMPLT                                                  FLM144C
205   IF(NEDATA(NP,1).EQ.N.AND.NEDATA(NP,2).EQ.LL) NPL = .TRUE.             FLM145C
210   CONTINUE                                                              FLM146C
      IF(LL.NE.1) GO TO 220                                                 FLM147C
      S = 0.0                                                               FLM148C
      T = 0.0                                                               FLM149C
      GO TO 230                                                             FLM150C
220   S = STOW(1,LL-1)                                                      FLM151C
      T = STOW(2,LL-1)                                                      FLM152C
      WW = STOW(4,LL-1)                                                     FLM153C
230   CALL BRICK2(S,T,-1.,2,NEL,NSIDE)                                      FLM154C
```

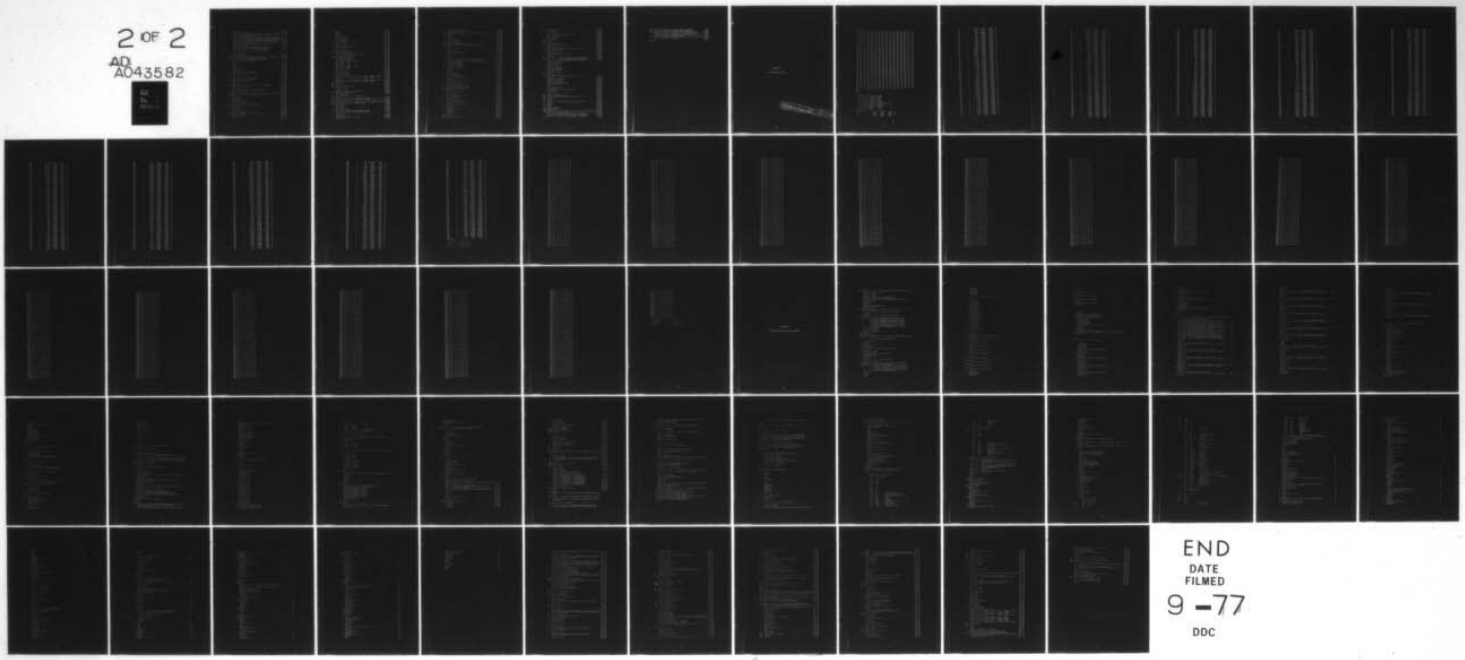
```
C**** COMPUTE THE PRESENT STRAIN AND STRAIN INCREMENT                    FLM155C
      DTH = 0.                                                          FLM156C
      THP = 0.                                                          FLM157C
      DO 330 I = 1,2                                                    FLM158C
      CC = 0.0                                                          FLM159C
      DO 300 K = 1,NEL                                                  FLM160C
300   CC = CC + SHAPE(4,K)*X(I,K)                                       FLM161C
      XX(I) = CC                                                        FLM162C
      DO 320 J = 1,2                                                    FLM163C
      CC = 0.                                                           FLM164C
      DD = 0.                                                           FLM165C
      DO 310 K = 1,NEL                                                  FLM166C
      CC = CC + SHAPE(I,K)*DUL(J,K) + SHAPE(J,K)*DUL(I,K)               FLM167C
310   DD = DD + SHAPE(I,K)*DUL(J,K) + SHAPE(J,K)*DUL(I,K)               FLM168C
      DEPS(I,J) = DD/XJAC                                               FLM169C
      DEPS(J+1,I) = CC/XJAC                                             FLM170C
      XIG(I,J) = GG*DEPS(J+1,I)                                         FLM171C
320   XIG(J+1,I) = XIG(I,J)                                             FLM172C
      DTH = DTH + DEPS(I,I)                                             FLM173C
330   THP = THP + DEPS(I+1,I)                                           FLM174C
      RR = XX(1)                                                        FLM175C
      ET = 0.                                                           FLM176C
      DET = 0.                                                          FLM177C
      DO 350 K = 1,NEL                                                  FLM178C
      CC = SHAPE(4,K)                                                   FLM179C
      ET = ET + CC*UL(1,K)                                              FLM180C
350   DET = DET + CC*DUL(1,K)                                           FLM181C
      ST = GG*ET*2.0                                                    FLM182C
      STP = ST                                                          FLM183C
      XIK = (THP + ET*2./RR)*(XK/2. - GG/3.)                            FLM184C
      YIK = XIK                                                         FLM185C
      DTH = DTH/2. + DET/RR                                             FLM186C
      DET = DET*2.0 - DTH*RR*TT                                         FLM187C
      DO 340 I = 1,2                                                    FLM188C
      DEPS(I,I) = DEPS(I,I) - DTH*TT                                    FLM189C
340   DEPS(I+1,I) = DEPS(I+1,I)/2.0                                     FLM190C
C**** UPDATE THE SHEAR HISTORY                                          FLM191C
      IG = 4                                                            FLM192C
      IF(NG.LE.0) GO TO 410                                             FLM193C
      DO 400 I = 1,NG                                                   FLM194C
      IG = IG + 2                                                       FLM195C
      DD = D(IG+1,MA)                                                   FLM196C
      CC = D(IG,MA)*HIST(DTP/DD)                                        FLM197C
      DD = EXP(-DT/DD)                                                  FLM198C
      DO 402 J = 1,2                                                    FLM199C
      DO 402 K = J,2                                                    FLM200C
      EE = H(NH) + CC*DEPS(J,K)                                         FLM201C
      XIG(J,K) = XIG(J,K) + EE                                          FLM202C
      EE = EE*DD                                                        FLM203C
      XIG(K+1,J) = XIG(K+1,J) + EE                                      FLM204C
      H(NH) = EE                                                        FLM205C
402   NH = NH + 1                                                       FLM206C
      EE = H(NH) + CC*DET                                               FLM207C
      ST = ST + EE                                                      FLM208C
      EE = EE*DD                                                        FLM209C
      H(NH) = EE                                                        FLM210C
      STP = STP + EE                                                    FLM211C
      NH = NH + 1                                                       FLM212C
400   CONTINUE                                                          FLM213C
C**** UPDATE THE BULK HISTORY                                           FLM214C
410   IF(NK.LE.0) GO TO 430                                             FLM215C
      DO 420 I = 1,NK                                                   FLM216C
      IG = IG + 2                                                       FLM217C
      DD = D(IG+1,MA)                                                   FLM218C
      EE = H(NH) + D(IG,MA)*HIST(DTP/DD)*DTH                            FLM219C
      XIK = XIK + EE                                                    FLM220C
      EE = EE*EXP(-DT/DD)                                               FLM221C
      YIK = YIK + EE                                                    FLM222C
      H(NH) = EE                                                        FLM223C
420   NH = NH + 1                                                       FLM224C
430   CONTINUE                                                          FLM225C
      DO 435 I = 1,2                                                    FLM226C
```

```
      XIG(I,I) = XIG(1,I) + XIK                                          FLM227C
435   XIG(I+1,I) = XIG(I+1,I) + YIK                                      FLM228C
      FT = FT/XX(I)                                                      FLM229C
      ST = ST/XX(I) + XIK                                                FLM230C
C**** OUTPUT THE STRESSES FOR THE CURRENT TIME STEP                      FLM231C
      IF(NOPRNT) GO TO 640                                               FLM232C
      MCT = MCT - 1                                                      FLM233C
      IF(MCT.GT.0) GO TO 630                                             FLM234C
      WRITE(IPE,2004) NEAD,TIME,IPG                                      FLM235C
      IPG = IPG + 1                                                      FLM236C
      MCT = 16                                                           FLM237C
630   WRITE(IPE,2003) N,(XX(I),I=1,2),((XIG(I,J),J=1,2),I=1,2),ST,       FLM238C
     X                 UM,MA,((DEPS(J+I-1),J=1,2),I=1,2),FT            FLM239C
640   IF(NPL) CALL PLDATA(NDIMEL,NDIM,N,LL,SMED,XX,XIG,FORCE)            FLM240C
      IF(IPLT2D .NE. 1) GO TO 641                                        FLM240C
      IF(LL.NE.1) GO TO 641                                              FLM240C
      VECT(N,1)= XIG(1,1)                                                FLM240C
      VECT(N,2)= XIG(2,2)                                                FLM240C
      VECT(N,3)= ST                                                      FLM240C
      VECT(N,4) = XIG(1,2)                                               FLM240C
      VECT(N,5)= DEPS(2,1)                                               FLM240C
      VECT(N,6)= DEPS(3,2)                                               FLM240C
      VECT(N,7)= ET                                                      FLM240C
      VECT(N,8)= DEPS(3,1)                                               FLM240C
641   CONTINUE                                                           FLM240C
      IF(ISW.EQ.4.OR.LL.EW.1) GO TO 500                                  FLM241C
C**** COMPUTE THE LOADS FOR THE NEXT TIME STEP                           FLM242C
      DD = XW*RR                                                         FLM243C
      XIG(1,1) = XIG(2,1)*DD                                             FLM244C
      XIG(1,2) = XIG(3,1)*DD                                             FLM245C
      XIG(2,2) = XIG(3,2)*DD                                             FLM246C
      XIG(2,1) = XIG(1,2)                                                FLM247C
      STF = (STF/RR + YIK)*XW*XJAC                                       FLM248C
      WW = XJAC*DD*D(3,MA)                                               FLM249C
      II =                                                               FLM250C
      DO 47  I = 1,NEL                                                   FLM251C
      CC = SHAPE(4,I)                                                    FLM252C
      FORCE(II+1,1) = FORCE(II+1,1) - STF*CC                             FLM253C
      EE = CC*WW                                                         FLM254C
      DO 46  J = 1,2                                                     FLM255C
      CC = 0.0                                                           FLM256C
      DO 45  K = 1,2                                                     FLM257C
450   CC = CC + SHAPE(K,I)*XIG(J,K)                                      FLM258C
460   FORCE(II+J,1) = FORCE(II+J,1) - CC -EE*UDL(J,I)                    FLM259C
470   II = II + NDF                                                      FLM260C
500   CONTINUE                                                           FLM261C
      RETURN                                                             FLM262C
C**** ELEMENT LOAD COMPUTATION FOR BODY TYPE FORCES, ETC.                FLM263C
5     CONTINUE                                                           FLM264C
      RETURN                                                             FLM265C
4000  IPG = 0                                                            FLM266C
      RETURN                                                             FLM267C
C**** FORMAT STATEMENTS                                                  FLM268C
1000  FORMAT(2I5,4F10.0)                                                 FLM269C
1001  FORMAT(8F10.0)                                                     FLM270C
2000  FORMAT(5X,57H LINEAR VISCOELASTIC MATERIAL, AXISYMMETRIC SOLID ELE FLM271C
     XMENT  //I10,12H SHEAR TERMS/                                       FLM272C
     X I10,12H BULK  TERMS//10X,12H G-INFINITY=,1PE12.4/10X,             FLM273C
     X 12H K-INFINITY=,1PE12.4/10X,12H DENSITY   =,1PE12.4/1X)          FLM274C
2001  FORMAT(10X,14H SHEAR MODULUS,6X,4HTIME/(10X,1P2E12.4))            FLM275C
2002  FORMAT(10X,14H BULK  MODULUS,6X,4HTIME/(10X,1P2E12.4))            FLM276C
2003  FORMAT(1H0,I7,0P2F12.3,1P4E12.3/8X,A5,I7,12X,1P4E12.3)             FLM277C
2004  FORMAT(1H1,12A6,E13.5,20X,4HPAGE,I3//5X,16HELEMENT STRESSES//      FLM278C
     1 1X,7HELEMENT,4X,6HR ORD.,6X,6HZ ORD.,5X,9HRR-STRESS,3X,          FLM279C
     2 9HZZ-STRESS,3X,9HZZ-STRESS,3X,9HTT-STRESS/35X,9HRR-STRAIN,3X,    FLM280C
     3 9HZZ-STRAIN,3X,9HZZ-STRAIN,3X,9HTT-STRAIN)                       FLM281C
2030  FORMAT(9H MATERIAL,I3,17H EXCEEDS STORAGE,,I5,13H SHEAR TERMS,,I5, FLM282C
     X 16H BULK TERMS NEED,I4,7H WORDS. )                               FLM283C
      END                                                                FLM284C
*DECK ELMT24                                                             FLM285C
      SUBROUTINE ELMT24(N,MA,NDIM,NDF,NEL,NEL1,NSTP,NSIZE,NREC,MCT,IPG,  FLM286C
     X XYZ,IX,SH,FORCE,ESTIF,U,VECT,ISW)                                FLM287C
```

AD-A043 582 CIVIL ENGINEERING LAB (NAVY) PORT HUENEME CALIF F/G 6/2
A FINITE ELEMENT HEAD INJURY MODEL. VOLUME II. COMPUTER PROGRAM--ETC(U)
JUL 77 T A SHUGAR

UNCLASSIFIED CEL-R-854-2 NL

2 OF 2
AD A043582



END
DATE FILMED
9 -77
DDC

```
      REAL LAML                                                          FLM  3C
      LOGICAL NPR,NPL,NOPRNT                                             FLM  4C
      DIMENSION D(63,1),FSTIF(NSTF,NSTF),FORCE(NSTF,2),STOW(4,4),XX(2),  FLM  5C
     X  IX(NEL+1),U(NDF,1),UEPS(3,2),XIG(3,2),H(1),SHED(6)               FLM  6C
      DIMENSION VECT(NSIZV,NVEC)                                         FLM  7C
      COMMON/DYNAMO/TIME,NSIG(7),NI,NSTEP,DT,NUMPLT,NEDATA(20,3),NPR,NPL FLM  8C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UM,RHED(6),RM FLM  9C
     X ,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)      FLM 10C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)      FLM 11C
      COMMON/TAPES/ ITP5,ITP6                                            FLM 12C
      COMMON/VISDAT/ DTP,NH,ISZH,DUL(6,20),UL(6,20),UDL(6,20),C6         FLM 13C
      COMMON/VTAPE/ ITP13,ITP14,ITRD,ITWR                                FLM 14C
      COMMON/PLOTS/ IPL120                                               FLM 15C
      DATA SHED/3H RR,3H ZZ,3H ZZ,3H TT,6HPRINC1,6HPRINC2/               FLM 16C
      DATA PI,TT/3.333333333333333,.666666666666667/                     FLM 17C
C
C     AXISYMMETRIC FLUID ELEMENT (ONE POINT QUADRATURE)
C
      IF(IS.GT.6) GO TO 4000                                             FLM 18C
      NSIDE =(NEL+1)/4                                                   FLM 19C
      GO TO (1,2,3,4,5,4), IS
C.... CHARACTERIZE THE MATERIAL FOR ISOTROPIC LINEAR VISCOELASTICITY      FLM 20C
1     READ(ITP5,1000) NG,NK,D(4,MA),D(5,MA),D(3,MA)                      FLM 21C
      WRITE(ITP6,2000) NG,NK,D(4,MA),D(5,MA),D(3,MA)                     FLM 22C
      CALL INTEGL(1,0,NDIM,LINT,STOW)
      CA = 0.0                                                           FLM 23C
      DT = 1.0                                                           FLM 24C
      DTP = 0.0                                                          FLM 25C
      NH = 1                                                             FLM 26C
      D(1,MA) = NG + .01                                                 FLM 27C
      D(2,MA) = NK + .01                                                 FLM 28C
      NU = 5 + 2*NG                                                      FLM 29C
      IF(NG.EQ.0) GO TO 100                                              FLM 30C
      READ(ITP5,1001) (D(I,MA),I=6,NU)                                   FLM 31C
      WRITE(ITP6,2001)(D(I,MA),I=6,NU)                                   FLM 32C
100   IF(NK.EQ.0) GO TO 200                                              FLM 33C
      NL = NU + 1                                                        FLM 34C
      NU = NU + 2*NK                                                     FLM 35C
      READ(ITP5,1001)  (D(I,MA),I=NL,NU)                                 FLM 36C
      WRITE(ITP6,2002) (D(I,MA),I=NL,NU)                                 FLM 37C
200   IF(NU.LT.64) RETURN                                                FLM 38C
      WRITE(ITP6,2030) MA,NG,NK,NU                                       FLM 39C
      IPG = 0                                                            FLM 40C
      RETURN                                                             FLM 41C
C.... CHECK MESH FOR NEGATIVE JACOBIANS AT ELEMENT CORNERS               FLM 42C
2     CALL CKMESH(NDIM,NEL,D4)                                           FLM 43C
      NG = D(1,MA)                                                       FLM 44C
      NK = D(2,MA)                                                       FLM 45C
      NH = NH + (LINT+1)*(NK+NG*((NDIM+1)*NDIM)/2+NG)                    FLM 46C
      CONI = 45./ATAN(1.)/2.                                             FLM 47C
      RETURN                                                             FLM 48C
C.... COMPUTE THE VISCOELASTIC STIFFNESS MATRICES FOR SHEAR AND BULK     FLM 49C
C.... SET UP THE MATERIAL PROPERTIES                                     FLM 50C
3     NL = 4                                                             FLM 51C
      NG = D(1,MA)                                                       FLM 52C
      GG = D(4,MA)                                                       FLM 53C
      RO = D(3,MA)                                                       FLM 54C
      IF(NG.LE.0) GO TO 150                                              FLM 55C
      NL = 2*NG + NL                                                     FLM 56C
      DO 14 I = 6,NL,2                                                   FLM 57C
140   GG = GG + D(I,MA)*HIST(DT/D(I+1,MA))                               FLM 58C
150   NK = D(2,MA)                                                       FLM 59C
      XK = D(5,MA)                                                       FLM 60C
      IF(NK.LE.0) GO TO 170                                              FLM 61C
      IL = NL + 2                                                        FLM 62C
      NL = NL + 2*NK                                                     FLM 63C
      DO 16 I = IL,NL,2                                                  FLM 64C
160   XK = XK + D(I,MA)*HIST(DT/D(I+1,MA))                               FLM 65C
170   C11 = XK + FT*GG                                                   FLM 66C
      C12 = XK - TT*GG                                                   FLM 67C
      C13 = C12                                                          FLM 68C
      C14 = 0.0                                                          FLM 69C
```

```
      C22 = C11                                                          FLM 70C
      C23 = C12                                                          FLM 71C
      C33 = C11                                                          FLM 72C
      C34 = C13                                                          FLM 73C
      C44 = GG                                                           FLM 74C
      DO 390 KK = 1,LINT                                                 FLM 75C
      S  = STOW(1,KK)                                                    FLM 76C
      T  = STOW(2,KK)                                                    FLM 77C
      WW = STOW(4,KK)                                                    FLM 78C
      CALL SHPCK2(S,T,-1.,2,NEL,NSIDE)                                   FLM 79C
      RR = 0.                                                            FLM 80C
      DO 180 I = 1,NEL                                                   FLM 81C
180   RR = RR + X(1,I)*SHAPE(4,I)                                        FLM 82C
      DVOL = WW*RR*XJAC                                                  FLM 83C
      I1 = 1                                                             FLM 84C
      DO 380 I = 1,NEL                                                   FLM 85C
C.... COMPUTE LUMPED MASS                                                FLM 86C
      TEMP = DVOL*SHAPE(4,I)*RHO                                         FLM 87C
      FORCE(I1  ,2) = FORCE(I1  ,2) + TEMP                               FLM 88C
      FORCE(I1+1,2) = FORCE(I1+1,2) + TEMP                               FLM 89C
C.... COMPUTE ELEMENT STIFFNESS                                          FLM 90C
      RN = SHAPE(1,I)/XJAC*DVOL                                          FLM 91C
      ZN = SHAPE(2,I)/XJAC*DVOL                                          FLM 92C
      ET = SHAPE(4,I)/RR*DVOL                                            FLM 93C
      A11 = C11*RN + C12*ET + C14*ZN                                     FLM 94C
      A12 = C13*ZN + C14*RN                                              FLM 95C
      A21 = C12*RN + C22*ET                                              FLM 96C
      A22 = C23*ZN                                                       FLM 97C
      A31 = C13*RN + C23*ET + C34*ZN                                     FLM 98C
      A32 = C33*ZN + C34*RN                                              FLM 99C
      A41 = C14*RN + C44*ZN                                              FLM100C
      A42 = C34*ZN + C44*RN                                              FLM101C
      J1 = I1                                                            FLM102C
      DO 375 J = 1,NEL                                                   FLM103C
      RN = SHAPE(1,J)/XJAC                                               FLM104C
      ZN = SHAPE(2,J)/XJAC                                               FLM105C
      ET = SHAPE(4,J)/RR                                                 FLM106C
      ESTIF(I1  ,J1  ) = ESTIF(I1  ,J1  ) + RN*A11 + ET*A21 + ZN*A41     FLM107C
      ESTIF(I1  ,J1+1) = ESTIF(I1  ,J1+1) + ZN*A31 + RN*A41              FLM108C
      IF(I.NE.J)                                                         FLM109C
     XESTIF(I1+1,J1  ) = ESTIF(I1+1,J1  ) + RN*A12 + ET*A22 + ZN*A42     FLM110C
      ESTIF(I1+1,J1+1) = ESTIF(I1+1,J1+1) + ZN*A32 + RN*A42              FLM111C
375   J1 = J1 + NDF                                                      FLM112C
380   I1 = I1 + NDF                                                      FLM113C
390   CONTINUE                                                           FLM114C
C.... CONSTRUCT SYMMETRIC ESTIF                                          FLM115C
      DO 395 I = 2,NSTF                                                  FLM116C
      K = I-1                                                            FLM117C
      DO 395 J = 1,K                                                     FLM118C
      ESTIF(I,J) = ESTIF(I,J) + ESTIF(J,I)                               FLM119C
395   ESTIF(J,I) = ESTIF(I,J)                                            FLM120C
      IF(C6.EQ.0.) RETURN                                                FLM121C
      DO 190 I=1,NSTF                                                    FLM122C
190   ESTIF(I,I) = ESTIF(I,I) + C6*FORCE(I,2)                            FLM123C
      RETURN                                                             FLM125C
C.... STRESS COMPUTATION FOR THE VISCOELASTIC ELEMENT                    FLM126C
C.... VISCOELASTIC LOAD HISTORY COMPUTATION AND ELEMENT ASSEMBLY FROM TP FLM127C
4     CONTINUE                                                           FLM128C
C.... COMPUTATION AND OUTPUT OF STRESS AND HISTORY OF VISCOELASTIC SOLN. FLM129C
C.... DO THE INTEGRATION AND OUTPUT THE STRESS                           FLM130C
      NG = D(1,MA)                                                       FLM131C
      NK = D(2,MA)                                                       FLM132C
      GG = D(4,MA)                                                       FLM133C
      XK = D(5,MA)                                                       FLM134C
C.... CHECK FOR H STORAGE                                                FLM135C
      LL = NH + (LINT*I)*(NK*NG*((NDIM*I)*NDIM)/2*NG)                    FLM136C
      IF(LL.GT.ISZH) CALL VBUFF(ITRD,I1WR,H,NH,ISZH)                     FLM137C
      NL = LINT                                                          FLM139C
      DO 500 LL = 1,NL                                                   FLM140C
      NOPRNT = NPR                                                       FLM141C
      IF(NSIG(LL).NE.0) NOPRNT = .TRUE.                                  FLM142C
      NPL = .FALSE.
```

```
      IF(NUMPLT.LE.0) GO TO 210                                          FLM143C
      DO 205 NP = 1,NUMPLT                                               FLM144C
205   IF(NEDATA(NP,1).EQ.N.AND.NEDATA(NP,2).EQ.LL) NPL = .TRUE.          FLM145C
210   CONTINUE                                                           FLM146C
      IF(LL.NE.1) GO TO 220                                              FLM147C
      S = 0.0                                                            FLM148C
      T = 0.0                                                            FLM149C
      GO TO 230                                                          FLM150C
220   S = STOR(1,LL-1)                                                   FLM151C
      T = STOR(2,LL-1)                                                   FLM152C
      WW = STOR(4,LL-1)                                                  FLM153C
230   CALL SHPCK2(S,T,-1.,2,NEL,NSIDE)                                   FLM154C
C.... COMPUTE THE PRESENT STRAIN AND STRAIN INCREMENT                    FLM155C
      DTH = 0.                                                           FLM156C
      THP = 0.                                                           FLM157C
      DO 33 I = 1,2                                                      FLM158C
      CC = 0.0                                                           FLM159C
      DO 300 K = 1,NEL                                                   FLM160C
300   CC = CC + SHAPE(4,K)*X(I,K)                                        FLM161C
      XX(I) = CC                                                         FLM162C
      DO 32 J = 1,2                                                      FLM163C
      CC = 0.                                                            FLM164C
      DD = 0.                                                            FLM165C
      DO 310 K = 1,NEL                                                   FLM166C
      CC = CC + SHAPE(I,K)*DLL(J,K) + SHAPE(J,K)*DUL(I,K)                FLM167C
310   DD = DD + SHAPE(I,K)*DUL(J,K) + SHAPE(J,K)*DUL(I,K)                FLM168C
      DEPS(I,J) = DD/XJAC                                                FLM169C
      DEPS(J+1,I) = CC/XJAC                                              FLM170C
      XIG(I,J) = GG*DEPS(J+1,I)                                          FLM171C
320   XIG(J+1,I) = XIG(I,J)                                              FLM172C
      DTH = DTH + DEPS(I,I)                                              FLM173C
330   THP = THP + DEPS(I+1,I)                                            FLM174C
      RR = XX(1)                                                         FLM175C
      ET = 0.                                                            FLM176C
      DET = 0.                                                           FLM177C
      DO 350 K = 1,NEL                                                   FLM178C
      CC = SHAPE(4,K)                                                    FLM179C
      ET = ET + CC*UL(1,K)                                               FLM180C
350   DET = DET + CC*DUL(1,K)                                            FLM181C
      ST = GG*ET*2.0                                                     FLM182C
      STF = ST                                                           FLM183C
      XIK = (THP + ET*2./RR)*(XK/2. - GG/3.)                             FLM184C
      YIK = XIK                                                          FLM185C
      DTH = DTH/2. + DET/RR                                              FLM186C
      DET = DET*2.0 - DTH*RR*TT                                          FLM187C
      DO 34 I = 1,2                                                      FLM188C
      DEPS(I,I) = DEPS(I,I) - DTH*TT                                     FLM189C
340   DEPS(I+1,I) = DEPS(I+1,I)/2.0                                      FLM190C
C.... UPDATE THE SHEAR HISTORY                                           FLM191C
      IG = 4                                                             FLM192C
      IF(NG.LE.3) GO TO 410                                              FLM193C
      DO 400 I = 1,NG                                                    FLM194C
      IG = IG + 2                                                        FLM195C
      DD = D(IG+1,MA)                                                    FLM196C
      CC = D(IG,MA)*HIST(DTP/DD)                                         FLM197C
      DD = EXP(-DT/DD)                                                   FLM198C
      DO 402 J = 1,2                                                     FLM199C
      DO 402 K = J,2                                                     FLM200C
      EE = H(NH) + CC*DEPS(J,K)                                          FLM201C
      XIG(J,K) = XIG(J,K) + EE                                           FLM202C
      EE = EE*DD                                                         FLM203C
      XIG(K+1,J) = XIG(K+1,J) + EE                                       FLM204C
      H(NH) = EE                                                         FLM205C
402   NH = NH + 1                                                        FLM206C
      EE = H(NH) + CC*DET                                                FLM207C
      ST = ST + EE                                                       FLM208C
      EE = EE*DD                                                         FLM209C
      H(NH) = EE                                                         FLM210C
      STF = STF + EE                                                     FLM211C
      NH = NH + 1                                                        FLM212C
400   CONTINUE                                                           FLM213C
C.... UPDATE THE BULK HISTORY                                            FLM214C
```

```
410   IF(NK.EQ.0) GO TO 430                                                  ELM215C
      DO 420 I = 1,NK                                                        ELM216C
      IG = IG + 2                                                            ELM217C
      DD = D(IG+1,MA)                                                        ELM218C
      EE = H(NH) + D(IG,MA)*HIST(DTP/DD)*DTH                                 ELM219C
      XIK = XIK + EE                                                         ELM220C
      EE = EE*EXP(-DT/DD)                                                    ELM221C
      YIK = YIK + EE                                                         ELM222C
      H(NH) = EE                                                             ELM223C
420   NH = NH + 1                                                            ELM224C
430   CONTINUE                                                               ELM225C
      DO 435 I = 1,2                                                         ELM226C
      XIG(I,I) = XIG(I,I) + XIK                                              ELM227C
435   XIG(I+1,I) = XIG(I+1,I) + YIK                                          ELM228C
      ET = ET/AX(I)                                                          ELM229C
      ST = ST/AX(I) + XIK                                                    ELM230C
C.... OUTPUT THE STRESSES FOR THE CURRENT TIME STEP                          ELM231C
      IF(NOPRNT) GO TO 640                                                   ELM232C
      MCT = MCT - 1                                                          ELM233C
      IF(MCT.GT.0) GO TO 630                                                 ELM234C
      WRITE(ITP6,2004) HEAD,TIME,IPG                                         ELM235C
      IPG = IPG + 1                                                          ELM236C
      MCT = 16                                                               ELM237C
630   WRITE(ITP6,2003) N,(XX(I),I=1,2),((XIG(I,J),J=I,2),I=1,2),ST,          ELM238C
     X                 DM,MA,((DEPS(J+1,I),J=I,2),I=1,2),ET                  ELM239C
640   IF(NPL) CALL PLDATA(NUMEL,NDIM,N,LL,SHED,XX,XIG,FORCE)                 ELM240C
      IF(IPLTZD .NE. 1) GO TO 641
      IF(LL.NE.1) GO TO 641
      VECT(N,1)= XIG(1,1)
      VECT(N,2)= XIG(2,2)
      VECT(N,3)= ST
      VECT(N,4) = XIG(1,2)
      VECT(N,5)= DEPS(2,1)
      VECT(N,6)= DEPS(3,2)
      VECT(N,7)= ET
      VECT(N,8)= DEPS(3,1)
641   CONTINUE
      IF(ISW.EQ.4) GO TO 500
C.... COMPUTE THE LOADS FOR THE NEXT TIME STEP                               ELM242C
      DD = WW*RR                                                             ELM243C
      XIG(1,1) = XIG(2,1)*DD                                                 ELM244C
      XIG(1,2) = XIG(3,1)*DD                                                 ELM245C
      XIG(2,2) = XIG(3,2)*DD                                                 ELM246C
      XIG(2,1) = XIG(1,2)                                                    ELM247C
      STF = (STF/RR + YIK)*WW*XJAC                                           ELM248C
      WW = XJAC*DD*D(3,MA)                                                   ELM249C
      II = 0                                                                 ELM250C
      DO 470 I = 1,NEL                                                       ELM251C
      CC = SHAPE(4,I)                                                        ELM252C
      FORCE(II+1,1) = FORCE(II+1,1) - STF*CC                                 ELM253C
      EE = CC*WW                                                             ELM254C
      DO 460 J = 1,2                                                         ELM255C
      CC = 0.0                                                               ELM256C
      DO 450 K = 1,2                                                         ELM257C
450   CC = CC + SHAPE(K,I)*XIG(J,K)                                          ELM258C
460   FORCE(II+J,1) = FORCE(II+J,1) - CC -EE*DDL(J,I)                        ELM259C
470   II = II + NDF                                                          ELM260C
500   CONTINUE                                                               ELM261C
      RETURN                                                                 ELM262C
C.... ELEMENT LOAD COMPUTATION FOR BODY TYPE FORCES, ETC.                    ELM263C
5     CONTINUE                                                               ELM264C
      RETURN                                                                 ELM265C
4000  IPG = 0                                                                ELM266C
      RETURN                                                                 ELM267C
C.... FORMAT STATEMENTS                                                      ELM268C
1000  FORMAT(2I5,4F10.0)                                                     ELM269C
1001  FORMAT(8F10.0)                                                         ELM270C
2000  FORMAT(5X,57H LINEAR VISCOELASTIC MATERIAL, AXISYMMETRIC SOLID ELE     ELM271C
     XMENT  //I10,12H SHEAR TERMS/                                           ELM272C
     X I10,12H BULK  TERMS//10X,12H G-INFINITY=,1PE12.4/10X,                 ELM273C
     X 12H K-INFINITY=,1PE12.4/10X,12H DENSITY   =,1PE12.4/1X)               ELM274C
2001  FORMAT(10X,14H SHEAR MODULUS,6X,4HTIME/(10X,1P2E12.4))                 ELM275C
```

```
 2002  FORMAT(10X,14H BULK  MODULUS,6X,4HTIME/(10X,1P2E12.4))                 FLM276C
 2003  FORMAT(1H0,I7,0P2F12.3,1P4E12.3/8X,A5,I7,12X,1P4E12.3)                  FLM277C
 2004  FORMAT(1H1,12A6,E13.5,20X,4HPAGE,I3//5X,16HELEMENT STRESSES//          FLM278C
      1 1X,7HELEMENT,4X,6HR ORD.,6X,6HZ ORD.,5X,9HRR-STRESS,3X,              FLM279C
      2 9HRZ-STRESS,3X,9HZZ-STRESS,3X,9HTT-STRESS/35X,9HRR-STRAIN,3X,        FLM280C
      3 9HRZ-STRAIN,3X,9HZZ-STRAIN,3X,9HTT-STRAIN)                           FLM281C
 2030  FORMAT(9H MATERIAL,I3,17H EXCEEDS STORAGE.,I5,13H SHEAR TERMS.,I5,     FLM282C
      X 16H BULK TERMS NEED,I4,7H WORDS. )                                   FLM283C
       END                                                                   FLM284C
*WEOR
```

Appendix G

SAMPLE INPUT DATA

PRECEDING PAGE BLANK-NOT FILMED

```
50000
     FEAPTS                                                                    1
   1  653  667  671  671  652  696  703  704  651  695  713  715  647  694  712
   2  716  645  690  709  714  638  680  699  700  617  658  661  662  650  666
   3  668  669  649  693  701  702  648  692  707  710  646  691  706  711  644
   4  689  715  708  637  679  697  698  616  657  659  660  643  663  664  665
   5  642  676  681  682  641  675  685  687  640  674  684  688  639  673  683
   6  686  636  672  677  678  615  654  655  656  611  618  619  620  610  625
   7  628  629  609  624  632  634  608  623  631  635  607  622  630  633  606
   8  621  626  627  605  612  613  614  531  535  536  537  530  542  545  546
   9  529  541  549  551  528  540  548  552  527  539  547  550  526  538  543
  10  544  525  532  533  534  595  600  603  604  594  599  601  602  593  596
  11  597  598  589  590  591  592  521  522  523  524  572  571  566  557  516
  12  436  439  440  576  569  565  556  515  435  443  445  568  567  564  555
  13  514  434  442  446  563  562  561  554  513  433  441  444  560  559  558
  14  553  512  432  437  438  517  518  519  520  573  574  575  576  577  580
  15  581  582  578  583  585  586  579  584  587  588  502  507  510  511  501
  16  506  508  509  500  503  504  505  496  497  498  499  488  489  490  491
  17  495  494  493  492  487  429  430  431  461  460  459  452  449  425  345
  18  347  458  457  456  451  448  424  344  348  455  454  453  450  447  423
  19  343  346  470  469  468  467  462  426  427  428  463  464  465  466  471
  20  472  473  474  475  478  479  480  476  481  483  484  477  482  485  486
  21  413  418  421  422  412  417  419  420  411  414  415  416  407  408  409
  22  410  398  400  401  402  406  405  404  403  398  390  391  392  397  396
  23  395  394  393  389  341  342  354  353  352  351  350  349  338  258  363
  24  362  361  360  359  355  339  340  372  371  370  369  364  356  357  358
  25  365  366  367  368  373  374  375  376  377  380  381  382  378  383  385
  26  386  379  384  387  388  328  333  336  337  327  332  334  335  326  329
  27  330  331  322  323  324  325  314  315  316  317  321  320  319  318  313
  28  305  306  307  312  311  310  304  308  304  296  297  303  302  301  300
  29  299  298  295  257  269  268  267  266  265  261  259  260  278  277  276
  30  275  270  262  263  264  271  272  273  274  279  280  281  282  283  286
  31  287  288  284  289  291  292  285  290  293  294  247  252  255  256  246
  32  251  253  254  245  248  249  250  241  242  243  244  233  234  235  236
  33  240  239  238  237  232  224  225  226  231  230  229  228  227  223  215
  34  216  222  221  220  219  218  217  214  206  213  212  211  210  209  208
  35  207  205  188  187  186  185  180  177  178  179  181  182  183  184  189
  36  190  191  192  193  196  197  198  194  199  201  202  195  200  203  204
  37  153  154  155  156  152  144  145  146  147  143  135  136  138  137  134
  38  126  129  128  127  125  120  119  118  117   97   98   99  100  161  162
  39  163  164  157   77   78   79  148   71   63   64  139   65   62   56  130
  40   58   57   55  121   51   50   49  101  102  103  104  165  168  169  170
  41  158   82   87   88  149   72   33   34  140   66   22   18  131   59   19
  42   17  122   52   43   44  105  108  109  110  166  171  173  175  159   83
  43   84   93  150   73   35   37  141   67   23    7  132   60   20   15  123
  44   53   45   47  106  111  113  114  167  172  174  176  160   84   90   94
  45  151   74   36   38  142   68   24    8  133   61   21   16  124   54   46
  46   48  127  112  115  116   81   86   92   95   80   85   91   96   76   75
  47   70   69   30   32   40   41   29   31   39   42   28   27   26   25    6
  48   10   12   13    5    9   11   14    4    3    2    1
9999
    3
    3
    8                350
    6 MATER1
    1 ELM21     HARD BONE--OUTER TABLE
            335570.     487600.     .00027
    2 ELM21     POROUS BONE--DIPLOE
            33557.      48760.      .00002
    3 ELM21     HARD BONE--INNER TABLE
            335570.     487600.     .00027
    4 ELM22     SHEARLESS SOLID--SUBARACHNOID SPACE
            3.227       305.        .0000937
    5 ELM22     SHEARLESS SOLID--BRAIN
            3.227       305000.     .0000937
    6 ELM21     NECK COVER
            355.        487.        .00027
  716 NODAL
    1           0         2.226      -1.455       3.469     5
    2           0         2.503      -1.017       3.535     5
    3           0         2.634       -.581       3.593     5
    4           1         2.717      -0.000       3.587     5
```

5	0	1.600	-1.723	3.617	5
6	0	1.648	-1.297	3.982	5
7	0	1.691	-.751	4.218	5
8	1	1.690	-0.000	4.378	5
9	0	.792	-2.096	3.603	5
10	0	.775	-1.523	4.212	5
11	0	.780	-.878	4.581	5
12	1	.774	-0.000	4.692	5
13	0	.081	-2.228	3.876	5
14	0	-.033	-1.619	4.346	5
15	0	-.038	-.819	4.687	5
16	1	-.033	-0.000	4.928	5
17	0	-.815	-2.243	3.839	5
18	0	-.835	-1.610	4.366	5
19	0	-.827	-.772	4.734	5
20	1	-.834	-0.000	4.936	5
21	0	-1.514	-1.961	3.785	5
22	0	-1.597	-1.439	4.236	5
23	0	-1.608	-.761	4.589	5
24	1	-1.587	-0.000	4.694	5
25	0	-2.072	-1.579	3.680	5
26	0	-2.287	-1.101	3.825	5
27	0	-2.323	-.594	3.993	5
28	1	-2.311	0.000	4.083	5
29	0	2.204	-1.429	3.439	5
30	0	2.480	-1.000	3.504	5
31	0	2.611	-.573	3.564	5
32	1	2.695	-0.000	3.563	5
33	0	1.589	-1.692	3.582	5
34	0	1.623	-1.267	3.939	5
35	0	1.668	-.736	4.176	5
36	1	1.667	-0.000	4.336	5
37	0	.774	-2.057	3.765	5
38	0	.761	-1.497	4.172	5
39	0	.765	-.861	4.528	5
40	1	.760	-0.000	4.647	5
41	0	.078	-2.183	3.832	5
42	0	-.037	-1.589	4.297	5
43	0	-.041	-.801	4.641	5
44	1	-.040	-0.000	4.871	5
45	0	-.801	-2.190	3.792	5
46	0	-.826	-1.575	4.312	5
47	0	-.820	-.752	4.678	5
48	1	-.823	-0.000	4.875	5
49	0	-1.481	-1.909	3.744	5
50	0	-1.568	-1.404	4.190	4
51	0	-1.576	-.743	4.529	4
52	1	-1.553	-0.000	4.633	4
53	0	-2.029	-1.535	3.642	4
54	0	-2.242	-1.076	3.789	4
55	0	-2.275	-.583	3.955	4
56	1	-2.268	-0.000	4.045	4
57	0	2.161	-1.376	3.380	4
58	0	2.433	-.967	3.441	4
59	0	2.564	-.558	3.508	4
60	1	2.652	-0.000	3.514	4
61	0	1.540	-1.631	3.514	4
62	0	1.574	-1.209	3.853	4
63	0	1.623	-.705	4.092	4
64	1	1.622	-0.000	4.252	4
65	0	.738	-1.979	3.688	4
66	0	.734	-1.446	4.091	4
67	0	.735	-.827	4.423	4
68	1	.733	-0.000	4.556	4
69	0	.073	-2.094	3.744	4
70	0	-.046	-1.530	4.199	4
71	0	-.047	-.767	4.547	4
72	1	-.053	-0.000	4.757	4
73	0	-.774	-2.083	3.697	4
74	0	-.809	-1.505	4.204	4
75	0	-.807	-.711	4.566	4
76	1	-.801	-0.000	4.752	4

77	0	-1.414	-1.803	3.661	4
78	0	-1.509	-1.335	4.097	4
79	0	-1.512	-.708	4.410	4
80	1	-1.484	-0.000	4.511	4
81	0	-1.443	-1.445	3.567	4
82	0	-2.154	-1.025	3.716	4
83	0	-2.178	-.561	3.878	4
84	1	-2.183	-0.000	3.968	4
85	0	2.140	-1.350	3.350	4
86	0	2.410	-.950	3.410	4
87	0	2.540	-.550	3.480	4
88	1	2.630	-0.000	3.490	4
89	0	1.520	-1.600	3.480	4
90	0	1.550	-1.180	3.810	4
91	0	1.600	-.690	4.050	4
92	1	1.600	-0.000	4.210	4
93	0	.720	-1.940	3.650	4
94	0	.720	-1.420	4.050	4
95	0	.720	-.810	4.370	4
96	1	.720	-0.000	4.510	4
97	0	.070	-2.050	3.700	4
98	0	-.050	-1.500	4.150	4
99	0	-.050	-.750	4.500	3
100	1	-.050	-0.000	4.700	3
101	0	-.760	-2.030	3.650	3
102	0	-.800	-1.470	4.150	3
103	0	-.800	-.690	4.510	3
104	1	-.790	-0.000	4.690	3
105	0	-1.340	-1.750	3.620	3
106	0	-1.480	-1.300	4.050	3
107	0	-1.480	-.690	4.350	3
108	1	-1.450	-0.000	4.450	3
109	0	-1.900	-1.400	3.530	3
110	0	-2.110	-1.000	3.680	3
111	0	-2.130	-.550	3.840	3
112	1	-2.140	-0.000	3.930	3
113	0	2.119	-1.324	3.320	3
114	0	2.387	-.933	3.379	3
115	0	2.516	-.542	3.452	3
116	1	2.608	-0.000	3.466	3
117	0	1.500	-1.569	3.446	3
118	0	1.526	-1.151	3.767	3
119	0	1.577	-.675	4.008	3
120	1	1.578	-0.000	4.168	3
121	0	.702	-1.901	3.612	3
122	0	.706	-1.394	4.009	3
123	0	.705	-.793	4.317	3
124	1	.707	-0.000	4.464	3
125	0	.067	-2.006	3.656	3
126	0	-.054	-1.470	4.101	3
127	0	-.053	-.733	4.453	3
128	1	-.067	-0.000	4.643	3
129	0	-.746	-1.977	3.603	3
130	0	-.791	-1.435	4.096	3
131	0	-.793	-.669	4.454	3
132	1	-.779	-0.000	4.628	3
133	0	-1.346	-1.697	3.579	3
134	0	-1.451	-1.265	4.003	3
135	0	-1.448	-.672	4.290	3
136	1	-1.416	-0.000	4.389	3
137	0	-1.857	-1.355	3.493	3
138	0	-2.066	-.975	3.644	3
139	0	-2.082	-.539	3.802	3
140	1	-2.097	-0.000	3.892	3
141	0	2.317	-1.751	3.066	3
142	0	2.685	-1.191	3.166	3
143	0	2.932	-.535	3.184	3
144	1	3.019	-0.000	3.153	3
145	0	2.288	-1.718	3.042	3
146	0	2.656	-1.174	3.144	3
147	0	2.899	-.526	3.163	3
148	1	2.981	-0.000	3.132	2

149	0	2.229	-1.653	2.994	2
150	0	2.599	-1.138	3.101	2
151	0	2.833	-.509	3.121	2
152	1	2.907	-0.000	3.091	2
153	0	2.200	-1.620	2.970	2
154	0	2.570	-1.120	3.080	2
155	0	2.800	-.500	3.100	2
156	1	2.870	-0.000	3.070	2
157	0	2.171	-1.587	2.946	2
158	0	2.541	-1.102	3.059	2
159	0	2.767	-.491	3.079	2
160	1	2.833	-0.000	3.049	2
161	0	1.683	-2.194	3.111	2
162	0	1.662	-2.158	3.083	2
163	0	1.621	-2.086	3.028	2
164	0	1.600	-2.050	3.000	2
165	0	1.579	-2.014	2.972	2
166	0	1.674	-1.301	3.161	2
167	0	1.768	-.648	3.287	2
168	1	1.797	-0.000	3.337	2
169	0	.777	-2.589	3.121	2
170	0	.763	-2.547	3.098	2
171	0	.734	-2.462	3.053	2
172	0	.720	-2.420	3.030	2
173	0	.706	-2.378	3.007	2
174	0	.817	-1.479	3.266	2
175	0	.863	-.735	3.443	2
176	1	.877	-0.000	3.514	2
177	0	.073	-2.712	3.220	2
178	0	.072	-2.659	3.197	2
179	0	.071	-2.553	3.152	2
180	0	.070	-2.500	3.130	2
181	0	.069	-2.447	3.108	2
182	0	.039	-1.531	3.335	2
183	0	.040	-.745	3.523	2
184	1	.040	-0.000	3.604	2
185	0	-.791	-2.622	3.217	2
186	0	-.776	-2.567	3.193	2
187	0	-.745	-2.456	3.144	2
188	0	-.730	-2.400	3.120	2
189	0	-.715	-2.344	3.096	2
190	0	-.732	-1.476	3.334	2
191	0	-.750	-.715	3.516	2
192	1	-.755	-0.000	3.593	2
193	0	-1.504	-2.328	3.159	2
194	0	-1.473	-2.279	3.137	2
195	0	-1.411	-2.180	3.092	2
196	0	-1.380	-2.130	3.070	2
197	0	-1.349	-2.080	3.048	1
198	0	-1.470	-1.315	3.271	1
199	0	-1.522	-.665	3.413	1
200	1	-1.543	-0.000	3.475	1
201	0	-1.934	-1.606	3.028	1
202	0	-2.274	-1.054	3.151	1
203	0	-2.339	-.598	3.184	1
204	1	-2.422	-0.000	3.276	1
205	0	-1.980	-1.650	3.050	1
206	0	-2.330	-1.080	3.170	1
207	0	-2.400	-.610	3.210	1
208	1	-2.480	-0.000	3.300	1
209	0	-2.026	-1.694	3.072	1
210	0	-2.386	-1.106	3.189	1
211	0	-2.461	-.622	3.236	1
212	1	-2.538	-0.000	3.324	1
213	0	-2.119	-1.782	3.116	1
214	0	-2.499	-1.157	3.228	1
215	0	-2.584	-.645	3.288	1
216	1	-2.653	-0.000	3.373	1
217	0	-2.165	-1.826	3.139	1
218	0	-2.556	-1.183	3.247	1
219	0	-2.645	-.656	3.313	1
220	1	-2.711	-0.000	3.397	1

221	0	2.409	-2.008	2.582	1
222	0	2.915	-1.380	2.676	1
223	0	3.181	-.630	2.704	1
224	1	3.307	-0.000	2.682	1
225	0	2.377	-1.969	2.562	1
226	0	2.886	-1.360	2.657	1
227	0	3.151	-.623	2.688	1
228	1	3.260	-0.000	2.664	1
229	0	2.312	-1.890	2.521	1
230	0	2.829	-1.320	2.619	1
231	0	3.090	-.608	2.656	1
232	1	3.167	-0.000	2.628	1
233	0	2.280	-1.850	2.500	1
234	0	2.800	-1.300	2.600	1
235	0	3.060	-.600	2.640	1
236	1	3.120	-0.000	2.610	1
237	0	2.248	-1.810	2.479	1
238	0	2.771	-1.280	2.581	1
239	0	3.036	-.592	2.624	1
240	1	3.073	-0.000	2.592	1
241	0	1.771	-2.400	2.538	1
242	0	1.753	-2.362	2.529	1
243	0	1.718	-2.287	2.510	1
244	0	1.700	-2.250	2.500	1
245	0	1.682	-2.213	2.490	1
246	0	1.811	-1.413	2.613	5
247	0	1.929	-.686	2.694	5
248	1	1.962	-0.000	2.715	5
249	0	.854	-2.737	2.448	5
250	0	.846	-2.690	2.448	5
251	0	.829	-2.597	2.449	5
252	0	.820	-2.550	2.450	5
253	0	.811	-2.503	2.451	4
254	0	.913	-1.538	2.642	4
255	0	.973	-.744	2.753	4
256	1	.990	-0.000	2.792	4
257	0	.065	-2.800	2.520	4
258	0	.066	-2.737	2.520	4
259	0	.069	-2.612	2.520	4
260	0	.070	-2.550	2.520	3
261	0	.071	-2.488	2.520	3
262	0	.091	-1.565	2.679	3
263	0	.102	-.758	2.786	3
264	1	.104	-0.000	2.826	3
265	0	-.733	-2.754	2.597	3
266	0	-.719	-2.695	2.593	3
267	0	-.693	-2.578	2.584	2
268	0	-.680	-2.520	2.580	2
269	0	-.667	-2.462	2.576	2
270	0	-.703	-1.518	2.692	2
271	0	-.735	-.736	2.782	2
272	1	-.747	-0.000	2.817	2
273	0	-1.490	-2.422	2.576	2
274	0	-1.463	-2.371	2.575	1
275	0	-1.408	-2.270	2.572	1
276	0	-1.380	-2.220	2.570	1
277	0	-1.352	-2.170	2.568	1
278	0	-1.493	-1.352	2.674	1
279	0	-1.585	-.686	2.742	1
280	1	-1.618	-0.000	2.765	1
281	0	-1.934	-1.768	2.544	5
282	0	-2.345	-1.064	2.661	4
283	0	-2.518	-.675	2.689	3
284	1	-2.585	-0.000	2.669	2
285	0	-1.980	-1.810	2.550	1
286	0	-2.410	-1.100	2.670	0
287	0	-2.580	-.690	2.710	0
288	1	-2.660	-0.000	2.690	0
289	0	-2.026	-1.852	2.556	0
290	0	-2.475	-1.136	2.679	0
291	0	-2.642	-.705	2.731	1
292	1	-2.735	-0.000	2.711	2

293	0	-2.117	-1.937	2.568	3
294	0	-2.615	-1.208	2.697	4
295	0	-2.767	-.734	2.773	5
296	1	-2.884	-0.000	2.753	5
297	0	-2.162	-1.980	2.574	4
298	0	-2.670	-1.245	2.705	3
299	0	-2.830	-.748	2.795	2
300	1	-2.958	-0.000	2.774	1
301	0	2.415	-2.157	2.101	0
302	0	3.064	-1.546	2.197	0
303	0	3.374	-.723	2.191	0
304	1	3.438	-0.000	2.187	0
305	0	2.341	-2.118	2.088	0
306	0	3.036	-1.524	2.185	1
307	0	3.343	-.717	2.183	2
308	1	3.384	-0.000	2.180	3
309	0	2.344	-2.039	2.063	4
310	0	2.979	-1.481	2.162	5
311	0	3.281	-.706	2.168	5
312	1	3.275	-0.000	2.167	4
313	0	2.320	-2.000	2.050	3
314	0	2.950	-1.460	2.150	2
315	0	3.250	-.700	2.160	1
316	1	3.220	-0.000	2.160	0
317	0	2.296	-1.961	2.037	0
318	0	2.921	-1.439	2.138	0
319	0	3.219	-.694	2.152	0
320	1	3.165	-0.000	2.153	0
321	0	1.766	-2.424	1.991	1
322	0	1.755	-2.380	1.993	2
323	0	1.732	-2.293	1.998	3
324	0	1.720	-2.250	2.000	4
325	0	1.708	-2.207	2.002	5
326	0	1.897	-1.460	2.113	5
327	0	2.031	-.719	2.169	4
328	1	2.051	-0.000	2.185	3
329	0	.898	-2.563	1.868	2
330	0	.891	-2.517	1.878	1
331	0	.877	-2.426	1.899	0
332	0	.870	-2.380	1.910	0
333	0	.863	-2.334	1.921	0
334	0	.977	-1.523	2.089	0
335	0	1.039	-.748	2.164	0
336	1	1.054	-0.000	2.188	1
337	0	.089	-2.749	1.979	2
338	0	.084	-2.687	1.982	3
339	0	.075	-2.562	1.987	4
340	0	.070	-2.500	1.990	5
341	0	.065	-2.438	1.993	5
342	0	.125	-1.558	2.098	4
343	0	.137	-.758	2.153	3
344	1	.137	-0.000	2.171	2
345	0	-.690	-2.752	2.030	1
346	0	-.675	-2.689	2.035	0
347	0	-.645	-2.563	2.045	0
348	0	-.630	-2.500	2.050	0
349	0	-.615	-2.437	2.055	0
350	0	-.681	-1.518	2.108	0
351	0	-.729	-.739	2.138	1
352	1	-.744	-0.000	2.152	2
353	0	-1.494	-2.381	2.043	3
354	0	-1.465	-2.326	2.050	4
355	0	-1.408	-2.215	2.063	5
356	0	-1.380	-2.160	2.070	1
357	0	-1.352	-2.105	2.077	1
358	0	-1.503	-1.357	2.108	1
359	0	-1.625	-.690	2.130	1
360	1	-1.661	-0.000	2.147	1
361	0	-1.940	-1.759	2.014	1
362	0	-2.369	-1.107	2.141	1
363	0	-2.655	-.666	2.169	2
364	1	-2.718	-0.000	2.236	2

365	0	-1.980	-1.800	2.010	2
366	0	-2.430	-1.150	2.150	2
367	0	-2.730	-.690	2.180	2
368	1	-2.780	-0.000	2.240	2
369	0	-2.020	-1.841	2.006	2
370	0	-2.491	-1.193	2.159	3
371	0	-2.805	-.714	2.191	3
372	1	-2.842	-0.000	2.244	3
373	0	-2.100	-1.923	1.997	3
374	0	-2.513	-1.278	2.177	3
375	0	-2.956	-.762	2.214	3
376	1	-2.967	-0.000	2.251	3
377	0	-2.140	-1.965	1.993	4
378	0	-2.674	-1.321	2.186	4
379	0	-3.032	-.785	2.228	4
380	1	-3.030	-0.000	2.254	4
381	0	2.475	-2.163	1.680	4
382	0	3.155	-1.581	1.685	4
383	0	3.419	-.719	1.731	4
384	1	3.470	-0.000	1.790	5
385	0	2.459	-2.133	1.678	5
386	0	3.124	-1.561	1.681	5
387	0	3.382	-.714	1.731	5
388	1	3.407	-0.000	1.790	5
389	0	2.426	-2.071	1.673	5
390	0	3.061	-1.520	1.674	5
391	0	3.307	-.705	1.730	5
392	1	3.282	-0.000	1.790	5
393	0	2.410	-2.040	1.670	5
394	0	3.030	-1.500	1.670	5
395	0	3.270	-.700	1.730	5
396	1	3.220	-0.000	1.790	5
397	0	2.394	-2.009	1.667	5
398	0	2.999	-1.480	1.666	4
399	0	3.233	-.695	1.730	4
400	1	3.157	-0.000	1.790	4
401	0	1.758	-2.367	1.554	4
402	0	1.748	-2.325	1.568	4
403	0	1.729	-2.242	1.596	4
404	0	1.720	-2.200	1.610	4
405	0	1.711	-2.158	1.624	3
406	0	1.935	-1.458	1.668	3
407	0	2.049	-.726	1.703	3
408	1	2.062	-0.000	1.717	3
409	0	.983	-2.526	1.447	3
410	0	.972	-2.469	1.448	3
411	0	.951	-2.356	1.449	3
412	0	.940	-2.300	1.450	2
413	0	.929	-2.244	1.451	2
414	0	1.025	-1.501	1.596	2
415	0	1.061	-.744	1.635	2
416	1	1.067	-0.000	1.650	2
417	0	.164	-2.749	1.500	2
418	0	.158	-2.687	1.500	2
419	0	.146	-2.562	1.500	1
420	0	.140	-2.500	1.500	1
421	0	.134	-2.438	1.500	1
422	0	.164	-1.544	1.568	1
423	0	.145	-.746	1.560	1
424	1	.138	-0.000	1.556	1
425	0	-.695	-2.689	1.538	1
426	0	-.679	-2.629	1.546	5
427	0	-.646	-2.510	1.562	4
428	0	-.630	-2.450	1.570	3
429	0	-.614	-2.390	1.578	2
430	0	-.664	-1.502	1.554	1
431	0	-.722	-.732	1.507	0
432	1	-.737	-0.000	1.499	0
433	0	-1.476	-2.288	1.569	0
434	0	-1.452	-2.239	1.574	0
435	0	-1.404	-2.146	1.585	0
436	0	-1.380	-2.100	1.590	1

437	0	-1.356	-2.054	1.595	2
438	0	-1.495	-1.369	1.520	3
439	0	-1.621	-.689	1.475	4
440	1	-1.633	-0.000	1.471	5
441	0	-1.834	-1.741	1.531	5
442	0	-2.429	-1.210	1.499	4
443	0	-2.719	-.638	1.478	3
444	1	-2.628	-0.000	1.506	2
445	0	-1.880	-1.800	1.530	1
446	0	-2.480	-1.250	1.500	0
447	0	-2.780	-.650	1.470	0
448	1	-2.730	-0.000	1.480	0
449	0	-1.926	-1.859	1.529	0
450	0	-2.531	-1.290	1.501	0
451	0	-2.841	-.662	1.462	1
452	1	-2.832	-0.000	1.454	2
453	0	-2.017	-1.978	1.526	3
454	0	-2.634	-1.370	1.502	4
455	0	-2.962	-.687	1.447	5
456	1	-3.035	-0.000	1.403	5
457	0	-2.063	-2.038	1.524	4
458	0	-2.685	-1.410	1.503	3
459	0	-3.023	-.699	1.440	2
460	111	-3.137	-0.000	1.378	1
461	0	2.475	-2.134	1.369	0
462	0	3.112	-1.568	1.198	0
463	0	3.505	-.784	1.113	0
464	111	3.638	-0.000	1.182	0
465	0	2.461	-2.100	1.379	0
466	0	3.089	-1.551	1.236	1
467	0	3.441	-.776	1.172	2
468	1	3.533	-0.000	1.224	3
469	0	2.434	-2.033	1.400	4
470	0	3.043	-1.517	1.312	5
471	0	3.314	-.759	1.291	5
472	1	3.324	-0.000	1.308	4
473	0	2.420	-2.000	1.410	3
474	0	3.020	-1.500	1.350	2
475	0	3.250	-.750	1.350	1
476	1	3.220	-0.000	1.350	0
477	0	2.406	-1.967	1.420	0
478	0	2.997	-1.483	1.388	0
479	0	3.186	-.741	1.409	0
480	1	3.116	-0.000	1.392	0
481	0	1.779	-2.184	1.212	1
482	0	1.764	-2.143	1.237	2
483	0	1.735	-2.061	1.286	3
484	0	1.720	-2.020	1.310	4
485	0	1.705	-1.979	1.334	5
486	0	1.927	-1.424	1.309	5
487	0	1.974	-.739	1.297	4
488	1	1.998	-0.000	1.271	3
489	0	1.015	-2.502	1.114	2
490	0	1.005	-2.464	1.118	1
491	0	.982	-2.388	1.126	0
492	0	.970	-2.350	1.130	0
493	0	.958	-2.312	1.134	0
494	0	1.042	-1.497	1.162	0
495	0	1.040	-.742	1.139	0
496	1	1.030	-0.000	1.166	1
497	0	.070	-2.658	1.104	2
498	0	.070	-2.618	1.110	3
499	0	.070	-2.539	1.123	4
500	0	.070	-2.500	1.130	5
501	0	.070	-2.461	1.137	1
502	0	.217	-1.517	1.099	1
503	0	.092	-.702	.941	1
504	1	.070	-0.000	.899	1
505	0	-.802	-2.585	1.018	1
506	0	-.784	-2.526	1.046	1
507	0	-.768	-2.409	1.102	1
508	0	-.730	-2.350	1.130	2

509	0	-.712	-2.291	1.158	2
510	0	-.633	-1.459	1.042	2
511	0	-.724	-.714	.814	2
512	1	-.740	-0.000	.803	2
513	0	-1.485	-2.298	1.106	2
514	0	-1.459	-2.236	1.110	2
515	0	-1.406	-2.112	1.117	3
516	0	-1.380	-2.050	1.120	3
517	0	-1.354	-1.988	1.123	3
518	0	-1.398	-1.405	.890	3
519	0	-1.534	-.709	.743	3
520	1	-1.532	-0.000	.723	3
521	0	-1.837	-1.795	.920	3
522	0	-2.269	-1.260	.786	4
523	0	-2.461	-.745	.688	4
524	1	-2.271	-0.000	.705	4
525	0	-1.880	-1.850	.920	4
526	0	-2.330	-1.300	.770	4
527	0	-2.530	-.750	.660	4
528	1	-2.380	-0.000	.650	4
529	0	-1.923	-1.905	.920	5
530	0	-2.391	-1.340	.754	5
531	0	-2.599	-.755	.632	5
532	1	-2.489	-0.000	.595	5
533	0	-2.010	-2.015	.920	5
534	0	-2.514	-1.420	.721	5
535	0	-2.738	-.766	.575	5
536	1	-2.708	-0.000	.484	5
537	0	-2.054	-2.070	.920	5
538	0	-2.575	-1.460	.704	5
539	0	-2.807	-.771	.547	5
540	1	-2.817	-0.000	.429	5
541	0	2.402	-1.869	1.197	5
542	0	2.4769	-1.2446	1.08	5
543	0	2.639	-.737	1.104	4
544	1	2.818	-0.000	.938	4
545	0	1.805	-1.825	1.111	4
546	0	1.870	-1.391	1.006	4
547	0	1.645	-.800	.950	4
548	1	1.834	-0.000	.760	4
549	0	1.007	-2.175	.782	4
550	0	1.326	-1.483	.695	3
551	0	.999	-.771	.636	3
552	1	.963	-0.000	.896	3
553	0	.004	-2.356	.827	3
554	0	.528	-1.440	.744	3
555	0	-.134	-.492	.134	3
556	1	-.194	-0.000	-.015	3
557	0	-.853	-2.008	.889	2
558	0	-.519	-1.323	.740	2
559	0	-.809	-.683	-.152	2
560	1	-.794	-0.000	.069	2
561	0	-1.474	-1.937	.573	2
562	0	-1.102	-1.646	.122	2
563	0	-1.468	-.698	-.130	2
564	1	-1.477	-0.000	-.128	1
565	0	-1.843	-1.755	.542	1
566	0	-2.060	-1.261	.105	1
567	0	-2.117	-.700	-.063	1
568	1	-1.936	-0.000	.044	1
569	0	2.426	-1.880	1.160	1
570	0	2.490	-1.250	1.050	1
571	0	2.650	-.750	1.070	5
572	1	2.850	-0.000	.870	4
573	0	1.820	-1.850	1.080	3
574	0	1.920	-1.400	.920	2
575	0	1.670	-.810	.880	1
576	1	1.840	-0.000	.700	0
577	0	1.020	-2.200	.750	0
578	0	1.350	-1.500	.600	0
579	0	1.060	-.750	.500	0
580	1	1.040	-0.000	.690	0

581	0	0.000	-2.380	.780	1
582	0	.520	-1.500	.580	2
583	0	-.180	-.500	.100	3
584	1	-.180	-0.000	-.050	4
585	0	-.900	-2.000	.800	5
586	0	-.480	-1.400	.650	5
587	0	-.780	-.720	-.240	4
588	1	-.780	-0.000	-.030	3
589	0	-1.480	-2.000	.550	2
590	0	-1.080	-1.690	.050	1
591	0	-1.480	-.720	-.240	0
592	1	-1.480	-0.000	-.240	0
593	0	-1.880	-1.800	.520	0
594	0	-2.080	-1.280	.080	0
595	0	-2.140	-.700	-.090	0
596	1	-1.980	-0.000	0.000	1
597	0	2.438	-1.891	1.123	2
598	0	2.503	-1.255	1.02	3
599	0	2.661	-.763	1.036	4
600	1	2.882	-0.000	.802	5
601	0	1.835	-1.875	1.049	5
602	0	1.970	-1.409	.834	4
603	0	1.695	-.820	.810	3
604	1	1.846	-0.000	.640	2
605	0	1.033	-2.225	.718	1
606	0	1.374	-1.517	.505	0
607	0	1.121	-.729	.364	0
608	1	1.117	-0.000	.484	0
609	0	-.004	-2.404	.733	0
610	0	.512	-1.560	.416	0
611	0	-.166	-.508	.066	1
612	1	-.166	-0.000	-.085	2
613	0	-.898	-2.046	.711	3
614	0	-.441	-1.477	.560	4
615	0	-.751	-.757	-.328	5
616	1	-.766	-0.000	-.129	5
617	0	-1.486	-2.063	.527	4
618	0	-1.058	-1.734	-.022	3
619	0	-1.492	-.742	-.350	2
620	1	-1.483	-0.000	-.352	1
621	0	-1.917	-1.845	.498	0
622	0	-2.100	-1.299	.055	0
623	0	-2.163	-.700	-.117	0
624	1	-2.024	-0.000	-.044	0
625	0	2.473	-1.912	1.049	0
626	0	2.529	-1.266	0.95	1
627	0	2.682	-.788	.969	2
628	1	2.946	-0.000	.666	3
629	0	1.864	-2.024	.986	4
630	0	2.069	-1.428	.661	5
631	0	1.745	-.839	.670	5
632	1	1.858	-0.000	.521	4
633	0	1.058	-2.276	.655	3
634	0	1.423	-1.350	.314	2
635	0	1.242	-.688	.093	1
636	1	1.270	-0.000	.071	0
637	0	-.011	-2.452	.640	0
638	0	.497	-1.680	.088	0
639	0	-.139	-.524	-.002	0
640	1	-.139	-0.000	-.155	0
641	0	-.894	-2.137	.533	1
642	0	-.363	-1.631	.379	2
643	0	-.692	-.830	-.505	3
644	1	-.737	-0.000	-.327	4
645	0	-1.499	-2.189	.481	5
646	0	-1.013	-1.823	-.166	1
647	0	-1.517	-.785	-.569	1
648	1	-1.488	-0.000	-.577	1
649	0	-1.992	-1.936	.454	1
650	0	-2.141	-1.336	.004	1
651	0	-2.208	-.700	-.170	1
652	1	-2.112	-0.000	-.133	1

```
653          0          2.491     -1.922      1.011     2
654          0          2.542     -1.272       0.90     2
655          0          2.692      -.801       .935     2
656          1          2.978     -0.000       .599     2
657          0          1.879     -2.048       .954     2
658          0           2.11    -1.4377      0.600     2
659          0          1.776      -.848       .600     2
660          1          1.864     -0.000       .461     3
661          0          1.970     -2.302       .623     3
662          0          1.448     -1.370       .218     3
663          0          1.303      -.667      -.043     3
664          1          1.346     -0.000      -.135     3
665          0           -.015    -2.476       .594     3
666          0           .489     -1.740      -.077     3
667          0           -.126      -.532      -.036     4
668          1           -.125     -0.000      -.189     4
669          0           -.891     -2.182       .444     4
670          0           -.324     -1.708       .289     4
671          0           -.663      -.866      -.593     4
672          1           -.722     -0.000      -.426     4
673          0          -1.505     -2.253       .458     4
674          0           -.991     -1.868      -.238     5
675          0          -1.530      -.807      -.679     5
676          1          -1.491     -0.000      -.690     5
677          0          -2.029     -1.981       .033     5
678          0          -2.162     -1.355      -.021     5
679          0          -2.231      -.700      -.196     5
680          1          -2.156     -0.000      -.178     5
681          0           2.820     -1.900      -.350     5
682          0           3.370     -1.500      -.500     5
683          0           3.420      -.800      -.350     5
684          1           3.520     -0.000      -.250     5
685          0           2.050     -2.000      -.350     5
686          0           2.020      -.700      -.350     5
687          0           2.820      -.300      -.250     5
688          1           3.220     -0.000      -.350     4
689          0           1.520     -2.200      -.450     4
690          0           1.520     -1.750      -.250     4
691          0            .420     -2.500      -.350     4
692          0            .620     -2.000      -.350     4
693          0           2.125     -1.548     -1.608     4
694          0           2.870     -1.000     -1.900     4
695          0           3.370      -.750     -2.000     3
696          1           3.620     -0.000     -2.000     3
697          0           1.820     -1.700     -1.500     3
698          0           1.820      -.700     -1.750     3
699          0           2.620      -.500     -1.750     3
700          1           3.120     -0.000     -2.000     3
701          0           1.320     -1.700     -1.450     3
702          0           1.320     -1.100     -1.450     2
703          0            .420     -1.750     -1.400     2
704          0            .420     -1.500     -1.400     2
705          0           1.620     -1.500     -3.250     2
706          0           2.220     -1.200     -3.650     2
707          0           2.820      -.700     -3.750     2
708          1           3.020     -0.000     -3.950     2
709          0           1.420     -1.600     -3.050     1
710          0           1.620     -1.000     -3.350     1
711          0           2.320      -.700     -3.750     1
712          1           2.520     -0.000     -3.950     1
713          0            .920     -1.600     -2.750     1
714          0            .920     -1.300     -2.750     1
715          0            .320     -1.600     -2.250     1
716          0            .320     -1.450     -2.250     5
555 ELEMEN
  1    1
 1   5    6    2   29   33   34   30
  2    1
 2   6    7    3   30   34   35   31
  3    1
 3   7    8    4   31   35   36   32
  4    1
```

```
 5   9  10   6  33  37  38  34
  5      1
 6  10  11   7  34  38  39  35
  6      1
 7  11  12   8  35  39  40  36
  7      1
 9  13  14  10  37  41  42  38
  8      1
10  14  15  11  38  42  43  39
  9      1
11  15  16  12  39  43  44  40
 10      1
13  17  18  14  41  45  46  42
 11      1
14  18  19  15  42  46  47  43
 12      1
15  19  20  16  43  47  48  44
 13      1
17  21  22  18  45  49  50  46
 14      1
18  22  23  19  46  50  51  47
 15      1
19  23  24  20  47  51  52  48
 16      1
21  25  26  22  49  53  54  50
 17      1
22  26  27  23  50  54  55  51
 18      1
23  27  28  24  51  55  56  52
 19      2
29  33  34  30  57  61  62  58
 20      2
30  34  35  31  58  62  63  59
 21      2
31  35  36  32  59  63  64  60
 22      2
33  37  38  34  61  65  66  62
 23      2
34  38  39  35  62  66  67  63
 24      2
35  39  40  36  63  67  68  64
 25      2
37  41  42  38  65  69  70  66
 26      2
38  42  43  39  66  70  71  67
 27      2
39  43  44  40  67  71  72  68
 28      2
41  45  46  42  69  73  74  70
 29      2
42  46  47  43  70  74  75  71
 30      2
43  47  48  44  71  75  76  72
 31      2
45  49  50  46  73  77  78  74
 32      2
46  50  51  47  74  78  79  75
 33      2
47  51  52  48  75  79  80  76
 34      2
49  53  54  50  77  81  82  78
 35      2
50  54  55  51  78  82  83  79
 36      2
51  55  56  52  79  83  84  80
 37      3
57  61  62  58  85  89  90  86
 38      3
58  62  63  59  86  90  91  87
 39      3
59  63  64  60  87  91  92  88
 40      3
```

```
61  65  66  62  89  93  94  90
 41     3
62  66  67  63  90  94  95  91
 42     3
63  67  68  64  91  95  96  92
 43     3
65  69  70  66  93  97  98  94
 44     3
66  70  71  67  94  98  99  95
 45     3
67  71  72  68  95  99 100  96
 46     3
69  73  74  70  97 101 102  98
 47     3
70  74  75  71  98 102 103  99
 48     3
71  75  76  72  99 103 104 100
 49     3
73  77  78  74 101 105 106 102
 50     3
74  78  79  75 102 106 107 103
 51     3
75  79  80  76 103 107 108 104
 52     3
77  81  82  78 105 109 110 106
 53     3
78  82  83  79 106 110 111 107
 54     3
79  83  84  80 107 111 112 108
 55     4
85  89  90  86 113 117 118 114
 56     4
86  90  91  87 114 118 119 115
 57     4
87  91  92  88 115 119 120 116
 58     4
89  93  94  90 117 121 122 118
 59     4
90  94  95  91 118 122 123 119
 60     4
91  95  96  92 119 123 124 120
 61     4
93  97  98  94 121 125 126 122
 62     4
94  98  99  95 122 126 127 123
 63     4
95  99 100  96 123 127 128 124
 64     4
97 101 102  98 125 129 130 126
 65     4
98 102 103  99 126 130 131 127
 66     4
99 103 104 100 127 131 132 128
 67     4
101 105 106 102 129 133 134 130
 68     4
102 106 107 103 130 134 135 131
 69     4
103 107 108 104 131 135 136 132
 70     4
105 109 110 106 133 137 138 134
 71     4
106 110 111 107 134 138 139 135
 72     4
107 111 112 108 135 139 140 136
 73     1
  1  29  30   2 141 145 146 142
 74     1
  2  30  31   3 142 146 147 143
 75     1
  3  31  32   4 143 147 148 144
 76     2
```

```
29  57  58  30 145 149 150 146
 77     2
30  58  59  31 146 150 151 147
 78     2
31  59  60  32 147 151 152 148
 79     3
57  85  86  58 149 153 154 150
 80     3
58  86  87  59 150 154 155 151
 81     3
59  87  88  60 151 155 156 152
 82     4
85 113 114  86 153 157 158 154
 83     4
86 114 115  87 154 158 159 155
 84     4
87 115 116  88 155 159 160 156
 85     1
 1   5  33  29 141 161 162 145
 86     2
29  33  61  57 145 162 163 149
 87     3
57  61  89  85 149 163 164 153
 88     4
85  89 117 113 153 164 165 157
 89     1
 5   9  37  33 161 169 170 162
 90     2
33  37  65  61 162 170 171 163
 91     3
61  65  93  89 163 171 172 164
 92     4
89  93 121 117 164 172 173 165
 93     1
 9  13  41  37 169 177 178 170
 94     2
37  41  69  65 170 178 179 171
 95     3
65  69  97  93 171 179 180 172
 96     4
93  97 125 121 172 180 181 173
 97     1
13  17  45  41 177 185 186 178
 98     2
41  45  73  69 178 186 187 179
 99     3
69  73 101  97 179 187 188 180
100     4
97 101 129 125 180 188 189 181
101     1
17  21  49  45 185 193 194 186
102     2
45  49  77  73 186 194 195 187
103     3
73  77 105 101 187 195 196 188
104     4
101 105 133 129 188 196 197 189
105     1
21  25  53  49 193 217 213 194
106     2
49  53  81  77 194 213 209 195
107     3
77  81 109 105 195 209 205 196
108     4
105 109 137 133 196 205 201 197
109     5
113 117 118 114 157 165 166 158
110     5
114 118 119 115 158 166 167 159
111     5
115 119 120 116 159 167 168 160
112     5
```

```
117 121 122 118 165 173 174 166
 113     5
118 122 123 119 166 174 175 167
 114     5
119 123 124 120 167 175 176 168
 115     5
121 125 126 122 173 181 182 174
 116     5
122 126 127 123 174 182 183 175
 117     5
123 127 128 124 175 183 184 176
 118     5
125 129 130 126 181 189 190 182
 119     5
126 130 131 127 182 190 191 183
 120     5
127 131 132 128 183 191 192 184
 121     5
129 133 134 130 189 197 198 190
 122     5
130 134 135 131 190 198 199 191
 123     5
131 135 136 132 191 199 200 192
 124     5
133 137 138 134 197 201 202 198
 125     5
134 138 139 135 198 202 203 199
 126     5
135 139 140 136 199 203 204 200
 127     4
137 109 110 138 201 205 206 202
 128     4
138 110 111 139 202 206 207 203
 129     4
139 111 112 140 203 207 208 204
 130     3
109  81  82 110 205 209 210 206
 131     3
110  82  83 111 206 210 211 207
 132     3
111  83  84 112 207 211 212 208
 133     2
 81  53  54  82 209 213 214 210
 134     2
 82  54  55  83 210 214 215 211
 135     2
 83  55  56  84 211 215 216 212
 136     1
 53  25  26  54 213 217 218 214
 137     1
 54  26  27  55 214 218 219 215
 138     1
 55  27  28  56 215 219 220 216
 139     1
141 145 146 142 221 225 226 222
 140     1
142 146 147 143 222 226 227 223
 141     1
143 147 148 144 223 227 228 224
 142     2
145 149 150 146 225 229 230 226
 143     2
146 150 151 147 226 230 231 227
 144     2
147 151 152 148 227 231 232 228
 145     3
149 153 154 150 229 233 234 230
 146     3
150 154 155 151 230 234 235 231
 147     3
151 155 156 152 231 235 236 232
 148     4
```

153 157 158 154 233 237 238 234
149 4
154 158 159 155 234 238 239 235
150 4
155 159 160 156 235 239 240 236
151 1
141 161 162 145 221 241 242 225
152 2
145 162 163 149 225 242 243 229
153 3
149 163 164 153 229 243 244 233
154 4
153 164 165 157 233 244 245 237
155 1
161 169 170 162 241 249 250 242
156 2
162 170 171 163 242 250 251 243
157 3
163 171 172 164 243 251 252 244
158 4
164 172 173 165 244 252 253 245
159 1
169 177 178 170 249 257 258 250
160 2
170 178 179 171 250 258 259 251
161 3
171 179 180 172 251 259 260 252
162 4
172 180 181 173 252 260 261 253
163 1
177 185 186 178 257 265 266 258
164 2
178 186 187 179 258 266 267 259
165 3
179 187 188 180 259 267 268 260
166 4
180 188 189 181 260 268 269 261
167 1
185 193 194 186 265 273 274 266
168 2
186 194 195 187 266 274 275 267
169 3
187 195 196 188 267 275 276 268
170 4
188 196 197 189 268 276 277 269
171 1
193 217 213 194 273 297 293 274
172 2
194 213 209 195 274 293 289 275
173 3
195 209 205 196 275 289 285 276
174 4
196 205 201 197 276 285 281 277
175 5
157 165 166 158 237 245 246 238
176 5
158 166 167 159 238 246 247 239
177 5
159 167 168 160 239 247 248 240
178 5
165 173 174 166 245 253 254 246
179 5
166 174 175 167 246 254 255 247
180 5
167 175 176 168 247 255 256 248
181 5
173 181 182 174 253 261 262 254
182 5
174 182 183 175 254 262 263 255
183 5
175 183 184 176 255 263 264 256
184 5

181 189 191 182 261 269 270 262
185 5
182 190 191 183 262 270 271 263
186 5
183 191 192 184 263 271 272 264
187 5
189 197 198 190 269 277 278 270
188 5
190 198 199 191 270 278 279 271
189 5
191 199 200 192 271 279 280 272
190 5
197 201 202 198 277 281 282 278
191 5
198 202 203 199 278 282 283 279
192 5
199 203 204 200 279 283 284 280
193 4
201 205 206 202 281 285 286 282
194 4
202 206 207 203 282 286 287 283
195 4
203 207 208 204 283 287 288 284
196 3
205 209 210 206 285 289 290 286
197 3
206 210 211 207 286 290 291 287
198 3
207 211 212 208 287 291 292 288
199 2
209 213 214 210 289 293 294 290
200 2
210 214 215 211 290 294 295 291
201 2
211 215 216 212 291 295 296 292
202 1
213 217 218 214 293 297 298 294
203 1
214 218 219 215 294 298 299 295
204 1
215 219 220 216 295 299 300 296
205 1
221 225 226 222 301 305 306 302
206 1
222 226 227 223 302 306 307 303
207 1
223 227 228 224 303 307 308 304
208 2
225 229 230 226 305 309 310 306
209 2
226 230 231 227 306 310 311 307
210 2
227 231 232 228 307 311 312 308
211 3
229 233 234 230 309 313 314 310
212 3
230 234 235 231 310 314 315 311
213 3
231 235 236 232 311 315 316 312
214 4
233 237 238 234 313 317 318 314
215 4
234 238 239 235 314 318 319 315
216 4
235 239 240 236 315 319 320 316
217 1
221 241 242 225 301 321 322 305
218 2
225 242 243 229 305 322 323 309
219 3
229 243 244 233 309 323 324 313
220 4

```
233 244 245 237 313 324 325 317
 221     1
241 249 250 242 321 329 330 322
 222     2
242 250 251 243 322 330 331 323
 223     3
243 251 252 244 323 331 332 324
 224     4
244 252 253 245 324 332 333 325
 225     1
249 257 258 250 329 337 338 330
 226     2
250 258 259 251 330 338 339 331
 227     3
251 259 260 252 331 339 340 332
 228     4
252 260 261 253 332 340 341 333
 229     1
257 265 266 258 337 345 346 338
 230     2
258 266 267 259 338 346 347 339
 231     3
259 267 268 260 339 347 348 340
 232     4
260 268 269 261 340 348 349 341
 233     1
265 273 274 266 345 353 354 346
 234     2
266 274 275 267 346 354 355 347
 235     3
267 275 276 268 347 355 356 348
 236     4
268 276 277 269 348 356 357 349
 237     1
273 297 293 274 353 377 373 354
 238     2
274 293 289 275 354 373 369 355
 239     3
275 289 285 276 355 369 365 356
 240     4
276 285 281 277 356 365 361 357
 241     5
237 245 246 238 317 325 326 318
 242     5
238 246 247 239 318 326 327 319
 243     5
239 247 248 240 319 327 328 320
 244     5
245 253 254 246 325 333 334 326
 245     5
246 254 255 247 326 334 335 327
 246     5
247 255 256 248 327 335 336 328
 247     5
253 261 262 254 333 341 342 334
 248     5
254 262 263 255 334 342 343 335
 249     5
255 263 264 256 335 343 344 336
 250     5
261 269 270 262 341 349 350 342
 251     5
262 270 271 263 342 350 351 343
 252     5
263 271 272 264 343 351 352 344
 253     5
269 277 278 270 349 357 358 350
 254     5
270 278 279 271 350 358 359 351
 255     5
271 279 280 272 351 359 360 352
 256     5
```

277 281 282 278 357 361 362 358
257 5
278 282 283 279 358 362 363 359
258 5
279 283 284 280 359 363 364 360
259 4
281 285 286 282 361 365 366 362
260 4
282 286 287 283 362 366 367 363
261 4
283 287 288 284 363 367 368 364
262 3
285 289 290 286 365 369 370 366
263 3
286 290 291 287 366 370 371 367
264 3
287 291 292 288 367 371 372 368
265 2
289 293 294 290 369 373 374 370
266 2
290 294 295 291 370 374 375 371
267 2
291 295 296 292 371 375 376 372
268 1
293 297 298 294 373 377 378 374
269 1
294 298 299 295 374 378 379 375
270 1
295 299 300 296 375 379 380 376
271 1
301 305 306 302 381 385 386 382
272 1
302 306 307 303 382 386 387 383
273 1
303 307 308 304 383 387 388 384
274 2
305 309 310 306 385 389 390 386
275 2
306 310 311 307 386 390 391 387
276 2
307 311 312 308 387 391 392 388
277 3
309 313 314 310 389 393 394 390
278 3
310 314 315 311 390 394 395 391
279 3
311 315 316 312 391 395 396 392
280 4
313 317 318 314 393 397 398 394
281 4
314 318 319 315 394 398 399 395
282 4
315 319 320 316 395 399 400 396
283 1
301 321 322 305 381 401 402 385
284 2
305 322 323 309 385 402 403 389
285 3
309 323 324 313 389 403 404 393
286 4
313 324 325 317 393 404 405 397
287 1
321 329 330 322 401 409 410 402
288 2
322 330 331 323 402 410 411 403
289 3
323 331 332 324 403 411 412 404
290 4
324 332 333 325 404 412 413 405
291 1
329 337 338 330 409 417 418 410
292 2

```
330 338 339 331 410 418 419 411
 293     3
331 339 340 332 411 419 420 412
 294     4
332 340 341 333 412 420 421 413
 295     1
337 345 346 338 417 425 426 418
 296     2
338 346 347 339 418 426 427 419
 297     3
339 347 348 340 419 427 428 420
 298     4
340 348 349 341 420 428 429 421
 299     1
345 353 354 346 425 433 434 426
 300     2
346 354 355 347 426 434 435 427
 301     3
347 355 356 348 427 435 436 428
 302     4
348 356 357 349 428 436 437 429
 303     1
353 377 373 354 433 457 453 434
 304     2
354 373 369 355 434 453 449 435
 305     3
355 369 365 356 435 449 445 436
 306     4
356 365 361 357 436 445 441 437
 307     5
317 325 326 318 397 405 406 398
 308     5
318 326 327 319 398 406 407 399
 309     5
319 327 328 320 399 407 408 400
 310     5
325 333 334 326 405 413 414 406
 311     5
326 334 335 327 406 414 415 407
 312     5
327 335 336 328 407 415 416 408
 313     5
333 341 342 334 413 421 422 414
 314     5
334 342 343 335 414 422 423 415
 315     5
335 343 344 336 415 423 424 416
 316     5
341 349 350 342 421 429 430 422
 317     5
342 350 351 343 422 430 431 423
 318     5
343 351 352 344 423 431 432 424
 319     5
349 357 358 350 429 437 438 430
 320     5
350 358 359 351 430 438 439 431
 321     5
351 359 360 352 431 439 440 432
 322     5
357 361 362 358 437 441 442 438
 323     5
358 362 363 359 438 442 443 439
 324     5
359 363 364 360 439 443 444 440
 325     4
361 365 366 362 441 445 446 442
 326     4
362 366 367 363 442 446 447 443
 327     4
363 367 368 364 443 447 448 444
 328     3
```

365 369 37J 3F6 445 449 450 446
329 3
366 370 371 367 446 450 451 447
330 3
367 371 372 368 447 451 452 448
331 2
369 373 374 37J 449 453 454 450
332 2
370 374 375 371 450 454 455 451
333 2
371 375 376 372 451 455 456 452
334 1
373 377 378 374 453 457 458 454
335 1
374 378 379 375 454 458 459 455
336 1
375 379 38U 376 455 459 460 456
337 1
381 385 386 382 461 465 466 462
338 1
382 386 387 383 462 466 467 463
339 1
383 387 388 384 463 467 468 464
340 2
385 389 39J 386 465 469 470 466
341 2
386 390 391 387 466 470 471 467
342 2
387 391 392 388 467 471 472 468
343 3
389 393 394 390 469 473 474 470
344 3
390 394 395 391 470 474 475 471
345 3
391 395 396 392 471 475 476 472
346 4
393 397 398 394 473 477 478 474
347 4
394 398 399 395 474 478 479 475
348 4
395 399 400 396 475 479 480 476
349 1
381 401 402 385 461 481 482 465
350 2
385 402 403 389 465 482 483 469
351 3
389 403 404 393 469 483 484 473
352 4
393 404 405 397 473 484 485 477
353 1
401 409 410 402 481 489 490 482
354 2
402 410 411 403 482 490 491 483
355 3
403 411 412 404 483 491 492 484
356 4
404 412 413 405 484 492 493 485
357 1
409 417 418 410 489 497 498 490
358 2
410 418 419 411 490 498 499 491
359 3
411 419 420 412 491 499 500 492
360 4
412 420 421 413 492 500 501 493
361 1
417 425 426 418 497 505 506 498
362 2
418 426 427 419 498 506 507 499
363 3
419 427 428 420 499 507 508 500
364 4

```
420 428 429 421 500 508 509 501
 365     1
425 433 434 426 505 513 514 506
 366     2
426 434 435 427 506 514 515 507
 367     3
427 435 436 428 507 515 516 508
 368     4
428 436 437 429 508 516 517 509
 369     1
433 457 453 434 513 537 533 514
 370     2
434 453 444 435 514 533 529 515
 371     3
435 449 445 436 515 529 525 516
 372     4
436 445 441 437 516 525 521 517
 373     5
397 405 406 398 477 485 486 478
 374     5
398 406 407 399 478 486 487 479
 375     5
399 407 408 400 479 487 488 480
 376     5
405 413 414 406 485 493 494 486
 377     5
406 414 415 407 486 494 495 487
 378     5
407 415 416 408 487 495 496 488
 379     5
413 421 422 414 493 501 502 494
 380     5
414 422 423 415 494 502 503 495
 381     5
415 423 424 416 495 503 504 496
 382     5
421 429 430 422 501 509 510 502
 383     5
422 430 431 423 502 510 511 503
 384     5
423 431 432 424 503 511 512 504
 385     5
429 437 438 430 509 517 518 510
 386     5
430 438 439 431 510 518 519 511
 387     5
431 439 440 432 511 519 520 512
 388     5
437 441 442 438 517 521 522 518
 389     5
438 442 443 439 518 522 523 519
 390     5
439 443 444 440 519 523 524 520
 391     4
441 445 446 442 521 525 526 522
 392     4
442 446 447 443 522 526 527 523
 393     4
443 447 448 444 523 527 528 524
 394     3
445 449 450 446 525 529 530 526
 395     3
446 450 451 447 526 530 531 527
 396     3
447 451 452 448 527 531 532 528
 397     2
449 453 454 450 529 533 534 530
 398     2
450 454 455 451 530 534 535 531
 399     2
451 455 456 452 531 535 536 532
 400     1
```

453 457 458 454 533 537 538 534
401 1
454 458 459 455 534 538 539 535
402 1
455 459 460 456 535 539 540 536
403 1
461 465 466 462 653 625 626 654
404 1
462 466 467 463 654 626 627 655
405 1
463 467 468 464 655 627 628 656
406 2
465 469 470 466 625 597 598 626
407 2
466 470 471 467 626 598 599 627
408 2
467 471 472 468 627 599 600 628
409 3
469 473 474 470 597 569 570 598
410 3
470 474 475 471 598 570 571 599
411 3
471 475 476 472 599 571 572 600
412 4
473 477 478 474 569 541 542 570
413 4
474 478 479 475 570 542 543 571
414 4
475 479 480 476 571 543 544 572
415 1
481 481 482 465 653 657 629 625
416 2
465 482 483 469 625 629 601 597
417 3
469 483 484 473 597 601 573 569
418 4
473 484 485 477 569 573 545 541
419 1
481 489 490 482 657 661 633 629
420 2
482 490 491 483 629 633 605 601
421 3
483 491 492 484 601 605 577 573
422 4
484 492 493 485 573 577 549 545
423 1
489 497 498 490 661 665 637 633
424 2
490 498 499 491 633 637 609 605
425 3
491 499 500 492 605 609 581 577
426 4
492 500 501 493 577 581 553 549
427 1
497 505 506 498 665 669 641 637
428 2
498 506 507 499 637 641 613 609
429 3
499 507 508 500 609 613 585 581
430 4
500 508 509 501 581 585 557 553
431 1
505 513 514 506 669 673 645 641
432 2
506 514 515 507 641 645 617 613
433 3
507 515 516 508 613 617 589 585
434 4
508 516 517 509 585 589 561 557
435 1
513 537 533 514 673 677 649 645
436 2

514 533 529 515 645 649 621 617
437 3
515 529 525 516 617 621 593 589
438 4
516 525 521 517 589 593 565 561
439 5
477 485 486 478 541 545 546 542
440 5
478 486 487 479 542 546 547 543
441 5
479 487 488 480 543 547 548 544
442 5
485 493 494 486 545 549 550 546
443 5
486 494 495 487 546 550 551 547
444 5
487 495 496 488 547 551 552 548
445 5
493 501 502 494 549 553 554 550
446 5
494 502 503 495 550 554 555 551
447 5
495 503 504 496 551 555 556 552
448 5
501 509 510 502 553 557 558 554
449 5
502 510 511 503 554 558 559 555
450 5
503 511 512 504 555 559 560 556
451 5
509 517 518 510 557 561 562 558
452 5
510 518 519 511 558 562 563 559
453 5
511 519 520 512 559 563 564 560
454 5
517 521 522 518 561 565 566 562
455 5
518 522 523 519 562 566 567 563
456 5
519 523 524 520 563 567 568 564
457 4
521 525 526 522 565 593 594 566
458 4
522 526 527 523 566 594 595 567
459 4
523 527 528 524 567 595 596 568
460 3
525 529 530 526 593 621 622 594
461 3
526 530 531 527 594 622 623 595
462 3
527 531 532 528 595 623 624 596
463 2
529 533 534 530 621 649 650 622
464 2
530 534 535 531 622 650 651 623
465 2
531 535 536 532 623 651 652 624
466 1
533 537 538 534 649 677 678 650
467 1
534 538 539 535 650 678 679 651
468 1
535 539 540 536 651 679 680 652
469 4
541 545 546 542 569 573 574 570
470 4
542 546 547 543 570 574 575 571
471 4
543 547 548 544 571 575 576 572
472 4

```
545 549 55[illegible] 546 573 577 578 574
 473     4
546 550 551 547 574 578 579 575
 474     4
547 551 552 548 575 579 580 576
 475     4
549 553 554 55[illegible] 577 581 582 578
 476     4
550 554 555 551 578 582 583 579
 477     4
551 555 55[illegible] 552 579 583 584 580
 478     4
553 557 55[illegible] 554 581 585 586 582
 479     4
554 558 554 555 582 586 587 583
 480     5
555 559 56[illegible] 556 583 587 588 584
 481     4
557 561 562 558 585 589 590 586
 482     4
558 562 563 559 586 59[illegible] 591 587
 483     5
559 563 564 560 587 591 592 588
 484     4
561 565 56[illegible] 562 589 593 594 590
 485     4
562 566 567 563 590 594 595 591
 486     4
563 567 568 564 591 595 596 592
 487     3
569 573 574 57[illegible] 597 601 602 598
 488     3
570 574 57[illegible] 571 598 602 603 599
 489     3
571 575 57[illegible] 572 599 603 604 600
 490     3
573 577 57[illegible] 574 601 605 606 602
 491     3
574 578 579 575 602 606 607 603
 492     3
575 579 58[illegible] 576 603 607 608 604
 493     3
577 581 582 578 605 609 610 606
 494     3
578 582 583 579 606 61[illegible] 611 607
 495     3
579 583 584 58[illegible] 607 611 612 608
 496     3
581 585 58[illegible] 582 609 613 614 610
 497     3
582 586 587 583 610 614 615 611
 498     5
583 587 588 584 611 615 616 612
 499     3
585 589 59[illegible] 586 613 617 618 614
 500     3
586 590 591 587 614 618 619 615
 501     5
587 591 59[illegible] 588 615 619 620 616
 502     3
589 593 594 590 617 621 622 618
 503     3
590 594 595 591 618 622 623 619
 504     3
591 595 596 592 619 623 624 620
 505     2
597 601 602 598 625 629 630 626
 506     2
598 602 60[illegible] 599 626 63[illegible] 631 627
 507     2
599 603 604 600 627 631 632 628
 508     2
```

```
601 605 606 602 629 633 634 630
 509     2
602 606 607 603 630 634 635 631
 510     2
603 607 608 604 631 635 636 632
 511     2
605 609 610 606 633 637 638 634
 512     2
606 610 611 607 634 638 639 635
 513     2
607 611 612 608 635 639 640 636
 514     2
609 613 614 610 637 641 642 638
 515     2
610 614 615 611 638 642 643 639
 516     5
611 615 616 612 639 643 644 640
 517     2
613 617 618 614 641 645 646 642
 518     2
614 618 619 615 642 646 647 643
 519     5
615 619 620 616 643 647 648 644
 520     2
617 621 622 618 645 649 650 646
 521     2
618 622 623 619 646 650 651 647
 522     2
619 623 624 620 647 651 652 648
 523     1
625 629 630 626 653 657 658 654
 524     1
626 630 631 627 654 658 659 655
 525     1
627 631 632 628 655 659 660 656
 526     1
629 633 634 630 657 661 662 658
 527     1
630 634 635 631 658 662 663 659
 528     1
631 635 636 632 659 663 664 660
 529     1
633 637 638 634 661 665 666 662
 530     1
634 638 639 635 662 666 667 663
 531     1
635 639 640 636 663 667 668 664
 532     1
637 641 642 638 665 669 670 666
 533     1
638 642 643 639 666 670 671 667
 534     6
639 643 644 640 667 671 672 668
 535     1
641 645 646 642 669 673 674 670
 536     1
642 646 647 643 670 674 675 671
 537     6
643 647 648 644 671 675 676 672
 538     1
645 649 650 646 673 677 678 674
 539     1
646 650 651 647 674 678 679 675
 540     1
647 651 652 648 675 679 680 676
 541     1
653 657 658 654 681 685 686 682
 542     1
654 658 659 655 682 686 687 683
 543     1
655 659 660 656 683 687 688 684
 544     1
```

```
657 661 662 658 685 689 690 686
 545     1
661 665 666 662 689 691 692 690
 546     1
681 685 686 682 693 697 698 694
 547     1
682 686 687 683 694 698 699 695
 548     1
683 687 688 684 695 699 700 696
 549     1
685 689 690 686 697 701 702 698
 550     1
689 691 692 690 701 703 704 702
 551     1
693 697 698 694 705 709 710 706
 552     1
694 698 699 695 706 710 711 707
 553     1
695 699 700 696 707 711 712 708
 554     1
697 701 702 698 709 713 714 710
 555     1
701 703 704 702 713 715 716 714
   4 FORCE
   4       -.9659                  -.2588
  12 PLOT3D
   1 IMPLICIT
0.001       11          1  716     1  555     1           .25
   2    2                0.01       500.       314.
   STOP
```

Appendix H

OTHER ASSOCIATED CODE LISTINGS

```
*DECK SKMSH
      PROGRAM SKMSH(INPUT,OUTPUT,TAPE5=INPUT,TAPE7=OUTPUT,TAPE6=OUTPUT,
     1   PUNCH)
      COMMON /JAC/ NJAC, ISA,NSA
      COMMON /X/ AA(3,20)
      DIMENSION DEN(12)
      DIMENSION  NHIG(15,15,15),NC(8),IPT(1215)
      DIMENSION X1(1215),Y1(1215),Z1(1215),CT(1215)
      COMMON ICONN(1120,10),X(1290),Y(1290),Z(1290),H(1290),HC(1290)
      DIMENSION ICORRC(8)
      INTEGER RC
      INTEGER P
      REAL MASS
      DATA ICORRC/1,2,4,3,5,6,8,7/
      A=0.577350269189626
      WRITE(6,2008)
 2008 FORMAT( 1H1,45X40HW E L C O M E  T O  S K U L L  M E S H  ///)
      READ(5,200) NSIZE
      READ (5,207) NDIM, NEL,NSA,NJAC,IHALF,JACON,MASSD
     *  ,IFACE
  207 FORMAT(8I5)
      WRITE(6,2012) NDIM,NEL,NSA,NJAC,IHALF,JACON,MASSD
     *  ,IFACE
 2012 FORMAT (40H NUMBER OF DIMENSIONS.....,I5,/
     X         40H NUMBER OF NODES FOR EACH ELEMENT.....,I5,/
     X         40H SUBARACHNOID SPACE SPECIFIED.....,I5,/
     X         40H CONSTANT THICKNESS TO BE COMPUTED.....,I5,/
     X         40H HALF SKULL SPECIFIED.....,I5,/
     X         40H JACOBIAN CHECK ONLY.....,I5,/
     X         40H MASS DISTRIBUTION PROPERTIES.....,I5,/
     X         40H FACIAL BONES TO BE SPECIFIED.....,I5,//)
      READ (5,4999) NUMMAT
      DO 5000 IM = 1,NUMMAT
      READ (5,5001) DEN(IM)
      WRITE (6,5002) IM,DEN(IM)
 5000 CONTINUE
 4999 FORMAT (I5)
 5001 FORMAT (/,29X,E11.4)
 5002 FORMAT ( 9H MATERIAL,I5,9H DENSITY=,E11.4,20H LB-SEC**2 PER IN**4)
C
C     CHECK JACOBIANS FOR COMPLETELY PRESCRIBED MESH
C
      DO 7002 I=1,1215
 7002 IPT(I)=0
      IF (JACON .EQ. 0) GO TO 2
      READ (5,3000) NPT
 3000 FORMAT (I5)
      DO 3 N1=1,NPT
      READ (5,91)   N,BC(N),X(N),Y(N),Z(N), IPT(N)
    3 CONTINUE
      READ (5,3000) NELEM
      DO 4 N1=1,NELEM
      READ (5,3001) N,IM,(ICONN(N,M),M=1,8)
 3001 FORMAT(2I5,/,8I4)
      ICONN(N,10) = IM
    4 CONTINUE
      GO TO 670
    2 READ (5,2010) SCALX, SCALY, SCALZ, SCALT
 2010 FORMAT (4F10.3)
      WRITE (6,2009) NSIZE
 2009 FORMAT(//33H ORDER OF BRAIN DISCRETIZATION,I5,2X,9HCUBED//)
      WRITE(6,2011) SCALX,SCALY,SCALZ,SCALT
 2011 FORMAT (40H X COORDINATE SCALE FACTOR.....,F5.3,/
     X         40H Y COORDINATE SCALE FACTOR.....,F5.3,/
     X         40H Z COORDINATE SCALE FACTOR.....,F5.3,/
     X         40H THICKNESS SCALE FACTOR.....,F5.3,//)
      NTOT=(NSIZE+1)**3+4*((NSIZE+1)**3-(NSIZE-1)**3)
      NPT=NTOT
      N1=1
      N4=4
      N5=5
      N6=6
```

```
      N8=NSIZE+4
      N9=NSIZE+5
      N10=NSIZE+6
      N11=NSIZE+7
      N12=NSIZE+8
      N13=NSIZE+9
C
C     BUILD A BASIC ARAY OF NODES WITHIN NBIG
C
      NODE=1
      DO 11 I=1,5
      DO 11 J =5,N9
      DO 11 K= 5,N9
      NODE = NODE+1
   11 NBIG(I,J,K)= NODE
      DO 14 I=6,N8
      DO 12 J=1,5
      DO 12 K=5,N9
      NODE = NODE+1
   12 NBIG(I,J,K) = NODE
      DO 13 J=N6,N8
      DO 13 K=1, N13
      NODE = NODE+1
   13 NBIG(I,J,K) = NODE
      DO 14 J=N9,N13
      DO 14 K=5,N9
      NODE = NODE+1
   14 NBIG(I,J,K) =NODE
      DO 15 I=N9,N13
      DO 15 J=5,N9
      DO 15 K=5,N9
      NODE =NODE+1
   15 NBIG(I,J,K)=NODE
      NTOT=NODE
C
C     FOLD IN THE EDGES
C
      DO 22 I=1,4
      IM=N13+1-I
C IM RUNS FROM 13 TO 10 AS I GOES FROM 1 TO 4
      IP= N9+I
C  IP RUNS FROM 10 TO 13 AS I GOES FROM 1 TO 4
      DO 21 J=5,N9
C     FRONT TOP
      NBIG(5,J,I)= NBIG(I,J,5)
C
C     FRONT BOTTOM
      NBIG(5,J,IM)= NBIG(I,J,N9)
C
C     BACK TOP
      NBIG(N9,J,I)= NBIG(IM,J,5)
C
C     BACK BOT
   21 NBIG(N9,J,IP)= NBIG(IP,J,N9)
      DO 22 K=5,N9
C
C LEFT FRONT
      NBIG(5,I,K)= NBIG(I,5,K)
C
C RIGHT FRONT
      NBIG(5,IM,K) = NBIG(I,N9,K)
C
C LEFT REAR
      NBIG(N9,I,K)= NBIG(IM,5,K)
C
C RIGHT REAR
      NBIG(N9,IP,K) = NBIG(IP,N9,K)
C
   22 CONTINUE
      DO 25 I=6,N8
      DO 25 J=1,4
      JM=N13+1-J
```

```
      JP=NN*J
C
C LEFT TOP
      NBIG(I,5,J)= NBIG(I,J,5)
C
C LEFT BOTTOM
      NBIG(I,5,JM) = NBIG(I,J,NN)
C
C RIGHT TOP
      NBIG(I,NN,J )= NBIG(I,JM,5)
C
C RIGHT BOTTOM
      NBIG(I,NN,JP)= NBIG(I,JP,NN)
   25 CONTINUE
C



      HSUM = 0
      KOUNT = 0
   70 READ(5,1000) XX,YY,ZZ,IP,NODE,HH
 1000 FORMAT(5X,3E10.0,I5,I9,E10.0)
      WRITE(6,1000) XX,YY,ZZ,IP,NODE,HH
      IF(NODE.EQ.0) GO TO 71
      KOUNT = KOUNT + 1
      X(NODE)=XX*SCALX
      Y(NODE)=YY* SCALY
      Z(NODE)=ZZ* SCALZ
      H(NODE)=HH* SCALH
      IF (NJAC .NE. 0) HSUM = HSUM + H(NODE)
      IP(NODE)=IP
      GO TO 70
   71 CONTINUE
      TSA =(HSUM/KOUNT)*.25
      IF (NJAC .NE. 0) WRITE (6,1001) TSA
 1001 FORMAT (41H CONSTANT SUBARACHNOID SPACE THICKNESS IS, F5.3,2X, 6HI
     NCHES//)


C
C
C  BUILD AND WRITE THE ICONN ARAY
C
      NELEM=0
      DO 33 I=1,4
      DO 33 J=5,NB
      DO 33 K=5,NB
      IMAT=1
      NELEM=NELEM+1
      CALL ISTORE(NELEM,I,J,K,NBIG,ICONN,IMAT)
   33 CONTINUE
      DO 37 I= 5,NB
      DO 34 J= 1,4
      DO 34 K=5,NB
      IMAT=J
      NELEM=NELEM+1
      CALL ISTORE(NELEM,I,J,K,NBIG,ICONN,IMAT)
   34 CONTINUE
      DO 35 J=5,NB
      DO 35 K=1,4
      IMAT=K
      NELEM=NELEM+1
      CALL ISTORE(NELEM,I,J,K,NBIG,ICONN,IMAT)
   35 CONTINUE
      DO 135 J=5,NB
      DO 135 K=5,NB
      IMAT=5
      NELEM=NELEM+1
      CALL ISTORE(NELEM,I,J,K,NBIG,ICONN,IMAT)
  135 CONTINUE
      DO 235 J=5,NB
```

```
      DO 235 K=N9,N12
      IMAT=N13-K
      NELEM=NELEM+1
      CALL ISTORE(NELEM,I,J,K,NBIG,ICONN,IMAT)
  235 CONTINUE
      DO 36 J=N9,N12
      DO 36 K=5,N8
      IMAT=N13-J
      NELEM=NELEM+1
      CALL ISTORE(NELEM,I,J,K,NBIG,ICONN,IMAT)
   36 CONTINUE
   37 CONTINUE
      DO 38 I= N9,N12
      DO 38 J=5,N8
      DO 38 K=5,N8
      IMAT=N13-1
      NELEM=NELEM+1
      CALL ISTORE(NELEM,I,J,K,NBIG,ICONN,IMAT)
   38 CONTINUE
  200 FORMAT (915)


C
C
C  FIND SURFACE NODE AND NEIGHBORS.  CODE=1 FOR CORNERS, 2 FOR EDGES, 3 FACES.
C
      CALL POSTS( 5,5,4, 5,6,4, 5,4,6, 6,5,4, 1,1,1,                          1
     1   5,  5,  5,   5,  5,  3,  5,  5,  2,   5,  5,  1,  1,NBIG,IPT)        1
      CALL POSTS(4,5,N9, 4,6,N9 ,6,4,N9 ,4,5,N8, 1,1,1,                       2
     1   5,  5, N9,   5,  5,N11,  5,  5,N12,   5,  5,N13,  1,NBIG,IPT)        2
      CALL POSTS(5,N9,4, 6,N9,4, 4,N9,6, 4,N8,5, 1,1,1,                       3
     1   5, N9,  5,   5, N9,  3,  5, N9,  2,   5, N9,  1,  1,NBIG,IPT)        3
      CALL POSTS(4,N9,N9, 6,N9,N10, 4,N8,N9, 4,N9,N8, 1,1,1,                  4
     1   5, N9, N9,   5, N9,N11,  5, N9,N12,   5, N9,N13,  1,NBIG,IPT)        4
      CALL POSTS(N9,4,5, N8,4,5, N9,4,6, N9,6,4, 1,1,1,                       5
     1  N9,  5,  5,  N9,  5,  3, N9,  5,  2,  N9,  5,  1,  1,NBIG,IPT)        5
      CALL POSTS(N9,4,N9, N8,4,N9, N10,6,N9, N9,4,N8, 1,1,1,                  6
     1  N9,  5, N9,  N9,  5,N11, N9,  5,N12,  N9,  5,N13,  1,NBIG,IPT)        6
      CALL POSTS(N9,N9,4, N9,N8,4, N10,N9,6, N8,N9,4, 1,1,1,                  7
     1  N9, N9,  5,  N9, N9,  3, N9, N9,  2,  N9, N9,  1,  1,NBIG,IPT)        7
      CALL POSTS(N9,N9,N10, N9,N8,N10, N8,N9,N10, N10,N9,N8, 1,1,1,           8
     1  N9, N9, N9,  N9, N9,N11, N9, N9,N12,  N9, N9,N13,  1,NBIG,IPT)        8
      DO 40 L=6,N8                                                            9
      M=L-1                                                                   9
      P=L+1                                                                   9
      CALL POSTS(4,5,L, 4,6,L, 4,5,P, 6,4,L, 4,5,M,                           9
     1   5,  5,  L,   3,  5,  L,  2,  5,  L,   1,  5,  L,  2,NBIG,IPT)        9
   40 CONTINUE                                                                10
      DO 41 L=6,N8                                                            10
      M=L-1                                                                   10
      P=L+1                                                                   10
      CALL POSTS(4,N9,L, 6,N10,L, 4,N9,P, 4,N8,L, 4,N9,M,                     10
     1   5, N9,  L,   3, N9,  L,  2, N9,  L,   1, N9,  L,  2,NBIG,IPT)        10
   41 CONTINUE                                                                11
      DO 42 L=6,N8                                                            11
      M=L-1                                                                   11
      P=L+1                                                                   11
      CALL POSTS(N9,4,L, N8,4,L, N9,4,P, N10,6,L, N9,4,M,                     11
     1  N9,  5,  L, N11,  5,  L,N12,  5,  L, N13,  5,  L,  2,NBIG,IPT)        11
   42 CONTINUE                                                                12
      DO 43 L=6,N8                                                            12
      P=L+1                                                                   12
      M=L-1                                                                   12
      CALL POSTS(N10,N9,L, N10,N8,L, N10,N9,P, N8,N10,L, N10,N9,M,            12
     1  N9, N9,  L, N11, N9,  L,N12, N9,  L, N13, N9,  L,  2,NBIG,IPT)        12
   43 CONTINUE                                                                13
      DO 44 L=6,N8                                                            13
      M=L-1                                                                   13
      P=L+1                                                                   13
      CALL POSTS(L,4,5, M,4,5, L,4,6, P,4,5, L,6,4,                           13
     1   L,  5,  5,   L,  5,  3,  L,  5,  2,   L,  5,  1,  2,NBIG,IPT)        13
   44 CONTINUE
```

```
      DO 45 L=A,NB
      M=L-1
      P=L+1
      CALL POSTS(L,N10,5, P,N10,5, L,N10,6, M,N10,5, L,NB,4,                14
     1   L, N9,  5,   L, N9,  3,  L, N9,  2,   L, N9,  1,  2,NBIG,IPT)       14
   45 CONTINUE
      DO 46 L=A,NB
      M=L-1
      P=L+1
      CALL POSTS(4,L,5, 4,P,5, 4,L,6, 4,M,5, 6,L,4,                         15
     1   5,  L,  5,   5,  L,  3,  5,  L,  2,   5,  L,  1,  2,NBIG,IPT)       15
   46 CONTINUE
      DO 47 L=A,NB
      M=L-1
      P=L+1
      CALL POSTS(N10,L,5, N10,M,5, N10,L,6, N10,P,5, NB,L,4,                16
     1  N9,  L,  5,  N9,  L,  3, N9,  L,  2,  N9,  L,  1,  2,NBIG,IPT)       16
   47 CONTINUE
      DO 48 L=A,NB
      M=L-1
      P=L+1
      CALL POSTS(L,4,N9, M,4,N9, L,6,N10, P,4,N9, L,4,NB,                   17
     1   L,  5, N9,   L,  5,N11,  L,  5,N12,   L,  5,N13,  2,NBIG,IPT)       17
   48 CONTINUE
      DO 49 L=A,NB
      M=L-1
      P=L+1
      CALL POSTS(L,N9,N10, P,N9,N10, L,NB,N10, M,N9,N10, L,N10,NB,          18
     1   L, N9, N9,   L, N9,N11,  L, N9,N12,   L, N9,N13,  2,NBIG,IPT)       18
   49 CONTINUE
      DO 50 L=A,NB
      M=L-1
      P=L+1
      CALL POSTS(4,L,N9, 4,P,N9, 6,L,N10, 4,M,N9, 4,L,NB,                   19
     1   5,  L, N9,   5,  L,N11,  5,  L,N12,   5,  L,N13,  2,NBIG,IPT)       19
   50 CONTINUE
      DO 51 L=A,NB
      M=L-1
      P=L+1
      CALL POSTS(N9,L,N10, N9,M,N10, NB,L,N10, N9,P,N10, N10,L,NB,          20
     1  N9,  L, N9,  N9,  L,N11, N9,  L,N12,  N9,  L,N13,  2,NBIG,IPT)       20
   51 CONTINUE
      DO 52 J=A,NB
      DO 52 K=A,NB
      JP=J+1
      JM=J-1
      KP=K+1
      KM=K-1
      CALL POSTS(4,J,K, 4,JP,K, 4,J,KP, 4,JM,K, 4,J,KM,                     21
     1   5,  J,  K,   3,  J,  K,  2,  J,  K,   1,  J,  K,  3,NBIG,IPT)       21
   52 CONTINUE
      DO 53 J=A,NB
      DO 53 K=A,NB
      JP=J+1
      JM=J-1
      KP=K+1
      KM=K-1
      CALL POSTS(N10,J,K, N10,JM,K, N10,J,KP, N10,JP,K, N10,J,KM,           22
     1  N9,  J,  K,  N11,  J,  K,N12,  J,  K, N13,  J,  K,  3,NBIG,IPT)       22
   53 CONTINUE
      DO 54 I=A,NB
      DO 54 K=A,NB
      IP=I+1
      IM=I-1
      KP=K+1
      KM=K-1
      CALL POSTS(I,4,K, IM,4,K, I,4,KP, IP,4,K, I,4,KM,                     23
     1   I,  5,  K,   I,  3,  K,  I,  2,  K,   I,  1,  K,  3,NBIG,IPT)       23
   54 CONTINUE
      DO 55 I=A,NB
      DO 55 K=A,NB
      IP=I+1
```

```
      IM= I-1
      KP= K+1
      KM= K-1
      CALL POSTS(I,N10,K, IP,N10,K, I,N10,KP, IM,N10,K, I,N10,KM,                24
     1    I, 09,  K,    I,N11,  K,  I,N12,  K,    I,N13,  K,  3,NBIG,IPT)      24
   55 CONTINUE
      DO 56 I=5,NB
      DO 56 J=5,NB
      IP = I+1
      IM= I-1
      JP= J+1
      JM= J-1
      CALL POSTS(I,J,4, I,JP,4, IM,J,4, I,JM,4, IP,J,4,                         25
     1    I,  J,  5,    I,  J,  3,  I,  J,  2,    I,  J,  1,  3,NBIG,IPT)      25
   56 CONTINUE
      DO 57 I =5,NB
      DO 57 J =5,NB
      IP= I+1
      IM= I-1
      JP= J+1
      JM= J-1
      CALL POSTS(I,J,N10, I,JP,N10, IP,J,N10, I,JM,N10, IM,J,N10,               26
     1    I,  J, N9,    I,  J,N11,  I,  J,N12,    I,  J,N13,  3,NBIG,IPT)      26
   57 CONTINUE


C
C     GENERATE COORDINATES FOR UNSPECIFIED INTERIOR NODES
C
  600 NMIS=0
      IF (NELEM .EQ. 0)GO TO 663
      DO 635 K=3,NPT
      IF(IPT(K) .NE. 0)GO TO 635
      X(K)=(X(1)+X(NPT))*.5
      Y(K)=(Y(1)+Y(NPT))*.5
      Z(K)=(Z(1)+Z(NPT))*.5
      NMIS=NMIS+1
  635 CONTINUE
      IF(NMIS .EQ. 0)GO TO 663
      DO 655 NN=1,NMIS
      IOT=0
      DO 637 K=1,NPT
      XT(K)=0.0
      YT(K)=0.0
      ZT(K)=0.0
  637 CT(K)=0.0
      DO 641 IE=1,NELEM
      DO 64. JJ=1,8
      N=8*(IE-1)
      M=N+JJ
      J=NOD(M)
      IF(IPT(J) .NE. 0)GO TO 640
      I=JJ-1
      L=I/4
      IF(4*L .EQ. I)I=4+I
      M=N+I
      I=NOD(M)
      K=JJ+1
      L=JJ/4
      IF(4*L .EQ. JJ)K=K-4
      M=N+K
      K=NOD(M)
      L=JJ*4
      IF(L .GT. 8)L=L-8
      M=N+L
      L=NOD(M)
      XT(J)=XT(J)+X(I)+X(K)+X(L)
      YT(J)=YT(J)+Y(I)+Y(K)+Y(L)
      ZT(J)=ZT(J)+Z(I)+Z(K)+Z(L)
  640 CT(J)=CT(J)+3.0
  641 CONTINUE
```

```
      DO 650 K=1,NPT
      D4=CT(K)
      IF(IPT(K) .GT. 0 .OR. D4     .EQ. 0.0)GO TO 650
      D1=X(K)
      D2=Y(K)
      D3=Z(K)
      X(K)=XT(K)/D4
      Y(K)=YT(K)/D4
      Z(K)=ZT(K)/D4
      D1=ABS (X(K)-D1)
      D2=ABS (Y(K)-D2)
      D3=ABS (Z(K)-D3)
      IF(D1+D2+D3 .GT..0001)IOT=1
  650 CONTINUE
      ICOUNT=ICOUNT+1
      IF(IOT .EQ. 0)GO TO 663
  655 CONTINUE
C
  663 CONTINUE
C
C     READ IN FACIAL BONES IF SPECIFIED
C
      IF(IFACE.NE.1) GO TO 680
C
C     READ IN THE FACIAL NODAL POINTS
      READ(5,3000) NPTF
      WRITE(6,3000) NPTF
      DO 681 I=1,NPTF
      READ(5,91) N,BC(N),X(N),Y(N),Z(N),IPT(N)
      WRITE(6,91) N,BC(N),X(N),Y(N),Z(N),IPT(N)
  681 CONTINUE
C
C     READ THE FACIAL CONNECTIVITY
C
      READ(5,3000) NELEMF
      WRITE(6,3000) NELEMF
      DO 682 I=1,NELEMF
      READ(5,3001) N,ICONN(N,10),(ICONN(N,J),J=1,8)
      WRITE(6,3001) N,ICONN(N,10),(ICONN(N,J),J=1,8)
  682 CONTINUE
C
C     UPDATE THE NUMBER OF NODES AND ELEMENTS
C
      NPT=NPT+NPTF
      NELEM=NELEM+NELEMF
  680 CONTINUE
C
C  IF SO SPECIFIED, HALVE THE MESH
C
      IF (IHALF .EQ. 1) CALL HALF(NPT,NELEM, NEL)
C
C  REFERENCE CO-ORDINATES TO FRANKFORT PLANE
C
      DO 669 N=1,NPT
      X(N)=5.62-X(N)
      Y(N) = -Y(N)
      Z(N)=Z(N)-5.75
  669 CONTINUE
C
C  CHECK MESH FOR NEGATIVE JACOBIANS
C
  670 DO 204 N = 1,NELEM
      DO 905 I=1,NEL
      K = ICONN(N,I)
      IF(K.GT.NPT  .OR.K.LT.0) GO TO
      IF(K.EQ.0) GO TO 905
      XA(1,I) = X(K)
      XA(2,I) = Y(K)
      XA(3,I) = Z(K)
  905 CONTINUE
      K1=ICONN(N,1)
      K2=ICONN(N,2)
```

```
      K3=ICONN(N,3)
      K4=ICONN(N,4)
      K5=ICONN(N,5)
      K6=ICONN(N,6)
      K7=ICONN(N,7)
      K8=ICONN(N,8)
      NSIDE = (NEL+1)/NDIM**2
      L=1
      IF(NDIM.EQ.2) L=2
      LJK = 0
      EVOL = 0.
      SS = 1.0
      DO 30 I = L,2
      SS = -SS
      TT = 1.0
      DO 30 J= 1,2
      TT = -TT
      UU = 1.0
      DO 30 K = 1,2
      UU = -UU
      CALL CHKJAC (UU,TT,SS,NDIM,NEL,NSIDE,XJAC)
      EVOL=EVOL+XJAC
C
C  COMPUTE MASS PROPERTIES OF SKULL
C
      LJK = LJK + 1
      IF(XJAC.GT.0.0) GO TO 30
      WRITE(6,208) N,XJAC,(ICONN(N,LL),LL=1,NEL)
  208 FORMAT(/2H ELEMENT,I5,2X, 8HJACOBIAN,E11.4/6H ICONN,8I4)
      JAC = -1
      WRITE(6,802) X(K1), X(K2), X(K3), X(K4), X(K5), X(K6), X(K7),X(K8)
      WRITE(6,803) Y(K1), Y(K2), Y(K3), Y(K4), Y(K5), Y(K6), Y(K7),Y(K8)
      WRITE(6,804) Z(K1), Z(K2), Z(K3), Z(K4), Z(K5), Z(K6), Z(K7),Z(K8)
  802 FORMAT (6H    X  ,8E14.4)
  803 FORMAT (6H    Y  ,8E14.4)
  804 FORMAT (6H    Z  ,8E14.4)
      LKJM=ICORRC(LJK)
      WRITE(6,206) ICONN(N,LKJM)
  206 FORMAT (/,5H NODE,I5,12H IS INVERTED)
   30 CONTINUE
      VOL = VOL + EVOL
      IM = ICONN(N,10)
      EMASS =EVOL*DEN(IM)
      MASS = MASS+EMASS
      IF(MASSD .NE.0) CALL MASSDIS(K1,K2,K3,K4,K5,K6,K7,K8,EMASS,N,
     XNELEM,MASS,IHALF)
  209 CONTINUE
      WRITE (6,99) VOL,MASS
   99 FORMAT(/9H VOLUME =,E11.4,10H INCHES**3/7H MASS =,E11.4,
     X 19H LB-SEC**2 PER INCH)
      IF (JACON .EQ. 1) GO TO 94
      IF (JAC.LT.0) GO TO 94
      WRITE(6,942)
  942 FORMAT (/55H MESH HAS BEEN CHECKED-NO NEGATIVE JACOBIANS WERE FOUN
     XD/)
   91 FORMAT (I5,10X, I1, 4X, 3F10.3,I5)
      DO 700 N=1,NPT
      PUNCH 91,    N,RC(N),X(N),Y(N),Z(N),IPT(N)
  700 WRITE (6,91) N,RC(N),X(N),Y(N),Z(N),IPT(N)
      PUNCH 93, (N,ICONN(N,10),(ICONN(N,M),M=1,8),N=1,NELEM)
      WRITE (6,93) (N,ICONN(N,10),(ICONN(N,M),M=1,8),N=1,NELEM)
   93 FORMAT (2I5,8I4)
      GO TO 94
  940 WRITE (6,941) K
  941 FORMAT (5H NODE,I5,2X,11HEXCEEDS NPT)
   94 STOP
      END
*DECK POSTS
      SUBROUTINE POSTS(LA,LB,LC,LD,LE,LF,LG,LH,LI,LJ,LK,LL,LM,LN,LO,LP,
     1   LQ,LR,LS,LT,LU,LV,LW,LX,LY,LZ,L1,ICODE,NBIG,IPT)
      COMMON /JAC/ NJAC, ISA,NSA
      COMMON ICONN(1120,10),X(1290),Y(1290),Z(1290),H(1290),RC(1290)
```

```
      INTEGER HC
      DIMENSION IPT(1215),NBIG(15,15,15),NS(4),NA(4)
      NODE=NBIG(LA,LB,LC)
      IPT(NODE)= 2
      NS(1)=NBIG(LD,LE,LF)
      K=NS(1)
      NS(2)= NBIG(LG,LH,LI)
      I=NS(2)
      NS(3)= NBIG(LJ,LK,LL)
      J=NS(3)
    3 NA(1)= NBIG(LP,LQ,LM)
      NA(2)=NBIG(LS,LT,LU)
      NA(3)=NBIG(LV,LW,LX)
      NA(4)=NBIG(LY,LZ,L1)
      NAN=NA(1)
      IPT(NAN)=1
      DO 40 II=2,4
      NAN=NA(II)
      IP=II+1
   40 IPT(NAN)=IP
      IF(ICODE-1) 4,4,2
    4 CONTINUE
      L=J
      NTRI=3
      GO TO 103
    2 NS(4)= NBIG(LM,LN,LO)
      L=NS(4)
      NTRI=4
  103 CONTINUE
      N=NODE
C
C  I,J,K,L ARE THE AUXILIARY NODES FOR NODE N
C
  104 C1 = 0
      C2 = 0
      C3 = 0
      DO 110 II = 1,NTRI
      XX = X(N)
      YY = Y(N)
      ZZ = Z(N)
      GO TO (105,106,107,108),II
  105 A1 = X(J) - XX
      A2 = Y(J) - YY
      A3 = Z(J) - ZZ
      B1 = X(I) - XX
      B2 = Y(I) - YY
      B3 = Z(I) - ZZ
      GO TO 109
  106 A1 = X(I) - XX
      A2 = Y(I) - YY
      A3 = Z(I) - ZZ
      B1 = X(K) - XX
      B2 = Y(K) - YY
      B3 = Z(K) - ZZ
      GO TO 109
  107 A1 = X(K) - XX
      A2 = Y(K) - YY
      A3 = Z(K) - ZZ
      B1 = X(L) - XX
      B2 = Y(L) - YY
      B3 = Z(L) - ZZ
      GO TO 109
  108 A1 = X(L) - XX
      A2 = Y(L) - YY
      A3 = Z(L) - ZZ
      B1 = X(J) - XX
      B2 = Y(J) - YY
      B3 = Z(J) - ZZ
  109 C1 = (A2*B3 - A3*B2)/NTRI + C1
      C2 = (A3*B1 - A1*B3)/NTRI + C2
      C3 = (A1*B2 - A2*B1)/NTRI + C3
  110 CONTINUE
```

```
C
C  DIRECTION COSINES
C
      CL =  (C1*C1 + C2*C2  + C3*C3)**.5
      COAL =     (C1/CL)
      COHE =     (C2/CL)
      COGA =     (C3/CL)
      COSCH = (COAL*COAL+COHE*COHE+COGA*COGA)**.5
C
C  IF SO SPECIFIED-INCLUDE THE SUBARACHNOID SPACE
C
      IF (NSA .EQ. 0) GO TO 116
C
C  CORRECT FOR NEG JACOBIANS BY SPECIFYING AVERAGE THICKNESS TSA
C
      IF (NJAC .NE. 0) GO TO 917
      TSA=H(NODE)*.25
  917 CONTINUE
      N=NA(1)
      X(N) = XX - TSA*COAL
      Y(N) = YY - TSA*COHE
      Z(N) = ZZ - TSA*COGA
  116 N=NODE
C
C  DIPLOE LAYER THICKNESS IS ASSUMED  ONE HALF THE TOTAL SKULL THICKNESS
C
  117 T = H(N)*.25
      N=NA(2)
      X (N)=XX + T*COAL
      Y (N)=YY + T*COHE
      Z (N)=ZZ + T*COGA
      N=NA(3)
      X (N)=XX + 3.*T*COAL
      Y (N)=YY + 3.*T*COHE
      Z (N)=ZZ + 3.*T*COGA
      N=NA(4)
      X (N)=XX + 4.*T*COAL
      Y (N)=YY + 4.*T*COHE
      Z (N)=ZZ + 4.*T*COGA
      RETURN
      END
*DECK NOD
      FUNCTION NOD(M)
      COMMON ICONN(1120,10),X(1290),Y(1290),Z(1290),H(1290),HC(1290)
      INTEGER HC
      NM1=(M-1)/8
      N=NM1+1
      JM1=M-NM1*8
      J=JM1
      NOD=ICONN(N,J)
      RETURN
      END
*DECK ISTORE
      SUBROUTINE ISTORE(NELEM,I,J,K,NBIG,ICONN,IMAT)
      DIMENSION NBIG(15,15,15),ICONN(1120,10)
      ICONN(NELEM,1)=NBIG(I  ,J  ,K  )
      ICONN(NELEM,2)=NBIG(I  ,J+1,K)
      ICONN(NELEM,3)=NBIG(I  ,J+1,K+1)
      ICONN(NELEM,4)=NBIG(I  ,J  ,K+1)
      ICONN(NELEM,5)=NBIG(I+1,J  ,K  )
      ICONN(NELEM,6)=NBIG(I+1,J+1,K)
      ICONN(NELEM,7)=NBIG(I+1,J+1,K+1)
      ICONN(NELEM,8)=NBIG(I+1,J  ,K+1)
      ICONN(NELEM, 9)=0
      ICONN(NELEM,10)=IMAT
      RETURN
      END
*DECK HALF
      SUBROUTINE HALF (NP1,NELEM,NEL)
      COMMON ICONN(1120,10),X(1290),Y(1290),Z(1290),H(1290),HC(1290)
      INTEGER HC
      DIMENSION ICROS(1215)
```

```
      DO 5 N = 1, NPT
    5 ICROS(N) = 0
      WRITE (6,130) NPT,NELEM,NEL
  130 FORMAT (3I5)
      N = 0
C
C  SET TO ZERO ALL Y COORDINATES ON MIDSAGITTAL PLANE AND
C  DELETE NODES WITH NEGATIVE Y COORDINATES
C
      DO 10 I = 1,NPT
      YI = Y(I)
      YI = ABS(Y(I))
      IF (YI .LT..009) GO TO 7
      IF (Y(I) .LT. 0.) GO TO 10
      GO TO 8
    7 YI = 0.
    8 N = N + 1
      ICROS(I)=N
      X(N) = X(I)
      Y(N) = YI
      Z(N) = Z(I)
      IF (YI .EQ. 0.) BC(N) = 1
   10 CONTINUE
      N = N+1
      DO 20 I = N,NPT
      X(I) = 0.
      Y(I) = 0.
   20 Z(I) = 0.
      NPT = N-1
C
C     FORM CONNECTIVITY FOR HALF SKULL
C
      KOUNT = 0
      DO 30 J = 1,NELEM
      DO 25 K = 1,NEL
      JK = ICONN(J,K)
      IF (ICROS(JK) .EQ. 0) GO TO 30
      ICONN(J,K)      = ICROS(JK)
   25 CONTINUE
      KOUNT = KOUNT + 1
      DO 27 K=1,NEL
   27 ICONN(KOUNT,K) = ICONN(J,K)
   30 CONTINUE
      NELEM = KOUNT
      WRITE (6,130) NPT,NELEM,NEL
      RETURN
      END
*DECK CHKJAC
      SUBROUTINE CHKJAC(R,S,T,NDIM,NEL,NSIDE,XJAC)
      COMMON /X/ XA(3,20)
      DIMENSION SHAPE(4,20),SG(3,3), SR(3,3)
      DIMENSION N1(12),N2(12),N3(12),NODE(12),S2ORD(12),S3ORD(12),SS(3)   HRI  2C
      DIMENSION COLM(3)                                                   HRI  3C
      DATA NODE/2,4,6,8,9,10,11,12,14,16,18,20/                           HRI  5C
      DATA    N1/1,2,1,2,3,3,3,3,1,2,1,2/ ,N2/2,3,2,3,1,1,1,1,2,3,2,3/    HRI  6C
      DATA    N3/3,1,3,1,2,2,2,2,3,1,3,1/                                 HRI  7C
      DATA S2ORD/-.5,-.5, .5,-.5,-.5, .5, .5,-.5,-.5, .5, .5,-.5/         HRI  8C
      DATA S3ORD/-.5, .5,-.5,-.5,-.5,-.5, .5, .5, .5, .5, .5,-.5/        HRI  9C
      SS(1) = R                                                           HRI 10C
      SS(2) = S                                                           HRI 11C
      SS(3) = T                                                           HRI 12C
      K1 = 4                                                              HRI 13C
      L1 = 1                                                              HRI 14C
      IF(NSIDE.EQ.1) GO TO 110                                            HRI 15C
      K1 = 12                                                             HRI 16C
      L1 = 2                                                              HRI 17C
C**** FORM MIDSIDE SHAPE FUNCTIONS                                        HRI 18C
      DO 100 L=1,12                                                       HRI 19C
      N = NODE(L)                                                         HRI 20C
      I = N1(L)                                                           HRI 21C
      J = N2(L)                                                           HRI 22C
      K = N3(L)                                                           HRI 23C
```

```
      SJ = S2ORD(L)                                                     HRI 24C
      SL = S3ORD(L)                                                     HRI 25C
      RP = 1. -SS(I)**2                                                 HRI 26C
      SP = .5 + SJ*SS(J)                                                HRI 27C
      TP = .5 + SL*SS(K)                                                HRI 28C
      SHAPE(I,N) = -2.0*SS(I)*SP*TP                                     HRI 29C
      SHAPE(J,N) = SJ*RP*TP                                             HRI 30C
      SHAPE(K,N) = SL*RP*SP                                             HRI 31C
100   SHAPE(4,N) = RP*SP*TP                                             HRI 32C
C**** FORM CORNER SHAPE FUNCTIONS                                       HRI 33C
110   SJ = -0.5                                                         HRI 34C
      K = 8                                                             HRI 35C
      DO 250 J = 1,2                                                    HRI 36C
      L = 1                                                             HRI 37C
      DO 200 I = 5,8                                                    HRI 38C
      RP = .5 + S2ORD(I)*K                                              HRI 39C
      SP = .5 + S3ORD(I)*S                                              HRI 40C
      TP = .5 + SJ*T                                                    HRI 41C
      SHAPE(1,L + K) = S2ORD(I)*SP*TP                                   HRI 42C
      SHAPE(2,L + K) = RP*S3ORD(I)*TP                                   HRI 43C
      SHAPE(3,L + K) = RP*SP*SJ                                         HRI 44C
      SHAPE(4,L + K) = RP*SP*TP                                         HRI 45C
200   L = L + 4                                                         HRI 46C
      K = K1                                                            HRI 47C
250   SJ = -0.5                                                         HRI 48C
      IF(NSIDE.EQ.1) GO TO 360                                          HRI 49C
C**** CORRECT BASIC CORNER FUNCTIONS BY PROPORTIONS OF MIDSIDE FUNCTIONS HRI 50C
      K = 8                                                             HRI 51C
      L = 9                                                             HRI 52C
      DO 350 I = 1,8,2                                                  HRI 53C
      DO 300 J = 1,4                                                    HRI 54C
      SHAPE(J,I) = SHAPE(J,I)-0.5*(SHAPE(J,I+1)+SHAPE(J,K)+SHAPE(J,L))  HRI 55C
300   SHAPE(J,I+12)=SHAPE(J,I+12)-0.5*(SHAPE(J,I+13)+SHAPE(J,K+12))     HRI 56C
     X  + SHAPE(J,L))                                                   HRI 57C
      K = I + 1                                                         HRI 58C
350   L = L + 1                                                         HRI 59C
C**** FORM THE JACOBIAN DETERMINANT                                     HRI 60C
360   DO 370 I = 1,3                                                    HRI 61C
      DO 370 J = 1,3                                                    HRI 62C
370   SK(I,J) = 0.0                                                     HRI 63C
      DO 400 I = 1,NDIM                                                 HRI 64C
      DO 400 J = 1,NDIM                                                 HRI 65C
      DO 400 K = 1,NEL                                                  HRI 66C
400   SK(I,J) = SK(I,J) + SHAPE(J,K)*XA(I,K)
      IF(NDIM.EQ.2) SK(3,3) = 1.0                                       HRI 68C
      SG(1,1) =  SK(2,2)*SK(3,3) - SK(2,3)*SK(3,2)                      HRI 69C
      SG(2,2) =  SK(1,1)*SK(3,3) - SK(1,3)*SK(3,1)                      HRI 70C
      SG(3,3) =  SK(1,1)*SK(2,2) - SK(1,2)*SK(2,1)                      HRI 71C
      SG(1,2) = -SK(1,2)*SK(3,3) + SK(1,3)*SK(3,2)                      HRI 72C
      SG(1,3) =  SK(1,2)*SK(2,3) - SK(1,3)*SK(2,2)                      HRI 73C
      SG(2,1) = -SK(2,1)*SK(3,3) + SK(2,3)*SK(3,1)                      HRI 74C
      SG(2,3) = -SK(1,1)*SK(2,3) + SK(2,1)*SK(1,3)                      HRI 75C
      SG(3,1) =  SK(2,1)*SK(3,2) - SK(2,2)*SK(3,1)                      HRI 76C
      SG(3,2) = -SK(1,1)*SK(3,2) + SK(1,2)*SK(3,1)                      HRI 77C
      XJAC = 0.0                                                        HRI 78C
      DO 600 I = 1,NDIM                                                 HRI 79C
600   XJAC = XJAC + SG(1,I)*SK(1,I)                                     HRI 80C
      RETURN                                                            HRI 89C
      END                                                               HRI 90C
*DECK MASSDIS
      SUBROUTINE  MASSDIS(K1,K2,K3,K4,K5,K6,K7,K8,EMASS,N,NELEM,MASS,
     XIHALF)
C
C  MASS MOMENT OF INERTIA TENSOR IS COMPUTED VIA PARALLEL AXIS THEOREM
C
      COMMON ICONN(1120,16),X(1290),Y(1290),Z(1290),H(1290),HC(1290)
      REAL MASS
C
C  COMPUTE ELEMENT CG CO-ORDINATES
C
      XCG = (X(K1)+X(K2)+X(K3)+X(K4)+X(K5)+X(K6)+X(K7)+X(K8))/8.0
      YCG = (Y(K1)+Y(K2)+Y(K3)+Y(K4)+Y(K5)+Y(K6)+Y(K7)+Y(K8))/8.0
```

```
      ZCG = (Z(K1)+Z(K2)+Z(K3)+Z(K4)+Z(K5)+Z(K6)+Z(K7)+Z(K8))/8.0
      RHOX =  YCG*YCG + ZCG*ZCG
      RHOY =  XCG*XCG + ZCG*ZCG
      RHOZ =  XCG*XCG + YCG*YCG
C
C  ACCUMULATE FIRST MOMENT OF ELEMENT MASS WRT REFERENCE AXES
C
      FMOY = XCG*EMASS+FMOY
      FMOX = ZCG*EMASS+FMOX
C
C  ACCUMULATE SECOND MOMENT OF ELEMENT MASS WRT REFERENCE AXES
C
      SMOXX = RHOX*EMASS+SMOXX
      SMOYY = RHOY*EMASS+SMOYY
      SMOZZ = RHOZ*EMASS+SMOZZ
      SMOXY = 0.
      SMOXZ = XCG*ZCG*EMASS+SMOXZ
      SMOYZ = 0.
C
      IF (N .LT. NELEM) RETURN
C
C     COMPUTE CG LOCATION WRT REFERENCE AXES
C
      WRITE (6,93)
   93 FORMAT (1H1,29H MASS DISTRIBUTION PROPERTIES//)
      IF (IHALF .EQ. 1) WRITE(6,94)
   94 FORMAT (/30H VALUES ARE FOR A HALVED SKULL/)
      XBAR = FMOY/MASS
      YBAR = 0.
      ZBAR = FMOX/MASS
      WRITE (6,98) XBAR,YBAR,ZBAR
      WRITE (6,99) SMOXX,SMOXY,SMOXZ,SMOXY,SMOYY,SMOYZ,SMOXZ,SMOYZ,SMOZZ
C
C  COMPUTE MASS MOMENT OF INERTIA WRT TO CG REFERENCE
C
      SMOXX = SMOXX - MASS*ZBAR*ZBAR
      SMOYY = SMOYY - MASS*(XBAR*XBAR + ZBAR*ZBAR)
      SMOZZ = SMOZZ - MASS*XBAR*XBAR
      SMOXY = 0.
      SMOXZ = SMOXZ - MASS*XBAR*ZBAR
      SMOYZ = 0.
      WRITE (6,96) SMOXX,SMOXY,SMOXZ,SMOXY,SMOYY,SMOYZ,SMOXZ,SMOYZ,SMOZZ
C
C     PRINCIPAL MASS MOMENTS OF INERTIA WRT TO CG REFERENCE
C
      AVE = (SMOXX+SMOZZ)*.5
      BASE = (SMOXX-SMOZZ)*.5
      RAD = (BASE*BASE + SMOXZ*SMOXZ)**.5
      SMOXX = AVE + RAD
      SMOZZ = AVE - RAD
      DUM = SMOXZ
      SMOXZ = 0.
      WRITE (6,95) SMOXX,SMOXY,SMOXZ,SMOXY,SMOYY,SMOYZ,SMOXZ,SMOYZ,SMOZZ
      THETA = .5*ATAN(DUM/BASE)
      THETA = 180.*THETA/3.14159265
      WRITE (6,97) THETA
   97 FORMAT(//55H PRINCIPAL INERTIA AXIS ORIENTATION WRT FRANKFORT PLAN
     XE/F8.1,4H DEGREES)
   98 FORMAT (/,21H CG LOCATION (INCHES)/6H XBAR=,F10.3/
     X6H YBAR=,F10.3/6H ZBAR=,F10.3)
   99 FORMAT(//34H INERTIA TENSOR WRT REFERENCE AXES/
     X3E11.3/3E11.3,5X,13H LB-SEC**2-IN/3E11.3)
   96 FORMAT(//52H INERTIA TENSOR WRT PARALLEL AXES PASSING THROUGH CG/
     X3E11.3/3E11.3,5X,13H LB-SEC**2-IN/3E11.3)
   95 FORMAT (//32H PRINCIPAL INERTIA TENSOR WRT CG/
     X3E11.3/3E11.3,5X,13H LB-SEC**2-IN/3E11.3)
      RETURN
      END
```

```
      PROGRAM BANDS (INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     *  TAPE7=PUNCH)
C
CXXXX
C  PREPROCESSOR FOR FEAP  (BANDWIDTH MINIMIZING)
C
C     INPUT
C
C     CARD) (81E)  NUMNP,IPUNCH,IPRINT
C     ELEMENT CARDS
C     EOF CARD
C
C  IF IPUNCH AND/OR IPRINT FLAGS ARE ON, THE NODE RESEQUENCING
C  ARRAY IS PUNCHED AND/OR PRINTED.  THIS ARRAY IS THEN INPUT
C  IN THE FEAP PROGRAM IMMEDIATELY FOLLOWING THE FEAP CARD.
C  A 1 IN CC 79 OF THE FEAP CARD SIGNALS THAT THE RESEQUENCING
C  ARRAY IS INCLUDED IN THE FEAP INPUT.
C
C  THE FORMAT FOR THE RESEQUENCING PUNCH IS :
C
C     CC 1-5, A SEQUENCE NUMBER  (TO MAINTAIN CARD ORDER)
C     CC 11-80,  15 ENTRIES IN THE RESEQUENCING ARRAY (15I5)
C     THE PUNCH IS TERMINATED WITH A 9999 IN CC 1-5
CXXXX
C
C
      COMMON /FILES/  IT1,IT2,IT3
      COMMON /CONTRL/ JNK(7),INUM,INEX,IRENUM,NTUTD
      COMMON /ERRS/   IERR(33)
      COMMON /OPUT/   MED(21)
      COMMON /SIG1/   MSEP(91)
      COMMON /MISC/   IASTER,ZERO,RAD
      COMMON /KKACOM/ KKK,KNT
      COMMON /TPOLGY/ II,JJ(3),NEXT,LAST,KORIG,KNEW
      COMMON /S/      NN,MM,IM,IB
      COMMON /P/      IMB,IME
      COMMON /TIME/   STIME,NCM
C
      COMMON          IA(1)
CXX
      READ (5,50) INUM,IPUNCH,IPRINT
   50 FORMAT (3I5)
CXX
      INEX=INUM
      IRENUM=2
      NTUTD=0
      NN=INUM
      MM=0
      LAST=3*INUM
      NEXT=INUM+1
C
      N1=1
      N2=N1+INEX
      N3=N2+INUM
      N4=N3+INUM
      N5=N4+INUM
      N6=N5+INUM
      N7=N6+INUM
      N8=N7+INUM
      N9=N8+LAST
C
      CALL FIELD (IA(N9))
      DO 10 I=1,INEX
   10 IA(I)=I
      DO 20 I=N2,N9
   20 IA(I)=0
C
      CALL ELIN (IA(N1),IA(N7),IA(N8),INUM,4)
C
      N9=N8+LAST+2*MM+1
      CALL FIELD (IA(N9))
      CALL SCHEME (IA(N8),IA(N1),IA(N2),IA(N3),IA(N4),IA(N5),IA(N6),
```

```
     *  IA(N7),IA(N8+LAST))
C
      WRITE (6,100) KORIG,KNEW
  100 FORMAT (1H1,15H ORIGINAL BW  =,I5,15H    FINAL BW  =,I5  )
C
      IF(IPRINT.NE.0) WRITE (6,800)
  800 FORMAT (22H0OLD NODE     NEW NODE   //)
      N7M=N7+INCM-1
      J=0
      IM=N7-1
      K=0
      DO 40 I=N7,N7M
      J=J+1
      IF(IPUNCH.EQ.0) GO TO 30
      IF(MOD(J,15).NE.1) GO TO 30
      IL=IM+1
      IM=IM+15
      IF(IM.GT.N7M) IM=N7M
      K=K+1
      WRITE (7,60) K,(IA(L),L=IL,IM)
   30 IF(IPRINT.NE.0) WRITE (6,70, J,IA(I)
   40 CONTINUE
   60 FORMAT (16I5)
   70 FORMAT (1X,I8,5X,I8)
      J=9999
      IF(IPUNCH.NE.0) WRITE (7,60) J
      STOP
      END
      SUBROUTINE FIELD (IA)
      DIMENSION IA(1)
      DATA LAST/0/,MAX/320000B/,MIN/55000B/
      NOW=((LOCF(IA(1))/100B)+1)*100B
      IF(NOW.GT.MAX) GO TO 100
      IF(NOW.LT.MIN) NOW=MIN
      IF(NOW.GT.LAST) GO TO 50
      IF(NOW.LT.LAST-10000) GO TO 50
      GO TO 75
   50 CALL AREL (NOW)
      LAST=NOW
   75 RETURN
  100 WRITE (6,200) NOW
  200 FORMAT (36H0*** ERROR  EXCESSIVE CORE REQUESTED  /
     *       11X,12HJOB REQUESTS, I7, 7H WORDS  )
      STOP
      END
          IDENT  LOAD  (I,LH,LL,IN,IF)
          ENTRY  LOAD
          VFD    24/4HLOAD,36/5
LOAD      SB7    1
          SB1    1
          SA2    A1+B1
          SA3    A2+B1
          SA4    A3+B1
          SA5    A4+B1
          MX1    1
          SX2    X2
          SX3    X3
          SX4    X4
          SX1    X1
          SX5    X5
          SB3    X2              LH TO B3
          IAS    X2-X3           LH=LL (MASK WIDTH LESS 1)
          SB1    X3-1            NO OF LFT SHFTS FOR IN
          SB2    X5              NO OF RGHT SHFTS TO BUILD MASK
          LX4    B1,X4           POSN IN
          SB4    60              B4=60
          MXA    B2,X0           FORM MASK
          SB4    B4-B1           LFT SHFTS TO PUT (IF) IN LO BITS
          LAA    B3,X6           POSN MASK
          BXA7   X6*X1           GET OLD VALUE
          BXA1   X1-X7           ZERO-OUT OLD VALUE
          LX7    B4,X7           POSN (IF) IN LO BITS
```

```
          BXA     X1*X4           MASK IN NEW VALUE
          SA7     B5              STORE IF
          SA6     A1              STORE I
          EQ      LOAD            ALL DONE
          END
          IDENT   UNLOAD  (I,LH,LL,IN)
          ENTRY   UNLOAD
          VFD     36/6HUNLOAD,24/4
UNLOAD    BSSZ    1
          SB1     1
          SA2     A1+B1
          SA3     A2+B1
          SA4     A3+B1
          MX0     1
          SA2     X2
          SA3     X3
          SA1     X1
          SB4     X4
          SB3     X2              LH TO B3
          IX5     X2-X3           LH-LL (MASK WIDTH LESS 1)
          SB2     X5              NO OF RGHT SHFTS TO BUILD MASK-
          IX7     X5-X2           -LL TO X7
          BX6     B2,X0           FORM MASK
          SB1     X7+B1           NO OF LFT SHFTS TO PACK IN RGHT
          LX6     B3,X6           POSN MASK
          BX7     X6*X1           EXTRACT VALUE
          LX6     B1,X7           SHFT IN TO LO ORDER POSN
          SA6     B4              STORE IN
          EQ      UNLOAD          ALL DONE
          END
      SUBROUTINE ELIN (NODIN,MXJ,ID,NMNP,NSEC)
      COMMON /FILES/  III,II2,II3
      COMMON /CONTRL/ NLOAD,NCOMB,IMODE,IPLOT,IRSTRT,ICORE,INSYS,INUM,
     1                INEX,IRENUM,NTOTD
      COMMON /ERRS/   IERR(25),KARD,KARDS,NSECT,KOLS(2),IERR,NERRS,
     1                MAINI
      COMMON /OPUT/   MED1(8),MED2(8),NLINE,NPAGE,LSTOFF,LSPS,LSP
      COMMON /STGI/   MSEP(25),LSEP(25),IORD(25),ICARD(8),NSEP,NSEP1,
     1                LOSEP,MXERR,NSP,NK,LBLNK,IT
      COMMON /MISC/   IASTER,ZERO,RAD
      COMMON /KRKCOM/ KRK,ANT
      DIMENSION       NODIN(1),MXJ(1),ID(1),IJ(8),JI(24)
      DIMENSION       IAT(8),IX(8)
      NODM=NMNP-NTOTD
      KARD=1
      MAXN=8
      IEL=0
   15 READ (5,115) IELN,IAT
  115 FORMAT (I5,15X,8I3)
      IF(EOF(5)) 99,20
   20 IF(IELN.LE.0) GO TO 99
      IF(IELN.GT.IEL+1) GO TO 60
      IF(IELN.LE.IEL) GO TO 98
   25 READ (5,120) IJ
  120 FORMAT (8I4)
      IEL=IELN
      IH=0
      DO 30 I=1,MAXN
      IF(IJ(I).LE.0) GO TO 35
      IH=IH+1
      IX(IH)=IAT(IH)
      IF(IX(IH).EQ.0) IX(IH)=1
   30 CONTINUE
   35 M=0
      DO 45 I=1,IH
      L=IJ(I)
      IF(L.GT.NMNP) GO TO 97
      M=M+1
      JI(M)=L
   45 CONTINUE
      IF(M.GT.0) CALL PACK (JI,MXJ,ID,M)
      GO TO 15
```

```
   60 IF(IEL.EQ.0) GO TO 96
      NG=IELN-IEL-1
      DO 80 N=1,NG
      M=0
      IEL=IEL+1
      DO 70 I=1,IM
      IJ(I)=IJ(I)+IX(I)
      L=IJ(I)
      IF(L.GT.NMNP) GO TO 97
      M=M+1
      JI(M)=L
   70 CONTINUE
      IF(M.GT.0) CALL PACK (JI,MXJ,ID,M)
   80 CONTINUE
      GO TO 25
   96 WRITE (6,600)
  600 FORMAT (50H0** ERROR **  FIRST ELEMENT OMITTED.                    )
      STOP 1
   97 WRITE (6,610) IEL
  610 FORMAT (50H0** ERROR **  NODE NUMBER OUT OF RANGE. ELEM NO.  ,I5 )
      STOP 2
   98 WRITE (6,620) IELN
  620 FORMAT (50H0** ERROR **  ELEMENT OUT OF SEQUENCE.  ELEM NO.  ,I5 )
      STOP 3
   99 RETURN
      END
      SUBROUTINE PACK (JI,MXJ,IJ,N)
      COMMON /TPOLGY/ II,JJ,J2,J3,NEXT,LAST
      COMMON /CONTRL/ IDUMMY(7),INUM
      DIMENSION       JI(1),MXJ(1),IJ(1)
      NN=N
      DO 20 I=1,N
      J=JI(I)
      IF(MXJ(J).EQ.0) GO TO 20
      MXJJ=MXJ(J)
      CALL UNLOAD (MXJJ,45,31,KI)
      CALL UNLOAD (MXJJ,30,16,KK)
      CALL UNLOAD (MXJJ,15, 1,KJ)
      L=I
      IF(KJ.EQ.0) GO TO 10
      NN=NN+1
      L=NN
   10 JI(L)=KI
      IF(KK.EQ.0) GO TO 20
      NN=NN+1
      JI(NN)=KK
   20 CONTINUE
   30 IF(NEXT+NN+NN.LT.LAST) GO TO 50
      LAST=LAST+INUM
      CALL FIELD (IJ(LAST))
      IL=LAST-INUM+1
      DO 40 I=IL,LAST
      IJ(I)=0
   40 CONTINUE
      GO TO 30
   50 DO 100 I=1,NN
      II=JI(I)
      DO 90 J=1,NN
      JJ=JI(J)
      IF(II.EQ.JJ) GO TO 90
      CALL PK (IJ(II))
   90 CONTINUE
  100 CONTINUE
      RETURN
      END
          IDENT   PK     (IJ(I))
          ENTRY   PK
          VFD     12/2HPK,48/1
PK        BSS     1
          USE     /TPOLGY/
I         BSS     1
JI        BSS     1
```

```
J2          BSS     1
J3          BSS     1
NEXT        BSS     1
            USE
            USE     /S/
NUMY        BSS     1
MM          BSS     1
            USE
            SA2     I               I TO X2
            SA4     NEXT            NEXT TO X4
            SA3     J1              J1 TO X3
            SA1     X1
            SB1     B1
            LXC     14              14 BT LO MSK IN X0
            SB3     46              INITIAL SHFT VALUE
            SB2     X2              1 TO B2
            SB4     14              B4=14
            SB1     B1-B2           ADDR(IJ(0)) TO B1
            SB7     B0              INIT LOOP COUNTER
            SA5     I
            SA2     MM              MM TO X2
            SX7     X4+1            FORM NEW NEXT IN X7
            SB2     B3              INIT B2
            SB6     X3              J1 TO B6
LOOP        LXB     B4,X1           SHFT IJ INTO X6
            SB7     B7+A5
            BXB     X6*X0           EXTRACT J
            SB5     X6              MOVE J TO B5
            ZR      X5,PACK12       IS J ZERO
            EQ      B5,B6,PK        DONE IF J=J1
            LXB     14              SHFT IJ
            SB2     B2-B4           DEC SHFT REG
            BXB     X6*X0           EXTRACT J
            SB5     X6              MOVE J TO B5
            ZR      X5,PACK12       IS J ZERO
            EQ      B5,B6,PK        DONE IF J=J1
            LXB     14              SHFT IJ
            SB2     B2-B4           DEC SHFT REG
            BXB     X6*X0           EXTRACT J
            SB5     X6              MOVE J TO B5
            ZR      X5,PACK3        IS J ZERO
            EQ      B5,B6,PK        DONE IF J=J1
            SB1     B1+X1           FETCH NEXT IJ TO X1
            SB2     B3              INIT SHFT REG
            EQ      LOOP            BACK TO TOP TO CONSIDER NEXT IJ
PACK3       LX4     42              SHFT CURRENT NEXT
            SA7     A4              STORE NEW NEXT
            BXB     X6+X4           PACK CURRENT NEXT INTO ADDRESS PORTION
PACK12      BXB     X6+X3           PACK J1 INTO IJ
            LXB     B2,X6           REPOSN IJ
            SB6     X2              MOVE MM TO B6
            SA6     A1              STORE UPDATED IJ
            GT      B6,B7,PK        DONE IF MM.GT.LOOP COUNT
            SX7     B7              NEW MM TO X7
            SA7     A2              STORE NEW MM
            EQ      PK              DONE
            END
            IDENT   UNPK  (IJ(I))
            ENTRY   UNPK
            VFD     24/4HUNPK,36/1
UNPK        BSS     1

            USE     /TPOLGY/
I           BSS     1
J1          BSS     1
J2          BSS     1
J3          BSS     1
NEXT        BSS     1
            USE
            MX0     14              BUILD MASK
            SA1     X1
            LX0     14              14 BT LO MASK
```

```
      LA1     I4              SHFT IJ(I)
      BA6     X1*X0           GET J1
      LA1     I4              SHFT  IJ(I)
      SA6     J1              STORE J1
      BA7     X1*X0           GET J2
      LA1     I4              SHFT IJ(I)
      SA7     J2              STORE J2
      BA6     X1*X0           GET J3
      LA1     I8              SHFT IJ(I)
      SA6     J3              STORE J3
      SB7     X1              GET NEXT
      SA7     B7              PUT NEXT IN X7
      SA7     NEXT            STORE NEXT
      EQ      UNPK            DONE
      END
      SUBROUTINE SCHEME (IG,IC,IDEG,IDIS,IW,NEW,ICC,ILD,IPP)
      DIMENSION  IG(1),IC(1),IDEG(1),IDIS(1),IW(1),NEW(1),ICC(1),ILD(1),
     *           IPP(1),NODESL(100)
      COMMON /TPOLGY/ II,JI(3),NEXT,LAST,KORIG,KNEW
      COMMON /S/ NN,MM,IM,IH                                          *
      COMMON /P/ IH0,IME                                              *
      COMMON /TIME/ STIME,NCM                                         *
      EQUIVALENCE (IM,ATIME)                                          *
      NI=MIND(IG,NN)
      NUM=1
      NOM=2
      IO=2
      IP=0
      CALL DEGREE (IG,IDEG)
      MODD=MODF(IDEG,IPP)                                             *
      NCM=COMPNT(IG,IC,IDEG,IW,ICC)
      MAXD=MAXDGR(0,IC,IDEG)                                          *
      MM=MAXD                                                         *
      DO 30 I=1,NN                                                    *
      NEW(I)=I                                                        *
   30 ILD(I)=I                                                        *
      IS=MAXBND(0,IG,IC,IDEG,NEW,ILD)
      KORIG=IS                                                        *
      IH0=IM                                                          *
      DO 35 I=1,NN                                                    *
      NEW(I)=0                                                        *
   35 ILD(I)=0                                                        *
      DO 500 NC=1,NCM                                                 *
      MI=MINDEG(NC,IC,IDEG)                                           *
      MAD=MI                                                          *
      IF(NUM.EQ.0) GO TO 91
   90 MA=MAXDGR(NC,IC,IDEG)                                           *
      MAD=MI+((MA-MI)*NUM)/NOM                                        *
      MAD=MIND(MAD,MODD-1)                                            *
      MAD=MAX0(MAD,MI)                                                *
   91 CALL DIAM(NC,MAD,NL,NODESL,MAXLEV,IG,IC,IDEG,IDIS,IW,ICC)
      JMAX=MIND(NI,NL)                                                *
      IM=9000000000                                                   *
      IMM=IM                                                          *
      DO 400 J=1,JMAX                                                 *
      CALL RELABL(1,NODESL(J),IG,IC,IDEG,IDIS,IW,NEW,ICC,ILD)
      IH=MAXBND(NC,IG,IC,IDEG,NEW,ILD)
      IE=ICC(NC+1)-1                                                  *
      IF(IM-IH) 400,350,300                                           *
  300 IM=IH                                                           *
      IMM=IM                                                          *
      IJ=J                                                            *
      GO TO 400                                                       *
  350 IF(IMM.LE.IH) GO TO 400                                         *
      IMM=IH                                                          *
      IJ=J                                                            *
  400 CONTINUE
      CALL RELABL(1,NODESL(IJ),IG,IC,IDEG,IDIS,IW,NEW,ICC,ILD)
  500 CONTINUE                                                        *
      CALL STACK(IDEG,NEW,ILD,IW)                                     *
      IH=MAXBND(0,IG,IC,IDEG,NEW,ILD)
      IF(IH-IS) 715,742,744                                           *
```

```
742 IF(IM.LT.IM0) GO TO 715                                              *
744 DO 712 I=1,NN                                                        *
    ILD(I)=I                                                             *
712 NEW(I)=I                                                             *
    CALL STACK(IDEG,NEW,ILD,IM)                                          *
    IH=IS                                                                *
    IH=IM0                                                               *
715 IHE=IM                                                               *
    CALL REVERS(NEW,ILD)                                                 *
    IB=MAXBND(0,IG,IC,IDEG,NEW,ILD)                                      *
717 IF(IM.LT.IHE) GO TO 720                                              *
    CALL REVERS(NEW,ILD)                                                 *
    IH=MAXBND(0,IG,IC,IDEG,NEW,ILD)                                      *
720 IHE=IM                                                               *
    NNEW=IB                                                              *
600 RETURN                                                               *
    END                                                                  *
    SUBROUTINE DEGREE (IG,IDEG)
    DIMENSION IG(1),IDEG(1)
    COMMON /TPOLGY/ II,J1,J2,J3,NEXT
    COMMON /S/ NN,MM,IH,IB
    DO 50 I=1,NN
    NEXT=1
    J=0
 10 CALL UNPK (IG(NEXT))
    IF(J3.EQ.0) GO TO 20
    J=J+3
    GO TO 10
 20 IF(J2.EQ.0) GO TO 30
    J=J+2
    GO TO 40
 30 IF(J1.EQ.0) GO TO 40
    J=J+1
 40 IDEG(I)=J
    MM=MAX0(MM,J)
 50 CONTINUE
    RETURN
    END
    FUNCTION MODE(IDEG,MODD)                                             *
    COMMON /S/ NN,MM                                                     *
    DIMENSION IDEG(1),MODD(1)                                            *
    DO 10 I=1,MM                                                         *
 10 MODD(I)=0                                                            *
    DO 20 I=1,NN                                                         *
    K=IDEG(I)                                                            *
    IF(K.EQ.0) GO TO 20
    MODD(K)=MODD(K)+1
 20 CONTINUE
    MODE=0                                                               *
    MAX=0                                                                *
    DO 30 I=1,MM                                                         *
    K=MODD(I)                                                            *
    IF(K.LE.MAX) GO TO 30                                                *
    MAX=K                                                                *
    MODE=I                                                               *
 30 CONTINUE                                                             *
    RETURN                                                               *
    END                                                                  *
    FUNCTION COMPNT (IG,IC,IDEG,IW,ICC)
    DIMENSION IG(1),IC(1),IDEG(1),IW(1),ICC(1)
    COMMON /S/ NN,MM,IH,IB                                               *
    COMMON /TPOLGY/ II,JJ(3),NEXT
    DO 100 I=1,NN                                                        *
    ICC(I)=0                                                             *
    IC(I)=0                                                              *
100 CONTINUE                                                             *
    NC=0                                                                 *
    ICC(1)=1                                                             *
105 DO 110 I=1,NN                                                        *
    IF(IC(I).EQ.0) GO TO 120
    COMPNT=NC
110 CONTINUE
```

```
      RETURN
  120 NC=NC+1
      KI=0
      KO=1
      IW(1)=I
      IC(I)=NC
      IF(NC.LE.0)  GO TO 130
      IS=ICC(NC)+1
      ICC(NC+1)=IS
  130 KI=KI+1
      II=IW(KI)
      N=IDEG(II)
      IF(N.EQ.0) GO TO 105
      NEXT=II
  140 CALL UNPK (IG(NEXT))
      DO 180 I=1,3
      IA=JJ(I)
      IF(IA.EQ.0) GO TO 200
      IF(IC(IA).NE.0) GO TO 180
      IC(IA)=NC
      KO=KO+1
      IW(KO)=IA
      IS=ICC(NC+1)+1
      ICC(NC+1)=IS
  180 CONTINUE
      GO TO 140
  200 IF(KO.LE.KI) GO TO 105
      GO TO 130
      END
      FUNCTION MAXDGR(NC,IC,IDEG)
      DIMENSION IC(1),IDEG(1)
      COMMON /S/ NN,MM,IH,IB
      M=0
      IF(NC.NE.0) GO TO 40
      DO 30 I=1,NN
      IF(IDEG(I).LE.M) GO TO 30
      M=IDEG(I)
   30 CONTINUE
      GO TO 100
   40 DO 50 I=1,NN
      IF(IC(I).NE.NC) GO TO 50
      IF(IDEG(I).LE.M) GO TO 50
      M=IDEG(I)
   50 CONTINUE
  100 MAXDGR=M
      RETURN
      END
      FUNCTION MAXRND(NC,IG,IC,IDEG,NEW,ILD)
      DIMENSION IG(1),IC(1),IDEG(1),NEW(1),ILD(1)
      COMMON /S/ NN,MM,IH,IB
      COMMON /TPOLGY/ II,JJ(3),NEXT
      IH=0
      M=0
      DO 100 I=1,NN
      MX=0
      IA=NEW(I)
      IF(NC.EQ.0) GO TO 50
      IF(IA.EQ.0) GO TO 100
      IF(NC.NE.IC(IA)) GO TO 100
   50 N=IDEG(IA)
      IF(N.LE.0) GO TO 100
      NEXT=IA
   60 CALL UNPK (IG(NEXT))
      DO 80 J=1,3
      II=JJ(J)
      IF(II.EQ.0) GO TO 90
      IH=MAX0(0+1-ILD(II))
      IF(IH.GT.MX) MX=IH
   80 CONTINUE
      GO TO 90
   90 IF(MX.GT.M) M=MX
      IH=IH+MX
```

```
100 CONTINUE
    MAXMND=M
    RETURN
    END
    FUNCTION MINDEG(NC,IC,IDEG)
    DIMENSION IC(1),IDEG(1)
    COMMON /S/ NN,MM,IM,IH
    M=IDEG(1)
    IF(NC.NE.1) GO TO 40
    DO 30 I=1,NN
    IF(M.LE.IDEG(I)) GO TO 30
    M=IDEG(I)
 30 CONTINUE
    GO TO 100
 40 DO 60 I=1,NN
    IF(IC(I).NE.NC) GO TO 60
    IF(M.LE.IDEG(I)) GO TO 60
    M=IDEG(I)
 60 CONTINUE
100 MINDEG=M
    RETURN
    END
    SUBROUTINE DIAM(NC,MAXDEG,NL,NODESL,MAXLEV,
   *     IG,IC,IDEG,IDIS,IW,ICC)
    DIMENSION IG(1),IDIS(1),IW(1),ICC(1),IC(1),IDEG(1)
    DIMENSION NODESL(1)
    COMMON /S/ NN,MM,IM,IH
    NL=0
    MAXLEV=10000
    DO 110 I=1,NN
    IF(NC.NE.IC(I)) GO TO 100
    IF(MAXDEG.LT.IDEG(I)) GO TO 100
105 MD=IDIST(I,NL,MAXLEV,IG,IC,IDEG,IDIS,IW,ICC)
    IF(MD.LE.0) GO TO 115
 56 IF(NL-MAXLEV)58,64,100
 58 MAXLEV=NL
    NL=1
    NODESL(1)=I
    GO TO 100
 64 IF(NL.GE.100) GO TO 100
    NL=NL+1
    NODESL(NL)=I
100 CONTINUE
    GO TO 110
115 NL=1
    NODESL(1)=I
    MAXLEV=0
110 RETURN
    END
    FUNCTION IDIST(NS,ML,MAXLEV,IG,IC,IDEG,IDIS,IW,ICC)
    DIMENSION IG(1),IC(1),IDEG(1),IDIS(1),IW(1),ICC(1)
    COMMON /S/ NN,MM,IM,IH
    COMMON /TPOLGY/ II,JJ(3),NEXT
    ICN=IC(NS)
    NAC=ICC(ICN+1)-ICC(ICN)
    DO 50 I=1,NN
    IF(IC(I).NE.IC(NS)) GO TO 50
    IDIS(I)=0
 50 CONTINUE
    LL=1
    L=0
    KI=0
    KO=1
    ML=0
    IW(1)=NS
    IDIS(NS)=-1
130 KI=KI+1
    IF(KI.NE.LL) GO TO 135
132 L=L+1
    LL=KO+1
    K=KO-KI+1
    IF(K.LE.ML) GO TO 135
```

```
133 ML=K                                                                *
    IF(ML.GT.MAXLEV) GO TO 220
135 II=IW(KI)
    N=IDEG(II)                                                          *
    IF(N.EQ.0) GO TO 215
    NEXT=II
140 CALL UNPK (IG(NEXT))
    DO 190 I=1,3
    IA=JJ(I)
    IF(IA.EQ.0) GO TO 200
    IF(IDIS(IA).NE.0) GO TO 190
    IDIS(IA)=L
    KO=KO+1                                                             *
    IW(KO)=IA                                                           *
190 CONTINUE
    GO TO 140
200 IF(KO.LT.NNC) GO TO 130
205 IDIST=L                                                             *
    IDIS(NS)=0                                                          *
    K=KO-KI                                                             *
    IF(K.GT.ML) ML=K
    GO TO 230
215 L=0                                                                 *
    GO TO 205                                                           *
220 IDIST=1                                                             *
230 RETURN                                                              *
    END                                                                 *
    SUBROUTINE RELABL(NS,NODES,IG,IC,IDEG,IDIS,IW,NEW,ICC,ILD)
    INTEGER X                                                           *
    DIMENSION IG(1),IC(1),IDEG(1),IDIS(1),IW(1),NEW(1),ICC(1)
    DIMENSION ILD(1)                                                    *
    DIMENSION NODES(1),IAJ(50)                                          *
    COMMON /S/ KN,MM,IH,IB                                              *
    COMMON /TPOLGY/ II,JI(3),NEXT
    I=NODES(1)                                                          *
    ICN=IC(I)                                                           *
    NT=ICC(ICN)-1                                                       *
    DO 50 I=1,NN                                                        *
    IF(IC(I).NE.ICN) GO TO 50
 40 IDIS(I)=0
 50 CONTINUE
    DO 100 J=1,NS                                                       *
    JJ=NODES(J)                                                         *
    IDIS(JJ)=-1                                                         *
    JT=J+NT                                                             *
    NEW(JT)=JJ                                                          *
100 ILD(JJ)=JT
    KI=NT                                                               *
    KO=NS+NT                                                            *
    LL=KO                                                               *
    L=1                                                                 *
    J=KO                                                                *
    NNC=ICC(ICN+1)-1                                                    *
130 KI=KI+1
    IF(KI.NE.LL) GO TO 135
132 L=L+1
    LL=KO+1                                                             *
135 II=NEW(KI)
    N=IDEG(II)                                                          *
    IF(N.EQ.0) GO TO 255
    IJ=0
    NEXT=II
140 CALL UNPK(IG(NEXT))
    DO 190 I=1,3
    IA=JI(I)
    IF(IA.EQ.0) GO TO 200
    IF(IDIS(IA).NE.0) GO TO 190
150 IJ=IJ+1
    IDIS(IA)=L                                                          *
    KO=KO+1                                                             *
    IAJ(IJ)=IA                                                          *
    IW(IJ)=IDEG(IA)                                                     *
```

```
190 CONTINUE
    GO TO 140
200 IF(IJ-1) 250,210,220
210 J=KO
    IZ=IAJ(1)                                                          *
    NEW(KO)=IZ                                                         *
    ILD(IZ)=KO                                                         *
    GO TO 250                                                          *
220 X=0
221 DO 230 I=2,IJ
    IF(IW(I).GE.IW(I-1)) GO TO 230
    X=IW(I)                                                            *
    IW(I)=IW(I-1)                                                      *
    IW(I-1)=X                                                          *
225 X=IAJ(I)
    IAJ(I)=IAJ(I-1)                                                    *
    IAJ(I-1)=X                                                         *
230 CONTINUE
    IF(X.GT.0) GO TO 220
235 DO 240 I=1,IJ
    J=J+1                                                              *
    IZ=IAJ(I)                                                          *
    NEW(J)=IZ                                                          *
    ILD(IZ)=J                                                          *
240 CONTINUE
250 IF(KO+LI.NE.NNC)GO TO 130
255 RETURN
    END                                                                *
    SUBROUTINE STACK(IDEG,NEW,ILD,IW)                                  *
    DIMENSION IDEG(1),NEW(1),ILD(1),IW(1)                              *
    COMMON /S/ NN                                                      *
    COMMON //ERO/ KT                                                   *
    KT=0                                                               *
    NN1=NN-1                                                           *
    DO 10 I=1,NN                                                       *
    IF(IDEG(I).GT.0) GO TO 10                                          *
    KT=KT+1                                                            *
    IW(KT)=ILD(I)                                                      *
 10 CONTINUE                                                           *
    IF(KT.LE.0) GO TO 70                                               *
    CALL SORT(IW,KT)                                                   *
    DO 40 L=1,KT                                                       *
    I=IW(L)-L+1                                                        *
    K=NEW(I)                                                           *
    IF(I.GE.NN) GO TO 30                                               *
    DO 20 J=I,NN1                                                      *
 20 NEW(J)=NEW(J+1)                                                    *
 30 NEW(NN)=K                                                          *
 40 CONTINUE                                                           *
 70 DO 80 I=1,NN                                                       *
    K=NEW(I)                                                           *
 80 ILD(K)=I                                                           *
    RETURN                                                             *
    END                                                                *
    SUBROUTINE SORT(LIST,NL)                                           *
    DIMENSION LIST(1)                                                  *
    IF(NL.LE.1) RETURN                                                 *
    NL1=NL-1                                                           *
    DO 20 I=1,NL1                                                      *
    K=NL-I                                                             *
    KFLAG=0                                                            *
    DO 10 J=1,K                                                        *
    IF(LIST(J).LE.LIST(J+1)) GO TO 10                                  *
    KFLAG=1                                                            *
    L=LIST(J)                                                          *
    LIST(J)=LIST(J+1)                                                  *
    LIST(J+1)=L                                                        *
 10 CONTINUE                                                           *
    IF(KFLAG.EQ.0) RETURN                                              *
 20 CONTINUE                                                           *
    RETURN                                                             *
    END                                                                *
```

```
      SUBROUTINE REVERS(NEW,ILD)
      DIMENSION NEW(1),ILD(1)
      COMMON /S/ NN
      COMMON //ZERO/ KT
      J=(NN-KT)/2
      LL=NN-KT+1
      DO 10 I=1,J
      L=LL-I
      K=NEW(L)
      NEW(L)=NEW(I)
   10 NEW(I)=K
      DO 20 I=1,NN
      K=NEW(I)
   20 ILD(K)=I
      RETURN
      END
```

```
      SUBROUTINE ELMT02(N,MA,NDIM,NDF,NEL,NEL1,NSTF,NSIZV,NVEC,MCT,DM,D,  ELM   1C
     X XYZ,IX,F,FORCE,ESTIF,U,VECT,ISW)                                   ELM   2C
      REAL LAHL                                                           ELM   3C
      LOGICAL NPR,NPL,NOPRNT                                              ELM   4C
      DIMENSION AH(9,33),CC(9),SG0(3,3),D(63,1),XX(3),S1(7),S2(7),S3(7),  ELM   5C
     1   EPS(6),SIG(6),TAB1(4),TAB2(4),THED(6),DV(3,3),STUW(4,4),AW(3),   ELM   6C
     2   FORCE(NSTF,2),ESTIF(NSTF,NSTF)                                   ELM   7C
      DIMENSION UDL(6,16)                                                 ELM   8C
      COMMON/SHAPE/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)      ELM   9C
      COMMON/DYNAMD/TIME,NSIG(7),NI,NSTEP,DT,NUMPLT,NEDATA(20,3),NPR,NPL  ELM  10C
      COMMON/LABELS/ LABL(6),XHED(3),XM,FHED(6),FM,UHED(6),UM,RHED(6),RM  ELM  11C
     1,AWORD1,AWORD2,AWORD3,HEAD(12),START,CEASE,IPG,NSTR,WORD(30)         ELM  12C
      COMMON/TAPES/ ITP5,ITP6                                             ELM  13C
      COMMON/VISDAT/ DTP,NM,ISZM,DUL(6,20),UL(6,20),UDDL(6,20),C6         ELM  14C
      DATA S1/ 0.,-1.,  1.,  0.,  0.,  0.,  0./                           ELM  15C
      DATA S2/ 0.,  0.,  0.,-1.,  1.,  0.,  0./                           ELM  16C
      DATA S3/ 0.,  0.,  0.,  0.,  0.,-1.,  1./                           ELM  17C
      DATA SM,FM,BLANK/6HSTRESS,6HSTRAIN,6H      /                        ELM  18C
      DATA TAB1/10H(1H0,I7,0P,1H0,10HF12.3,1P6E,5M12.3)/                  ELM  19C
      DATA TAB2/10H(8X,A5,I7,,1H0,10H(12X),1P6E,5M12.3)/,AW/1H1,1H2,1H3/  ELM  20C
      DATA THED/6H   XX-,6H   XY-,6H   YY-,6H   YZ-,6H   ZZ-,6H   ZX-/  ELM  21C
      NSIDE =(NEL+1)/NDIM**2                                              ELM  22C
      GO TO (1,2,3,3,5,3),ISW                                             ELM  23C
C**** LINEAR ISOTROPIC 3-D ELASTIC CHARACTERIZATION                       ELM  24C
C**** FOR TWO DIMENSIONAL PLANE STRESS PROBLEMS AA SHOULD BE SET TO 1.0   ELM  25C
1     READ(ITP5,1000) E,XNU,D(3,MA),AA                                    ELM  26C
      IF (AA .NE. 1.0) AA = 2.0                                           ELM  27C
      D(1,MA) = E/(1.-AA*XNU)/(1.+AA)                                     ELM  28C
      D(2,MA) = E/2./(1.+XNU)                                             ELM  29C
      WRITE(ITP6,2000)E,XNU,D(3,MA),AA                                    ELM  30C
      D(4,MA) = D(1,MA) - 2.*D(2,MA)/(1.+AA)                              ELM  31C
      D(5,MA) = D(4,MA) + 2.*D(2,MA)                                      ELM  32C
      C6 = 1.                                                             ELM  33C
      CON1 = 45./ATAN(1.)/2.                                              ELM  34C
C**** SET UP HEADS FOR PRINTOUTS                                          ELM  35C
      TAB1(2) = AW(NDIM)                                                  ELM  36C
      TAB2(2) = AW(NDIM-1)                                                ELM  37C
      CALL INTEGL(2 ,0,NDIM,LINT,STUW)                                    ELM  38C
      RETURN                                                              ELM  39C
2     CALL CKBRIK(NDIM,NEL,DM)                                            ELM  40C
      DM = DM + 1.E-25                                                    ELM  41C
      IF(NEL.EQ.4.AND.NDIM.EQ.2.OR.NEL.EQ.8.AND.NDIM.EQ.3) RETURN         ELM  42C
      DM = -1.0                                                           ELM  43C
      WRITE(ITP6,2003)                                                    ELM  44C
      RETURN                                                              ELM  45C
C**** SET UP MATERIAL PROPERTIES                                          ELM  46C
3     XK = D(1,MA)                                                        ELM  47C
      GG = D(2,MA)                                                        ELM  48C
      IF(ISW.NE.3) GO TO 4                                                ELM  49C
      CALL VSTIF(NDIM,NDF,NEL,NSIDE,ESTIF(NDF+1,1),FORCE(1,2),STUW,LINT)  ELM  50C
      CALL ASSEMH(NDIM,NDF,NEL,NSTF,ESTIF(NDF+1,1),ESTIF,FORCE(1,2),GG,   ELM  51C
     1   D(4,MA),C6*D(3,MA))                                              ELM  52C
4     SX = NDIM - 3.0                                                     ELM  53C
      CALL BRICK2(0.,0.,SX,NDIM,NEL,NSIDE)                                ELM  54C
      DO 400 I = 1,NDIM                                                   ELM  55C
      DO 400 J = 1,NDIM                                                   ELM  56C
400   SG0(I,J) = SG(I,J)                                                  ELM  57C
      MM = NDIM*NDF                                                       ELM  58C
      NSTF = NSTF + MM                                                    ELM  59C
      DO 401 I = 1,MM                                                     ELM  60C
      CC(I) = 0.                                                          ELM  61C
      DO 401 J = 1,NSTF                                                   ELM  62C
401   AH(I,J) = 0.0                                                       ELM  63C
      CALL INSTIF(AH,CC,SG0,STUW,LINT,NDF,NDIM,NEL,NSIDE,MM,              ELM  64C
     1   D(5,MA),D(4,MA),GG)                                              ELM  65C
      NSTF = NEL*NDF+1                                                    ELM  66C
      K = 1                                                               ELM  67C
      IF(ISW.NE.3) K = 3                                                  ELM  68C
      CALL INCOMP(AH(1,NSTF),AH,CC,ESTIF,FORCE,MM,NSTF,K)                 ELM  69C
      IF(ISW.EQ.3) RETURN                                                 ELM  70C
      K = 1                                                               ELM  71C
      DO 402 I = 1,NEL                                                    ELM  72C
```

```
      DO 402 J = 1,NDF                                                  FLM 73C
      FORCE(K,1) = 0.0                                                  FLM 74C
      UDL(K,1) = UL(J,I)                                                FLM 75C
402   K = K + 1                                                         FLM 76C
      CALL INCOMP(AR(1,NSIP),AM,CC,ESTIF,UDL,MM,NSIF,2)                 FLM 77C
      K = 1                                                             FLM 78C
      DO 403 I = 1,NDIM                                                 FLM 79C
      DO 403 J = 1,NDF                                                  FLM 80C
      UL(J,I,NEL) = CC(K)                                               FLM 81C
403   K = K + 1                                                         FLM 82C
      NSIP = NEL * NDIM                                                 FLM 83C
      NL = (NDIM*(NDIM+1))/2                                            FLM 84C
      XK  = D(4,MA)                                                     FLM 85C
      GG  = 2.0*D(2,MA)                                                 FLM 86C
      DO 413 II = 1,LINT                                                FLM 87C
      CALL BRICK2(STUW(1,II),STUW(2,II),STUW(3,II),NDIM,NEL,NSIDE)      FLM 88C
      CALL INSHAP(STUW(1,II),SG,SHAPE(1,NEL+1),NDIM)                    FLM 89C
      DO 411 K = 1,NL                                                   FLM 90C
      UDL(K,II) = SIG(K)                                                FLM 91C
411   UDL(K,II+LINT) = EPS(K)                                           FLM 92C
      IF(ISW.EQ.4) GO TO 413                                            FLM 93C
C**** FORM DISPLACEMENT GRADIENTS                                       FLM 94C
      DO 405 I = 1,NDF                                                  FLM 95C
      DO 405 J = 1,NDIM                                                 FLM 96C
      TEMP = 0.                                                         FLM 97C
      DO 404 K = 1,NSIP                                                 FLM 98C
404   TEMP = TEMP + SHAPE(J,K)*UL(I,K)                                  FLM 99C
405   DV(I,J) = TEMP/XJAC                                               FLM100C
C**** FORM STRESS - STRAINS                                             FLM101C
      DO 407 I = 1,NDIM                                                 FLM102C
      XX(I) = 0.                                                        FLM103C
      DO 406 J = 1,NEL                                                  FLM104C
406   XX(I) = XX(I) + X(I,J)*SHAPE(4,J)                                 FLM105C
407   UDL(I,II) = XX(I)                                                 FLM106C
      EPS(1) = DV(1,1)                                                  FLM107C
      EPS(2) = 0.5*(DV(1,2)+DV(2,1))                                    FLM108C
      EPS(3) = DV(2,2)                                                  FLM109C
      TEMP = EPS(1) + EPS(3)                                            FLM110C
      IF(NDIM.EQ.2) GO TO 408                                           FLM111C
      EPS(4) = 0.5*(DV(2,3)+DV(3,2))                                    FLM112C
      EPS(5) = DV(3,3)                                                  FLM113C
      EPS(6) = 0.5*(DV(1,3)+DV(3,1))                                    FLM114C
      TEMP = TEMP + EPS(5)                                              FLM115C
408   DO 409 I = 1,NL                                                   FLM116C
409   SIG(I) =  GG*EPS(I)                                               FLM117C
      DO 410 I = 1,NL,2                                                 FLM118C
410   SIG(I) = SIG(I) +  XK*TEMP                                        FLM119C
      WU = STUW(4,II)                                                   FLM120C
      TEMP = WU*D(3,MA)*XJAC                                            FLM121C
      II = 1                                                            FLM122C
      DO 412 I = 1,NEL                                                  FLM123C
      SX = SHAPE(1,I)*WU                                                FLM124C
      SY = SHAPE(2,I)*WU                                                FLM125C
      SM = SHAPE(4,I)*TEMP                                              FLM126C
      FORCE(II,  1) = FORCE(II  ,1) - SIG(1)*SX - SIG(2)*SY-SM*UDDL(1,I) FLM127C
      FORCE(II+1,1) = FORCE(II+1,1) - SIG(2)*SX - SIG(3)*SY-SM*UDDL(2,I) FLM128C
      IF(NDIM.EQ.2) GO TO 412                                           FLM129C
      SZ = SHAPE(3,I)*WU                                                FLM130C
      FORCE(II  ,1)=FORCE(II  ,1) - SZ*SIG(6)                           FLM131C
      FORCE(II+1,1)=FORCE(II+1,1) - SZ*SIG(4)                           FLM132C
      FORCE(II+2,1)=FORCE(II+2,1) - SZ*SIG(5)-SY*SIG(4)-SX*SIG(6)       FLM133C
     C                    -SM*UDDL(3,I)                                FLM134C
412   II = II + NDF                                                     FLM135C
413   CONTINUE                                                          FLM136C
      NSIP = NDIM * NDIM + 1                                            FLM137C
      DO 421 J = 1,NSIP                                                 FLM138C
      NOPRNT = NPR                                                      FLM139C
      IF(NS(G(J) .GT.0) NOPRNT = .TRUE.                                 FLM140C
      NPL = .FALSE.                                                     FLM141C
      IF(NUMPLT.LE.0) GO TO 41                                          FLM142C
      DO 40 NP = 1,NUMPLT                                               FLM143C
40    IF(NEDATA(NP,1).EQ.N.AND.NEDATA(NP,2).EQ.J) NPL = TRUE            FLM144C
```

```
41    IF(NOPRNT.AND..NOT.NPL) GO TO 421                                    FLM145C
      DO 414 I = 1,NDIM                                                    FLM146C
414   XX(I) = 0.0                                                          FLM147C
      DO 415 I = 1,NL                                                      FLM148C
      SIG(I) = 0.0                                                         FLM149C
415   EPS(I) = 0.0                                                         FLM150C
      DO 417 L = 1,LINT                                                    FLM151C
      E = (1.+SI(J)/STUW(1,L))*(1.+S2(J)/STUW(2,L))/4.0                    FLM152C
      IF(NDIM.EQ.3) E = E*(1.+S3(J)/STUW(3,L))/2.0                         FLM153C
      DO 416 I = 1,NDIM                                                    FLM154C
416   XX(I) = XX(I) + E*DVOL(I,L)                                          FLM155C
      DO 417 I = 1,NL                                                      FLM156C
      SIG(I) = SIG(I) + E*UDL(I,L)                                         FLM157C
417   EPS(I) = EPS(I) + E*UDL(I,L+LINT)                                    FLM158C
      IF(NOPRNT) GO TO 420                                                 FLM159C
C**** COMPUTE INVARIANTS FOR TWO D PROBLEMS                                FLM160C
      IF(NDIM.NE.2) GO TO 418                                              FLM161C
      CALL PSTRES(NDIM,SIG,SIG(4),SIG(5),SIG(6))                           FLM162C
      CALL PSTRES(NDIM,EPS,EPS(4),EPS(5),EPS(6))                           FLM163C
418   MCT = MCT - 1                                                        FLM164C
      IF(MCT.GT.0) GO TO 419                                               FLM165C
      WRITE(IPR,2002) U,HEAD,TIME,IPG,(XHED(I),XH,I=1,NDIM),               FLM166C
     X (IHED(I),SH,I=1,NL)                                                 FLM167C
      WRITE(IPR,2001) (BLANK,BLANK,I=1,NDIM),(IHED(I),EM,I=1,NL)           FLM168C
      IF(NDIM.EQ.2) WRITE(IPR,2004)                                        FLM169C
      IPG = IPG + 1                                                        FLM170C
      MCT = 15                                                             FLM171C
419   WRITE(IPR,1AH1) N,(XX(I),I=1,NDIM), SIG                              FLM172C
      WRITE(IPR,1AH2) DM,MA,EPS                                            FLM173C
420   IF(NPL) CALL PLDATA(N,NDIM,N,J,IHED,XX,SIG,FORCE)                    FLM174C
421   CONTINUE                                                             FLM175C
5     RETURN                                                               FLM176C
C**** FORMAT STATEMENTS                                                    FLM177C
1000  FORMAT(4F10.0,I10)                                                   FLM178C
2000  FORMAT(5X,54HLINEAR ISOTROPIC ELASTIC TWO/THREE DIMENSIONAL ELEMEN   FLM179C
     1T//5X,3HE =,E15.5,5X,4HNU =,F10.5,5X,5HRHO =,F10.5/5X,4HAA =,F10.5   FLM180C
     2,2X5H(1.0 FOR PLANE STRESS, 2.0 FOR PLANE STRAIN AND 3D)/5X)         FLM181C
2001  FORMAT(1X,9(2A6))                                                    FLM182C
2002  FORMAT(A1,12A6,E13.5,17X,4HPAGE,I3//5X,16HELEMENT STRESSES//         FLM183C
     1 1X,7HELEMENT,9(2A6))                                               FLM184C
2003  FORMAT(5X,9(4H**FATAL ERROR 1200 ATTEMPT TO USE INCOMPATIMLE MODE E  FLM185C
     1LEMENT WITH WRONG NUMBER OF CONNECTED NODES)                        FLM186C
2004  FORMAT(72X,8H1-VALUES,4X,8H2-VALUES,7X,6HANGLE  )                    FLM187C
      END                                                                  FLM188C
      SUBROUTINE VSTIF(NDIM,NDF,NEL,NSIDE,ST,FORCE,STUW,LINT)              VST  1C
C********** VSTIF REWRITTEN****7/20/73**********************************  VST  2C
      DIMENSION STUW(4,27),ST(1),FORCE(NDF,1),SHP(81)                      VST  3C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SK(3,3),X(3,20),LD(120)        VST  4C
      EQUIVALENCE(SHP,SHAPE)                                               VST  5C
      N=NEL*NDIM                                                           VST  6C
      DO 200 II=1,LINT                                                     VST  7C
      SS=STUW(1,II)                                                        VST  8C
      TT=STUW(2,II)                                                        VST  9C
      UU=STUW(3,II)                                                        VST 10C
      WU=STUW(4,II)                                                        VST 11C
      CALL BRICK2(SS,TT,UU,NDIM,NEL,NSIDE)                                VST 12C
      DVOL=WU/XJAC                                                         VST 13C
      XJAC=XJAC*WU                                                         VST 14C
      NS=0                                                                 VST 15C
      DO 50 J=1,NEL                                                        VST 16C
      CC=SHAPE(4,J)*XJAC                                                   VST 17C
      DO 50 I=1,NDIM                                                       VST 18C
      NS=NS+1                                                              VST 19C
      SHP(NS)=SHAPE(I,J)                                                   VST 20C
50    FORCE(I,J)=FORCE(I,J)+CC                                             VST 21C
      NS=0                                                                 VST 22C
      DO 150 J=1,N                                                         VST 23C
      CC=SHP(J)*DVOL                                                       VST 24C
      DO 100 I=1,J                                                         VST 25C
100   ST(I+NS)=ST(I+NS)+CC*SHP(I)                                          VST 26C
150   NS=NS+J                                                              VST 27C
200   CONTINUE                                                             VST 28C
```

```
      RETURN                                                            VST 29C
      END                                                               VST 30C
      SUBROUTINE ASSEMB(NDIM,NDF,NEL,NSTF,ST,ESTIF,FORCE,GG,XX,RO)      ASS  1C
C***** ASSEMB      ****** 7/03/73 ***************************************** ASS  2C
      DIMENSION ESTIF(NSTF,NSTF),ST(1),FORCE(1)                         ASS  3C
C**** ASSEMBLY OF ELEMENT INTO THE STIFNESS                             ASS  4C
      NU = NEL*NDIM                                                     ASS  5C
      NS = (NU*(NU+1))/2                                                ASS  6C
      J1 = NDF*(NEL-1)                                                  ASS  7C
      ND1 = NDIM + 1                                                    ASS  8C
      DO 230 J = 1,NEL                                                  ASS  9C
      DO 220 M = 1,NDIM                                                 ASS 10C
      M1 = ND1 - M                                                      ASS 11C
      I1 = J1                                                           ASS 12C
      DO 210 I = J,NEL                                                  ASS 13C
      L = 1                                                             ASS 14C
      IF(I.EQ.J) L = M                                                  ASS 15C
      DO 200 M = L,NDIM                                                 ASS 16C
      N1 = ND1 - M                                                      ASS 17C
      CC = ST(NS)                                                       ASS 18C
      ST(NS) = 0.0                                                      ASS 19C
      ESTIF(I1+M1,J1+N1) = ESTIF(I1+M1,J1+N1) + XX*CC                   ASS 20C
      CC = GG*CC                                                        ASS 21C
      ESTIF(I1+N1,J1+M1) = ESTIF(I1+N1,J1+M1) + CC                      ASS 22C
      IF(M1.NE.N1) GO TO 200                                            ASS 23C
      DO 100 K = 1,NDIM                                                 ASS 24C
  100 ESTIF(I1+K,J1+K) = ESTIF(I1+K,J1+K) + CC                          ASS 25C
  200 NS = NS - 1                                                       ASS 26C
  210 I1 = I1 - NDF                                                     ASS 27C
  220 CONTINUE                                                          ASS 28C
  230 J1 = J1 - NDF                                                     ASS 29C
      NU = NEL*NDF                                                      ASS 30C
      DO 300 J = 2,NU                                                   ASS 31C
      K = J - 1                                                         ASS 32C
      DO 300 I = 1,K                                                    ASS 33C
      ESTIF(I,J) = ESTIF(I,J) + ESTIF(J,I)                              ASS 34C
  300 ESTIF(J,I) = ESTIF(I,J)                                           ASS 35C
      DO 400 I=1,NU                                                     ASS 36C
  400 ESTIF(I,I)= ESTIF(I,I)+RO*FORCE(I)                                ASS 37C
      RETURN                                                            ASS 38C
      END                                                               ASS 39C
      SUBROUTINE INSHAP(S,SG,SHAPE,NDIM)                                INS  1C
      DIMENSION SHAPE(4,1), SG(3,3),S(3)                                INS  2C
      DO 101 I = 1,NDIM                                                 INS  3C
      SHAPE(4,I) = 1. - S(I)**2                                         INS  4C
      DO 101 J = 1,NDIM                                                 INS  5C
  101 SHAPE(I,J) = -2.0*SG(J,I)*S(J)                                    INS  6C
      RETURN                                                            INS  7C
      END                                                               INS  8C
      SUBROUTINE INCOMP(A,M,C,S,F,ME,NE,ISW)                            INC  1C
      DIMENSION A( 9,ME),B( 9,NE),C(ME),S(NE,NE),F(NE)                  INC  2C
      GO TO (100,200,100), ISW                                          INC  3C
C**** REDUCE ARRAYS                                                     INC  4C
  100 DO 105 N=1,ME                                                     INC  5C
      P = A(N,N)                                                        INC  6C
      IF(P.EQ.0.0) GO TO 105                                            INC  7C
      IF(N.EQ.ME) GO TO 103                                             INC  8C
      NP = N+1                                                          INC  9C
      DO 102 I=NP,ME                                                    INC 10C
      AP = A(I,N)/P                                                     INC 11C
      IF(AP.EQ.0.0) GO TO 103                                           INC 12C
      C(I) = C(I) - AP*C(N)                                             INC 13C
      DO 101 J=I,ME                                                     INC 14C
      A(I,J) = A(I,J) - A(N,J)*AP                                       INC 15C
  101 A(J,I) = A(I,J)                                                   INC 16C
      DO 102 J=1,NE                                                     INC 17C
  102 B(I,J) = B(I,J) - B(N,J)*AP                                       INC 18C
  103 CONTINUE                                                          INC 19C
      IF(ISW.NE.1) GO TO 105                                            INC 20C
      DO 104 I = 1,NE                                                   INC 21C
      AP = B(N,1)/P                                                     INC 22C
      F(I) = F(I) - AP*C(N)                                             INC 23C
```

```
      DO 104 J = I,NE                                                    INC 24C
      S(I,J) = S(I,J) - H(N,J)*AP                                        INC 25C
104   S(J,I) = S(I,J)                                                    INC 26C
105   CONTINUE                                                           INC 27C
      RETURN                                                             INC 28C
C.... RECOVER INCOMPATIBLE DISPLACEMENTS                                 INC 29C
200   DO 201 I = 1,NE                                                    INC 30C
      DO 201 J = 1,NE                                                    INC 31C
201   C(I) = C(I) - H(I,J)*F(J)                                          INC 32C
C.... BACKSUBSTITUTE                                                     INC 33C
      N = NE                                                             INC 34C
202   C(N) = C(N)/A(N,N)                                                 INC 35C
      IF(N.LE.1) RETURN                                                  INC 36C
      I = N                                                              INC 37C
      N = N-1                                                            INC 38C
      DO 203 J = I,NE                                                    INC 39C
203   C(N) = C(N) - A(N,J)*C(J)                                          INC 40C
      GO TO 202                                                          INC 41C
      END                                                                INC 42C
      SUBROUTINE INSTIF(ESTIF,FORCE,SG0,STUW,LINT,NDF,NDIM,NEL,NSIDE,MM,  INS  1C
     1   C11,C12,C33)                                                    INS  2C
      DIMENSION ESTIF( 9,1),FORCE(1),SG0(3,3),STUW(4,1)                  INS  3C
      COMMON/SHAP/ XJAC,SHAPE(4,20),SG(3,3),SR(3,3),X(3,20),LD(120)      INS  4C
      ZN = 0.0                                                           INS  5C
      A42 = 0.0                                                          INS  6C
      DO 305 L = 1,LINT                                                  INS  7C
      CALL BRICK2(STUW(1,L),STUW(2,L),STUW(3,L),NDIM,NEL,NSIDE)          INS  8C
      CALL INSHAP(STUW(1,L),SG0,SHAPE(1,NEL+1),NDIM)                     INS  9C
      DV = STUW(4,L)*XJAC                                                INS 10C
      J1 = 1                                                             INS 11C
      NELP = NEL + NDIM                                                  INS 12C
      DO 304 J = 1,NELP                                                  INS 13C
      I1 = 1                                                             INS 14C
      XN = SHAPE(1,J)*DV                                                 INS 15C
      YN = SHAPE(2,J)*DV                                                 INS 16C
      A11 = C11*XN                                                       INS 17C
      A12 = C12*YN                                                       INS 18C
      A21 = C33*YN                                                       INS 19C
      A22 = C33*XN                                                       INS 20C
      A31 = C12*XN                                                       INS 21C
      A32 = C11*YN                                                       INS 22C
      IF(NDIM.EQ.2) GO TO 301                                            INS 23C
      ZN = SHAPE(3,J)*DV                                                 INS 24C
      A13 = C12*ZN                                                       INS 25C
      A42 = C33*ZN                                                       INS 26C
      A53 = C11*ZN                                                       INS 27C
301   DO 303 I = 1,NDIM                                                  INS 28C
      XN = SHAPE(1,I+NEL)                                                INS 29C
      YN = SHAPE(2,I+NEL)                                                INS 30C
      IF(NDIM.EQ.3) ZN = SHAPE(3,I+NEL)                                  INS 31C
      ESTIF(I1  ,J1  ) = ESTIF(I1  ,J1  ) + XN*A11 + YN*A21 + ZN*A42     INS 32C
      ESTIF(I1  ,J1+1) = ESTIF(I1  ,J1+1) + XN*A12 + YN*A22              INS 33C
      ESTIF(I1+1,J1  ) = ESTIF(I1+1,J1  ) + XN*A21 + YN*A31              INS 34C
      ESTIF(I1+1,J1+1) = ESTIF(I1+1,J1+1) + XN*A22 + YN*A32 + ZN*A42     INS 35C
      IF(NDIM.EQ.2) GO TO 303                                            INS 36C
      ESTIF(I1  ,J1+2) = ESTIF(I1  ,J1+2) + XN*A13 + ZN*A22              INS 37C
      ESTIF(I1+1,J1+2) = ESTIF(I1+1,J1+2) + YN*A13 + ZN*A21              INS 38C
      ESTIF(I1+2,J1  ) = ESTIF(I1+2,J1  ) + ZN*A31 + XN*A42              INS 39C
      ESTIF(I1+2,J1+1) = ESTIF(I1+2,J1+1) + YN*A42 + ZN*A12              INS 40C
      ESTIF(I1+2,J1+2) = ESTIF(I1+2,J1+2) + YN*A21 + ZN*A53 + XN*A22     INS 41C
303   I1 = I1 + NDF                                                      INS 42C
304   J1 = J1 + NDF                                                      INS 43C
305   CONTINUE                                                           INS 44C
      RETURN                                                             INS 45C
      END                                                                INS 46C
      SUBROUTINE PSTRES(NDIM,SIG,P1,P2,P3)                               PST  1C
      DIMENSION SIG(6)                                                   PST  2C
      DATA PI2/1.41421356237309/,PI23/2.09439510239321/                  PST  3C
C.... STRESSES MUST BE STORED IN ARRAY SIG(6) IN THE ORDER               PST  4C
C          TAU-XX,TAU-XY,TAU-XZ,TAU-YY,TAU-YZ,TAU-ZZ FOR 3-D PROBLEMS    PST  5C
      IF(NDIM.LT.2.OR.NDIM.GT.3) RETURN                                  PST  6C
      IF(NDIM.EQ.3) GO TO 100                                            PST  7C
```

```
C**** COMPUTE PRINCIPLE STRESSES FOR 2-D PROBLEMS                    PST  8C
      TEMP = 22.5/ATAN(1.0)                                          PST  9C
      XI1 = (SIG(1) + SIG(3))/2.                                     PST 10C
      XI2 = (SIG(1) - SIG(3))/2.                                     PST 11C
      RHO = SQRT(XI2*XI2 + SIG(2)*SIG(2))                            PST 12C
      P1 = XI1 + RHO                                                 PST 13C
      P2 = XI1 - RHO                                                 PST 14C
      P3 = 45.0                                                      PST 15C
      IF(XI2.NE.0.0) P3 = TEMP*ATAN2(SIG(2),XI2)                     PST 16C
      RETURN                                                         PST 17C
C**** COMPUTE PRINCIPLE STRESSES FOR 3-D PROBLEMS.                   PST 18C
100   RHO = 0.                                                       PST 19C
      XI1 = (SIG(1) + SIG(4) + SIG(6))/3.0                           PST 20C
      XI2 = SIG(1)*(SIG(4)+SIG(6))+SIG(4)*SIG(6)-SIG(2)**2-SIG(3)**2 PST 21C
     1     -SIG(5)**2                                                PST 22C
      XI3 = SIG(1)*SIG(4)*SIG(6)+2.*SIG(2)*SIG(3)*SIG(5)             PST 23C
     1     -SIG(1)*SIG(5)**2-SIG(4)*SIG(3)**2-SIG(6)*SIG(2)**2       PST 24C
      DI2 = 3.*XI1*XI1 - XI2                                         PST 25C
      TAU0 = 0.                                                      PST 26C
      IF(DI2.EQ.0.0) GO TO 200                                       PST 27C
      TAU0 = SQRT(2.*DI2/3.)                                         PST 28C
      TEMP = (XI3 + (DI2 - XI1*XI1)*XI1)*RT2/TAU0**3                 PST 29C
      RHO = SQRT(ABS(1.-TEMP*TEMP))                                  PST 30C
      RHO = ATAN2(RHO,TEMP)/3.                                       PST 31C
C**** COMPUTE AND RETURN PRINCIPAL STRESSES                          PST 32C
200   P1 = XI1 + TAU0*RT2*COS(RHO)                                   PST 33C
      P2 = XI1 + TAU0*RT2*COS(RHO - PI23)                            PST 34C
      P3 = XI1 + TAU0*RT2*COS(RHO + PI23)                            PST 35C
      RETURN                                                         PST 36C
      END                                                            PST 37C
```