Ph.D. THESIS

APPENDIX B

The Computer Programmes

H.E. ERMUTLU.

APPENDIX B

THE COMPUTER PROGRAMME

B.1 INTRODUCTION

B.1.1 In this appendix the computer programmes used to calculate the results of Chapters 2, 3 and 4 are briefly described and listed. All of these programmes given in the appendix are written in FORTRAN language and are prepared in accordance with the specifications given for the ATLAS computer at Chilton, Berkshire, U.K.

This appendix includes three complete computer programmes, namely the static analysis of arch dams (chapter 2); calculation of natural frequencies and mode shapes (chapter 3); and, lastly, the response calculations by means of random vibration analysis method (chapter 4). Each of these programmes is firstly described briefly, then the list of variables is given. This is followed by the list of subroutines used in each programme and the input specifications are given.

The specifications of each subroutine are given altogether after the brief description of each programme. To provide a detailed specificationfor each subroutine would take an undue amount of space. Hence the specifications only give the use of the routines, the quantities passed to the routines in the argument list, brief details of the computational procedure and any other relevant details such as the data required, the output from the routine and error stops. However, in describing each programme some general information which applies to many of the routines is given and so this information is not repeated for each individual routine.

Each subroutine is numbered in sequence and the FORTRAN listings of subroutines are contained in the final part of this appendix in this order.

Unless the contrary is stated in the specification, the routines that follow assume tetrahedral finite elements with four corners and three unknown displacements per corner being used.

B.1.2 As the finite element analysis may usually consist of hundreds of nodal points and thousands of elements, for structures which have geometrical regularities, the input data must be minimized in order to reduce the data preparation task and also to minimize the data reading time. For an arch dam which has circular arches, this economy can easily be achieved by using the special subroutines prepared. In order to use these subroutines the structure should be idealized and the nodal points should be numbered in certain ways. This idealization and the order of numbering the nodal points are shown for the half of an arch dam and the necessary information is given in Figure B.1. By using this type of idealization, the input data is reduced to consist of only one card per each elevation considered in the idealization. The variables necessary for such an elevation are shown in Figure B.2.

If such a regular idealization is impossible, then the data would consist of one card per each element and also one card per each nodal point.

B.2 COMPUTER PROGRAMME FOR STATIC ANALYSIS OF ARCH DAMS

B.2.1 The basic flow diagram of this programme is shown in Figure 2.14 and is described in Section 2.11 briefly.

Two magnetic tapes (recognised in the listings as "magnetic tape 23" and "magnetic tape 24") are used as intermediate storage.

The application of iterative method presented in Appendix A.2 leads to a special type of storage which reduces the necessary storage for the overall stiffness matrix to minimum. This matrix consists of a three dimensional array. As each nodal point stiffness submatrix has nine elements, this three dimensional array consists of nine rectangular matrices. One of these matrices is given in Figure B.3 with the other main arrays used in this programme.

B.2.2 Scalars and arrays used in the programme

ND	Maximum number of elements
MD	Maximum number of nodal points
LD	Maximum number of boundary nodal points
ID	Maximum number of nodal points adjacent to a nodal point (for only one layer of elements ID = 18, for more layers ID = 27).
JD	Given as JD = ID + 1
ATITLE	Title of the problem (Max. 80 characters)
NUMEL	Actual number of elements
NUMNP	Actual number of nodal points
NUMBC	Actual number of restrained boundary points
NCPIN	Cycle interval for the print of the force unbalance
NOPIN	Cycle interval for the print of the displacements
NCYCM	Maximum number of cycles problem may run
TOLER	Convergence limite for unbalanced forces (Kg)
XFAC	Over relaxation factor
NCONT	Non-zero integer to suppress printing of input data
XORD (MD)	X-ordinates of nodal points (cm)
YORD (MD)	Y-ordinates of nodal points (cm)
ZORD (MD)	Z-ordinates of nodal points (cm)
DSX (MD)	X-component of displacement of nodal point (cm)
DSY (MD)	Y-component of displacement of nodal point (cm)
DSZ (MD)	Z-component of displacement of nodal point (cm)
XLOAD (MD)	X-component of force acting at nodal point (Kg)
YLOAD (MD)	Y-component of force acting at nodal point (Kg)
ZLOAD (MD)	Z-component of force acting at nodal point (Kg)

NPI (ND)	Nodal point number of corner i of element
NPJ (ND)	Nodal point number of corner j of element
NPK (ND)	Nodal point number of corner k of element
NPL (ND)	Nodal point number of corner ℓ of element
ET(ND)	Modulus of elasticity of element (Kg/cm2)
XU (ND)	Poisson's ratio of element
RO (ND)	Unit weight of element (Kg/cm3)
NPB (LD)	Constrained nodal point numbers
NFIX (LD)	Type of constraint for each corresponding boundary nodal point as follows :

0 if nodal point is fixed in all directions
1 if nodal point is fixed in X-direction
2 if nodal point is fixed in Y-direction
3 if nodal point is fixed in Z-direction

R(9,ID,MD)	Three dimensional overall stiffness array (Figure B.3).
NP(MD,JD)	Matrix containing the information about the adjacent nodal point numbers (Figure B.3)
NAP (MD)	Number of adjacent nodal points (Figure B.3)
S(12,12)	Stiffness matrix for an element
A(12,12)	) Matrices used internally for element stiffness matrix
BTDB(12,12)	) calculations.
LM (4)	Used for temporary allocation of element corners.

B.2.3 Special scalars and arrays used for arranging nodal point and element arrays (as given in Figure B.1)

K1	Total number of vertical planes from left to right.
K2	Total number of vertical planes through thickness
K3	Total number of horizontal planes from top to bottom
NPM(K1,K2)	Matrix for the total number of nodal points contained in each column (to be given as input).
NOD(K1,K2, K3)	Matrix contains the positions of nodal points in the idealization (to be generated automatically as shown in Figure B.1)
K4	Total number of irregular elements
K5	Total number of irregular nodal points
K6	Total number of nodal points with non-zero initial displacements
K7	Total number of irregular boundary nodal points.

B.2.4 Subroutines used in the programme

S/R No.	SUBROUTINE	
1		ELEAR2
2		NOKTA
3		NOKTB
4		CORDLD
5		SINIR
6		MODSTF
7		INVNP2
8		BOUND2
9		ITERA2
10		REACT2
11		FORCE

B.2.5 Input data specifications

(1) ATITLE (10A8)

(2) NUMEL, NUMNP, NUMBC, NCPIN, NOPIN, NCYCM, TOLER, XFAC, NCONT
(6I4, 2F12.6, 1I1)

(3) K1, K2, K3, K4, K5, K6, K7 (7I4)

(4) ELMOD, POISON, UNIDOM (3F15.5) (from S/R ELEAR 2)

(5) ((NPM(I,J), J=1, K2), I=1, K1)(20I4) (from S/R ELEAR 2)

(6) HLEVEL, RAD, CENTER, THICK, ALPHA, NDIV, HLOAD, VLOAD
(4F8.1, 1F8.3, 1I4, 2F12.1) (from S/R CORDLD)
(one card for each level considered).

B.3 COMPUTER PROGRAMME FOR CALCULATING NATURAL FREQUENCIES AND MODE SHAPES

B.3.1 The basic flow diagram of this programme is shown in Figure 3.1 and is described in Section 3.11 briefly. In order to reduce the computing time, this programme is divided into two stages: the first stage (assembly stage) consists of calculation of stiffness and mass matrices and writing this information into a private magnetic tape (the first three parts introduced in Section 3.11). The second stage (analysis stage) is the evaluation of frequency and mode shape (the fourth part of Section 3.11) which uses directly the information stored in the private magnetic tape; so that the stage of preparing the mass and stiffness matrices is not repeated for each evaluation of a mode.

The private magnetic tape is recognised in the listing as "magnetic tape 23" and another tape, recognised as "magnetic tape 25" is also used as an intermediate storage in the mass and stiffness preparation stage.

In this programme, as very large matrices are involved, the same storage is used for overall stiffness and mass matrices and also the product matrix $(K - \omega^2 M)$ in successive stages and both K and M matrices are kept in magnetic tape 23 during the eigenvalue evaluation stage.

The most important feature of this programme is the use of the planar type storage for the overall matrices. The special eigenvalue routine PRVS2 was written at the Institute of Sound and Vibration Research of Southampton University by Drs. Mercer and Petyt and Miss Seavey to accommodate planar matrices. The details of the planar type of storage are given in the ISVR Structures Group Computation Subroutine Library Specifications.

The idealization of a three-dimensional structure, the structural planes assumed and the order of numbering the nodal points required for utilizing the planar storage are shown in Figure B.4. The overall matrix in planar form and the zero and non-zero coefficients for the idealized structure given in Figure B.4 are shown in Figures B.5 and B.6.

B.3.2 Scalars and arrays used in the programme

NP	Number of nodal points
NE	Number of elements
NC	Actual number of constraints
NCA	Maximum number of constraints
NPP	Number of planes in structural idealization
NMAX	Actual number of submatrices (assembly stage)
MMAX	Maximum number of submatrices (including working space, used in the analysis stage)
NNN	Size of the largest submatrix = (max) NPN*NDIM
NDIM	Number of unknown displacements per nodal point
NCORN	Number of corners of an element
KKK	Size of individual element mass or stiffness matrix = NDIM*NCORN
NST	Number of maximum steps programme may run
IEPS	Significant figures for accuracy
XAM	Initial estimate of the root
DLAM	Increment of the estimate
NCONT	Number of frequencies and mode shapes to be analyzed.
G(NP,3)	Coordinates of nodal points (m)
NGE(NE,4)	Nodal point numbers given to corners of elements
ET(NE)	Modulus of elasticity of element (Tons/m^2)
XU(NE)	Poisson's ratio of element
RO(NE)	Unit mass of element (Tons.sec^2/m^4)
VOL(NE)	Volume of element (m3)
MCN(NCA,2)	Matrix of constraint numbers (as given in S/R APCONP, ISVR S/R No.5)
DM(NNN, NNN,MMAX)	Overall stiffness or mass matrix stored in planar fashion. (In the assembly stage the third dimension can be NMAX).
NSUB(MMAX,2)	Matrix contains the dimension information of each submatrix (plane) in DM. (In the assembly stage the first dimension can be NMAX)
VECT(NNN, NPP)	The modal vector
NPN(NPP)	Contains the number of nodal points in each plane of the complete structure.
S(12,12)	Element stiffness or mass matrix
A(12,12) B(12,12)	Matrices used internally for element stiffness matrix calculations.
NPA(NCORN) NA(NCORN)	Used internally
K1,K2,K3,K4,K5 NPM(K1,K2) NOD(K1,K2,K3)	Special scalars and arrays used for arranging nodal point and element arrays as given in Section B.2.3

B.3.3 Subroutines used in the programme

B.3.3.1 Assembly stage

S/R No. 12	SUBROUTINE	ELEARM	
13		NOKTAM	
14		NOKTBM	
15		CORDLM	
16		DIMNP	
17		CLEAR	
18		STIFF	
19		CONMAS	
20		HYDNMC	
21		TRSFGP	(ISVR S/R No.35)
22		APCONP	(" " No.5)
23		RSHIFT	(" " No.5)
24		CSHIFT	(" " No.5)
25		WTPME	

B.3.3.2 Analysis stage

S/R No. 17	SUBROUTINE	CLEAR	
26		PRVS2	(ISVR S/R No.17)
27		MXFRTP	(" " No.10)
28		DETSCL	(" " No.16)
29		PLVECT	(" " No.18)
30		UNAPGE	
31		VCTWRT	

B.3.4 Input data specifications

B.3.4.1 Assembly stage

(1) NP, NE, NC, NPP, NMAX, MMAX, NNN, NCA, NDIM, NCORN, KKK, NST, IEPS, XAM, DLAM (13I4, 2F10.2).

(2) (NPN(K), K=1,NPP) (20I4)

(3) K1, K2, K3, K4, K5 (5I4)

(4) ELMOD, POISON, UNIDOM (3F15.5) (from S/R ELEARM)

(5) ((NPM(I,J), J=1,K2), I=1,K1) (20I4) (from S/R ELEARM)

(6) HLEVEL, RAD, CENTER, THICK, ALPHA, NDIV
(5F8.3, 1I4) (from S/R CORDLM)
(One card for each level considered)

(7) ((MCN(L,I), I=1,2), L=1,NC) (20I4)

If reservoir is in full condition :

(8) (HYDNC(M), M=1, NPH) (16F5.3) (from S/R HYDNMC)

(9) ALPHA, ALOAD, NDIV (1F8.3, 1F12.3, 1I4) (from S/R HYDNMC)
(one card for each level considered).

B.3.4.2 Analysis Stage

(1) NP, NE, NC, NPP, NMAX, MMAX, NNN, NCA, NDIM, NCORN, KKK, NST, IEPS, XAM, DLAM, NCONT (13I4, 2F10.2, 1I4)

(2) (NPN(K), K=1, NPP) (20I4)

(3) ((MCN(L,I), I=1,2), L=1,NC) (20I4)

B.4 COMPUTER PROGRAMME FOR RESPONSE CALCULATION USING RANDOM VIBRATION ANALYSIS METHODS

B.4.1 The basic flow diagram ofthis programme is shown in Figure 4.6 and is described in Section 4.8 briefly.

The whole structure is taken into consideration in these calculations, and only one of the possible idealizations is used.

In order to reduce the input time, the frequencies and mode shapes are stored in private magnetic tape No.23 and are read from this tape for each analysis (S/R VCTRD is used for this purpose).

The overall inertia matrix DM is a square matrix of order NP, where NP is the number of nodal points. Only one of the three components of the inertia per nodal point is assumed. The element consistent mass matrix given in Appendix A.4 by equation A.4.11 is reduced to 4 x 4 matrix for one component of inertia. The overall inertia matrix DM is combined by simple transferring of each element mass matrix.

B.4.2 Scalars and arrays used in the programme

NP,NE G(NP,3), NGE(NE,4) RO(NE), VOL(NE)	As given in Section B.3.2
K1,K2,K3,K4,K5 NPM(K1,K2) NOD(K1,K2,K3)	Special scalars and arrays used for arranging nodal point and element arrays as given in Section B.2.3.
NPP, NDIM NPN (NPP)	As given in Section B.3.2 and are only used if the eigen-vectors are read from a magnetic tape (S/R VCTRD).

MR	Number of modes considered in the analysis
NS	Number of power spectral density ordinates (must be an odd number)
DELTA	Frequency increment considered in the power spectral density.
CPS	Power spectral intensity of ground acceleration (cm^2/sec^4) as given by equation (5.1).
VEL	Velocity of earthquake waves (m/sec)
NC1	=0 for reservoir in empty condition =1 for reservoir in full condition.
NC2	Number of loading conditions to be analyzed.
NC3	=0 for velocity of earthquake waves is infinite =1 for velocity of earthquake waves is finite
NC4	=1 for forces in X-direction =2 for forces in Y-direction =3 for forces in Z-direction
NC5	Number of nodal points where the response is to be analyzed.
NC6	=1 for displacement in X-direction =2 for displacement in Y-direction =3 for displacement in Z-direction
AREA	Variance of displacement
SIGMA	The RMS value of displacement
ARSQRT	= $\left(\int_0^\infty f^2 S_x(f) \, df \right)^{\frac{1}{2}}$ as given by equation (4.80)
DM(NP,NP)	Overall inertia matrix as described in Section B.4.1.
FX(NP,MR)	Modal matrix contains only X-direction displacements
FY(NP,MR)	Modal matrix contains only Y-direction displacements
FZ(NP,MR)	Modal matrix contains only Z-direction displacements
HX(NP)	Hydrodynamic forces in X-direction
HY(NP)	Hydrodynamic forces in Y-direction
HZ(NP)	Hydrodynamic forces in Z-direction
BX(MR,NP)	Given by the matrix product $[FX]^T [DM]$
BY(MR,NP)	Given by the matrix product $[FY]^T [DM]$
BZ(MR,NP)	Given by the matrix product $[FZ]^T [DM]$
FREQ(MR)	Natural frequency (Hz)
C(MR)	Damping ratio
GMASS(MR)	Generalized mass, as given by equation (4.27)
P(MR)	Modal force, as given by equation (4.96a)
SFR(NS,MR)	Unit modal receptances, as given by equation (4.95)
SPQ(NS,MR)	Modal response spectral densities, as given by equation (4.97a or b).
APS(NS)	Normalized power spectral density of ground acceleration.
A(NS)	Power spectral density of displacement as given by equation (4.98).
B(NS), D(NS)	Used in S/R ASUMP internally.

B.4.3 Subroutines used in the programme

S/R No.32	SUBROUTINE	ELEREM
33		NOKSYM
34		CORSYM
35		RESMAS
36		HYDREM
37		VCTRD
38		GENMAS
39		RECEPT
40		MDCYRC
41		DCYFRC
42		MDFRCE
43		MODRCP
44		SPECTR
45		SIMPSN
46		ASUMP
47		MAGFAC

B.4.4 Input data specifications

(1) NPP, NDIM (2I4)

(2) (NPN(K), K=1, NPP) (20I4)

(3) NP, NE, MR, NC1, NC2, NS, DELTA (6I4, 1F6.3)

(4) K1, K2, K3, K4, K5 (5I4)

(5) UNIDOM (1F15.5) (from S/R ELEREM)

(6) ((NPM(I,J), J=1,K2), I=1,K1) (20I4) (from S/R ELEREM)

(7) HLEVEL, RAD, CENTER, THICK, ALPHA, NDIV
(5F8.3, 1I4) (from S/R CORSYM)
(one card for each level considered)

If <u>reservoir</u> is in <u>full</u> condition :

(8) (HYDNC(M), M=1, NPH) (16F5.3) (from S/R HYDREM)

(9) ALPHA, ALOAD, NDIV (1F8.3, 1F12.3, 1I4) (from S/R HYDREM)
(one card for each level considered)

(10) (FREQ(I), I=1,MR) (10F8.3)

(11) (C(I), I=1, MR) (10F8.3)

(12) (APS(I), I=1,NS) (10F8.1)

<u>Repeat</u> the following <u>NC2</u> times :

(13) NC3, NC4, NC5, CPS, VEL (3I4, 1F8.3, 1F8.1)

Repeat the following NC5 times

(14) NC6, N (2I4)

Note 1: If the eigenvectors are not kept in the private magnetic tape, data cards (1) and (2) should be ignored.

Note 2: If reservoir is in empty condition, the data in lines (8) and (9) should be ignored.

B.5 SUBROUTINE SPECIFICATIONS

The specifications of the subroutines used in the programmes described in the previous sections, are given below in the order of numbering. In these specifications, only the arguments which are not included in the main programmes are to be given and the others, which are defined in the previous sections are not repeated.

The precision of all of the subroutines is single and the language used is ATLAS FORTRAN.

The subroutines, which are already existing in the ISVR Subroutine Library are not repeated herein and only the ISVR subroutine numbers are given.

1. Subroutine ELEAR 2

1. Purpose

Arranges the nodal point and element arrays and applies the initial values to nodal point forces and displacements (steps 3 to 6 of Figure 2.14)

2. Method

Controls the other subroutines which generate the nodal point and element arrays for a structure idealized as given in Figure B.1.

3. Other routines used

This routine calls Subroutines NOKTA and CORDLD.

4. Data required

Section B.2.5, data numbered (4) and (5), where ELMOD is the elasticity modulus for all the elements, POISON is the Poisson's ratio for all the elements and UNIDOM is the unit weight for all the elements. NPM is given in section B.2.3 and in Figure B.1.

2. Subroutine NOKTA

1. Purpose

Generates the nodal point matrix NOD and arranges the first idealization of the structure by means of tetrahedral elements (step 6 of Figure 2.14).

2. Method

The nodal point matrix NOD is generated according to the numbering orders given in Figure B.1. Five tetrahedral elements are used to construct a brick like block (the first possible combination with A-type brick like blocks, section 2.3.2).

3. Subroutine NOKTB

1. Purpose and method

The second possible combination with A-type brick like blocks, by using five tetrahedral elements, is arranged (section 2.3.2). (step 6 of Figure 2.14)

4. Subroutine CORDLD

1. Purpose

Arranges nodal point arrays, calculates the nodal point coordinates and hydrostatic loads (step 4 of Figure 2.14).

2. Method

This routine can only be used if the arch dam consists of circular, symmetrical arches. The variables shown in Figure B.2 are used for each elevation considered and the nodal point coordinates and hydrostatic forces are calculated.

3. Data required

Section B.2.5, data numbered (6), where HLEVEL, RAD, CENTER, THICK, ALPHA and NDIV are given in Figure B.2. HLOAD is the horizontal hydrostatic load calculated for a vertical surface area around a nodal point at that particular level and VLOAD is an external vertical load which might be existing.

5. Subroutine SINIR

1. Purpose

Arranges the boundary point arrays.

2. Method

Generates the arrays NPB and NFIX, puts "0" for the points on the boundary and "1" for the points on the plane of symmetry in the array NFIX. This routine can only be used for the half of a symmetrical arch dam (Figure B.1).

6. Subroutine MODSTF

1. Purpose

Calculates element stiffness matrices, transfers the half of the stiffnesses to overall stiffness matrix and calculates gravity loads for each nodal point (steps 8 to 11 of Figure 2.14).

2. Method

See Sections 2.4, 2.5 and 2.6, and also Appendix A.1, the flow diagram in Figure 2.14 and the arrays shown in Figure B.3.

3. Magnetic Tape

Common magnetic tape number 23 must be supplied, onto each individual element stiffness matrix is written.

4. Error stops

The execution is terminated if the volume of an element is found to be zero and if the number of adjacent points exceeds the prescribed value of ID as given in Section B.2.2.

5. Miscellaneous

(a) If the volume of an element is found negative, then the order of nodal points is changed so that the volume becomes positive and a message is printed.

(b) This routine is used for each of two possible idealizations by means of A-type brick like blocks.

7. Subroutine INVNP2

1. Purpose

Inverts nodal point stiffness submatrix into nodal point flexibility matrix (steps 12 to 13 of Figure 2.14).

2. Method

See section 2.7 and Appendix A.2.

3 x 3 matrix is inverted explicitly.

8. Subroutine BOUND2

1. Purpose

Applies constraints to boundary nodal points. (Steps 14 to 15 of Figure 2.14).

2. Method

See section 2.8 and Appendix A.3. The effective flexibility matrices are calculated for each boundary nodal point.

9. Subroutine ITERA2

1. Purpose

Performs a cycle of iteration (step 17 of Figure 2.14).

2. Method

See section 2.7 and Appendix A.3. A cycle of iteration is performed for all nodal points and the total unbalanced forces, SUM, is calculated.

10. Subroutine REACT2

1. Purpose

Calculates the nodal point reactions (steps 21 to 22 of Figure 2.14).

2. Method

See section 2.9 and Figure 2.11a. The element reactions are calculated and are combined in order to calculate the nodal point forces.

3. Magnetic tape

This routine uses the information written on magnetic tape 23 by the subroutine MODSTF.

4. Printing

This routine prints the resultant nodal point reactions as shown in Figure 2.11a.

11. Subroutine FORCE

1. Purpose

Calculates the nodal point reactions in the cross-sections of the structure (step 22 of Figure 2.14).

2. Method

See section 2.9 and Figures 2.11b and 2.11c.

3. Printing

This routine prints the nodal point reactions for each force component direction and for each cut through the nodal points as shown in Figures 2.11b and 2.11c.

12. Subroutine ELEARM

1. Purpose and method

This routine is similar to S/R ELEAR2 (No.1) but modified in order to be used in the programme for calculating natural frequencies and mode shapes. (Steps 3-4 of Figure 3.1).

2. Other routines used

This routine calles Subroutines NOKTAM and CORDLM.

3. Data required

Section B.3.4.1, data numbered (4) and (5), where ELMOD is the elasticity modulus, POISON is the Poisson's ratio and UNIDOM is the unit mass for all the elements. NPM is given in Section B.2.3 and in Figure B.1.

13. Subroutine NOKTAM

1. Purpose and method

This routine is similar to S/R NOKTA (No.2) but modified according to S/R ELEARM (step 6 of Figure 3.1).

2. Argument list

L(NE,4) is the nodal point number given to corners of elements.

14. Subroutine NOKTBM

1. Purpose and method

This routine is similar to S/R NOKTB (No.3) but modified in order to be used in the dynamic analysis programme (step 6 of Figure 3.1).

2. Argument list

L(NE,4) is as given in S/R NOKTAM (No.13).

15. Subroutine CORDLM

1. Purpose

Calculates the nodal point coordinates (step 4 of Figure 3.1).

2. Method

As given in S/R CORDLD (No.4).

3. Data required

Section B.3.4.1, data numbered (6), where HLEVEL, RAD, CENTER, THICK, ALPHA, and NDIV are given in Figure B.2.

16. Subroutine DIMNP

1. Purpose and method

Calculates the dimensions of each submatrix of the overall matrix DM (see Figure B.6), by using the information given by NPN(NPP), NSUB (MMAX,2) matrix is generated (steps 8 and 19 of Figure 3.1).

17. Subroutine CLEAR

1. Purpose and method

Clears the matrix DM after each stage of computation is completed (step 18 of Figure 3.1).

18. Subroutine STIFF

1. Purpose

Calculates element stiffness matrix (step 13 of Figure 3.1).

2. Method

See sections 2.4, 2.5 and also Appendix A.1.

3. Error stops

The execution is terminated if the volume of an element is found to be zero.

4. Argument list

BTDB (12,12) is used internally and N is the element number for which the stiffness matrix is computed.

5. Miscellaneous

(a) This routine is used in a loop for each element.

(b) If the volume of an element is found negative, then the order of nodal points is changed so that the volume becomes positive and a message is printed.

19. Subroutine CONMAS

1. Purpose

Calculates element consistent mass matrix (step 24 of Figure 3.1).

2. Method

See sections 2.5, 3.3 and Appendix A.4.

3. Argument list

N is the element number for which the mass matrix is computed, C(12,12) is the consistent mass matrix of the element.

4. Miscellaneous

This routine is used in a loop for each element.

20. Subroutine HYDNMC

1. Purpose

Calculates the hydrodynamic forces of the reservoir water and transfers to overall inertia matrix (step 28 of Figure 3.1).

2. Method

See sections 3.7, 3.8 and 3.9.

3. Argument list

A(NNN,NNN,MMAX) is the overall matrix stored in planar fashion, NPH is the number of nodal points on the water face of the arch dam (in case of one element layer used through thickness NPH = NP/2).

4. Data required

Section B.3.4.1, data numbered (8) and (9), where HYDNC (NPH) is the hydrodynamic pressure coefficient for each nodal point on the water face of the arch dam (as given in Figure 3.14); ALPHA and NDIV are as given in Figure B.2, ALOAD is the horizontal hydrostatic load calculated for a vertical surface area around a nodal point at that particular level.

21. Subroutine TRSFGP

ISVR subroutine library, S/R No.35. (used at steps 14 and 25 of Figure 3.1).

22. Subroutine APCONP

ISVR subroutine library, S/R No.5. (Used at steps 16 and 29 of Figure 3.1). Calls subroutines RSHIFT and CSHIFT.

23. Subroutine RSHIFT

Included in subroutine APCONP, ISVR subroutine library S/R No.5.

24. Subroutine CSHIFT

Included in subroutine APCONP, ISVR subroutine library S/R No.5.

25. Subroutine WTPME

1. Purpose and method

Writes the overall stiffness or mass matrices onto private magnetic tape (steps 17 and 30 of Figure 3.1).

2. Argument list

A(NNN,NNN,KMAX) is the overall stiffness or mass matrix stored in planar fashion and KMAX is the maximum number of submatrices.

3. Magnetic tape

Private magnetic tape number 23 must be supplied, onto the overall stiffness and mass matrices are written.

26. Subroutine PRVS2

ISVR subroutine library, S/R No.17 (used at step 31 of Figure 3.1). Calls subroutines MXFRTP, DETSCL and PLVECT.

27. Subroutine MXFRTP

ISVR Subroutine library, S/R No.10.

28. Subroutine DETSCL

ISVR subroutine library, S/R No.16.

29. Subroutine PLVECT

ISVR subroutine library, S/R No.18.

30. Subroutine UNAPGE

1. Purpose and method

Re-inserts the constrained zero displacements into the constrained vector by means of simply expanding the vector and inserting zeros (step 32 of Figure 3.1).

31. Subroutine VCTWRT

1. Purpose and method

Separates the displacement components of the eigenvector and prints the results in a pleasing form (step 33 of Figure 3.1).

32. Subroutine ELEREM

1. Purpose and method

This routine is similar to S/R ELEAR2 (No.1) but modified in order to be used in theprogramme for response calculations (steps 3 to 5 of Figure 4.6).

2. Other routines used

This routine calls subroutines NOKSYM and CORSYM.

3. Data required

Section B.4.4, data numbered (5) and (6), where UNIDOM is the unit mass for all the elements and NPM is given in Section B.2.3 and in Figure B.1.

33. Subroutine NOKSYM

1. Purpose and method

This routine is similar to S/R NOKTAM (No.13), but is modified in order to idealize the whole structure (steps 3 and 5 of Figure 4.6).

34. Subroutine CORSYM

1. Purpose and method

This routine is similar to S/R CORDLM (No.15), but is modified in order to calculate the coordinates of all of the nodal points in the whole structure (step 4 of Figure 4.6).

2. Data required

Section B.4.4, data numbered (7), where HLEVEL, RAD, CENTER, THICK, ALPHA and NDIV are given in Figure B.2.

35. Subroutine RESMAS

1. Purpose

Calculates the overall mass matrix (step 6 of Figure 4.6).

2. Method

See sections 2.5, 3.3 and Appendices A.1 and A.4, and also Section B.4.1.

3. Error stops

The execution is terminated if the volume of an element is found to be zero.

4. Miscellaneous

(a) This routine firstly calculates mass matrix of each element and transfers to overall mass matrix in turn.

(b) If the volume of an element is found negative, then the order of nodal points is changed so that the volume becomes positive and a message is printed.

36. Subroutine HYDREM

1. Purpose

Calculates the hydrodynamic forces of the reservoir water and arranges the hydrodynamic matrix (step 8 of Figure 4.6).

2. Method, argument list and data required

They are similar to S/R HYDNMC (No.20), except this routine can be used not for the half of the dam but for the whole structure and the hydrodynamic forces calculated are not transferred to overall inertia matrix, but the hydrodynamic arrays HX, HY and HZ are generated. The data required in section B.4.4, data numbered (8) and (9) are the same as given for S/R HYDNMC (No.20).

37. Subroutine VCTRD

1. Purpose

Reads the eigenvectors from the private magnetic tape where they have been written (step 9 of Figure 4.6).

2. Magnetic tape

Private magnetic tape 23 must be supplied onto which the eigenvectors are already written.

3. Miscellaneous

If the eigenvectors are read by means of another peripheral, this routine and the magnetic tape are not necessary.

38. Subroutine GENMAS

1. Purpose

Calculates the generalized masses for each mode considered (step 10 of Figure 4.6).

2. Method

See section 4.4, equation (4.27).

39. Subroutine RECEPT

1. Purpose

Calculates the unit modal receptances of each mode considered (step 11 of Figure 4.6).

2. Method

See sections 4.5 and 4.8, equation (4.95).

40. Subroutine MDCYRC

1. Purpose

Calculates the modal response power spectral densities for each mode considered in the case of earthquake wave velocity is finite (steps 16 and 17 of Figure 4.6).

2. Method

See sections 4.5 and 4.8, equation (4.97b).

3. Other routines used

This routne calls subroutine DCYFRC.

4. Argument list

QA(MR,NP) takes the value of one of the three arrays BX, BY and BZ according to the forcing direction.

41. Subroutine DCYFRC

1. Purpose

Calculates the modal forces in the case of earthquake wave velocity being finite (step 16 of Figure 4.6).

2. Method

See sections 4.5 and 4.8, equation (4.96b).

3. Argument list

F is the frequency, M is the mode number, P is used in S/R MDCYRC internally and QA(MR,NP) is as given in S/R MDCYRC.

42. Subroutine MDFRCE

1. Purpose

Calculates the modal forces for each mode considered in the case of earthquake wave velocity being infinite (step 19 of figure 4.6).

2. Method

See sections 4.5 and 4.8, equation (4.96a).

3. Argument list

QA(MR,NP) is as described in S/R MDCYRC (No.40).

43. Subroutine MODRCP

1. Purpose

Calculates the modal response power spectral densities for each mode considered in the case of earthquake wave velocity being infinite (step 20 of Figure 4.6).

2. Method

See sections 4.5 and 4.8, equation (4.97a).

44. Subroutine SPECTR

1. Purpose

Calculates the power spectral density of response (step 24 of Figure 4.6).

2. Method

See sections 4.5 and 4.8, equation (4.98).

3. Argument list

N is the nodal point number where the response is calculated, FA(NP,MR) takes the value of one of the three arrays FX, FY and FZ according to the direction of the response component.

45. Subroutine SIMPSN

1. Purpose

Calculates the area under a function (used at step 25 of Figure 4.6).

2. Method

Uses the very well known "Simpson Rule" for the area calculation.

46. Subroutine ASUMP

1. Purpose

Calculates the values used to simplify the Rice's general formula.

2. Method

See section 4.5.6, equations 4.69 and 4.70. This routine is only used to check the validity of the assumptions made in section 4.5.6 in order to simplify the Rice's general formula.

47. Subroutine MAGFAC

1. Purpose

Calculates the magnification factor and the mean maxima of responses for several earthquake durations (step 25 of figure 4.6).

2. Method

See section 4.5.7 and equation 4.90. The mean maxima of responses are calculated for earthquake duration from 10 to 60 seconds.

B.6 LISTINGS OF THE FORTRAN PROGRAMMES AND SUBROUTINES

In the following, firstly the main routines of the four programmes described in Sections B.2, B.3 and B.4 are given, then the listings of the subroutines are given in the same order of Section B.5.

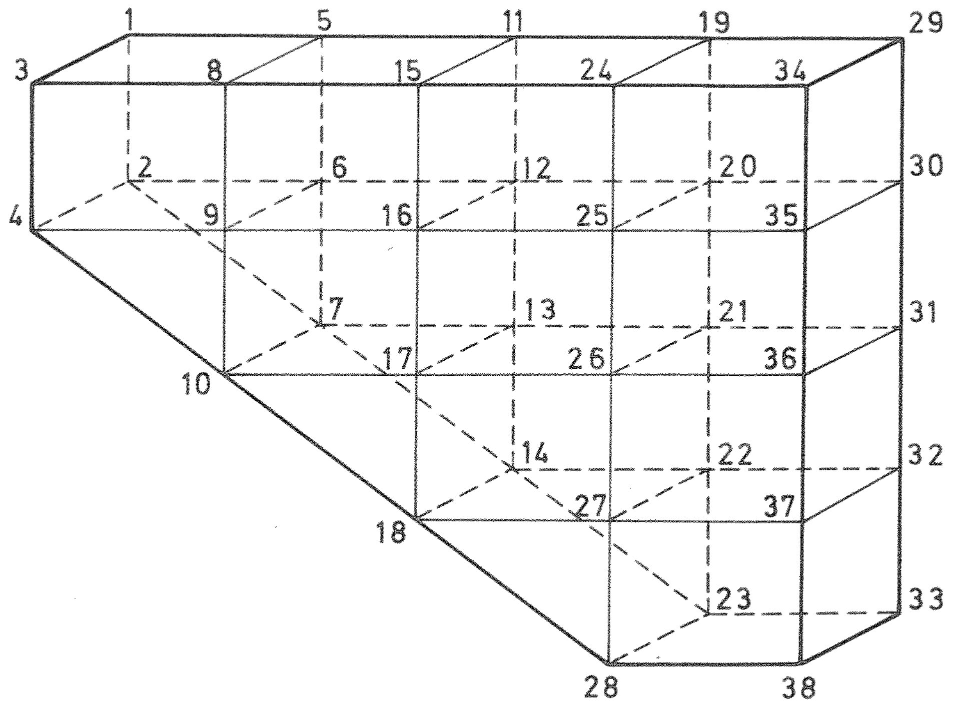
The main routines are :

(I) The programme for static analysis of arch dams (given in section B.2).

(II-1) Assembly stage programme for calculating natural frequencies and mode shapes (given in Section B.3).

(II-2) Analysis stage programme for calculating natural frequencies and mode shapes (given in Section B.3).

(III) The programme for response calculation using random vibration analysis methods (given in Section B.4).



Total number of vertical planes from left to right, K1 = 5

" " " " " through thickness, K2 = 2

" " " horizontal " from top to bottom, K3 = 5

Matrix for the total number of nodal points contained in each column, dimensioned NPM (K1,K2)

$$NPM = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \\ 5 & 5 \end{bmatrix}$$

Fig. B.1. Necessary information for numbering the nodal points and arranging the element arrays (For 4x1x4 mesh type)

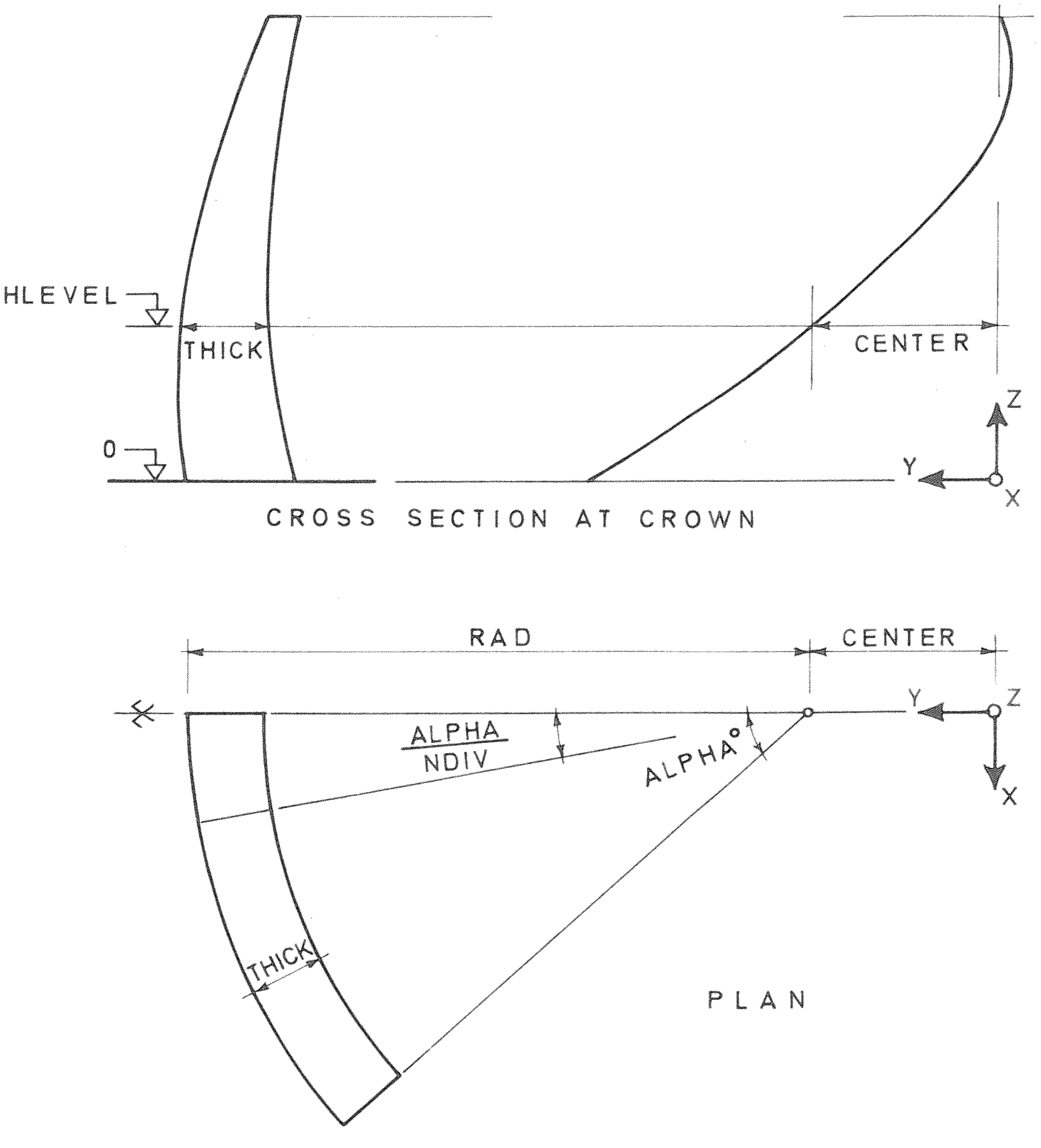


Fig. B.2. Double curvature dam with circular arches, the scalars used in computations

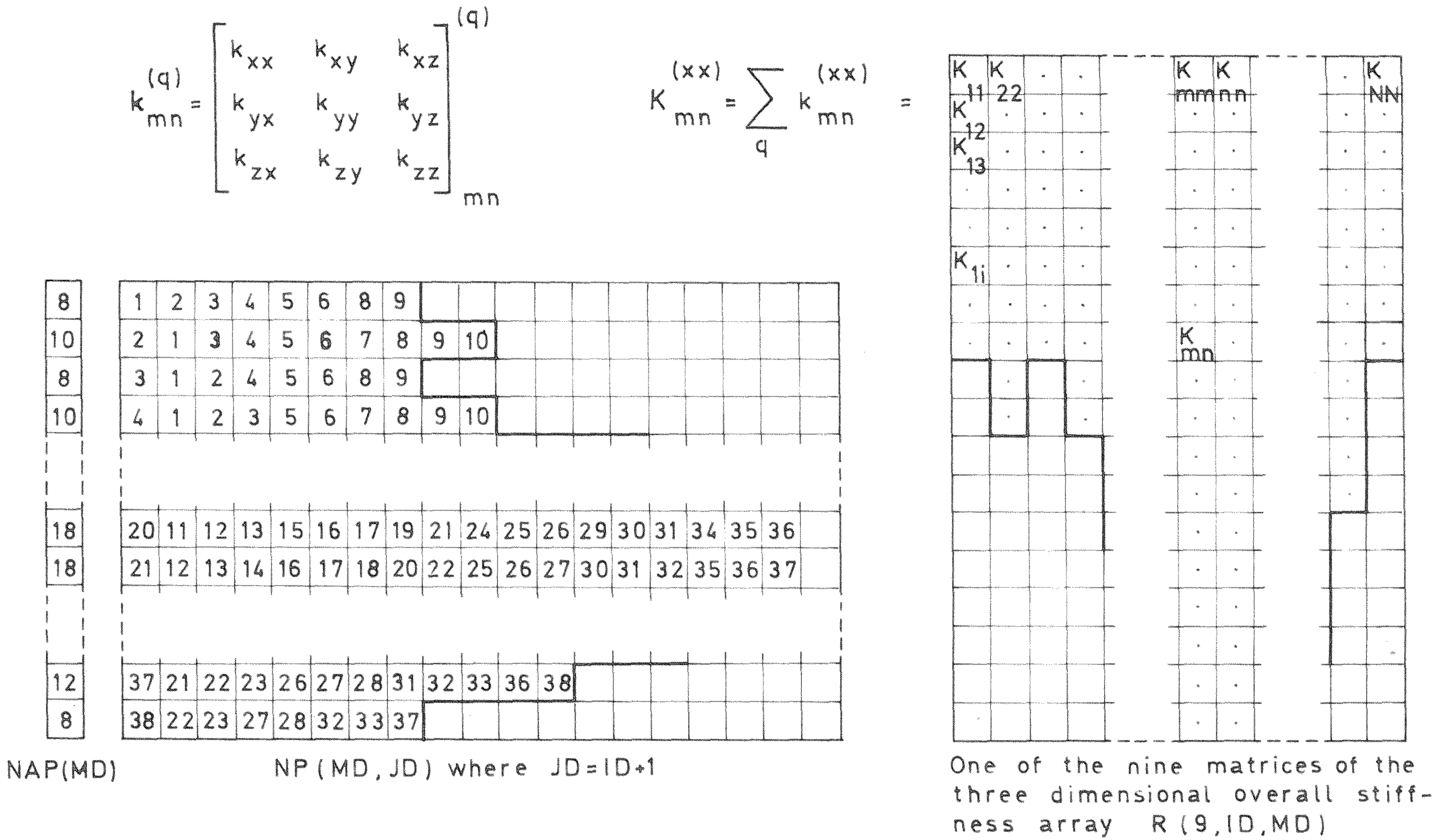


Fig. B.3. Main arrays required for static analysis by means of iterative method

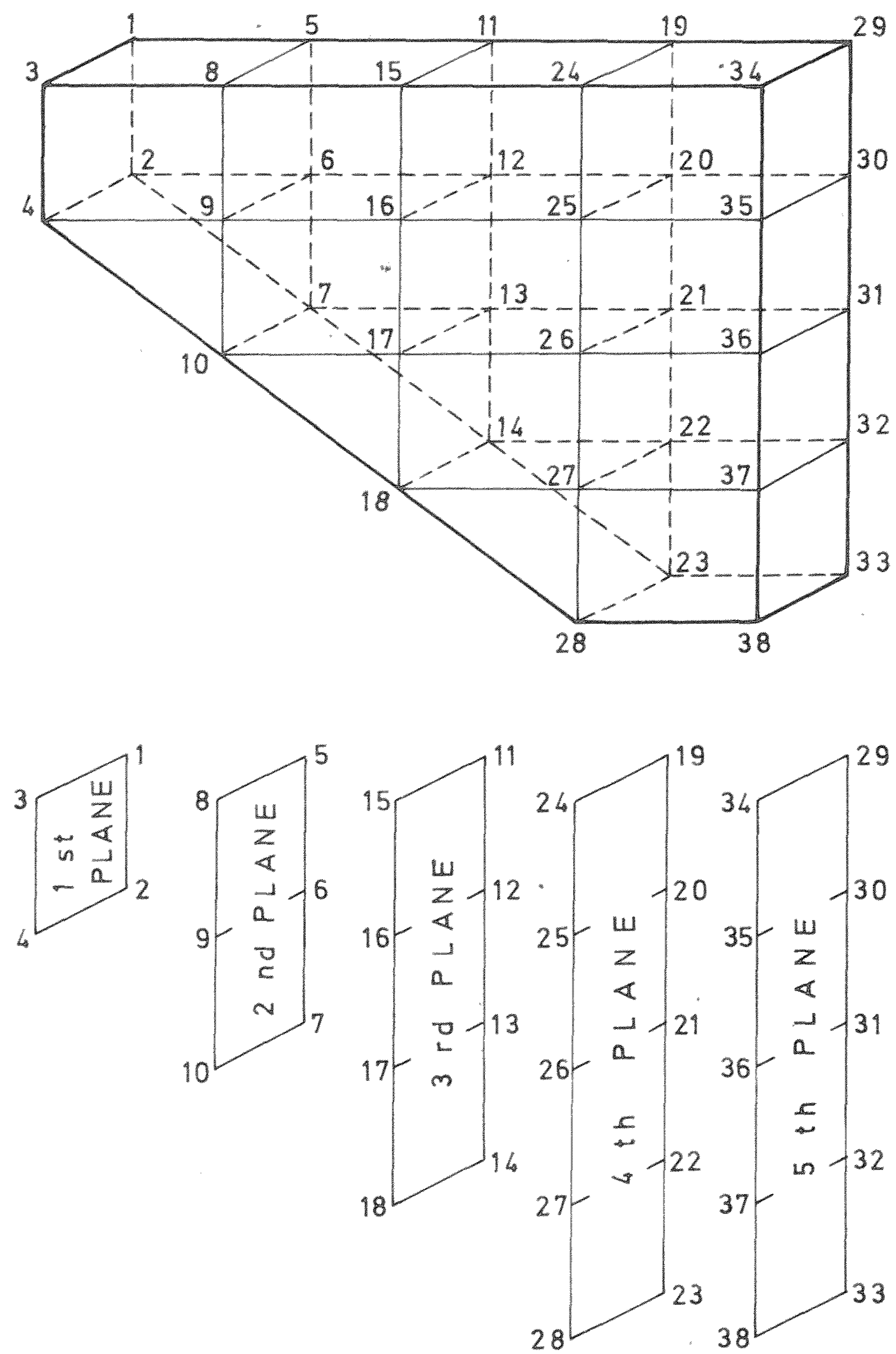


Fig. B.4. Order of numbering the nodal points and assumed planes required for planar storage

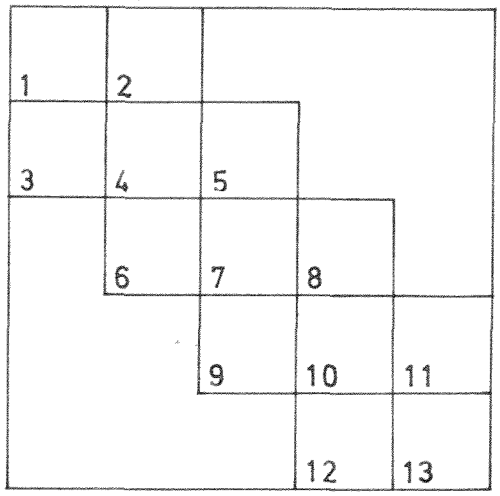


Fig. B.5. The form of the overall stiffness or mass matrix required in planar storage

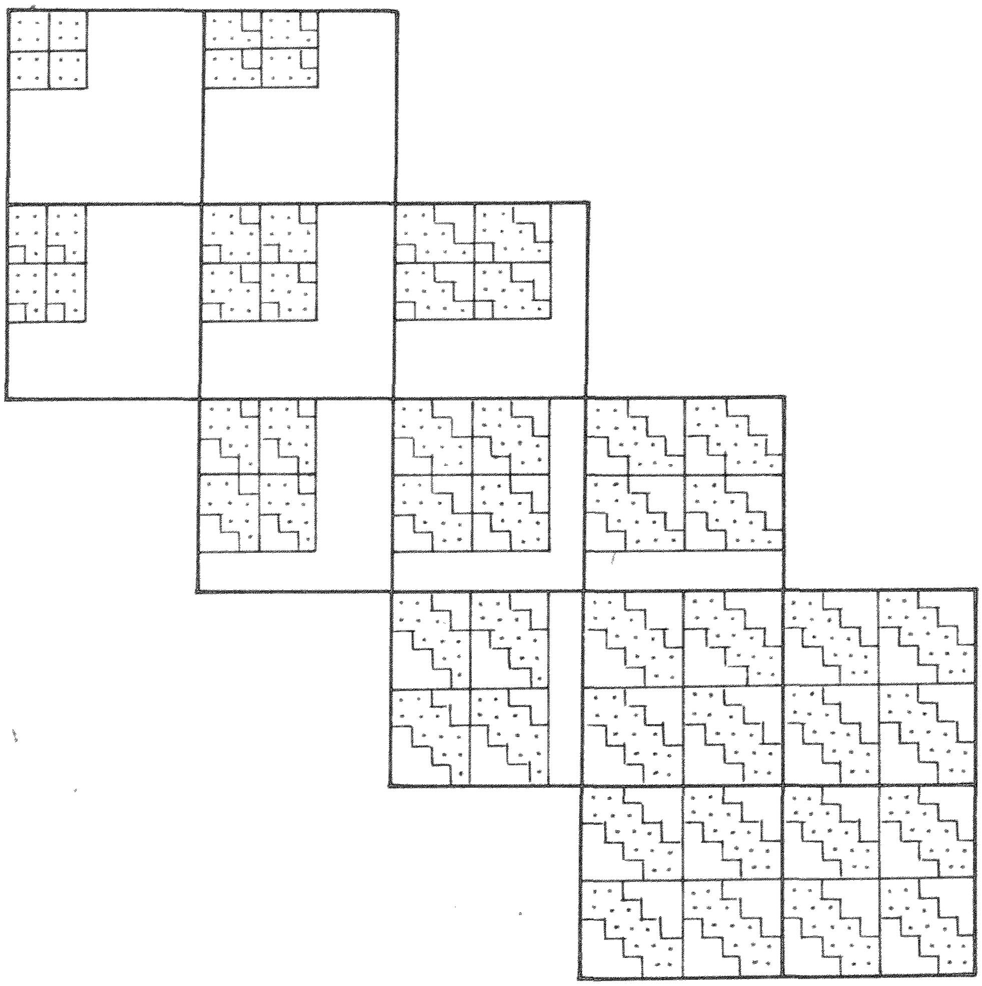


Fig. B.6. Planar storage required for the structure shown in Fig. B.4. Zero and non-zero coefficients

(I) MAIN ROUTINE FOR STATIC ANALYSIS OF ARCH DAMS

```
C     H.E.ERMUTLU, SOUTHAMPTON UNIVERSITY
C     THREE DIMENSIONAL STRESS ANALYSIS
      DIMENSION              XORD(422),YORD(422),ZORD(422),
     1 DSX(422),DSY(422),DSZ(422),XLOAD(422),YLOAD(422),ZLOAD(422),
     2 NP(422,19),NAP(422),ATITLE(10)
      DIMENSION              NPI(926),NPJ(926),NPK(926),NPL(926),ET(926),
     1 XU(926),RO(926)
      DIMENSION NPB(74),NFIX(74),LM(4),S(12,12),A(12,12),BTDB(12,12)
      DIMENSION NOD(17, 2,17),NPM(17, 2)
      COMMON R(9,18,422)
      MD=422
      ND=926
      LD=74
      ID=18
      READ 100, ATITLE
      PRINT 99
      PRINT 100, ATITLE
      READ    1 ,NUMEL,NUMNP,NUMBC,NCPIN,NOPIN,NCYCM,TOLER,XFAC,NCONT
      PRINT    101, NUMEL
      PRINT    102, NUMNP
      PRINT    103, NUMBC
      PRINT    104, NCPIN
      PRINT    105, NOPIN
      PRINT    106, NCYCM
      PRINT    107, TOLER
      PRINT    108, XFAC
      READ   8, K1,K2,K3,K4,K5,K6,K7
      CALL        ELEAR2(NUMEL,NUMNP,      NPI,NPJ,NPK,NPL,ET,XU,RO,
     1        XORD,YORD,ZORD,XLOAD,YLOAD,ZLOAD,DSX,DSY,DSZ,
     2 NOD,NPM,K1,K2,K3,K4,K5,K6,ND,MD)
      CALL        SINTR(NPB,NFIX,NOD,NPM,K1,K2,K3,K7,LD)
      IF (NCONT) 160,155,160
155   PRINT 110
      PRINT    2, (       N ,NPI(N),NPJ(N),NPK(N),NPL(N),ET(N),RO(N),
     1 XU(N),N=1,NUMEL)
      PRINT 111
      PRINT    109, (        M ,XORD(M),YORD(M),ZORD(M),XLOAD(M),YLOAD(M),
     1 ZLOAD(M),DSX(M),DSY(M),DSZ(M),M=1,NUMNP)
      PRINT 112
      PRINT    4, (NPB(L),NFIX(L),L=1,NUMBC)
160   NCYCLE=0
      NUMPT=NCPIN
      NUMOPT=NOPIN
      JD=ID+1
      DO 165 M=1,NUMNP
      DO 165 L=1,ID
      DO 165 K=1,9
165   R(K,L,M)=0.0
      DO 175 L=1,NUMNP
      DO 170 M=1,ID
170   NP(L,M)=0
      NP(L,JD)=0
175   NP(L,1)=L
      REWIND 23
      REWIND 24
      WRITE TAPE 24, (NPI(N),NPJ(N),NPK(N),NPL(N),N=1,NUMEL)
```

```
      PRINT 021
      CALL       MODSTF (NUMEL,NPI,NPJ,NPK,NPL,XORD,YORD,ZORD,RO,ET,
     1 XU,ZLOAD,BTOB,A,S,LM,NP,R,ND,MD,ID,JD)
      CALL       NOKTB(NPI,NPJ,NPK,NPL,NPM,NOD,K1,K2,K3,ND)
      WRITE TAPE 24, (NPI(N),NPJ(N),NPK(N),NPL(N),N=1,NUMEL)
      END FILE 24
      PRINT 021
      CALL       MODSTF (NUMEL,NPI,NPJ,NPK,NPL,XORD,YORD,ZORD,RO,ET,
     1 XU,ZLOAD,BTOB,A,S,LM,NP,R,ND,MD,ID,JD)
      END FILE 23
      DO 206 M=1,NUMNP
      MX=1
  205 MX=MX+1
      IF (NP(M,MX)) 206,206,205
  206 NAP(M)=MX-1
      CALL       INVNP2(NUMNP,R,MD,ID)
      CALL       BOUND2(NUMBC,NPB,NFIX,NP,R,LD,MD,ID,JD)
  243 PRINT 119
  244 SUM=0.0
      CALL       ITERA2 (SUM,NUMNP,NAP,XLOAD,YLOAD,ZLOAD,DSX,DSY,DSZ,
     1 NP,R,XFAC,MD,ID,JD)
      NCYCLE=NCYCLE+1
      IF (NCYCLE-NUMPT) 305,300,300
  300 NUMPT=NUMPT+NOPIN
      PRINT    120, NCYCLE,SUM
  305 IF (SUM-TOLER) 400,400,310
  310 IF(NCYCM-NCYCLE) 400,400,315
  315 IF (NCYCLE-NUMOPT) 244,320,320
  320 NUMOPT=NUMOPT+NOPIN
  400 PRINT 99
      PRINT 121
      PRINT    122, (     M ,DSX(M),DSY(M),DSZ(M),M=1,NUMNP)
      IF (SUM-TOLER) 440,440,430
  430 IF (NCYCM-NCYCLE) 440,440,450
  450 CONTINUE
      CALL OUTBRK(0)
      PRINT 825
      GO TO 243
  440 CONTINUE
      CALL       REACT2 (NUMEL,NUMNP,NPI,NPJ,NPK,NPL,DSX,DSY,DSZ,
     1 XLOAD,YLOAD,ZLOAD,S,LM,ND,MD,
     2 R,ID,NOD,NPM,K1,K2,K3)
      CALL       FORCE  (NUMNP,R,ID,MD)
      PRINT 11
      GO TO EXIT
    1 FORMAT (6I4,2F12.6,1I1)
    2 FORMAT (5I4,1F12.1,2F12.5)
    4 FORMAT (2I4)
    6 FORMAT(7I4)
   11 FORMAT (///37H THE END OF JOB VS030 , H.E.ERMUTLU .////////)
   99 FORMAT (1H1)
  100 FORMAT(10A8)
  101 FORMAT (29H0NUMBER OF ELEMENTS           =1I4/)
  102 FORMAT (29H NUMBER OF NODAL POINTS       =1I4/)
  103 FORMAT (29H NUMBER OF BOUNDARY POINTS    =1I4/)
  104 FORMAT (29H CYCLE WRITE INTERVAL         =1I4/)
  105 FORMAT (29H OUTPUT INTERVAL OF RESULTS   =1I4/)
  106 FORMAT (29H CYCLE LIMIT                  =1I4/)
  107 FORMAT (29H TOLERANCE LIMIT              =1F12.6/)
  108 FORMAT (29H OVER RELAXATION FACTOR       =1F6.3/)
  109 FORMAT (1I8,6F12.1,3F12.8)
```

```
110 FORMAT(55H1EL.    I    J    K    L    ELAS.MOD.       DENSITY      POISSON)
111 FORMAT(114H1        NP          X-ORD           Y-ORD           Z-ORD          X-LOA
   1D          Y-LOAD          Z-LOAD     X-DISP        Y-DISP          Z-DISP)
112 FORMAT (20H1BOUNDARY CONDITIONS)
119 FORMAT (34H0              CYCLE         FORCE UNBALANCE)
120 FORMAT(1I12,1F25.10)
121 FORMAT(55H0NODAL POINT    X-DISP.           Y-DISP.           Z-DISP.    )
122 FORMAT(1I12,3F15.10)
321  FORMAT(1H1,55X,45H EL.   I    J    K    L    VOLUME OF TETRAHEDRON )
325  FORMAT (31H0 JOB RE-ENTERED AFTER OUTBREAK)
     END
```

(II-1) ASSEMBLY STAGE MAIN ROUTINE FOR CALCULATING NATURAL FREQUENCIES AND MODE SHAPES

```
     H.E.ERMUTLU, SOUTHAMPTON UNIVERSITY, JAN.1967
     SOLUTION OF EIGENVALUE PROBLEM WITH SPACE FINITE ELEMENT METHOD
     DIMENSION DM(54,54,25),NSUB(25,2),NGE(233,4),MCN(114,2),
    1           G(120,3),S(12,12),A(12,12),B(12,12),ET(233),
    2 XU(233),RO(233),VOL(233),NPN(9),NPA(4),NA(4)
     DIMENSION NOD(9,2,9),NPM(9,2)
     COMMON DM
     READ 1,NP,NE,NC,NPP,NMAX,MMAX,NNN,NCA,NDIM,NCORN,KKK,
    1 NST,IEPS,XAM,DLAM
     READ 5,(NPN(K),K=1,NPP)
     READ 9, K1,K2,K3,K4,K5
     CALL       ELEARM(NE,NP,NGE,G,ET,XU,RO,NOD,NPM,K1,K2,K3,K4,K5)
     READ 4,((MCN(L,I),I=1,2),L=1,NC)
     PRINT 10
     PRINT 11,NP
     PRINT 12,NE
     PRINT 13,NC
     PRINT 14,NPP
     PRINT 15,NMAX
     PRINT 16,MMAX
     PRINT 17,NNN
     PRINT 18,NCA
     PRINT 19,NDIM
     PRINT 20,NCORN
     PRINT 21,NST
     PRINT 22,IEPS
     PRINT 23,XAM
     PRINT 24,DLAM
     PRINT 25
     PRINT 5,(NPN(K),K=1,NPP)
     PRINT 26
     PRINT 6,(N,(NGE(N,I),I=1,NCORN),ET(N),XU(N),RO(N),N=1,NE)
     PRINT 27
     PRINT 7,(M,(G(M,J),J=1,NDIM),M=1,NP)
     PRINT 28
     PRINT 8,((MCN(L,I),I=1,2),L=1,NC)
     DO 33 I=1,MMAX
     DO 33 J=1,2
  33 NSUB(I,J)=0
     CALL DIMNP(NSUB,NPN,MMAX,NPP)
     CALL CLEAR(DM,NNN,MMAX)
     PRINT 29
```

```
      DO 31 IE=1,NE
      CALL STIFF(NGE,G,S,A,B,ET,XU,VOL,NE,NP,IE)
      CALL TRSFGP(DM,S,NGE,NSUB,NPN,NPA,NA,NE,NPP,NNN,KKK,MMAX,NDIM,
     1 NCORN,IE)
   31 CONTINUE
      REWIND 25
      WRITE TAPE 25, ((NGE(N,I),I=1,4),VOL(N),N=1,NE)
      END FILE 25
      CALL     NOKTBM (NGE,NPM,NOD,K1,K2,K3,NE)
      PRINT 29
      DO131 IE=1,NE
      CALL STIFF(NGE,G,S,A,B,ET,XU,VOL,NE,NP,IE)
      CALL TRSFGP(DM,S,NGE,NSUB,NPN,NPA,NA,NE,NPP,NNN,KKK,MMAX,NDIM,
     1 NCORN,IE)
  131 CONTINUE
      DO 231 K=1,MMAX
      DO 232 J=1,NNN
      DO 233 I=1,NNN
  233 DM(I,J,K)=0.5*DM(I,J,K)
  232 CONTINUE
  231 CONTINUE
      CALL APCONP(DM,NSUB,MCN,NC,NNN,MMAX,NCA)
      REWIND 23
      WRITE TAPE 23,NMAX
      WRITE TAPE 23,((NSUB(I,J),I=1,NMAX),J=1,2)
      CALL WTPME(DM,NSUB,NNN,MMAX,NMAX)
      CALL CLEAR(DM,NNN,MMAX)
      CALL DIMNP(NSUB,NPN,MMAX,NPP)
      DO 32 IE=1,NE
      CALL CONMAS(S,VOL,RO,NE,IE)
      CALL TRSFGP(DM,S,NGE,NSUB,NPN,NPA,NA,NE,NPP,NNN,KKK,MMAX,NDIM,
     1 NCORN,IE)
   32 CONTINUE
      REWIND 25
      READ  TAPE 25, ((NGE(N,I),I=1,4),VOL(N),N=1,NE)
      DO132 IE=1,NE
      CALL CONMAS(S,VOL,RO,NE,IE)
      CALL TRSFGP(DM,S,NGE,NSUB,NPN,NPA,NA,NE,NPP,NNN,KKK,MMAX,NDIM,
     1 NCORN,IE)
  132 CONTINUE
      DO 331 K=1,MMAX
      DO 332 J=1,NNN
      DO 333 I=1,NNN
  333 DM(I,J,K)=0.5*DM(I,J,K)
  332 CONTINUE
  331 CONTINUE
      NPH=NP/2
      CALL       HYDNMC(DM,NSUB,G,NPN,NOD,NNN,MMAX,NP,NPP,K1,K2,K3,NPH)
      CALL APCONP(DM,NSUB,MCN,NC,NNN,MMAX,NCA)
      CALL WTPME(DM,NSUB,NNN,MMAX,NMAX)
      END FILE 23
      PRINT 57
      GO TO EXIT
    1 FORMAT(13I4,2F10.2)
    4 FORMAT (20I4)
    5 FORMAT (20I4)
    6 FORMAT(5I4,1F12.1,2F12.5)
    7 FORMAT(1I8,3F12.3)
    8 FORMAT(2I4)
```

```
  9 FORMAT(5I4)
 10 FORMAT(55H1 JOB VS030, H.E.ERMUTLU, EIGENVALUE PROBLEM, JAN.1967 )
 11 FORMAT(41H0NUMBER OF NODAL POINTS                          =1I4//)
 12 FORMAT(41H NUMBER OF ELEMENTS                              =1I4//)
 13 FORMAT(41H ACTUAL NUMBER OF CONSTRAINTS                    =1I4//)
 14 FORMAT(41H NUMBER OF PLANES                                =1I4//)
 15 FORMAT(41H ACTUAL NUMBER OF SUBMATRICES                    =1I4//)
 16 FORMAT(41H MAXIMUM NUMBER OF SUBMATRICES                   =1I4//)
 17 FORMAT(41H SIZE OF THE LARGEST SUBMATRIX                   =1I4//)
 18 FORMAT(41H MAXIMUM NUMBER OF CONSTRAINTS                   =1I4//)
 19 FORMAT(41H NUMBER OF UNKNOWN DISPLACEMENTS OF N.P.=1I4//)
 20 FORMAT(41H NUMBER OF CORNERS OF AN ELEMENT                 =1I4//)
 21 FORMAT(41H NUMBER OF MAXIMUM STEPS ALLOWED                 =1I4//)
 22 FORMAT(41H SIGNIFICANT FIGURES FOR ACCURACY                =1I4//)
 23 FORMAT(41H INITIAL ESTIMATE OF THE ROOT                    =1F12.5//)
 24 FORMAT(41H INCREMENT OF THE ESTIMATE                       =1F12.5//)
 25 FORMAT(55H NUMBERS OF NODAL POINTS IN EACH PLANES ARE FOLLOWED    )
 26 FORMAT(55H1EL.   I    J    K    L    ELAS.MOD.      POISSON  UNIT MASS)
 27 FORMAT (49H1        NP          X-ORD           Y-ORD          Z-ORD      )
 28 FORMAT(33H1APPLIED CONSTRAINTS ARE FOLLOWED)
 29 FORMAT(1H1,55X,45H EL.   I    J    K    L    VOLUME OF TETRAHEDRON )
 57 FORMAT(////21H THE END OF JOB VS030//////)
    END
```

(II-2) ANALYSIS STAGE MAIN ROUTINE FOR CALCULATING NATURAL FREQUENCIES AND MODE SHAPES

```
C     H.E.ERMUTLU, SOUTHAMPTON UNIVERSITY, JAN.1967
C     SOLUTION OF EIGENVALUE PROBLEM WITH SPACE FINITE ELEMENT METHOD
      DIMENSION DM(54,54,33),NSUB(33,2),MCN(114,2),VECT(54,9),NPN(9)
      COMMON DM
      READ 1,NP,NE,NC,NPP,NMAX,MMAX,NNN,NCA,NDIM,NCORN,KKK,
     1 NST,IEPS,XAM,DLAM,NCONT
      READ 5,(NPN(K),K=1,NPP)
      READ 4,((MCN(L,I),I=1,2),L=1,NC)
      PRINT 10
      PRINT 21,NST
      PRINT 22,IEPS
      PRINT 23,XAM
      PRINT 24,DLAM
      PRINT 25
      PRINT 5,(NPN(K),K=1,NPP)
      PRINT 28
      PRINT 8,((MCN(L,I),I=1,2),L=1,NC)
      MST=NST
      DO 61 KF=1,NCONT
      CALL CLEAR(DM,NNN,MMAX)
      DO 34 J=1,NPP
      DO 34 I=1,NNN
   34 VECT(I,J)=0.0
   64 CALL PRVS2(XAM,DLAM,MST,IEPS,NNN,MMAX,NSUB,DM,VECT,NPP)
      IF (MST) 62,63,63
   62 MST=NST
      XAM=XAM+DLAM
      GO TO 64
   63 PRINT 51, XAM
      FREQ=SQRTF(XAM)/6.28318
      PRINT 52,FREQ
```

```
      PRINT 53
      CALL UNAPGE (MCN,VECT,NC,NCA,NNN,NPP)
      PRINT 58
      CALL VCTWRT (VECT,NPN,NNN,NPP,NDIM)
      XAM=XAM+DLAM
      CALL OUTBRK(0)
      PRINT 59, KF
   61 CONTINUE
      PRINT 57
      GO TO EXIT
    1 FORMAT(13I4,2F10.2,1I4)
    4 FORMAT (20I4)
    5 FORMAT (20I4)
    8 FORMAT(2I4)
   10 FORMAT(55H1 JOB VS030, H.E.ERMUTLU, EIGENVALUE PROBLEM, JAN.1967 )
   21 FORMAT(41H NUMBER OF MAXIMUM STEPS ALLOWED              =1I4//)
   22 FORMAT(41H SIGNIFICANT FIGURES FOR ACCURACY             =1I4//)
   23 FORMAT(41H INITIAL ESTIMATE OF THE ROOT                 =1F12.5//)
   24 FORMAT(41H INCREMENT OF THE ESTIMATE                    =1F12.5//)
   25 FORMAT(55H NUMBERS OF NODAL POINTS IN EACH PLANES ARE FOLLOWED    )
   28 FORMAT(33H1APPLIED CONSTRAINTS ARE FOLLOWED)
   51 FORMAT(25H1THE ROOT OF DETERMINANT=1F16.8)
   52 FORMAT(25H0THE NATURAL FREQUENCY  =1F16.8)
   53 FORMAT(17H0THE MODAL VECTOR)
   57 FORMAT(////21H THE END OF JOB VS030//////)
   58 FORMAT(55H0N.POINT    X-DIRECTION    Y-DIRECTION    Z-DIRECTION   )
   59 FORMAT(13H0END OF PHASE,1I3/20H0START OF NEXT PHASE)
      END
```

(III) MAIN ROUTINE FOR RESPONSE CALCULATION USING RANDOM VIBRATION ANALYSIS METHODS

```
C      H.E.ERMUTLU,      POWER SPECTRUM METHOD
      DIMENSION DM(164,164),NGE(384,4),G(164,3),RO(384),VOL(384),
     1 NOD(17,2,6),NPM(17,2),FX(164,5),FY(164,5),FZ(164,5),HX(164),
     2 HY(164),HZ(164),BX(5,164),BY(5,164),BZ(5,164),C(5),FREQ(5),P(5),
     3 GMASS(5),APS(101),SFR(101,5),SPQ(101,5),A(101),R(101),D(101)
      DIMENSION NPN(17)
      COMMON DM,NGE,G,RO,VOL
      READ 70, NPP,NDIM
   70 FORMAT (2I4)
      READ 71, (NPN(K),K=1,NPP)
   71 FORMAT (20I4)
      READ 1, NP,NE,MR,NC1,NC2,NS,DELTA
    1 FORMAT (6I4,1F6.3)
      READ 2, K1,K2,K3,K4,K5
    2 FORMAT (5I4)
      CALL ELEREM (NE,NP,NGE,G,RO,NOD,NPM,K1,K2,K3,K4,K5)
      DO 10 J=1,NP
      DO 10 I=1,NP
   10 DM(I,J)=0.0
      PRINT 29
   29 FORMAT(1H1,55X,45H EL.    I    J    K    L   VOLUME OF TETRAHEDRON )
      CALL RESMAS (NGE,G,DM,VOL,RO,NE,NP)
      IF (NC1) 11,11,12
   12 NPH=NP/2
```

```
      DO 13 J=1,NP
      HX(J)=0.0
      HY(J)=0.0
      HZ(J)=0.0
   13 CONTINUE
      CALL HYDREM (G,NOD,NP,K1,K2,K3,NPH,HX,HY,HZ)
   11 CONTINUE
      REWIND 23
      DO 14 K=1,MR
      READ TAPE 23, M
      CALL        VCTRD (FX,FY,FZ,NP,MR,NPN,NPP,NDIM,M)
   14 CONTINUE
      READ 3, (FREQ(I),I=1,MR)
      READ 3, (C(I),I=1,MR)
    3 FORMAT (10F8.3)
      DO 15 M=1,MR
      PRINT 4, M,FREQ(M),C(M)
    4 FORMAT (13H1MODE NUMBER=1I2//12H  FREQUENCY=1F8.3,16H  DAMPING RAT
     1IO=1F6.3//)
      PRINT 5, (I,FX(I,M),FY(I,M),FZ(I,M),I=1,NP)
    5 FORMAT (1I4,3F12.6)
   15 CONTINUE
      CALL        GENMAS(BX,BY,BZ,FX,FY,FZ,DM,HX,HY,HZ,GMASS,MR,NP,NC1)
      READ 21, (APS(I),I=1,NS)
   21 FORMAT (10F8.1)
      CALL        RECEPT (FREQ,C,SFR,DELTA,MR,NS)
      DO 101 NCONT=1,NC2
      READ 22, NC3,NC4,NC5,CPS,VEL
   22 FORMAT (3I4,1F8.3,1F8.1)
      IF (NC3) 103,103,104
  104 CONTINUE
      IF (NC4-2) 401,402,403
  401 CALL        MDCYRC(GMASS,SPQ,SFR,APS,G,BX,CPS,VEL,NC4,MR,NP,NS,
     1 DELTA)
      GO TO 107
  402 CALL        MDCYRC(GMASS,SPQ,SFR,APS,G,BY,CPS,VEL,NC4,MR,NP,NS,
     1 DELTA)
      GO TO 107
  403 CALL        MDCYRC(GMASS,SPQ,SFR,APS,G,BZ,CPS,VEL,NC4,MR,NP,NS,
     1 DELTA)
      GO TO 107
  103 IF (NC4-2) 201,202,203
  201 CALL        MDFRCE (BX,P,MR,NP)
      GO TO 105
  202 CALL        MDFRCE (BY,P,MR,NP)
      GO TO 105
  203 CALL        MDFRCE (BZ,P,MR,NP)
  105 CONTINUE
      CALL        MODRCP (P,GMASS,SPQ,SFR,APS,CPS,MR,NS,NC4)
  107 CONTINUE
      DO 106 NCT=1,NC5
      READ 23, NC6,N
   23 FORMAT (2I4)
      IF (NC6-2) 301,302,303
  301 CALL        SPECTR (FX,SPQ,A,N,MR,NP,NS,NC6)
      GO TO 304
  302 CALL        SPECTR (FY,SPQ,A,N,MR,NP,NS,NC6)
      GO TO 304
  303 CALL        SPECTR (FZ,SPQ,A,N,MR,NP,NS,NC6)
  304 CONTINUE
```

```
      CALL        SIMPSN (AREA,DELTA,A,NS)
      SIGMA=SQRTF(AREA)
      PRINT 305, AREA,SIGMA
  305 FORMAT (10H0VARIANCE=1F20.6,9H  R.M.S.=1F20.6//)
      CALL        ASUMP(A,B,D,NS,DELTA,AREA,ARSQRT)
      CALL        MAGFAC (ARSQRT,SIGMA)
  106 CONTINUE
  101 CONTINUE
      PRINT 108
  108 FORMAT (29H0THE END OF JOB VS030 ERMUTLU////)
      GO TO EXIT
      END
```

(1) SUBROUTINE ELEAR2

```
      SUBROUTINE ELEAR2(NUMEL,NUMNP,     NPI,NPJ,NPK,NPL,ET,XU,RO,
     1        XORD,YORD,ZORD,XLOAD,YLOAD,ZLOAD,DSX,DSY,DSZ,
     2 NOD,NPM,K1,K2,K3,K4,K5,K6,ND,MD)
      DIMENSION            NPI(ND),NPJ(ND),NPK(ND),NPL(ND),ET(ND),
     1 XU(ND),RO(ND),            XORD(MD),YORD(MD),ZORD(MD),DSX(MD),DSY
     2 (MD),DSZ(MD),XLOAD(MD),YLOAD(MD),ZLOAD(MD),NOD(K1,K2,K3),
     3 NPM(K1,K2)
      READ   41, ELMOD,POISON,UNIDOM
   41 FORMAT(3F15.5)
      READ   7, ((NPM(I,J),J=1,K2),I=1,K1)
    7 FORMAT(20I4)
      CALL        NOKTA(NPI,NPJ,NPK,NPL,NPM,NOD,K1,K2,K3,ND)
      IF (K4) 51,51,52
   52 READ 70, N
   70 FORMAT(1I4)
      DO 75 I=1,K4
      READ 9, NPI(N),NPJ(N),NPK(N),NPL(N)
    9 FORMAT(4X,4I4)
   75 N=N+1
   51 DO 76 N=1,NUMEL
      ET(N)=ELMOD
      XU(N)=POISON
   76 RO(N)=UNIDOM
      CALL        CORDLD(XORD,YORD,ZORD,XLOAD,YLOAD,ZLOAD,NOD,K1,K2,K3,
     1 MD)
      IF (K5) 53,53,54
   54 READ 70, M
      DO 55 I=1,K5
      READ 56, XORD(M),YORD(M),ZORD(M),XLOAD(M),YLOAD(M),ZLOAD(M)
   56 FORMAT(4X,3F8.1,3F12.1)
   55 M=M+1
   53 DO 57 M=1,NUMNP
      DSX(M)=0.0
      DSY(M)=0.0
   57 DSZ(M)=0.0
      IF (K6) 58,58,59
   59 READ 70, M
      DO 60 I=1,K6
      READ 61, DSX(M),DSY(M),DSZ(M)
   61 FORMAT(4X,3F8.4)
   60 M=M+1
   58 RETURN
      END
```

(2) SUBROUTINE NOKTA

```
      SUBROUTINE NOKTA(NPI,NPJ,NPK,NPL,NPM,NOD,K1,K2,K3,ND)
      DIMENSION NPI(ND),NPJ(ND),NPK(ND),NPL(ND),NPM(K1,K2),
     1 NOD(K1,K2,K3)
      M=0
      DO 1 I=1,K1
      DO 1 J=1,K2
      KN=NPM(I,J)
      DO 1 K=1,KN
      M=M+1
      NOD(I,J,K)=M
    1 CONTINUE
C
      N=0
      K1A=K1-1
      K2A=K2-1
      DO 2 I=1,K1A
      DO 2 J=1,K2A
      KNA=NPM(I,J)-1
      DO 5 K=1,KNA
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 3,3,4
    3 N=N+1
      NPI(N)=NOD(I,J,K)
      NPJ(N)=NOD(I,J+1,K)
      NPK(N)=NOD(I+1,J,K)
      NPL(N)=NOD(I,J,K+1)
      N=N+1
      NPI(N)=NOD(I,J+1,K+1)
      NPJ(N)=NOD(I,J,K+1)
      NPK(N)=NOD(I+1,J+1,K+1)
      NPL(N)=NOD(I,J+1,K)
      N=N+1
      NPI(N)=NOD(I+1,J,K+1)
      NPJ(N)=NOD(I+1,J+1,K+1)
      NPK(N)=NOD(I,J,K+1)
      NPL(N)=NOD(I+1,J,K)
      N=N+1
      NPI(N)=NOD(I+1,J+1,K)
      NPJ(N)=NOD(I+1,J,K)
      NPK(N)=NOD(I,J+1,K)
      NPL(N)=NOD(I+1,J+1,K+1)
      N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J,K)
      NPK(N)=NOD(I,J,K+1)
      NPL(N)=NOD(I+1,J+1,K+1)
      GO TO 5
    4 N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J+1,K)
      NPK(N)=NOD(I,J,K)
      NPL(N)=NOD(I,J+1,K+1)
```

```
      N=N+1
      NPI(N)=NOD(I,J,K+1)
      NPJ(N)=NOD(I+1,J,K+1)
      NPK(N)=NOD(I,J+1,K+1)
      NPL(N)=NOD(I,J,K)
      N=N+1
      NPI(N)=NOD(I+1,J,K)
      NPJ(N)=NOD(I,J,K)
      NPK(N)=NOD(I+1,J+1,K)
      NPL(N)=NOD(I+1,J,K+1)
      N=N+1
      NPI(N)=NOD(I+1,J+1,K+1)
      NPJ(N)=NOD(I,J+1,K+1)
      NPK(N)=NOD(I+1,J,K+1)
      NPL(N)=NOD(I+1,J+1,K)
      N=N+1
      NPI(N)=NOD(I,J,K)
      NPJ(N)=NOD(I,J+1,K+1)
      NPK(N)=NOD(I+1,J+1,K)
      NPL(N)=NOD(I+1,J,K+1)
    5 CONTINUE
      IF (NPM(I,J)-NPM(I+1,J)) 6,2,2
    6 K=NPM(I,J)
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 7,7,8
    7 N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J,K)
      NPK(N)=NOD(I,J,K)
      NPL(N)=NOD(I+1,J,K+1)
      N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J+1,K)
      NPK(N)=NOD(I+1,J,K)
      NPL(N)=NOD(I+1,J+1,K+1)
      N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J+1,K+1)
      NPK(N)=NOD(I+1,J,K)
      NPL(N)=NOD(I+1,J,K+1)
      GO TO 2
    8 N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J+1,K)
      NPK(N)=NOD(I,J,K)
      NPL(N)=NOD(I+1,J+1,K+1)
      N=N+1
      NPI(N)=NOD(I,J,K)
      NPJ(N)=NOD(I+1,J+1,K)
      NPK(N)=NOD(I+1,J,K)
      NPL(N)=NOD(I+1,J,K+1)
      N=N+1
      NPI(N)=NOD(I,J,K)
      NPJ(N)=NOD(I+1,J+1,K+1)
      NPK(N)=NOD(I+1,J+1,K)
      NPL(N)=NOD(I+1,J,K+1)
    2 CONTINUE
      RETURN
      END
```

(3) SUBROUTINE NOKTB

```
      SUBROUTINE NOKTB(NPI,NPJ,NPK,NPL,NPM,NOD,K1,K2,K3,ND)
      DIMENSION NPI(ND),NPJ(ND),NPK(ND),NPL(ND),NPM(K1,K2),
     1 NOD(K1,K2,K3)
      N=0
      K1A=K1=1
      K2A=K2=1
      DO 2 I=1,K1A
      DO 2 J=1,K2A
      KNA=NPM(I,J)=1
      DO 5 K=1,KNA
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 3,3,4
    4 N=N+1
      NPI(N)=NOD(I,J,K)
      NPJ(N)=NOD(I,J+1,K)
      NPK(N)=NOD(I+1,J,K)
      NPL(N)=NOD(I,J,K+1)
      N=N+1
      NPI(N)=NOD(I,J+1,K+1)
      NPJ(N)=NOD(I,J,K+1)
      NPK(N)=NOD(I+1,J+1,K+1)
      NPL(N)=NOD(I,J+1,K)
      N=N+1
      NPI(N)=NOD(I+1,J,K+1)
      NPJ(N)=NOD(I+1,J+1,K+1)
      NPK(N)=NOD(I,J,K+1)
      NPL(N)=NOD(I+1,J,K)
      N=N+1
      NPI(N)=NOD(I+1,J+1,K)
      NPJ(N)=NOD(I+1,J,K)
      NPK(N)=NOD(I,J+1,K)
      NPL(N)=NOD(I+1,J+1,K+1)
      N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J,K)
      NPK(N)=NOD(I,J,K+1)
      NPL(N)=NOD(I+1,J+1,K+1)
      GO TO 5
    3 N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J+1,K)
      NPK(N)=NOD(I,J,K)
      NPL(N)=NOD(I,J+1,K+1)
      N=N+1
      NPI(N)=NOD(I,J,K+1)
      NPJ(N)=NOD(I+1,J,K+1)
      NPK(N)=NOD(I,J+1,K+1)
      NPL(N)=NOD(I,J,K)
      N=N+1
      NPI(N)=NOD(I+1,J,K)
      NPJ(N)=NOD(I,J,K)
      NPK(N)=NOD(I+1,J+1,K)
      NPL(N)=NOD(I+1,J,K+1)
      N=N+1
      NPI(N)=NOD(I+1,J+1,K+1)
      NPJ(N)=NOD(I,J+1,K+1)
      NPK(N)=NOD(I+1,J,K+1)
      NPL(N)=NOD(I+1,J+1,K)
```

```
      N=N+1
      NPI(N)=NOD(I,J,K)
      NPJ(N)=NOD(I,J+1,K+1)
      NPK(N)=NOD(I+1,J+1,K)
      NPL(N)=NOD(I+1,J,K+1)
    5 CONTINUE
      IF (NPM(I,J)-NPM(I+1,J)) 6,2,2
    6 K=NPM(I,J)
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 7,7,8
    8 N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J,K)
      NPK(N)=NOD(I,J,K)
      NPL(N)=NOD(I+1,J,K+1)
      N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J+1,K)
      NPK(N)=NOD(I+1,J,K)
      NPL(N)=NOD(I+1,J+1,K+1)
      N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J+1,K+1)
      NPK(N)=NOD(I+1,J,K)
      NPL(N)=NOD(I+1,J,K+1)
      GO TO 2
    7 N=N+1
      NPI(N)=NOD(I,J+1,K)
      NPJ(N)=NOD(I+1,J+1,K)
      NPK(N)=NOD(I,J,K)
      NPL(N)=NOD(I+1,J+1,K+1)
      N=N+1
      NPI(N)=NOD(I,J,K)
      NPJ(N)=NOD(I+1,J+1,K)
      NPK(N)=NOD(I+1,J,K)
      NPL(N)=NOD(I+1,J,K+1)
      N=N+1
      NPI(N)=NOD(I,J,K)
      NPJ(N)=NOD(I+1,J+1,K+1)
      NPK(N)=NOD(I+1,J+1,K)
      NPL(N)=NOD(I+1,J,K+1)
    2 CONTINUE
      RETURN
      END
```

(4) SUBROUTINE CORDLD

```
      SUBROUTINE CORDLD(XORD,YORD,ZORD,XLOAD,YLOAD,ZLOAD,NOD,K1,K2,K3,
     1 MD)
      DIMENSION XORD(MD),YORD(MD),ZORD(MD),NOD(K1,K2,K3)
     1 ,XLOAD(MD),YLOAD(MD),ZLOAD(MD)
      DO 1 K=1,K3
      READ   2, HLEVEL,RAD,CENTER,THICK,ALPHA,NDIV,HLOAD,VLOAD
    2 FORMAT (4F8.1,1F8.3,1I4,2F12.1)
      T=THICK/(K2-1)
      FA=ALPHA*3.14159/180.0
      FD=FA/NDIV
```

```
      KA=K1-NDIV
      NS=0
      DO 1 J=1,K2
      R=RAD-NS*T
      NS=NS+1
      F=FA
      DO 1 I=KA,K1
      SN=SINF(F)
      CS=COSF(F)
      N=NOD(I,J,K)
      XORD(N)=R*SN
      YORD(N)=R*CS+CENTER
      ZORD(N)=HLEVEL
      IF (J-1) 3,3,4
    3 IF (I-K1) 6,7,7
    7 XLOAD(N)=0.0
      YLOAD(N)=HLOAD/2.0
      IF (K-1) 8,8,9
    8 ZLOAD(N)=VLOAD/2.0
      GO TO 5
    9 ZLOAD(N)=HLOAD/2.0*(YORD(N)-YORD(N-1))/(ZORD(N-1)-ZORD(N))
     1 +VLOAD/2.0
      GO TO 5
    6 XLOAD(N)=HLOAD*SN
      YLOAD(N)=HLOAD*CS
      IF (K-1) 10,10,11
   10 ZLOAD(N)=VLOAD
      GO TO 5
   11 DY=YORD(N)-YORD(N-1)
      DX=XORD(N)-XORD(N-1)
      DZ=ZORD(N-1)-ZORD(N)
      IF (DY) 12,13,13
   12 DXY=DX*DX+DY*DY
      ZLOAD(N)=-HLOAD*SQRTF(DXY)/DZ+VLOAD
      GO TO 5
   13 DXY=DX*DX+DY*DY
      ZLOAD(N)= HLOAD*SQRTF(DXY)/DZ+VLOAD
      GO TO 5
    4 XLOAD(N)=0.0
      YLOAD(N)=0.0
      ZLOAD(N)=0.0
    5 F=F-FD
    1 CONTINUE
      RETURN
      END
```

(5) SUBROUTINE SINIR

```
      SUBROUTINE SINIR(NPB,NFIX,NOD,NPM,K1,K2,K3,K7,LD)
      DIMENSION NPB(LD),NFIX(LD),NOD(K1,K2,K3),NPM(K1,K2)
      N=0
      DO 1 J=1,K2
      KN=NPM(1,J)
      DO 1 K=1,KN
      N=N+1
      NPB(N)=NOD(1,J,K)
    1 NFIX(N)=0
```

```
      DO 2 I=2,K1
      DO 2 J=1,K2
      KN=NPM(I,J)
      N=N+1
      NPB(N)=NOD(I,J,KN)
    2 NFIX(N)=0
      DO 3 J=1,K2
      KNA=NPM(K1,J)-1
      DO 3 K=1,KNA
      N=N+1
      NPB(N)=NOD(K1,J,K)
    3 NFIX(N)=1
      IF (K7) 4,4,5
    5 DO 6 L=1,K7
      N=N+1
    6 READ    7, NPB(N),NFIX(N)
    7 FORMAT(2I4)
    4 RETURN
      END
```

(6) SUBROUTINE MODSTF

```
      SUBROUTINE MODSTF (NUMEL,NPI,NPJ,NPK,NPL,XORD,YORD,ZORD,RO,ET,
     1 XU,ZLOAD,BTDB,A,S,LM,NP,R,ND,MD,ID,JD)
      DIMENSION  NPI(ND),NPJ(ND),NPK(ND),NPL(ND),XORD(MD),YORD(MD),
     1 ZORD(MD),RO(ND),ET(ND),XU(ND),BTDB(12,12),S(12,12),A(12,12),
     2 LM(4),NP(MD,JD),R(9,ID,MD),ZLOAD(MD)
C     MODIFICATION OF LOADS AND ELEMENT DIMENSIONS
      DO 200 N=1,NUMEL
 153  I=NPI(N)
      J=NPJ(N)
      K=NPK(N)
      L=NPL(N)
      AJ   =XORD(J)-XORD(I)
      BJ   =YORD(J)-YORD(I)
      CJ   =ZORD(J)-ZORD(I)
      AK   =XORD(K)-XORD(I)
      BK   =YORD(K)-YORD(I)
      CK   =ZORD(K)-ZORD(I)
      AL   =XORD(L)-XORD(I)
      BL   =YORD(L)-YORD(I)
      CL   =ZORD(L)-ZORD(I)
      SIXVOL= AJ*BK*CL-AJ*BL*CK+AK*BL*CJ-AK*BJ*CL+AL*BJ*CK-AL*BK*CJ
      IF (SIXVOL)     701,703,177
 701  PRINT    711,        N ,NPJ(N),NPK(N)
 711  FORMAT(12H ELEMENT NO.,I4,31H INTERCHANGED NODAL POINT NO.S.2I4)
      K=NPJ(N)
      J=NPK(N)
      NPJ(N)=J
      NPK(N)=K
      GO TO 153
 703  PRINT    713, N
 713  FORMAT(21H0ZERO VOLUME. EL.NO.=1I4)
      GO TO EXIT
 177  VOL=SIXVOL    /6.0
      PRINT     6,         N ,NPI(N),NPJ(N),NPK(N),NPL(N),VOL
    6 FORMAT (56X,5I4,1F25.6)
```

```
      DL=VOL*RO(N)/4.
      DL=0.5*DL
      ZLOAD(I)=ZLOAD(I)-DL
      ZLOAD(J)=ZLOAD(J)-DL
      ZLOAD(K)=ZLOAD(K)-DL
  180 ZLOAD(L)=ZLOAD(L)-DL
C     FORMATION OF STIFFNESS ARRAY
      COMM=ET(N)/((1.+XU(N))*(1.-2.*XU(N))*36.*VOL)
      D1=(1.-XU(N))*COMM
      D2=XU(N)*COMM
      D3=(0.5-XU(N))*COMM
      DO 2001 J=1,12
      DO 2001 I=1,12
 2001 BTDB(I,J)=0.0
      BTDB(2,2)=D1
      BTDB(7,7)=D1
      BTDB(12,12)=D1
      BTDB(2,7)=D2
      BTDB(2,12)=D2
      BTDB(7,2)=D2
      BTDB(7,12)=D2
      BTDB(12,2)=D2
      BTDB(12,7)=D2
      BTDB(3,3)=D3
      BTDB(3,6)=D3
      BTDB(4,4)=D3
      BTDB(4,10)=D3
      BTDB(6,3)=D3
      BTDB(6,6)=D3
      BTDB(8,8)=D3
      BTDB(8,11)=D3
      BTDB(10,4)=D3
      BTDB(10,10)=D3
      BTDB(11,8)=D3
      BTDB(11,11)=D3
      A21=BJ*CL-BJ*CK+BK*CJ-BK*CL+BL*CK-BL*CJ
      A22=BK*CL-BL*CK
      A23=BL*CJ-BJ*CL
      A24=BJ*CK-BK*CJ
      A31=AJ*CK-AJ*CL+AK*CL-AK*CJ+AL*CJ-AL*CK
      A32=AL*CK-AK*CL
      A33=AJ*CL-AL*CJ
      A34=AK*CJ-AJ*CK
      A41=AJ*BL-AJ*BK+AK*BJ-AK*BL+AL*BK-AL*BJ
      A42=AK*BL-AL*BK
      A43=AL*BJ-AJ*BL
      A44=AJ*BK-AK*BJ
      DO 2002 J=1,12
      DO 2002 I=1,12
 2002 A(I,J)=0.
      A(1,1)=1.0
      A(5,2)=1.0
      A(9,3)=1.0
      A(2,1)=A21
      A(6,2)=A21
      A(10,3)=A21
      A(3,1)=A31
      A(7,2)=A31
      A(11,3)=A31
      A(4,1)=A41
      A(8,2)=A41
```

```
      A(12,3)=A41
      A(2,4)=A22
      A(6,5)=A22
      A(10,6)=A22
      A(3,4)=A32
      A(7,5)=A32
      A(11,6)=A32
      A(4,4)=A42
      A(8,5)=A42
      A(12,6)=A42
      A(2,7)=A23
      A(6,8)=A23
      A(10,9)=A23
      A(3,7)=A33
      A(7,8)=A33
      A(11,9)=A33
      A(4,7)=A43
      A(8,8)=A43
      A(12,9)=A43
      A(2,10)=A24
      A(6,11)=A24
      A(10,12)=A24
      A(3,10)=A34
      A(7,11)=A34
      A(11,12)=A34
      A(4,10)=A44
      A(8,11)=A44
      A(12,12)=A44
      DO 181 J=1,12
      DO 181 I=1,12
      S(I,J)=0.
      DO 181 K=1,12
  181 S(I,J)=S(I,J)+BTDB(I,K)*A(K,J)
      DO 182 J=1,12
      DO 182 I=1,12
  182 BTDB(J,I)=A(I,J)
      DO 183 J=1,12
      DO 183 I=1,12
  183 A(I,J)=S(I,J)
      DO 184 J=1,12
      DO 184 I=1,12
      S(I,J)=0.
      DO 184 K=1,12
  184 S(I,J)=S(I,J)+BTDB(I,K)*A(K,J)
      WRITE TAPE 23, ((S(I,J),I=1,12),J=1,12)
      DO 86 J=1,12
      DO 87 I=1,12
   87 S(I,J)=0.5*S(I,J)
   86 CONTINUE
      LM(1)=NPI(N)
      LM(2)=NPJ(N)
      LM(3)=NPK(N)
      LM(4)=NPL(N)
      DO 200 L=1,4
      DO 200 M=1,4
      LX=LM(L)
      MX=0
  185 MX=MX+1
      IF (NP(LX,MX)-LM(M)) 190,195,190
  190 IF (NP(LX,MX)) 185,195,185
  195 NP(LX,MX)=LM(M)
```

```
      IF (MX-JD) 196,702,702
  702 PRINT    712, (LX)
  712 FORMAT (31H0MORE N.P. ADJACENT TO N.P. NO.1I4)
      GO TO EXIT
  196 R(1,MX,LX)=R(1,MX,LX)+S(3*L-2,3*M-2)
      R(2,MX,LX)=R(2,MX,LX)+S(3*L-2,3*M-1)
      R(3,MX,LX)=R(3,MX,LX)+S(3*L-2,3*M)
      R(4,MX,LX)=R(4,MX,LX)+S(3*L-1,3*M-2)
      R(5,MX,LX)=R(5,MX,LX)+S(3*L-1,3*M-1)
      R(6,MX,LX)=R(6,MX,LX)+S(3*L-1,3*M)
      R(7,MX,LX)=R(7,MX,LX)+S(3*L,3*M-2)
      R(8,MX,LX)=R(8,MX,LX)+S(3*L,3*M-1)
  200 R(9,MX,LX)=R(9,MX,LX)+S(3*L,3*M)
      RETURN
      END
```

(7) SUBROUTINE INVNP2

```
      SUBROUTINE INVNP2(NUMNP,R,MD,ID)
      DIMENSION R(9,ID,MD)
C     INVERSION OF NODAL POINT STIFFNESS
      DO 210 M=1,NUMNP
      COMM=R(1,1,M)*R(5,1,M)*R(9,1,M)+R(4,1,M)*R(8,1,M)*R(3,1,M)+
     1 R(7,1,M)*R(2,1,M)*R(6,1,M)-R(3,1,M)*R(5,1,M)*R(7,1,M)-
     2 R(1,1,M)*R(8,1,M)*R(6,1,M)-R(4,1,M)*R(2,1,M)*R(9,1,M)
      S11=( R(5,1,M)*R(9,1,M)-R(8,1,M)*R(6,1,M))/COMM
      S12=(-R(2,1,M)*R(9,1,M)+R(8,1,M)*R(3,1,M))/COMM
      S13=( R(2,1,M)*R(6,1,M)-R(5,1,M)*R(3,1,M))/COMM
      S21=(-R(4,1,M)*R(9,1,M)+R(7,1,M)*R(6,1,M))/COMM
      S22=( R(1,1,M)*R(9,1,M)-R(7,1,M)*R(3,1,M))/COMM
      S23=(-R(1,1,M)*R(6,1,M)+R(4,1,M)*R(3,1,M))/COMM
      S31=( R(4,1,M)*R(8,1,M)-R(7,1,M)*R(5,1,M))/COMM
      S32=(-R(1,1,M)*R(8,1,M)+R(7,1,M)*R(2,1,M))/COMM
      S33=( R(1,1,M)*R(5,1,M)-R(4,1,M)*R(2,1,M))/COMM
      R(1,1,M)=S11
      R(2,1,M)=S12
      R(3,1,M)=S13
      R(4,1,M)=S21
      R(5,1,M)=S22
      R(6,1,M)=S23
      R(7,1,M)=S31
      R(8,1,M)=S32
  210 R(9,1,M)=S33
      RETURN
      END
```

(8) SUBROUTINE BOUND2

```
      SUBROUTINE BOUND2(NUMBC,NPB,NFIX,NP,R,LD,MD,ID,JD)
      DIMENSION NPB(LD),NFIX(LD),NP(MD,JD),R(9,ID,MD)
C     MODIFICATION OF BOUNDARY FLEXIBILITIES
      DO 240 L=1,NUMBC
      M=NPB(L)
      NP(M,1)=0
```

```
      IF (NFIX(L)-1) 225,220,215
215   IF(NFIX(L)-3) 230,235,240
220   R(5,1,M)=R(5,1,M)-R(4,1,M)*R(2,1,M)/R(1,1,M)
      R(6,1,M)=R(6,1,M)-R(4,1,M)*R(3,1,M)/R(1,1,M)
      R(8,1,M)=R(8,1,M)-R(7,1,M)*R(2,1,M)/R(1,1,M)
      R(9,1,M)=R(9,1,M)-R(7,1,M)*R(3,1,M)/R(1,1,M)
      R(1,1,M)=0.0
      R(2,1,M)=0.0
      R(3,1,M)=0.0
      R(4,1,M)=0.0
      R(7,1,M)=0.0
      GO TO 240
230   R(1,1,M)=R(1,1,M)-R(2,1,M)*R(4,1,M)/R(5,1,M)
      R(3,1,M)=R(3,1,M)-R(2,1,M)*R(6,1,M)/R(5,1,M)
      R(7,1,M)=R(7,1,M)-R(8,1,M)*R(4,1,M)/R(5,1,M)
      R(9,1,M)=R(9,1,M)-R(8,1,M)*R(6,1,M)/R(5,1,M)
      R(2,1,M)=0.0
      R(4,1,M)=0.0
      R(5,1,M)=0.0
      R(6,1,M)=0.0
      R(8,1,M)=0.0
      GO TO 240
235   R(1,1,M)=R(1,1,M)-R(3,1,M)*R(7,1,M)/R(9,1,M)
      R(2,1,M)=R(2,1,M)-R(3,1,M)*R(8,1,M)/R(9,1,M)
      R(4,1,M)=R(4,1,M)-R(6,1,M)*R(7,1,M)/R(9,1,M)
      R(5,1,M)=R(5,1,M)-R(6,1,M)*R(8,1,M)/R(9,1,M)
      GO TO 236
225   R(1,1,M)=0.0
      R(2,1,M)=0.0
      R(4,1,M)=0.0
      R(5,1,M)=0.0
236   R(3,1,M)=0.0
      R(6,1,M)=0.0
      R(7,1,M)=0.0
      R(8,1,M)=0.0
      R(9,1,M)=0.0
240   CONTINUE
      RETURN
      END
```

(9) SUBROUTINE ITERA2

```
      SUBROUTINE ITERA2 (SUM,NUMNP,NAP,XLOAD,YLOAD,ZLOAD,DSX,DSY,DSZ,
     1 NP,R,XFAC,MD,ID,JD)
      DIMENSION NAP(MD),XLOAD(MD),YLOAD(MD),ZLOAD(MD),DSX(MD),DSY(MD),
     1 DSZ(MD),NP(MD,JD),R(9,ID,MD)
C     ITERATION ON NODAL POINT DISPLACEMENTS
      DO 290 M=1,NUMNP
      NUM=NAP(M)
      IF (R(1,1,M)+R(5,1,M)+R(9,1,M)) 275,290,275
275   FRX=XLOAD(M)
      FRY=YLOAD(M)
      FRZ=ZLOAD(M)
```

```
      DO 280 L=2,NUM
      N=NP(M,L)
      FRX=FRX-R(1,L,M)*DSX(N)-R(2,L,M)*DSY(N)-R(3,L,M)*DSZ(N)
      FRY=FRY-R(4,L,M)*DSX(N)-R(5,L,M)*DSY(N)-R(6,L,M)*DSZ(N)
  280 FRZ=FRZ-R(7,L,M)*DSX(N)-R(8,L,M)*DSY(N)-R(9,L,M)*DSZ(N)
      DX=R(1,1,M)*FRX+R(2,1,M)*FRY+R(3,1,M)*FRZ-DSX(M)
      DY=R(4,1,M)*FRX+R(5,1,M)*FRY+R(6,1,M)*FRZ-DSY(M)
      DZ=R(7,1,M)*FRX+R(8,1,M)*FRY+R(9,1,M)*FRZ-DSZ(M)
      DSX(M)=DSX(M)+XFAC*DX
      DSY(M)=DSY(M)+XFAC*DY
      DSZ(M)=DSZ(M)+XFAC*DZ
      IF (NP(M,1)) 285,290,285
  285 SUM=SUM+ABSF(DX/R(1,1,M))+ABSF(DY/R(5,1,M))+ABSF(DZ/R(9,1,M))
  290 CONTINUE
      RETURN
      END
```

(10) SUBROUTINE REACT2

```
      SUBROUTINE REACT2 (NUMEL,NUMNP,NPI,NPJ,NPK,NPL,DSX,DSY,DSZ,
     1 XLOAD,YLOAD,ZLOAD,S,LM,ND,MD,
     2 R,ID,NOD,NPM,K1,K2,K3)
      DIMENSION NPI(ND),NPJ(ND),NPK(ND),NPL(ND),DSX(MD),DSY(MD),DSZ(MD),
     1 XLOAD(MD),YLOAD(MD),ZLOAD(MD),S(12,12),LM(4)
      DIMENSION          R(9,ID,MD),NOD(K1,K2,K3),NPM(K1,K2),LN(8)
      PRINT 500
  500 FORMAT(1H1)
      REWIND 23
      REWIND 24
      DO 165 M=1,NUMNP
      DO 166 L=1,8
      DO 167 K=1,3
  167 R(K,L,M)=0.0
  166 CONTINUE
  165 CONTINUE
      DO 510 M=1,NUMNP
      XLOAD(M)=0.0
      YLOAD(M)=0.0
      ZLOAD(M)=0.0
  510 CONTINUE
      DO 999 NCONT=1,2
      READ  TAPE 24, (NPI(N),NPJ(N),NPK(N),NPL(N),N=1,NUMEL)
      N=0
      K1A=K1-1
      K2A=K2-1
      DO 1 I=1,K1A
      DO 2 J=1,K2A
      KNA=NPM(I,J)
      DO 3 K=1,KNA
      IF (K-KNA) 15,25,25
   25 IF (NPM(I,J)-NPM(I+1,J)) 45,2,2
   45 MLP=3
```

```
      LN(2)=0
      LN(4)=0
      GO TO 35
   15 MLP=5
      LN(2)=NOD(I,J,K+1)
      LN(4)=NOD(I,J+1,K+1)
   35 LN(1)=NOD(I,J,K)
      LN(3)=NOD(I,J+1,K)
      LN(5)=NOD(I+1,J,K)
      LN(6)=NOD(I+1,J,K+1)
      LN(7)=NOD(I+1,J+1,K)
      LN(8)=NOD(I+1,J+1,K+1)
      DO 4 KLM=1,MLP
      N=N+1
      READ TAPE 23, ((S(IJ,JI),IJ=1,12),JI=1,12)
      LM(1)=NPI(N)
      LM(2)=NPJ(N)
      LM(3)=NPK(N)
      LM(4)=NPL(N)
      DO 530 M=1,4
      MX=LM(M)
      XFORCE=0.0
      YFORCE=0.0
      ZFORCE=0.0
      DO 540 L=1,4
      LX=LM(L)
      XFORCE=XFORCE+S(3*M-2,3*L-2)*DSX(LX)+S(3*M-2,3*L-1)*DSY(LX)
     1 +S(3*M-2,3*L)*DSZ(LX)
      YFORCE=YFORCE+S(3*M-1,3*L-2)*DSX(LX)+S(3*M-1,3*L-1)*DSY(LX)
     1 +S(3*M-1,3*L)*DSZ(LX)
      ZFORCE=ZFORCE+S(3*M,3*L-2)*DSX(LX)+S(3*M,3*L-1)*DSY(LX)
     1 +S(3*M,3*L)*DSZ(LX)
  540 CONTINUE
      XLOAD(MX)=XLOAD(MX)+XFORCE*0.5
      YLOAD(MX)=YLOAD(MX)+YFORCE*0.5
      ZLOAD(MX)=ZLOAD(MX)+ZFORCE*0.5
      NO=1
    7 IF (LN(NO)=MX) 6,55,6
    6 NO=NO+1
      GO TO 7
   55 R(1,NO,MX)=R(1,NO,MX)+XFORCE*0.5
      R(2,NO,MX)=R(2,NO,MX)+YFORCE*0.5
      R(3,NO,MX)=R(3,NO,MX)+ZFORCE*0.5
  530 CONTINUE
    4 CONTINUE
    3 CONTINUE
    2 CONTINUE
    1 CONTINUE
  999 CONTINUE
      PRINT 503
  503 FORMAT(57H1NODAL POINT      X-REACTION      Y-REACTION      Z-REACTIO
     1N)
      PRINT 504, (M,XLOAD(M),YLOAD(M),ZLOAD(M),M=1,NUMNP)
  504 FORMAT(1I12,3F15.1)
      RETURN
      END
```

(11) SUBROUTINE FORCE

```
      SUBROUTINE FORCE  (NUMNP,R,ID,MD)
      DIMENSION R(9,ID,MD)
      DO 1 I=1,3
      PRINT 2
    2 FORMAT (1H1)
      DO 3 M=1,NUMNP
      UPWARD=R(I,2,M)+R(I,4,M)+R(I,6,M)+R(I,8,M)
      DOWNWD=R(I,1,M)+R(I,3,M)+R(I,5,M)+R(I,7,M)
      FORWRD=R(I,3,M)+R(I,4,M)+R(I,7,M)+R(I,8,M)
      BACKWD=R(I,1,M)+R(I,2,M)+R(I,5,M)+R(I,6,M)
      FLEFTW=R(I,5,M)+R(I,6,M)+R(I,7,M)+R(I,8,M)
      FRIGHT=R(I,1,M)+R(I,2,M)+R(I,3,M)+R(I,4,M)
      PRINT 4, I,M,UPWARD,DOWNWD,FORWRD,BACKWD,FLEFTW,FRIGHT
    4 FORMAT (I2,I6,6F12.1)
    3 CONTINUE
    1 CONTINUE
      RETURN
      END
```

(12) SUBROUTINE ELEARM

```
      SUBROUTINE ELEARM(NE,NP,NGE,G,ET,XU,RO,NOD,NPM,K1,K2,K3,K4,K5)
      DIMENSION NGE(NE,4),G(NP,3)ET(NE),XU(NE),RO(NE),NOD(K1,K2,K3),
     1 NPM(K1,K2)
      READ 41, ELMOD,POISON,UNIDOM
   41 FORMAT(3F15.5)
      READ 7, ((NPM(I,J),J=1,K2),I=1,K1)
    7 FORMAT(20I4)
      CALL NOKTAM(NGE,NPM,NOD,K1,K2,K3,NE)
      IF (K4) 51,51,52
   52 READ 70, N
   70 FORMAT(1I4)
      DO 75 I=1,K4
      READ 9, (NGE(N,L),L=1,4)
    9 FORMAT(4X,4I4)
   75 N=N+1
   51 DO 76 N=1,NE
      ET(N)=ELMOD
      XU(N)=POISON
   76 RO(N)=UNIDOM
      CALL CORDLM(G,NOD,K1,K2,K3,NP)
      IF (K5) 53,53,54
   54 READ 70, M
      DO 55 I=1,K5
      READ 56, (G(M,L),L=1,3)
   56 FORMAT(4X,3F8.3)
   55 M=M+1
   53 RETURN
      END
```

(13) SUBROUTINE NOKTAM

```
      SUBROUTINE NOKTAM(L,NPM,NOD,K1,K2,K3,NE)
      DIMENSION L(NE,4),NPM(K1,K2),NOD(K1,K2,K3)
      M=0
      DO 1 I=1,K1
      DO 1 J=1,K2
      KN=NPM(I,J)
      DO 1 K=1,KN
      M=M+1
      NOD(I,J,K)=M
    1 CONTINUE
C
      N=0
      K1A=K1-1
      K2A=K2-1
      DO 2 I=1,K1A
      DO 2 J=1,K2A
      KNA=NPM(I,J)-1
      DO 5 K=1,KNA
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 3,3,4
    3 N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K+1)
      L(N,2)=NOD(I,J,K+1)
      L(N,3)=NOD(I+1,J+1,K+1)
      L(N,4)=NOD(I,J+1,K)
      N=N+1
      L(N,1)=NOD(I+1,J,K+1)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I,J,K+1)
      L(N,4)=NOD(I+1,J,K)
      N=N+1
      L(N,1)=NOD(I+1,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J+1,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J,K+1)
      L(N,4)=NOD(I+1,J+1,K+1)
      GO TO 5
    4 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I,J+1,K+1)
```

```
      N=N+1
      L(N,1)=NOD(I,J,K+1)
      L(N,2)=NOD(I+1,J,K+1)
      L(N,3)=NOD(I,J+1,K+1)
      L(N,4)=NOD(I,J,K)
      N=N+1
      L(N,1)=NOD(I+1,J,K)
      L(N,2)=NOD(I,J,K)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I+1,J+1,K+1)
      L(N,2)=NOD(I,J+1,K+1)
      L(N,3)=NOD(I+1,J,K+1)
      L(N,4)=NOD(I+1,J+1,K)
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I,J+1,K+1)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
    5 CONTINUE
      IF (NPM(I,J)-NPM(I+1,J)) 6,2,2
    6 K=NPM(I,J)
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 7,7,8
    7 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      GO TO 2
    8 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
    2 CONTINUE
      RETURN
      END
```

(14) SUBROUTINE NOKTBM

```
      SUBROUTINE NOKTBM(L,NPM,NOD,K1,K2,K3,NE)
      DIMENSION L(NE,4),NPM(K1,K2),NOD(K1,K2,K3)
      N=0
      K1A=K1-1
      K2A=K2-1
      DO 2 I=1,K1A
      DO 2 J=1,K2A
      KNA=NPM(I,J)-1
      DO 5 K=1,KNA
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 3,3,4
    4 N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K+1)
      L(N,2)=NOD(I,J,K+1)
      L(N,3)=NOD(I+1,J+1,K+1)
      L(N,4)=NOD(I,J+1,K)
      N=N+1
      L(N,1)=NOD(I+1,J,K+1)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I,J,K+1)
      L(N,4)=NOD(I+1,J,K)
      N=N+1
      L(N,1)=NOD(I+1,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J+1,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J,K+1)
      L(N,4)=NOD(I+1,J+1,K+1)
      GO TO 5
    3 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J,K+1)
      L(N,2)=NOD(I+1,J,K+1)
      L(N,3)=NOD(I,J+1,K+1)
      L(N,4)=NOD(I,J,K)
      N=N+1
      L(N,1)=NOD(I+1,J,K)
      L(N,2)=NOD(I,J,K)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I+1,J+1,K+1)
      L(N,2)=NOD(I,J+1,K+1)
      L(N,3)=NOD(I+1,J,K+1)
      L(N,4)=NOD(I+1,J+1,K)
```

```
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I,J+1,K+1)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
    5 CONTINUE
      IF (NPM(I,J)-NPM(I+1,J)) 6,2,2
    6 K=NPM(I,J)
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 7,7,8
    8 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      GO TO 2
    7 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
    2 CONTINUE
      RETURN
      END
```

(15) SUBROUTINE CORDLM

```
      SUBROUTINE CORDLM(G,NOD,K1,K2,K3,NP)
      DIMENSION G(NP,3),NOD(K1,K2,K3)
      DO 1 K=1,K3
      READ 2, HLEVEL,RAD,CENTER,THICK,ALPHA,NDIV
    2 FORMAT(5F8.3,1I4)
      T=THICK/(K2-1)
      FA=ALPHA*3.14159/180.0
      FD=FA/NDIV
      KA=K1-NDIV
```

```
      NS=0
      DO 1 J=1,K2
      R=RAD-NS*T
      NS=NS+1
      F=FA
      DO 1 I=KA,K1
      N=NOD(I,J,K)
      G(N,1)=R*SINF(F)
      G(N,2)=R*COSF(F)+CENTER
      G(N,3)=HLEVEL
      F=F-FD
    1 CONTINUE
      RETURN
      END
```

(16) SUBROUTINE DIMNP

```
      SUBROUTINE DIMNP(NSUB,NPN,NMAX,NPP)
      DIMENSION NSUB(NMAX,2),NPN(NPP)
      NPPI=NPP-1
      DO 1 I=1,NPPI
      NSUB(3*I-1,1)=3*NPN(I)
      NSUB(3*I-1,2)=3*NPN(I+1)
      NSUB(3*I,1)=3*NPN(I+1)
      NSUB(3*I,2)=3*NPN(I)
    1 CONTINUE
      DO 2 I=1,NPP
      NSUB(3*I-2,1)=3*NPN(I)
      NSUB(3*I-2,2)=3*NPN(I)
    2 CONTINUE
      RETURN
      END
```

(17) SUBROUTINE CLEAR

```
      SUBROUTINE CLEAR(DM,NNN,MMAX)
      DIMENSION DM(NNN,NNN,MMAX)
      DO 1 K=1,MMAX
      DO 1 I=1,NNN
      DO 1 J=1,NNN
    1 DM(I,J,K)=0.0
      RETURN
      END
```

(18) SUBROUTINE STIFF

```
      SUBROUTINE STIFF (NGE,G,S,A,BTDB,ET,XU,VOL,NE,NP,N)
      DIMENSION NGE(NE,4),G(NP,3),S(12,12),A(12,12),BTDB(12,12),
     1 ET(NE),XU(NE),VOL(NE)
   12 NI=NGE(N,1)
      NJ=NGE(N,2)
      NK=NGE(N,3)
      NL=NGE(N,4)
      AJ=G(NJ,1)-G(NI,1)
      BJ=G(NJ,2)-G(NI,2)
      CJ=G(NJ,3)-G(NI,3)
      AK=G(NK,1)-G(NI,1)
      BK=G(NK,2)-G(NI,2)
      CK=G(NK,3)-G(NI,3)
      AL=G(NL,1)-G(NI,1)
      BL=G(NL,2)-G(NI,2)
      CL=G(NL,3)-G(NI,3)
      VOL(N)=(AJ*BK*CL-AJ*BL*CK+AK*BL*CJ-AK*BJ*CL+AL*BJ*CK-AL*BK*CJ)/6.0
      IF (VOL(N)) 13,14,15
   13 PRINT 711,N,NJ,NK
      NK=NGE(N,2)
      NJ=NGE(N,3)
      NGE(N,2)=NJ
      NGE(N,3)=NK
      GO TO 12
   14 PRINT 713,N
      GO TO EXIT
   15 PRINT  6,    N,NI,NJ,NK,NL,VOL(N)
      COMM=ET(N)/((1.+XU(N))*(1.-2.*XU(N))*36.*VOL(N))
      D1=(1.-XU(N))*COMM
      D2=XU(N)*COMM
      D3=(0.5-XU(N))*COMM
      DO 2001 J=1,12
      DO 2001 I=1,12
 2001 BTDB(I,J)=0.0
      BTDB(2,2)=D1
      BTDB(7,7)=D1
      BTDB(12,12)=D1
      BTDB(2,7)=D2
      BTDB(2,12)=D2
      BTDB(7,2)=D2
      BTDB(7,12)=D2
      BTDB(12,2)=D2
      BTDB(12,7)=D2
      BTDB(3,3)=D3
      BTDB(3,6)=D3
      BTDB(4,4)=D3
      BTDB(4,10)=D3
      BTDB(6,3)=D3
      BTDB(6,6)=D3
      BTDB(8,8)=D3
      BTDB(8,11)=D3
      BTDB(10,4)=D3
      BTDB(10,10)=D3
      BTDB(11,8)=D3
      BTDB(11,11)=D3
```

```
      A21=BJ*CL-BJ*CK+BK*CJ-BK*CL+BL*CK-BL*CJ
      A22=BK*CL-BL*CK
      A23=BL*CJ-BJ*CL
      A24=BJ*CK-BK*CJ
      A31=AJ*CK-AJ*CL+AK*CL-AK*CJ+AL*CJ-AL*CK
      A32=AL*CK-AK*CL
      A33=AJ*CL-AL*CJ
      A34=AK*CJ-AJ*CK
      A41=AJ*BL-AJ*BK+AK*BJ-AK*BL+AL*BK-AL*BJ
      A42=AK*BL-AL*BK
      A43=AL*BJ-AJ*BL
      A44=AJ*BK-AK*BJ
      DO 2002 J=1,12
      DO 2002 I=1,12
 2002 A(I,J)=0.
      A(1,1)=1.0
      A(5,2)=1.0
      A(9,3)=1.0
      A(2,1)=A21
      A(6,2)=A21
      A(10,3)=A21
      A(3,1)=A31
      A(7,2)=A31
      A(11,3)=A31
      A(4,1)=A41
      A(8,2)=A41
      A(12,3)=A41
      A(2,4)=A22
      A(6,5)=A22
      A(10,6)=A22
      A(3,4)=A32
      A(7,5)=A32
      A(11,6)=A32
      A(4,4)=A42
      A(8,5)=A42
      A(12,6)=A42
      A(2,7)=A23
      A(6,8)=A23
      A(10,9)=A23
      A(3,7)=A33
      A(7,8)=A33
      A(11,9)=A33
      A(4,7)=A43
      A(8,8)=A43
      A(12,9)=A43
      A(2,10)=A24
      A(6,11)=A24
      A(10,12)=A24
      A(3,10)=A34
      A(7,11)=A34
      A(11,12)=A34
      A(4,10)=A44
      A(8,11)=A44
      A(12,12)=A44
      DO 181 J=1,12
      DO 181 I=1,12
      S(I,J)=0.
      DO 181 K=1,12
 181  S(I,J)=S(I,J)+BTDB(I,K)*A(K,J)
      DO 182 J=1,12
      DO 182 I=1,12
 182  BTDB(J,I)=A(I,J)
```

```
      DO 183 J=1,12
      DO 183 I=1,12
183   A(I,J)=S(I,J)
      DO 184 J=1,12
      DO 184 I=1,12
      S(I,J)=0.
      DO 184 K=1,12
184   S(I,J)=S(I,J)+BTDB(I,K)*A(K,J)
    6 FORMAT (56X,5I4,1F25.6)
711   FORMAT(12H ELEMENT NO.1I4,31H INTERCHANGED NODAL POINT NO.S.2I4)
 713  FORMAT(21H0ZERO VOLUME, EL.NO.=1I4)
      RETURN
      END
```

(19) SUBROUTINE CONMAS

```
      SUBROUTINE CONMAS (C,VOL,RO,NE,N)
      DIMENSION C(12,12),VOL(NE),RO(NE)
      TMASS=VOL(N)*RO(N)
      C1=0.1*TMASS
      C2=0.05*TMASS
      DO 11 J=1,12
      DO 11 I=1,12
  11  C(I,J)=0.0
      DO 12 I=1,12
  12  C(I,I)=C1
      C(1,4 )=C2
      C(1,7 )=C2
      C(1,10)=C2
      C(2,5 )=C2
      C(2,8 )=C2
      C(2,11)=C2
      C(3,6 )=C2
      C(3,9 )=C2
      C(3,12)=C2
      C(4,1 )=C2
      C(4,7 )=C2
      C(4,10)=C2
      C(5,2 )=C2
      C(5,8 )=C2
      C(5,11)=C2
      C(6,3 )=C2
      C(6,9 )=C2
      C(6,12)=C2
      C(7,1 )=C2
      C(7,4 )=C2
      C(7,10)=C2
      C(8,2 )=C2
      C(8,5 )=C2
      C(8,11)=C2
      C(9,3 )=C2
      C(9,6 )=C2
      C(9,12)=C2
      C(10,1)=C2
      C(10,4)=C2
      C(10,7)=C2
      C(11,2)=C2
      C(11,5)=C2
```

```
      C(11,8)=C2
      C(12,3)=C2
      C(12,6)=C2
      C(12,9)=C2
      RETURN
      END
```

(20) SUBROUTINE HYDNMC

```
      SUBROUTINE HYDNMC(A,NSUB,G,NPN,NOD,NNN,MMAX,NP,NPP,K1,K2,K3,NPH)
      DIMENSION A(NNN,NNN,MMAX),NSUB(MMAX,2),G(NP,3),NPN(NPP),
     1NOD(K1,K2,K3),HYDNC(60)
C     EVALUATION OF HYDRODYNAMIC LOADS
      NW=0
      READ 21,(HYDNC(M),M=1,NPH)
   21 FORMAT (16F5.3)
      PRINT 111
  111 FORMAT (1H1)
      DO 1 K=1,K3
      READ 2, ALPHA,ALOAD,NDIV
    2 FORMAT (1F8.3,1F12.3,1I4)
      FA=ALPHA*3.14159/180.0
      FD=FA/NDIV
      KA=K1-NDIV
      F=FA
      DO 1 I=KA,K1
      N=NOD(I,1,K)
      SN=SINF(F)
      CS=COSF(F)
      NW=NW+1
      HLOAD=ALOAD*HYDNC(NW)
      IF (I-K1) 6,7,7
    7 XLOAD=0.0
      YLOAD=HLOAD/2.0
      IF (K-1) 8,8,9
    8 ZLOAD=0.0
      GO TO 5
    9 ZLOAD=ABSF(HLOAD/2.0*(G(N,2)-G(N-1,2))/(G(N-1,3)-G(N,3)))
      GO TO 5
    6 XLOAD=ABSF(HLOAD*SN)
      YLOAD=ABSF(HLOAD*CS)
      IF (K-1) 10,10,11
   10 ZLOAD=0.0
      GO TO 5
   11 DX=G(N,1)-G(N-1,1)
      DY=G(N,2)-G(N-1,2)
      DZ=G(N,3)-G(N-1,3)
      DXY=DX*DX+DY*DY
      ZLOAD=ABSF(HLOAD*SQRTF(DXY)/DZ)
    5 F=F-FD
      PRINT112,ALPHA,ALOAD,NDIV,N,NW.HYDNC(NW),F,SN,CS,XLOAD,YLOAD,ZLOAD
  112 FORMAT (1F8.3,1F12.3,3I5,4F8.5,3F12.3)
C     TRANSFER TO THE OVERALL MASS MATRIX
      IM=0
      NPS=0
    3 IM=IM+1
      NPS =NPS+NPN(IM)
```

```
      NI=NPS-N
      IF (NI) 3,4,4
    4 NRL=3*IM-2
      IR=NSUB(NPL,1)-3*(NI+1)+1
      A(IR,IR,NPL)=A(IR,IR,NPL)+XLOAD
      A(IR+1,IR+1,NPL)=A(IR+1,IR+1,NPL)+YLOAD
      A(IR+2,IR+2,NPL)=A(IR+2,IR+2,NPL)+ZLOAD
    1 CONTINUE
      RETURN
      END
```

(25) SUBROUTINE WTPME

```
      SUBROUTINE WTPME(A,NSUB,NNN,KMAX,NMAX)
      DIMENSION A(NNN,NNN,KMAX),NSUB(KMAX,2)
      DO 1 MA=1,NMAX
      NR=NSUB(MA,1)
      NC=NSUB(MA,2)
      IF (NR*NC) 1,1,2
    2 WRITE TAPE 23,((A(I,J,MA),I=1,NR),J=1,NC)
    1 CONTINUE
      RETURN
      END
```

(30) SUBROUTINE UNAPGE

```
      SUBROUTINE UNAPGE(MCN,VECT,NC,NCA,NNN,NPP)
      DIMENSION MCN(NCA,2),VECT(NNN,NPP)
      DO 1 N=1,NC
      NP=(MCN(N,2)+2)/3
      NR=MCN(N,1)
      NTR=NNN-NR
      DO 2 KR=1,NTR
      MT=NNN-KR
    2 VECT(MT+1,NP)=VECT(MT,NP)
    1 VECT(NR,NP)=0.
      RETURN
      END
```

(31) SUBROUTINE VCTWRT

```
      SUBROUTINE VCTWRT(VECT,NPN,NNN,NPP,NDIM)
      DIMENSION VECT(NNN,NPP),NPN(NPP)
      DO 1 NP=1,NPP
      NPS=0
      DO 2 JP=1,NP
    2 NPS=NPS+NPN(JP)
      NM=NPN(NP)
      NPM=NPS-NM
      DO 1 I=1,NM
      MP=NPM+I
      NI=NDIM*(I-1)
      PRINT  3, MP,(VECT(NI+J,NP),J=1,NDIM)
    1 CONTINUE
    3 FORMAT(1I8,3F15.10)
      RETURN
      END
```

(32) SUBROUTINE ELEREM

```
      SUBROUTINE ELEREM(NE,NP,NGE,G,RO,NOD,NPM,K1,K2,K3,K4,K5)
      DIMENSION NGE(NE,4),G(NP,3),RO(NE),NOD(K1,K2,K3),NPM(K1,K2)
      READ 41, UNIDOM
   41 FORMAT(1F15.5)
      READ 7, ((NPM(I,J),J=1,K2),I=1,K1)
    7 FORMAT(20I4)
      CALL NOKSYM(NGE,NPM,NOD,K1,K2,K3,NE)
      IF (K4) 51,51,52
   52 READ 70, N
   70 FORMAT(1I4)
      DO 75 I=1,K4
      READ 9, (NGE(N,L),L=1,4)
    9 FORMAT(4X,4I4)
   75 N=N+1
   51 DO 76 N=1,NE
   76 RO(N)=UNIDOM
      CALL        CORSYM(G,NOD,K1,K2,K3,NP)
      IF (K5) 53,53,54
   54 READ 70, M
      DO 55 I=1,K5
      READ 56, (G(M,L),L=1,3)
   56 FORMAT(4X,3F8.3)
   55 M=M+1
   53 RETURN
      END
```

(33) SUBROUTINE NOKSYM

```
      SUBROUTINE NOKSYM(L,NPM,NOD,K1,K2,K3,NE)
      DIMENSION L(NE,4),NPM(K1,K2),NOD(K1,K2,K3)
      M=0
      DO 1 I=1,K1
      DO 1 J=1,K2
      KN=NPM(I,J)
      DO 1 K=1,KN
      M=M+1
      NOD(I,J,K)=M
    1 CONTINUE
C
      N=0
      K1A=K1-1
      K2A=K2-1
      DO 2 I=1,K1A
      DO 2 J=1,K2A
      IF (NPM(I,J)-NPM(I+1,J)) 9,9,10
    9 KNA=NPM(I,J)-1
      GO TO 11
   10 KNA=NPM(I,J)-2
   11 DO 5 K=1,KNA
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 3,3,4
    3 N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K+1)
      L(N,2)=NOD(I,J,K+1)
      L(N,3)=NOD(I+1,J+1,K+1)
      L(N,4)=NOD(I,J+1,K)
      N=N+1
      L(N,1)=NOD(I+1,J,K+1)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I,J,K+1)
      L(N,4)=NOD(I+1,J,K)
      N=N+1
      L(N,1)=NOD(I+1,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J+1,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J,K+1)
      L(N,4)=NOD(I+1,J+1,K+1)
      GO TO 5
    4 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I,J+1,K+1)
```

```
      N=N+1
      L(N,1)=NOD(I,J,K+1)
      L(N,2)=NOD(I+1,J,K+1)
      L(N,3)=NOD(I,J+1,K+1)
      L(N,4)=NOD(I,J,K)
      N=N+1
      L(N,1)=NOD(I+1,J,K)
      L(N,2)=NOD(I,J,K)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I+1,J+1,K+1)
      L(N,2)=NOD(I,J+1,K+1)
      L(N,3)=NOD(I+1,J,K+1)
      L(N,4)=NOD(I+1,J+1,K)
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I,J+1,K+1)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
    5 CONTINUE
      IF (NPM(I,J)-NPM(I+1,J)) 6,2,12
    6 K=NPM(I,J)
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 7,7,8
    7 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      GO TO 2
    8 N=N+1
      L(N,1)=NOD(I,J+1,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I,J,K)
      L(N,4)=NOD(I+1,J+1,K+1)
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I+1,J+1,K)
      L(N,3)=NOD(I+1,J,K)
      L(N,4)=NOD(I+1,J,K+1)
      N=N+1
      L(N,1)=NOD(I,J,K)
      L(N,2)=NOD(I+1,J+1,K+1)
      L(N,3)=NOD(I+1,J+1,K)
      L(N,4)=NOD(I+1,J,K+1)
      GO TO 2
   12 K=NPM(I,J)-1
      NEXP=I+J+K
      NTYPE=(-1)**NEXP
      IF (NTYPE) 13,13,14
```

```
13 N=N+1
   L(N,1)=NOD(I,J,K)
   L(N,2)=NOD(I,J+1,K)
   L(N,3)=NOD(I+1,J,K)
   L(N,4)=NOD(I,J,K+1)
   N=N+1
   L(N,1)=NOD(I,J+1,K)
   L(N,2)=NOD(I,J+1,K+1)
   L(N,3)=NOD(I,J,K+1)
   L(N,4)=NOD(I+1,J,K)
   N=N+1
   L(N,1)=NOD(I,J+1,K)
   L(N,2)=NOD(I,J+1,K+1)
   L(N,3)=NOD(I+1,J,K)
   L(N,4)=NOD(I+1,J+1,K)
   GO TO 2
14 N=N+1
   L(N,1)=NOD(I,J+1,K)
   L(N,2)=NOD(I,J,K)
   L(N,3)=NOD(I+1,J+1,K)
   L(N,4)=NOD(I,J+1,K+1)
   N=N+1
   L(N,1)=NOD(I,J,K)
   L(N,2)=NOD(I,J,K+1)
   L(N,3)=NOD(I,J+1,K+1)
   L(N,4)=NOD(I+1,J+1,K)
   N=N+1
   L(N,1)=NOD(I,J,K)
   L(N,2)=NOD(I+1,J,K)
   L(N,3)=NOD(I+1,J+1,K)
   L(N,4)=NOD(I,J,K+1)
 2 CONTINUE
   RETURN
   END
```

(34) SUBROUTINE CORSYM

```
   SUBROUTINE CORSYM(G,NOD,K1,K2,K3,NP)
   DIMENSION G(NP,3),NOD(K1,K2,K3)
   DO 1 K=1,K3
   READ 2, HLEVEL,RAD,CENTER,THICK,ALPHA,NDIV
 2 FORMAT(5F8.3,1I4)
   T=THICK/(K2-1)
   FA=ALPHA*3.14159/180.0
   FD=FA/NDIV
   KA=(K1+1)/2-NDIV
   KB=K1+1-KA
   NS=0
   DO 1 J=1,K2
   R=RAD-NS*T
   NS=NS+1
   F=FA
   DO 1 I=KA,KB
   N=NOD(I,J,K)
   G(N,1)=R*SINF(F)
   G(N,2)=R*COSF(F)+CENTER
   G(N,3)=HLEVEL
   F=F-FD
 1 CONTINUE
   RETURN
   END
```

(35) SUBROUTINE RESMAS

```
      SUBROUTINE RESMAS(NGE,G,DM,VOL,RO,NE,NP)
      DIMENSION NGE(NE,4),G(NP,3),DM(NP,NP),VOL(NE),RO(NE)
      DO 31 N=1,NE
   12 NI=NGE(N,1)
      NJ=NGE(N,2)
      NK=NGE(N,3)
      NL=NGE(N,4)
      AJ=G(NJ,1)-G(NI,1)
      BJ=G(NJ,2)-G(NI,2)
      CJ=G(NJ,3)-G(NI,3)
      AK=G(NK,1)-G(NI,1)
      BK=G(NK,2)-G(NI,2)
      CK=G(NK,3)-G(NI,3)
      AL=G(NL,1)-G(NI,1)
      BL=G(NL,2)-G(NI,2)
      CL=G(NL,3)-G(NI,3)
      VOL(N)=(AJ*BK*CL-AJ*BL*CK+AK*BL*CJ-AK*BJ*CL+AL*BJ*CK-AL*BK*CJ)/6.0
      IF (VOL(N)) 13,14,15
   13 PRINT 711,N,NJ,NK
  711 FORMAT(12H ELEMENT NO.1I4,31H INTERCHANGED NODAL POINT NO.S.2I4)
      NK=NGE(N,2)
      NJ=NGE(N,3)
      NGE(N,2)=NJ
      NGE(N,3)=NK
      GO TO 12
   14 PRINT 713,N
  713 FORMAT(21H0ZERO VOLUME. EL.NO.=1I4)
      GO TO EXIT
   15 PRINT  6,    N,NI,NJ,NK,NL,VOL(N)
    6 FORMAT (56X,5I4,1F25.6)
      TMASS=VOL(N)*RO(N)
      C1=0.10*TMASS
      C2=0.05*TMASS
      DM(NI,NI)=DM(NI,NI)+C1
      DM(NI,NJ)=DM(NI,NJ)+C2
      DM(NI,NK)=DM(NI,NK)+C2
      DM(NI,NL)=DM(NI,NL)+C2
      DM(NJ,NI)=DM(NJ,NI)+C2
      DM(NJ,NJ)=DM(NJ,NJ)+C1
      DM(NJ,NK)=DM(NJ,NK)+C2
      DM(NJ,NL)=DM(NJ,NL)+C2
      DM(NK,NI)=DM(NK,NI)+C2
      DM(NK,NJ)=DM(NK,NJ)+C2
      DM(NK,NK)=DM(NK,NK)+C1
      DM(NK,NL)=DM(NK,NL)+C2
      DM(NL,NI)=DM(NL,NI)+C2
      DM(NL,NJ)=DM(NL,NJ)+C2
      DM(NL,NK)=DM(NL,NK)+C2
      DM(NL,NL)=DM(NL,NL)+C1
   31 CONTINUE
      RETURN
      END
```

(36) SUBROUTINE HYDREM

```
      SUBROUTINE HYDREM(G,NOD,NP,K1,K2,K3,NPH,HX,HY,HZ)
      DIMENSION G(NP,3),NOD(K1,K2,K3),HX(NP),HY(NP),HZ(NP),HYDNC(82)
C     EVALUATION OF HYDRODYNAMIC LOADS
      NW=0
      READ 21,(HYDNC(M),M=1,NPH)
   21 FORMAT (16F5.3)
      PRINT 111
  111 FORMAT (1H1)
      DO 1 K=1,K3
      READ 2, ALPHA,ALOAD,NDIV
    2 FORMAT (1F8.3,1F12.3,1I4)
      FA=ALPHA*3.14159/180.0
      FD=FA/NDIV
      KA=(K1+1)/2-NDIV
      KB=K1+1-KA
      F=FA
      DO 1 I=KA,KB
      N=NOD(I,1,K)
      SN=SINF(F)
      CS=COSF(F)
      NW=NW+1
      HLOAD=ALOAD*HYDNC(NW)
    6 XLOAD=ABSF(HLOAD*SN)
      YLOAD=ABSF(HLOAD*CS)
      IF (K-1) 10,10,11
   10 ZLOAD=0.0
      GO TO 5
   11 DX=G(N,1)-G(N-1,1)
      DY=G(N,2)-G(N-1,2)
      DZ=G(N,3)-G(N-1,3)
      DXY=DX*DX+DY*DY
      ZLOAD=ABSF(HLOAD*SQRTF(DXY)/DZ)
    5 F=F-FD
      PRINT112,ALPHA,ALOAD,NDIV,N,NW,HYDNC(NW),F,SN,CS,XLOAD,YLOAD,ZLOAD
  112 FORMAT (1F8.3,1F12.3,3I5,4F8.5,3F12.3)
      HX(N)=XLOAD
      HY(N)=YLOAD
      HZ(N)=ZLOAD
    1 CONTINUE
      RETURN
      END
```

(37) SUBROUTINE VCTRD

```
      SUBROUTINE VCTRD (FX,FY,FZ,ND,MR,NPN,NPP,NDIM,M)
      DIMENSION FX(ND,MR),FY(ND,MR),FZ(ND,MR),NPN(NPP).DUM(3)
      DO 1 NP=1,NPP
      NPS=0
      DO 2 JP=1,NP
    2 NPS=NPS+NPN(JP)
      NM=NPN(NP)
      NPM=NPS-NM
      DO 1 I=1,NM
      MP=NPM+I
      READ TAPE 23, (DUM(J),J=1,NDIM)
      FX(MP,M)=DUM(1)
      FY(MP,M)=DUM(2)
      FZ(MP,M)=DUM(3)
    1 CONTINUE
      RETURN
      END
```

(38) SUBROUTINE GENMAS

```
      SUBROUTINE GENMAS(BX,BY,BZ,FX,FY,FZ,DM,HX,HY,HZ,GMASS,MR,NP,NC1)
      DIMENSION BX(MR,NP),BY(MR,NP),BZ(MR,NP),FX(NP,MR),FY(NP,MR),
     1 FZ(NP,MR),DM(NP,NP),HX(NP),HY(NP),HZ(NP),GMASS(MR)
      DO 11 M=1,MR
      DO 12 J=1,NP
      BX(M,J)=0.0
      BY(M,J)=0.0
      BZ(M,J)=0.0
      DO 12 I=1,NP
      BX(M,J)=BX(M,J)+FX(I,M)*DM(I,J)
      BY(M,J)=BY(M,J)+FY(I,M)*DM(I,J)
      BZ(M,J)=BZ(M,J)+FZ(I,M)*DM(I,J)
   12 CONTINUE
      IF (NC1) 11,11,13
   13 DO 14 J=1,NP
      BX(M,J)=BX(M,J)+HX(J)*FX(J,M)
      BY(M,J)=BY(M,J)+HY(J)*FY(J,M)
      BZ(M,J)=BZ(M,J)+HZ(J)*FZ(J,M)
   14 CONTINUE
   11 CONTINUE
      DO 21 M=1,MR
      GMASS(M)=0.0
      DO 21 J=1,NP
      GMASS(M)=GMASS(M)+BX(M,J)*FX(J,M)+BY(M,J)*FY(J,M)+BZ(M,J)*FZ(J,M)
   21 CONTINUE
      RETURN
      END
```

(39) SUBROUTINE RECEPT

```
      SUBROUTINE RECEPT (FREQ,C,SFR,DELTA,MR,NS)
      DIMENSION FREQ(MR),C(MR),SFR(NS,MR)
      DO 1 M=1,MR
      FR=FREQ(M)*FREQ(M)
      CF=4.0*C(M)*C(M)*FR
      F=0.0
      DO 2 N=1,NS
      FF=F*F
      R=FR-FF
      SFR(N,M)=1.0/(R*R+CF*FF)
      F=F+DELTA
    2 CONTINUE
    1 CONTINUE
      PRINT 3
    3 FORMAT (23H1UNIT MODAL RECEPTANCES//)
      PRINT 4, (N,(SFR(N,M),M=1,MR),N=1,NS)
    4 FORMAT (1I4,5F20.10)
      RETURN
      END
```

(40) SUBROUTINE MDCYRC

```
      SUBROUTINE MDCYRC(GMASS,SPQ,SFR,APS,G,QA,CPS,VEL,NC4,MR,NP,NS,
     1 DELTA)
      DIMENSION GMASS(MR),SPQ(NS,MR),SFR(NS,MR),APS(NS),G(NP,3),
     1 QA(MR,NP)
      PI=3.14159
      CA=16.0*PI*PI*PI*PI
      CP=CPS/CA
      DO 1 M=1,MR
      CB=CP/(GMASS(M)*GMASS(M))
      F=0.0
      DO 1 N=1,NS
      CALL        DCYFRC(F,VEL,G,QA,P,NC4,M,MR,NP)
      SPQ(N,M)=CB*P*SFR(N,M)*APS(N)
      F=F+DELTA
    1 CONTINUE
      PRINT 2
    2 FORMAT (22H1MODAL POWER SPECTRUMS//)
      PRINT 5, NC4,CPS,VEL
    5 FORMAT (17H0FORCE DIRECTION=1I2,22H   INTENSITY CONSTANT=1F5.3,
     1 12H   VELOCITY=1F7.1//)
      PRINT 3, (GMASS(M),M=1,MR)
    3 FORMAT (5F20.10//)
      PRINT 4, (N,(SPQ(N,M),M=1,MR),N=1,NS)
    4 FORMAT (1I4,5F20.10)
      RETURN
      END
```

(41) SUBROUTINE DCYFRC

```
      SUBROUTINE DCYFRC(F,VEL,G,QA,P,NC4,M,MR,NP)
      DIMENSION G(NP,3),QA(MR,NP)
      AF=6.28318*F/VEL
      PS=0.0
      PC=0.0
      DO 1 J=1,NP
      FI=AF*G(J,NC4)
      PS=PS+QA(M,J)*SINF(FI)
      PC=PC+QA(M,J)*COSF(FI)
    1 CONTINUE
      P=PS*PS+PC*PC
      RETURN
      END
```

(42) SUBROUTINE MDFRCE

```
      SUBROUTINE MDFRCE (QA,P,MR,NP)
      DIMENSION QA(MR,NP),P(MR)
      DO 1 M=1,MR
      P(M)=0.0
      DO 1 J=1,NP
      P(M)=P(M)+QA(M,J)
    1 CONTINUE
      RETURN
      END
```

(43) SUBROUTINE MODRCP

```
      SUBROUTINE MODRCP (P,GMASS,SPQ,SFR,APS,CPS,MR,NS.NC4)
      DIMENSION P(MR),GMASS(MR),SPQ(NS,MR),SFR(NS,MR),APS(NS)
      PI=3.14159
      CA=16.0*PI*PI*PI*PI
      CP=CPS/CA
      DO 1 M=1,MR
      CB=P(M)*P(M)*CP/(GMASS(M)*GMASS(M))
      DO 1 N=1,NS
      SPQ(N,M)=CB*SFR(N,M)*APS(N)
    1 CONTINUE
      PRINT 2
    2 FORMAT (22H1MODAL POWER SPECTRUMS//)
      PRINT 5, NC4,CPS
    5 FORMAT (17H0FORCE DIRECTION=1I2,22H   INTENSITY CONSTANT=1F5.3//)
      PRINT 3, (GMASS(M),M=1,MR),(P(M),M=1,MR)
    3 FORMAT (5F20.10//)
      PRINT 4, (N,(SPQ(N,M),M=1,MR),N=1,NS)
    4 FORMAT (1I4,5F20.10)
      RETURN
      END
```

(44) SUBROUTINE SPECTR

```
      SUBROUTINE SPECTR (FA,SPQ,A,N,MR,NP,NS,NC6)
      DIMENSION FA(NP,MR),SPQ(NS,MR),A(NS),FF(10)
      DO 1 M=1,MR
      FF(M)=FA(N,M)*FA(N,M)
    1 CONTINUE
      DO 2 I=1,NS
      A(I)=0.0
      DO 2 M=1,MR
      A(I)=A(I)+SPQ(I,M)*FF(M)
    2 CONTINUE
      PRINT 3, N,NC6
    3 FORMAT(13H1NODAL POINT=1I4,11H DIRECTION=1I2//31H POWER SPECTRUM O
     1F DISPLACEMENT//)
      PRINT 4, (A(I),I=1,NS)
    4 FORMAT (10F12.6)
      RETURN
      END
```

(45) SUBROUTINE SIMPSN

```
      SUBROUTINE SIMPSN (AREA,DELTA,A,NS)
      DIMENSION A(NS)
      B=A(1)+A(NS)
      C=0.0
      N1=NS-1
      DO 1 I=2,N1,2
      C=C+A(I)
    1 CONTINUE
      D=0.0
      N2=NS-2
      DO 2 I=3,N2,2
      D=D+A(I)
    2 CONTINUE
      AREA=(B+4.0*C+2.0*D)*DELTA/3.0
      RETURN
      END
```

(46) SUBROUTINE ASUMP

```
      SUBROUTINE ASUMP(A,B,D,NS,DELTA,AREA,ARSQRT)
      DIMENSION A(NS),B(NS),D(NS)
      F=0.0
      DO 1 I=1,NS
      B(I)=A(I)*F*F
      F=F+DELTA
    1 CONTINUE
      CALL SIMPSN (ARSQ,DELTA,B,NS)
      ARSQRT=SQRTF(ARSQ)
      PRINT 3, ARSQ,ARSQRT
    3 FORMAT (6H0ARSQ=1F20.3,8H ARSQRT=1F20.3)
      F=0.0
      DO 2 I=1,NS
      D(I)=B(I)*F*F
      F=F+DELTA
    2 CONTINUE
      CALL SIMPSN (ARQT,DELTA,D,NS)
      PRINT 4, ARQT
    4 FORMAT (6H0ARQT=1F20.3)
      RRR=AREA*ARQT/(ARSQ*ARSQ)
      AKSQ=1.0/(RRR-1.0)
      AKSQRT=SQRTF(AKSQ)
      PRINT 5, AKSQ,AKSQRT
    5 FORMAT (6H0AKSQ=1F20.8,8H AKSQRT=1F20.8)
      RETURN
      END
```

(47) SUBROUTINE MAGFAC

```
      SUBROUTINE MAGFAC (ARSQRT,SIGMA)
      PRINT 1
    1 FORMAT(33H0DURATION   MAG.COEF.    MEAN MAX.)
      DT=10.0
      DO 2 I=1,6
      T=I*DT
      DF=LOGF(ARSQRT*T/SIGMA)
      CN=SQRTF(2.0*DF)
      XMAX=CN*SIGMA
      PRINT 3, T,CN,XMAX
    3 FORMAT (1F9.2,1F12.3,1F12.4)
    2 CONTINUE
      RETURN
      END
```