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 specification for 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
NPX (ND) Nodal point number of corner l of element
EY (YN) Modulus of elasticity of element (Kg/cm²)
UX (YN) Poisson’s ratio of element
RO (YN) Unit weight of element (Kg/cm3)
NPF (LD) Constrained nodal point numbers
NFI (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(ND, JD) Matrix containing the information about the adjacent nodal
point numbers (Figure B.3)
NAP (MD) Number of adjacent nodal points (Figure B.3)
G(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) Matrix contains the positions of nodal points in the idealization
K3) (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

<table>
<thead>
<tr>
<th>S/R No.</th>
<th>SUBROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ELEAR2</td>
</tr>
<tr>
<td>3</td>
<td>NOKTA</td>
</tr>
<tr>
<td>4</td>
<td>NOKTB</td>
</tr>
<tr>
<td>5</td>
<td>CORDLD</td>
</tr>
<tr>
<td>6</td>
<td>SINIR</td>
</tr>
<tr>
<td>7</td>
<td>MODSTF</td>
</tr>
<tr>
<td>8</td>
<td>INVNP2</td>
</tr>
<tr>
<td>9</td>
<td>BOUND2</td>
</tr>
<tr>
<td>10</td>
<td>ITERA2</td>
</tr>
<tr>
<td>11</td>
<td>REACT2</td>
</tr>
<tr>
<td></td>
<td>FORCE</td>
</tr>
</tbody>
</table>
B.2.5 Input data specifications

1. ATITLE (10A8)
2. NUMEL, NUMNP, NUMBC, NCPIN, NOPIN, NCYCM, TOLER, XFACE, NCONNT
   (6I4, 2F12.6, 1I1)
3. K1, K2, K3, K4, K5, K6, K7 (7I4)
4. EIMOD, 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. ELEVEL, RAD, CENTER, THICK, ALPHA, NDIV, HLOAD, VLOAD
   (4F8.1, 1F8.3, 11I4, 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 \((K - \omega^2M)\) 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
NCONN Number of corners of an element
KGG 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 (m^3)
MCN(NCA,2) Matrix of constraint numbers (as given in S/R APOCP, ISVR S/R No.5)
      (in the assembly stage the third dimension can be NMAX).
DM(NNN, NMAX) 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

S(12,12) Element stiffness or mass matrix
A(12,12) Matrices used internally for element stiffness matrix calculations.
B(12,12) Matrices used internally
NPA(NCORN) Used internally
MA(NCORN) 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

<table>
<thead>
<tr>
<th>S/R No.</th>
<th>SUBROUTINE</th>
<th>S/R No.</th>
<th>SUBROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>ELEARM</td>
<td>13</td>
<td>NOKTAM</td>
</tr>
<tr>
<td>14</td>
<td>NOKTSM</td>
<td>15</td>
<td>CORDLM</td>
</tr>
<tr>
<td>16</td>
<td>DIMNP</td>
<td>17</td>
<td>CLEAR</td>
</tr>
<tr>
<td>18</td>
<td>STIFF</td>
<td>19</td>
<td>CONMAS</td>
</tr>
<tr>
<td>20</td>
<td>ITCLEAR</td>
<td>21</td>
<td>TRSFPGP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(ISVR S/R No.35)</td>
</tr>
<tr>
<td>22</td>
<td>STIFF</td>
<td>23</td>
<td>RSHIFT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(&quot; &quot; No.5 )</td>
</tr>
<tr>
<td>24</td>
<td>CSHIFT</td>
<td>25</td>
<td>WTPME</td>
</tr>
</tbody>
</table>

B.3.3.2 Analysis stage

<table>
<thead>
<tr>
<th>S/R No.</th>
<th>SUBROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>CLEAR</td>
</tr>
<tr>
<td>26</td>
<td>PRVS2</td>
</tr>
<tr>
<td>27</td>
<td>MXFRTP</td>
</tr>
<tr>
<td>28</td>
<td>DETSCL</td>
</tr>
<tr>
<td>29</td>
<td>PLVECT</td>
</tr>
<tr>
<td>30</td>
<td>UNAPGE</td>
</tr>
<tr>
<td>31</td>
<td>VCIWHT</td>
</tr>
</tbody>
</table>

B.3.4 Input data specifications

B.3.4.1 Assembly stage

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

(2) (NPM(K), K=1,NPP) (2014)

(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) (2014) (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) (2014)
If reservoir is in full condition:

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

(9) \(ALPHA, ALOAD, NDIV (1P8.3, 1F12.3, 1I4) \) (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) \(\text{NPN}(K), K=1, NPP) (20I4)\)

(3) \((\text{MCN}(L,I), I=1,2), L=1,NC) (20I4)\)

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

B.4.1 The basic flow diagram of this 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 VCTR 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 a \(4 \times 4\) 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

\[
\begin{align*}
\text{NP,NE} & \\
\text{G(NP,3), NGE(NE,4)} & \\
\text{RO(NE), VOL(NE)} & \\
\text{K1,K2,K3,K4,K5} & \\
\text{NPM(K1,K2)} & \\
\text{NOD(K1,K2,K3)} & \\
\text{NPP, NDIM} & \\
\text{NPW (NPP)} & 
\end{align*}
\]

As given in Section B.3.2

Special scalars and arrays used for arranging nodal point and element arrays as given in Section B.2.3.

As given in Section B.3.2 and are only used if the eigenvectors are read from a magnetic tape (S/R VCTR).
ME
NS
DELTA
CPS
VEL
NC1
NC2
NC3
NC4
NC5
NC6
AREA
SIGMA
ARSQRT
DM(NP, NP)
FX(NP, MR)
FY(NP, MR)
FZ(NP, MR)
HX(NP)
HY(NP)
HZ(NP)
EX(MR, NP)
BY(MR, NP)
BZ(MR, NP)
FREQ(MR)
C(MR)
CM(NR, MR)
P(MR)
SPR(NS, MR)
SPQ(NS, MR)
APS(NS)
A(NS)
B(NS), D(NS)

Number of modes considered in the analysis
Number of power spectral density ordinates (must be an odd number)
Frequency increment considered in the power spectral density.
Power spectral intensity of ground acceleration (cm²/sec²) as given by equation (5.1).
Velocity of earthquake waves (m/sec)
=0 for reservoir in empty condition
=1 for reservoir in full condition.
Number of loading conditions to be analyzed.
=0 for velocity of earthquake waves is infinite
=1 for velocity of earthquake waves is finite
=1 for forces in X-direction
=2 for forces in Y-direction
=3 for forces in Z-direction
Number of nodal points where the response is to be analyzed.
=1 for displacement in X-direction
=2 for displacement in Y-direction
=3 for displacement in Z-direction
Variance of displacement
The RMS value of displacement

\[ \text{ARSQRT} = \left( \int_0^\infty \sigma_x^2(f) \, df \right)^{\frac{1}{2}} \text{ as given by equation (4.80)} \]

Overall inertia matrix as described in Section B.4.1.
Modal matrix contains only X-direction displacements
Modal matrix contains only Y-direction displacements
Modal matrix contains only Z-direction displacements
Hydrodynamic forces in X-direction
Hydrodynamic forces in Y-direction
Hydrodynamic forces in Z-direction
Given by the matrix product \([FX]^T [DM] \)
Given by the matrix product \([FY]^T [DM] \)
Given by the matrix product \([FZ]^T [DM] \)
Natural frequency (Hz)
Damping ratio
Generalized mass, as given by equation (4.27)
Modal force, as given by equation (4.96a)
Unit modal receptances, as given by equation (4.95)
Modal response spectral densities, as given by equation (4.97a or b).
Normalized power spectral density of ground acceleration.
Power spectral density of displacement as given by equation (4.98).
Used in S/R ASUMP internally.

-8-
**B.4.3 Subroutines used in the programme**

<table>
<thead>
<tr>
<th>S/R No.</th>
<th>Subroutine</th>
<th>Subroutine</th>
<th>EBEREM</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>SUBROUTINE</td>
<td>ELEREM</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>NOKSYM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>CORSYM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>RESMAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>HYDREM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>VCTRD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>GENMAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>RECEPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>MDCYRC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>DCYFRC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>MDFRCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>MODCPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>SPECTR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>SIMPSN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>ASUMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>MAGFAC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 reservoir is in full 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) (10F8.3)

Repeat the following NC2 times:

13. NC3, NC4, NC5, CPS, VEL (3I4, 1F8.3, 1F8.1)
Repeat the following NC5 times

(14) NC6, N (214)

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 ELEAR

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 calls Subroutines NOKTAM and CORDIM.

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 NOETAM

1. Purpose and method

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

2. Argument list

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

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, ALPEA, 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(WPP), 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 COMMAS

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(ENN,NNN,NMAX) 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 APOENP

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 APOENP, ISVR subroutine library S/R No.5.

24. Subroutine CSHIFT

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

25. Subroutine WPME

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(NMN,NMN,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 MXPFTP, DETSCL and PLVECT.

27. Subroutine MXPFTP

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 VCTWRIT

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 ELEERM

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 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 NFM 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.3), but is modified in order to idealize the whole structure (steps 3 and 5 of Figure 4.6).
34. Subroutine C0R8YM

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 EYDREM

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 EX, EY and EZ 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 routine calls subroutine DCYFRC.

4. Argument list

QA(MR,NP) takes the value of one of the three arrays EX, 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

P 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 SIMFSN

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, $K_1 = 5$

through thickness, $K_2 = 2$

horizontal from top to bottom, $K_3 = 5$

Matrix for the total number of nodal points contained in each column, dimensioned $NPM(K1,K2)\begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \\ 5 & 5 \end{bmatrix}$

Fig.B.1. Necessary information for numbering the nodal points and arranging the element arrays (For $4\times1\times4$ mesh type)
Fig. B.2. Double curvature dam with circular arches, the scalars used in computations.
\[ k_{mn}^{(q)} = \begin{bmatrix} k_{xx}^{(q)} & k_{xy}^{(q)} & k_{xz}^{(q)} \\ k_{yx}^{(q)} & k_{yy}^{(q)} & k_{yz}^{(q)} \\ k_{zx}^{(q)} & k_{zy}^{(q)} & k_{zz}^{(q)} \end{bmatrix} \]

\[ K_{mn} = \sum_{q} k_{mn}^{(q)} \]

<table>
<thead>
<tr>
<th>NAP(MD)</th>
<th>NP(MD, JD) where JD = 1D + 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1 2 3 4 5 6 8 9</td>
</tr>
<tr>
<td>10</td>
<td>2 1 3 4 5 6 7 8 9 10</td>
</tr>
<tr>
<td>8</td>
<td>3 1 2 4 5 6 8 9</td>
</tr>
<tr>
<td>10</td>
<td>4 1 2 3 5 6 7 8 9 10</td>
</tr>
</tbody>
</table>

One of the nine matrices of the three dimensional overall stiffness array \( R(9, 1D, MD) \)

Fig.B.3. Main arrays required for static analysis by means of iterative method.
FIG. B.4. Order of numbering the modal 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
(A) MAIN ROUTINE FOR STATIC ANALYSIS OF
ARCH DARS

H.E. EMITTU, SOUTHAMPTON UNIVERSITY
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 NPK(422), L, NAP(422), ATITLE(10)
DIMENSION NPI(926), NPJ(926), NPK(926), NPL(926), ET(926),
1 X(1926), Y(1926)
DIMENSION NPJ(74), NFIX(74), LM(4), S(12, 12), A(12, 12), BTDB(12, 12)
DIMENSION NOD(17, 2, 17), NPM(17, 2)
COND N(11, 18, 422)
MD=422
ND=926
LD=74
ID=13
READ 100, ATITLE
PRINT 99
PRINT 100, ATITLE
READ 1, NUMFL, NUMNP, NUMBC, NCPIN, NOPIN, NCYCM, TOLER, XFAC, NCONT
PRINT 101, NUMFL
PRINT 103, NUMNP
PRINT 104, NUMBC
PRINT 105, NCPIN
PRINT 106, NOPIN
PRINT 107, NCYCM
PRINT 108, XFAC
READ 5, K1, K2, K3, K4, K5, K6, K7
CALL CLEAR2(NUMFL, NUMNP, NPI, NPJ, NPK, NPL, ET, XU, RO,
1 XORD, YORD, ZORD, XLOAD, YLOAD, ZLOAD, DSX, DSY, DSZ,
2 NOD, NPK, K1, K2, K3, K4, K5, K6, ND, MD)
CALL SINFIX(NPB, NFIX, NOD, NPM, K1, K2, K3, K6, K7, LD)
IF (NCONT) 160, 155, 160
155 PRINT 110
PRINT 1, ( M, NPI(N), NPJ(N), NPK(N), NPL(N), ET(N), RO(N),
1 XU(N), N=1, NUMFL)
PRINT 111
PRINT 108, ( 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, NUMBP)
160 NCYCLE=0
NUMPT=NCPIN
NCPIN=NCPIN
JD=10, 1
DO 150 M=1, NUMNP
DO 150 L=1, JU
DO 150 K=1, 0
159 N(K, L, M)=0.0
150 UD 159 L=1, NUMNP
DO 170 M=1, JU
160 NPI(L, K)=0
NPL(L, JU)=0
175 NPL(L, J)=L
READ 20
READ 21
WRITE TAPF 24, ( NPI(N), NPJ(N), NPK(N), NPL(N), N=1, NUMFL)
PRINT 021
CALL MOUSTF (NUMFL,NPI,NPJ,NPK,NPL,XORD,YORD,ZORD,RO,ET,
1 XU,ZLOAD,STUB,AS,LM,KP,PR,ND,MD,ID,JD)
CALL NUTE (NPI,NPJ,NPK,NPL,NPM,ND,K1,K2,K3,ND)
WRITE TAPE 24, (NPI(N),NPJ(N),NPK(N),NPL(N),N=1,NUMEL)
END FILE 24
PRINT 021
CALL MOUSTF (NUMFL,NPI,NPJ,NPK,NPL,XORD,YORD,ZORD,RO,ET,
1 XU,ZLOAD,STUB,AS,LM,KP,PR,ND,MD,ID,JD)
END FILE 23
IF NPL(M) MX=1
205 CALL INVP2 (NUMNP,R,MD,ID)
206 NPL=MX-1
CALL SOUND2 (NUMRC,NPR,NFIX,NP,LD,MD,ID,JD)
243 PRINT 119
244 SUM=0.0
CALL ITER2 (SUM,NUMNP,NAP,XLOAD,YLOAD,ZLOAD,DSX,DSY,DSZ,
1 NP,R,XFA,MU,ID,JD)
NCYCLE=NCYCLE+1
IF (NCYCLE=NUMAT) 305,300,300
300 NUMOPT=NUMOPT+1
PRINT 120, NCYCLE,SUM
305 IF (SUM> TOLER) 400,400,310
310 IF (NCYCLE=NCYCLE) 400,400,315
315 IF (NCYCLE=NUMOPT) 244,320,320
320 NUMOPT=NUMOPT+1
400 PRINT 93
412 PRINT 122, (M,DSX(M),DSY(M),DSZ(M),M=1,NUMNP)
IF (SUM> TOLER) 440,440,430
430 IF (NCYCLE=NCYCLE) 440,440,450
450 CONTINUE
CALL OUTFIRK(0)
PRINT 029
GO TO 245
440 CONTINUE
CALL RATE (NUMFL,NUMNP,NPI,NPJ,NPK,NPL,DSX,DSY,DSZ,
1 XLOAD,YLOAD,ZLOAD,SM,MD,ND)
2 R,ND,NP,MD,K1,K2,K3
CALL FORCE (NUMNP,R,MD,ND)
PRINT 11
GO TO 222
1 FORMAT (6I4,2F12.6,11)
2 FORMAT (6I4,1F12.1,2F12.5)
4 FORMAT (6I4)
5 FORMAT (714)
11 FORMAT (//3/H THE END OF JOB VS030 , H.E.R.E.MUTLU .////////)
99 FORMAT (1HC)
100 FORMAT (1UC)
101 FORMAT (6H NUMBER OF ELEMENTS =1I4/)
102 FORMAT (6H NUMBER OF MODAL POINTS =1I4/)
103 FORMAT (6H NUMBER OF BOUNDARY POINTS =1I4/)
104 FORMAT (6H CYCLE WRITE INTERVAL =1I4/)
105 FORMAT (6H OUTPUT INTERVAL OF RESULTS =1I4/)
106 FORMAT (6H CYCLE LIMIT =1I4/)
107 FORMAT (6H TOLERANCE LIMIT =1F12.6/)
108 FORMAT (6H OVER RELAXATION FACTOR =1F6.3/)
109 FORMAT (1I8,PF12.1,3F12.6)
(II-1) ASSEMBLY STAGE MAIN ROUTINE FOR CALCULATING
    NATURAL FREQUENCIES AND NODE SHAPES

H.E. ERMUTLU, SOUTHAMPTON UNIVERSITY, JAN. 1967

SOLUTION OF EIGENVALUE PROBLEM WITH SPACE FINITE ELEMENT METHOD

DIMENSION UM(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 NOU(9,2,9), NPM(9,9)

COMMUN DM

READ 1, NP, NE, NC, NPP, NMAX, MMAX, NNN, NCA, NDIM, NCORN, KKK, I
1 NST, EPS, XAM, DLAM

READ 5, (NPN(K), K=1, NPP)

READ 9, K1, K2, K3, K4, K5

CALL ELEARM(NE, NP, NGE, 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, EPS

PRINT 23, XAM

PRINT 24, DLAM

PRINT 25

PRINT 26

PRINT 6, (NGE(N, I), I=1, NCORN), ET(N), XU(N), RO(N), N=1, NE)

PRINT 27

PRINT 7, (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 STIRF(NGE,G,S,A,B,ET,XU,VOL,NE,NP,IE)
CALL TRSFGP(UM,S,NGE,NSUB,NPN,NPA,NA,NE,NPP,NNN,KKK,MMAX,NDIM,
1 NCORN,IE)
31 CONTINUE
REWIND 29
WRITE TAPE 29, ((NGE(N,I),I=1,4),VOL(N),N=1,NE)
END FILE 29
CALL NOKTM (NGE,NPM,NOD,K1,K2,K3,NE)
PRINT 29
DO13 IE=1,NE
CALL STIRF(NGE,G,S,A,B,ET,XU,VOL,NE,NP,IE)
CALL TRSFGP(UM,S,NGE,NSUB,NPN,NPA,NA,NE,NPP,NNN,KKK,MMAX,NDIM,
1 NCORN,IE)
13 CONTINUE
DO 231 K=1,MMAX
DO 232 J=1,NNN
DO 233 I=1,NNN
OM(I,J,K)=0.5*DM(I,J,K)
233 CONTINUE
232 CONTINUE
231 CONTINUE
CALL APCCNP(UM,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 NTPEM(DM,NSUB,NNN,MMAX,NMAX)
CALL CLEAR(DM,NNN,MMAX)
CALL DIMNP(NSUB,NPN,MMAX,NPP)
DO 33 IE=1,NE
CALL CONMAS(S,vOL,RO,NF,IE)
CALL TRSFGP(UM,S,NGE,NSUB,NPN,NPA,NA,NE,NPP,NNN,KKK,MMAX,NDIM,
1 NCORN,IE)
33 CONTINUE
REWIND 23
READ TAPE 23, ((NGE(N,I),I=1,4),VOL(N),N=1,NE)
DO132 IE=1,NE
CALL CONMAS(S,vOL,RO,NF,IE)
CALL TRSFGP(UM,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
OM(I,J,K)=0.5*DM(I,J,K)
333 CONTINUE
332 CONTINUE
331 CONTINUE
CALL HYDNMC(DM,NSUB,G,NPN,NOD,NNN,MMAX,NP,NPP,K1,K2,K3,NPH)
CALL APCCNP(UM,NSUB,MCN,NC,NNN,MMAX,NCA)
CALL NTPEM(DM,NSUB,NNN,MMAX,NMAX)
END FILE 23
PRINT >7
GO TO EXIT
1 FORMAT(13I4,2F10.2)
4 FORMAT (20I4)
5 FORMAT (20I4)
6 FORMAT(5I4,1F12.1,2F12.5)
7 FORMAT(11B,3F12.3)
8 FORMAT(214)
(II-2) ANALYSIS STAGE MAIN ROUTINE FOR CALCULATING
    NATURAL FREQUENCIES AND MODE SHAPES.

H.E. ERKMUTLU, SOUTHAMPTON UNIVERSITY, JAN. 1967

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 MST, IEPS, XAM, DLAM, NCONT
CALL HST,IEPS,XAM,DLAM,NST
READ 2,((NPN(K), K=1, NPP)
READ 4,((MCN(L, I), I=1, 2), L=1, NC)
PRINT 1
PRINT 21, MST
PRINT 22, IEPS
PRINT 23, XAM
PRINT 24, DLAM
PRINT 25
PRINT 5,((NPN(K), K=1, NPP)
PRINT 26
PRINT 8,((MCN(L, I), I=1, 2), L=1, NC)
MST=NST
DU 01 KF=1, NCONT
CALL CLEAR(DM, NNN, MMAX)
DU 04 J=1, NPP
DU 04 I=1, NNN
34 VECT(I, J)=1.0
64 CALL PRVS2(XAM, DLAM, MST, IEPS, NNN, MMAX, NSUB, DM, VECT, NPP)
IF (MST) 62, 63, 63
62 MST=NST
XAM=VECTOR(DLAM)
GO TO 64
63 PRINT 21, XAM
FREQ=SQR2F(XAM)/6.28318
PRINT 52, FREQ

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

H.E.ERMUTLU, POWER SPECTRUM METHOD

DIMENSION DM(164,164),NPS(384,4),G(164,3),RO(384),VOL(384),
1 N0DU7,2,6),NPM(17,2),FX(164,5),FY(164,5),FZ(164,5),HX(164),
2 HY(164),HZ(164),AX(5,164),AY(5,164),BZ(5,164),C(5),FREQ(5),P(5),
3 GHASS(5),APS(101),SFR(101,5),SPQ(101,5),A(101),R(101),D(101)

COMMON DM,NPS,G,RO,VOL
READ 70, NPP,NDIM

70 FORMAT (2I4)
    READ 71, (NPP(K),K=1,NPP)

71 FORMAT (2I4)
    READ 1, NP,NE,MR,NC1,NC2,NS,DEI TA

1 FORMAT (6I4,1F6,3)
    READ 2, K1,K2,K3,K4,K5

2 FORMAT (5I4)
    CALL ELEREM (NE,NP,NPS,G,RO,N0D,NPM,K1,K2,K3,K4,K5)
    DO 10 J=1,NE
    DO 10 I=1,NE

10 DM(I,J)=0.0
    PRINT 29

29 FORMAT(1H1,55X,45H EL. 1 J K L VOLUME OF TETRAHEDRON )
    CALL RESMAS (NPS,G,DM,VOL,RO,NE,NP)
    IF (NC1) 11,11,12

12 NPH=NPH/2
DO 10 J=1,NP
HX(J)=0.0
HY(J)=0.0
HZ(J)=0.0
10 CONTINUE
CALL HYDREM (G,N0D,NP,K1,K2,K3,NPH,HX,HY,HZ)
11 CONTINUE
REMAIN 23
DO 14 K=1,MR
READ TAPE 23, M
CALL VCTRD (FX,FY,FZ,NP,MR,NPN,NPP,NDIM,M)
14 CONTINUE
READ 4, (FREQ(I),I=1,MR)
READ 3, (C(I),I=1,MR)
3 FORMAT (10F8.3)
DO 12 M=1,MR
PRINT 4, M,FREQ(M),C(M)
4 FORMAT (13H1MODE NUMBER=12//12H FREQUENCY=1F8.3,16H DAMPING RAT
10=1F6.8//)
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 (1I4,3F12.6)
IF (NC4) 103,103,104
101 CONTINUE
IF (NC4-2) 401,402,403
401 CALL MDCYRC(GMASS,SP0,SFR,APS,G,BX,CPS,VEL,NC4,MR,NP,NS,
1 DELTA)
GO TO 107
402 CALL MDCYRC(GMASS,SP0,SFR,APS,G,BY,CPS,VEL,NC4,MR,NP,NS,
1 DELTA)
GO TO 107
403 CALL MDCYRC(GMASS,SP0,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 MODRCPO (P,GMASS,SP0,SFR,APS,CPS,MR,NS,NC4)
107 CONTINUE
DO 106 NCT=1,NC5
READ 23, NC6, N
26 FORMAT (214)
IF (NC6-2) 301,302,303
301 CALL SPECTR (FX,SP0,A,N,MR,NP,NS,NC6)
GO TO 304
302 CALL SPECTR (FY,SP0,A,N,MR,NP,NS,NC6)
GO TO 304
303 CALL SPECTR (FZ,SP0,A,N,MR,NP,NS,NC6)
304 CONTINUE
CALL SIMPSN (AREA, DELTA, A, NS)
SIGMA = SQRTF (AREA)
PRINT 305, AREA, SIGMA

305 FORMAT (10HOVARANCE=1F20.6, 9H R.M.S.=1F20.6//)
CALL ASUMP (A, B, D, NS, DELTA, AREA, ARSORT)
CALL MAGFAC (ARSORT, SIGMA)

106 CONTINUE
101 CONTINUE
PRINT 108
108 FORMAT (29H0 THE END OF JOB VS030 ERMITLU///)
GO TO EXIT
END

(1) SUBROUTINE ELEAR2

SUBROUTINE ELEAR2(NUMFL, 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 NPT(ND), NPJ(ND), NPK(ND), NPL(N), 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, K3)
READ 41, ELMOD, POISON, UNIDOM
41 FORMAT (3F19.5, 3)
READ 7, ((NPM(I, J), J=1, K2), I=1, K1)
7 FORMAT (4X, 20I4)
CALL NOKTA (NPI, NPJ, NPK, NPL, NOD, K1, K2, K3, ND)
IF (K4) GT 1, 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
XORD(N)=UNIDOM
CALL CORDLD (XORD, YORD, ZORD, XLOAD, YLOAD, ZLOAD, NOD, K1, K2, K3,
1 MD)
IF (K5) GT 3, 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, 3F8.4, 1, 3F12.1)
55 M=M+1
53 DO. 57 M=1, NUMNP
DSX(M)=0.0
DSY(M)=0.0
DSZ(M)=0.0
IF (K6) GT 8, 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
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=J
DU I=1, K1
DU J=1, K2
KN=NPM(I, J)
DU K=1, KN
M=M+1
NOD(I, J, K)=M
1 CONTINUE

N=J
K1A=K1-1
K2A=K2-1
DU I=1, K1A
DU J=1, K2A
KNA=NPM(I, J)-1
DU K=1, KNA
NEXP=1+J+K
NTYPE=(I-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+1, J+1, K)
N=N+1
NPI(N)=NOD(I, J+1, K-1)
NPJ(N)=NOD(I, J, K-1)
NPK(N)=NOD(I+1, J, K+1)
NPL(N)=NOD(I+1, J, K+1)
N=N+1
NPI(N)=NOD(I+1, J, K-1)
NPJ(N)=NOD(I+1, J+1, K)
NPK(N)=NOD(I, J, K+1)
NPL(N)=NOD(I, J+1, 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, K)
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, 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)
rv = N - r
\[ r \cdot r + I \cdot N = N_0 U(1, J, K + 1) \]
\[ r \cdot (N + 1) = N_0 U(1, J, K + 1) \]
\[ \text{CONTINUE } \]
\[ \text{IF (} NPM(I, J) - NPM(I+1, J) \text{) } \leq 2,2 \]
\[ N = N + 1 \]
\[ N = N + 1 \]
SUBROUTINE NOETB

DIMENSION NPI(ND),NPJ(ND),NPK(ND),NPL(ND),NOD(K1,K2,K3)

NOD(K1,K2,K3)

N = 0

K1A = K1 = 1
K2A = K2 = 1

DO 1 I = 1, K1A
DO 1 J = 1, K2A
KNA = NPM(I,J) = 1
DO 2 K = 1, KNA

NEXP = I*J*K
NTYPE = (-1)**NEXP

IF (NTYPE) 3, 3, 4

N = N + 1

NPI(N) = NOD(I,J,K)
NPJ(N) = NOD(I,J+1,K)
NPK(N) = NOD(I,J,K+1)
NPL(N) = NOD(I,J,K+1)

N = N + 1

NPI(N) = NOD(I,J+1,K+1)
NPJ(N) = NOD(I,J+1,K+1)
NPK(N) = NOD(I,J+1,K+1)
NPL(N) = NOD(I,J+1,K+1)

N = N + 1

NPI(N) = NOD(I+1,J,K)
NPJ(N) = NOD(I+1,J,K)
NPK(N) = NOD(I+1,J,K)
NPL(N) = NOD(I+1,J,K)

N = N + 1

NPI(N) = NOD(I+1,J+1,K)
NPJ(N) = NOD(I+1,J+1,K)
NPK(N) = NOD(I+1,J+1,K)
NPL(N) = NOD(I+1,J+1,K)

GO TO 5

3 N = N + 1

NPI(N) = NOD(I,J,K)
NPJ(N) = NOD(I,J,K)
NPK(N) = NOD(I,J,K)
NPL(N) = NOD(I,J,K)

N = N + 1

NPI(N) = NOD(I,J+1,K)
NPJ(N) = NOD(I,J+1,K)
NPK(N) = NOD(I,J+1,K)
NPL(N) = NOD(I,J+1,K)

N = N + 1

NPI(N) = NOD(I+1,J,K)
NPJ(N) = NOD(I+1,J,K)
NPK(N) = NOD(I+1,J,K)
NPL(N) = NOD(I+1,J,K)

N = N + 1

NPI(N) = NOD(I+1,J+1,K)
NPJ(N) = NOD(I+1,J+1,K)
NPK(N) = NOD(I+1,J+1,K)
NPL(N) = NOD(I+1,J+1,K)
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+1, J+1, K)

5 CONTINUE
IF (NPM(I, J) = NPM(I+1, J)) 6, 2, 2
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+1, 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+1, J+1, K+1)
NPL(N) = NOD(I+1, J+1, K+1)
GOTO 2

7 N = N + 1
NPI(N) = NOD(I, J, K)
NPJ(N) = NOD(I+1, J, K)
NPK(N) = NOD(I, J+1, K)
NPL(N) = NOD(I+1, J, K+1)
N = N + 1
NPI(N) = NOD(I, J, K)
NPJ(N) = NOD(I+1, J, K)
NPK(N) = NOD(I+1, J+1, K+1)
NPL(N) = NOD(I+1, J, K+1)
GOTO 2

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, RADIUS, CENTER, THICK, ALPHA, NDIV, HLOAD, VLOAD
2 FORMAT (4F8.1, 1F8.3, 1(F14.2, F12.1))
T = THICK/(K2-1)
FA = ALPHA*3.14159/180.0
FD = FA/NDIV

(12)
KA=K1-NDIV
NS=0
DU 1 J=1,K2
R=RAD-NS*T
NS=NS+1
F=FA
DU 1 I=KA,K1
SN=SNF(F)
CS=CSF(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) 8,6,9
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 UXY=DX*DX+DY*DY
ZLOAD(N)=HLOAD*SQRTF(DXY)/DZ+VLOAD
GO TO 5
13 UXY=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-DF-
1 CONTINUE
RETURN
END

(5) SUBROUTINE SINIR

SUBROUTINE SINIR(NPB,NFIX,NOD,NPM,K1,K2,K3,K7,LD)
DIMENSION NPB(LD),NFIK(LD),NOD(K1,K2,K3),NPM(K1,K2)
N=0
DU 1 J=1,K2
KN=NPM(1,J)
DU 1 K=1,KN
N=N+1
NPB(N)=NOD(1,J,K)
1 NFIK(N)=0
(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), NPM(NP, 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*BJ*CK+AK*BL*CJ-AK*BK*CL+AL*BJ*CK-AL*BK*CL
IF (SIXVOL) 701, 703, 177
701 PRINT 711, N, NPJ(N), NPK(N)
711 FORMAT('12H ELEMENT NO., 14, 31H INTERCHANGED NODAL POINT NO. S. 214)
K=NPJ(N)
J=NPK(N)
NPJ(N)=J
NPK(N)=K
GO TO 155
703 PRINT 713, N
713 FORMAT('12H ZERO VOLUME. EL. NO. =114)
GO TO EXIT
177 VOL=SIXVOL /6.0
PRINT 6, N, NPI(N), NPJ(N), NPK(N), NPL(N), VOL
6 FORMAT ('12X,5I4, 1F25.6)
\[DL = \frac{VOL \cdot R0(N)}{4};\]
\[DL = 0.5 \cdot DL\]

\[ZLOAD(I) = ZLOAD(I) - DL\]
\[ZLOAD(J) = ZLOAD(J) - DL\]
\[ZLOAD(K) = ZLOAD(K) - DL\]
\[ZLOAD(L) = ZLOAD(L) - DL\]

**FORMATION OF STIFFNESS ARRAY**

\[COMM = T(N) / ((1 + XU(N)) \cdot (1 - 2 \cdot XU(N)) \cdot 36 \cdot VOL)\]
\[D1 = (1 - XU(N)) \cdot COMM\]
\[D2 = XU(N) \cdot COMM\]
\[D3 = (0.5 - XU(N)) \cdot COMM\]

\[DO\]
\[J = 1, 12\]
\[I = 1, 12\]

\[BTDB(I, J) = 0.0\]
\[BTDB(1, 1) = D1\]
\[BTDB(1, 1) = D1\]
\[BTDB(12, 12) = D1\]
\[BTDB(2, 2) = D2\]
\[BTDB(7, 7) = D2\]
\[BTDB(12, 12) = D2\]
\[BTDB(2, 12) = D2\]
\[BTDB(7, 2) = D2\]
\[BTDB(7, 12) = D2\]
\[BTDB(12, 7) = D2\]
\[BTDB(3, 3) = D3\]
\[BTDB(3, 6) = D3\]
\[BTDB(4, 4) = D3\]
\[BTDB(4, 10) = D3\]
\[BTDB(6, 3) = D3\]
\[BTDB(6, 6) = D3\]
\[BTDB(8, 8) = D3\]
\[BTDB(11, 11) = D3\]
\[BTDB(10, 10) = D3\]
\[BTDB(11, 8) = D3\]

\[A21 = 3J \cdot CL - BJ \cdot CK + BK \cdot CJ - AK \cdot CL + BL \cdot CK - BL \cdot CJ\]
\[A22 = 3K \cdot CL - BL \cdot CK\]
\[A23 = 3L \cdot CJ - BJ \cdot CL\]
\[A24 = 3J \cdot CK - AK \cdot CJ\]
\[A31 = AJ \cdot CK - AK \cdot CJ\]
\[A32 = AL \cdot CK - AK \cdot CJ\]
\[A33 = AJ \cdot CL - AL \cdot CJ\]
\[A34 = AK \cdot CJ - AJ \cdot CK\]
\[A41 = AJ \cdot BL - AJ \cdot BK + AK \cdot BJ - AL \cdot BL + AL \cdot BK - AL \cdot BJ\]
\[A42 = AK \cdot BL - AL \cdot BK\]
\[A43 = AL \cdot BJ - AJ \cdot BL\]
\[A44 = AJ \cdot BK - AK \cdot BJ\]

\[DO\]
\[J = 1, 12\]
\[I = 1, 12\]
\[ A_{(12,3)} = A_{41} \]
\[ A_{(2,4)} = A_{22} \]
\[ A_{(6,5)} = A_{22} \]
\[ A_{(10,6)} = A_{22} \]
\[ A_{(3,4)} = A_{32} \]
\[ A_{(7,3)} = A_{32} \]
\[ A_{(11,6)} = A_{32} \]
\[ A_{(4,4)} = A_{42} \]
\[ A_{(8,5)} = A_{42} \]
\[ A_{(12,6)} = A_{42} \]
\[ A_{(2,7)} = A_{23} \]
\[ A_{(6,8)} = A_{23} \]
\[ A_{(10,9)} = A_{23} \]
\[ A_{(3,7)} = A_{33} \]
\[ A_{(7,8)} = A_{33} \]
\[ A_{(11,9)} = A_{33} \]
\[ A_{(4,7)} = A_{43} \]
\[ A_{(8,8)} = A_{43} \]
\[ A_{(12,9)} = A_{43} \]
\[ A_{(2,10)} = A_{24} \]
\[ A_{(6,11)} = A_{24} \]
\[ A_{(10,12)} = A_{24} \]
\[ A_{(3,10)} = A_{34} \]
\[ A_{(7,11)} = A_{34} \]
\[ A_{(11,12)} = A_{34} \]
\[ A_{(4,10)} = A_{44} \]
\[ A_{(8,11)} = A_{44} \]
\[ A_{(12,12)} = A_{44} \]

\[ \text{DO } 181 \quad J = 1, 12 \]
\[ \text{DO } 181 \quad I = 1, 12 \]
\[ S(I, J) = 0, \]
\[ \text{DO } 181 \quad K = 1, 12 \]
\[ S(I, J) = S(I, J) + A(I, K) \times A(K, J) \]
\[ \text{WRITE TAPE 23, \{ (S(I, J), I = 1, 12), J = 1, 12 \}} \]
\[ \text{DO 184 \quad J = 1, 12} \]
\[ \text{DO 184 \quad I = 1, 12} \]
\[ S(I, J) = 0,5 \times S(I, J) \]
\[ \text{CONTINUE} \]
\[ LM(1) = NP1(N) \]
\[ LM(2) = NPJ(N) \]
\[ LM(3) = NPK(N) \]
\[ LM(4) = NPL(N) \]
\[ \text{DO 200 \quad L = 1, 4} \]
\[ \text{DO 200 \quad M = 1, 4} \]
\[ LX = LM(L) \]
\[ MX = 0 \]
\[ \text{IF } (NP(LX, MX) - LM(M)) \quad 190, 195, 190 \]
\[ \text{IF } (NP[LX, MX]) \quad 185, 195, 185 \]
\[ NP(LX, MX) = LM(M) \]

(16)
IF (MX=JU) 196,702,702
PRINT 712, (LX)
FORMAT (3I10,MORE N,P, ADJACENT TO N.P. NO,14)
GO TO EXIT
196 R(1,MX,LX)=R(1,MX,LX)+R(3,1,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(9,1,M)+R(4,1,M)*R(8,1,M)*R(3,1,M)+
2 R(5,1,M)*R(9,1,M)+R(3,1,M)+R(7,1,M)*R(6,1,M)*R(9,1,M)+R(4,1,M)*R(3,1,M)+
S11=( R(5,1,M)*R(9,1,M)+R(4,1,M)*R(7,1,M)*R(6,1,M))/COMM
S12=(-R(2,1,M)*R(9,1,M)+R(4,1,M)*R(7,1,M)*R(6,1,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

(7) SUBROUTINE INVNP2

SUBROUTINE INVNP2(NUMNP,R,MD,ID)
DIMENSION R(9,ID,MD)
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(9,1,M)+R(4,1,M)*R(8,1,M)*R(3,1,M)+
2 R(5,1,M)*R(9,1,M)+R(3,1,M)+R(7,1,M)*R(6,1,M)*R(9,1,M)+R(4,1,M)*R(3,1,M)+
S11=( R(5,1,M)*R(9,1,M)+R(4,1,M)*R(7,1,M)*R(6,1,M))/COMM
S12=(-R(2,1,M)*R(9,1,M)+R(4,1,M)*R(7,1,M)*R(6,1,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(D),NP(MD,JD),R(9,ID,MD)
MODIFICATION OF BOUNDARY FLEXIBILITIES
DO 240 L=1,NUMBC
M=NPB(L)
NP(M,1)=0
(17)
IF (NFX(L) - 1) 225, 228, 215
228 \[ R(1,1,M) = R(5,1,M) - R(4,1,M) \times R(2,1,M) / R(1,1,M) \]
\[ R(2,1,M) = R(5,1,M) - R(4,1,M) \times R(3,1,M) / R(1,1,M) \]
\[ R(3,1,M) = R(5,1,M) - R(4,1,M) \times R(3,1,M) / R(1,1,M) \]
\[ R(4,1,M) = 0.0 \]
\[ R(5,1,M) = 0.0 \]
\[ R(6,1,M) = 0.0 \]
\[ R(7,1,M) = 0.0 \]
GO TO 240
240 CONTINUE
RETURN
END

(9) SUBROUTINE ITERA2

SUBROUTINE ITERA2 (SUM, NUMNP, NAP, XLOAD, YLOAD, ZLOAD, DSX, DSY, DSZ, NP, RFAC, MD, ID, JD)
DIMENSION NAP(MD), XLOAD(MD), YLOAD(MD), ZLOAD(MD), DSX(MD), DSY(MD), DSZ(MD), NP(MD, JD), R(9, ID, MD)

ITERATION ON NODAL POINT DISPLACEMENTS
DO 290 M = 1, NUMNP
IF (R(1,1,M) + R(5,1,M) + R(9,1,M)) 275, 275
FRX = XLOAD(M)
FRY = YLOAD(M)
FRZ = ZLOAD(M)
GO TO 290
275 CONTINUE
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)
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
SUM=SUM+ABSF(DX/R(1,1,M))+ABSF(DY/R(5,1,M))+ABSF(DZ/R(9,1,M))
285 CONTINUE
RETURN
END

(10) SUBROUTINE REACT2

SUBROUTINE REACT2 (NUMFL, NUMNP, NPI, NPJ, NPK, NPL, DSX, DSY, DSZ,
1 XLOAD, YLOAD, ZLOAD, S, 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 155 M=1,NUMNP
DO 156 L=1,B
DO 167 K=1,3
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
25 IF (K<KNA) 15,25,25
25 IF (NPM(I,J)=NPM(I+1,J)) 45,2,2
45 MLP=3
(19)
LN(2) = 0
LN(3) = 0
GO TO 35

15 MLP = 5
LN(2) = NOU(I, J, K + 1)
LN(3) = NOU(I, J, K + 1)
LN(4) = NOU(I, J + 1, K + 1)
LN(5) = NOU(I + 1, J, K)
LN(6) = NOU(I + 1, J + 1, K)
LN(7) = NOU(I + 1, J, K + 1)
LN(8) = NOU(I + 1, J + 1, K + 1)
DO 4 KLM = 1, MLP
N = N + 1
READ TAPE 23,
(LS(IJ, JJ), IJ = 1, 12), JJ = 1, 12)

LM(1) = NP1(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

530 CONTINUE:
XFORCE = XFORCE + ((3*M - 2, 3*L - 2)*DSX(LX) + (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) + (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) + (3*M, 3*L - 1)*DSY(LX))
1 + S(3*M, 3*L) * DSZ(LX)
DO 540 CONTINUE:
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) + YFORCE * 0.5
R(2, NO, MX) = R(2, NO, MX) + YFORCE * 0.5
R(3, NO, MX) = R(3, NO, MX) + YFORCE * 0.5
530 CONTINUE:
4 CONTINUE:
3 CONTINUE:
2 CONTINUE:
1 CONTINUE:
999 CONTINUE:
PRINT 503
503 FORMAT(57H1 NODEAL POINT: X-REACTION: Y-REACTION: Z-REACTION:
1N)
PRINT 504, (M, XLOAD(M), YLOAD(M), ZLOAD(M), M = 1, NUMNP)
504 FORMAT(11I12, 3F15.1)
RETURN
END
(11) SUBROUTINE FORCE

SUBROUTINE FORCE (NUMNP,R,ID,MU)
DIMENSION R(9,ID,MU)
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)
FORWARD=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)
LEFTW=R(I,5,M)+R(I,6,M)+R(I,7,M)+R(I,8,M)
RIGHT=R(I,1,M)+R(I,2,M)+R(I,3,M)+R(I,4,M)
PRINT 4, I,M,UPWARD,DOWNWD,FORWARD,BACKWD,LEFTW,RIGHT
4 FORMAT (I2,6F12.1)
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 (2014)
CALL NOKTAM(NGE,NPM,NOD,K1,K2,K3,NE)
IF (K4) 51,51,52
52 READ /O, N
53 FORMAT (14)
DO /5 1=1,K4
READ 9, (NGE(N,L),L=1,4)
9 FORMAT (4X,414)
54 FORMAT (4X,414)
55 N=N+1
51 DO /6 N=1,NE
ET(N)=ELMOD
XU(N)=POISON
56 HU(N)=UNIDOM
CALL CORDLM(G,NOD,K1,K2,K3,NP)
IF (K5) 53,53,54
54 READ /O, M
55 DO /5 1=1,K5
READ 50, (G(M,L),L=1,3)
56 FORMAT (4X,3F8.3)
55 M=M+1
53 RETURN
END
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
CONTINUE

K1A=K1-1
K2A=K2-1
DO 2 I=1,K1A
DO 2 J=1,K2A
KNA=NPM(I,J)-1
DO 2 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+1,J+1,K)
N=N+1
L(N,1)=NOD(I+1,J+1,K+1)
L(N,2)=NOD(I+1,J,K+1)
L(N,3)=NOD(I+1,J+1,K+1)
L(N,4)=NOD(I+1,J+1,K+1)
N=N+1
L(N,1)=NOD(I+1,J,K+1)
L(N,2)=NOD(I+1,J,K)
L(N,3)=NOD(I+1,J+1,K)
L(N,4)=NOD(I+1,J+1,K+1)
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+1,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)
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,J+1,K)
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)
(22)
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+1, J, K+1)
L(N,4) = NOD(I, J+1, K+1)
N = N + 1
L(N,1) = NOD(I+1, 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)
N = N + 1
L(N,1) = NOD(I+1, J+1, K+1)
L(N,2) = NOD(1, J+1, K+1)
L(N,3) = NOD(I+1, J+1, K)
L(N,4) = NOD(I+1, J, K+1)
CONTINUE
IF (NPM(I, J) - NPM(I+1, J)) 6, 2, 2
K = NPM(I, J)
NEXP = I+j+K
NTYPE = (-1)**NEXP
IF (NTYPE) 7, 7, 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, K+1)
GO TO 2
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+1, K+1)
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)
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, K+1)
L(N,3) = NOD(I+1, J+1, K)
L(N,4) = NOD(I+1, J+1, K+1)
CONTINUE
RETURN
END
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 1 I = 1, K1A
DO 1 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 + 1, J, K)
L(N, 3) = NOD(I, J + 1, K)
L(N, 4) = NOD(I + 1, 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, J + 1, K + 1)
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 + 1, J, 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 + 1, J, K + 1)
L(N, 4) = NOD(I, J + 1, K + 1)
N = N + 1
L(N, 1) = NOD(I + 1, J + 1, 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)
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 + 1, J, K + 1)
L(N, 4) = NOD(I + 1, J + 1, 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, K + 1)
GO TO 1
5 N = N + 1
L(N, 1) = NOD(I, J + 1, K)
L(N, 2) = NOD(I, J, K)
L(N, 3) = NOD(I, J, K)
L(N, 4) = NOD(I, J, K)
N = N + 1
L(N, 1) = NOD(I + 1, 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, 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 + 1, J, K + 1)
L(N, 4) = NOD(I + 1, J, K + 1)


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

(15) SUBROUTINE CORDIL

SUBROUTINE CORDIL(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
```

(25)
NS=0
DO 1 J=1,K2
R=RAD-NS*T
NS=NS+1
FF=FA
DO 1 I=K4,K1
NF=MOD(I,J,K)
G(N,1)=R*SINF(F)
G(N,2)=R*COSF(F)+CENTER
G(N,3)=LEVEL
FF=F-FF
1 CONTINUE
RETURN
END

(16) SUBROUTINE DIMNP

SUBROUTINE DIMNP(NSUB,NPN,NMAX,NPP)
DIMENSION NSUB(NMAX,2),NPN(NPP)
NPP1=NPP-1
DO 1 I=1,NPP1
NSUB(I,*I-1,1)=I*NPN(I)
NSUB(I,*I-1,2)=I*NPN(I+1)
NSUB(I,*I,1)=I*NPN(I+1)
NSUB(I,*I,2)=I*NPN(I)
1 CONTINUE
DO 2 I=1,NPP
NSUB(I,*I-2,1)=I*NPN(I)
NSUB(I,*I-2,2)=I*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
DM(I,J,K)=0.0
1 RETURN
END

(26)
SUBROUTINE STIFF

SUBROUTINE STIFF (NGE, R, S, A, BTDR, ET, XU, VOL, NE, NP, N)

DIMENSION G(NGE, 4), S(12, 12), A(12, 12), BTDR(12, 12),
1 ET(NE), XU(NE), VOL(NE)

12 N = NGE(N, 1)
N = NGE(N, 2)
N = NGE(N, 3)
N = NGE(N, 4)
AJ = G(NJ, 1) - G(NJ, 1)
BJ = G(NJ, 2) - G(NJ, 2)
CJ = G(NJ, 3) - G(NJ, 3)
AK = G(NK, 1) - G(NK, 1)
BK = G(NK, 2) - G(NK, 2)
CK = G(NK, 3) - G(NK, 3)
AL = G(NL, 1) - G(NL, 1)
BL = G(NL, 2) - G(NL, 2)
CL = G(NL, 3) - G(NL, 3)

VOL(N) = (AJ*BJ*CK - AL*BJ*CK + AK*BL*CJ - AK*BL*CJ)/6.0

14 PRINT 711, N, NJ,NK
N = NGE(N, 2)
N = NGE(N, 3)
N = NGE(N, 2) = N
N = NGE(N, 3) = NK

GO TO 12

15 PRINT 715, N
GO TO EXIT

2001 BTU(J, J) = 0.0
BTU(J, 2) = H1
BTU(J, 7) = H1
BTU(12, 12) = D1
BTU(2, 7) = H2
BTU(2, 12) = D2
BTU(7, 2) = H2
BTU(7, 12) = D2
BTU(12, 2) = D2
BTU(12, 7) = D2
BTU(12, 3) = H3
BTU(3, 6) = H3
BTU(9, 4) = H3
BTU(4, 10) = D3
BTU(6, 3) = H3
BTU(6, 9) = H3
BTU(6, 11) = D3
BTU(10, 4) = D3
BTU(10, 10) = D3
BTU(11, 8) = D3
BTU(11, 11) = D3
A21 = B0 + C0 + D0 + E0 + F0 + G0 + H0 + I0 + J0 + K0 + L0 + M0 + N0 + O0 + P0 + Q0 + R0 + S0 + T0 + U0 + V0 + W0 + X0 + Y0 + Z0
A22 = B1 + C1 + D1 + E1 + F1 + G1 + H1 + I1 + J1 + K1 + L1 + M1 + N1 + O1 + P1 + Q1 + R1 + S1 + T1 + U1 + V1 + W1 + X1 + Y1 + Z1
A23 = B2 + C2 + D2 + E2 + F2 + G2 + H2 + I2 + J2 + K2 + L2 + M2 + N2 + O2 + P2 + Q2 + R2 + S2 + T2 + U2 + V2 + W2 + X2 + Y2 + Z2
A24 = B3 + C3 + D3 + E3 + F3 + G3 + H3 + I3 + J3 + K3 + L3 + M3 + N3 + O3 + P3 + Q3 + R3 + S3 + T3 + U3 + V3 + W3 + X3 + Y3 + Z3

\[ S[I,J] = S[I,J] + BTD[I,K] * A(K,J) \]

(28)
SUBROUTINE CONMAS

DIMENSION C(12,12), VOL(NE), RO(NE)
TMASS = VOL(N) * RO(N)
C1 = 0.1 * TMASS
C2 = 0.0 * TMASS
DO 11 J = 1, 12
  DO 12 I = 1, 12
    C(I,J) = 0.0
  12 CONTINUE
11 CONTINUE

(19) SUBROUTINE CONMAS
SUBROUTINE HYDNMC

SUBROUTINE HYDNMC(A,NSUH,G,NPN,N0D,NNN,MMAX,NP,NPP,K1,K2,K3,NPH)
DIMENSION A(NNN,NNN,MMAX),NSUH(MMAX,2),G(NP,3),NPN(NPP),
NOD(K1,K2,K3),HYDNC(60)

EVALUATION OF HYDRODYNAMIC LOADS
NW=0
READ 21,(HYDNC(M),M=1,NPH)
21 FORMAT (16F5.3)
PRINT 111
111 FORMAT (IH1)
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
K0=K1-NDIV
FA=FA
DO 1 I=KA,K1
NiNOO(I,1,K) ,
SN=SNF(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)
IF=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)
TRANSFER TO THE OVERALL MASS MATRIX
IM=0
NRS=0
3 IM=IM+1
NRS =NPS+NPN(IM)

(30)
(25) SUBROUTINE WTPME

SUBROUTINE WTPME(A, NSUB, NNN, KMAX, NMAX)
DIMENSION A(NNN, NNN, KMAX), NSUB(KMAX, 2)
DO 1 MA = 1, NMAX
NH = NSUB(MA, 1)
NC = NSUB(MA, 2)
IF (NH = NC) 1, 1, 2
1 WRITE TAPE 23, ((A(I, J, MA), I=1, NR), J=1, NC)
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 VCTWR

SUBROUTINE VCTWR(VECT,NPN,NNN,NPP,NDIM)

DIMENSION VECT(NNN,NPP),NPN(NPP)

DO 1 NP=1,NPP
   NPS=1
   DO 2 JP=1,NP
      NPS=NPS+NPN(JP)
      NPM=NPS-NM
      DO 1 I=1,NM
         MP=NPM+1
         NI=NDIM*(I-1)
         PRINT(3,MP,(VECT(NI+J,NP),J=1,NDIM)
1 CONTINUE
3 FORMAT(118,3F15.10)
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(1F19.5)
READ 7, ((NPM(I,J),J=1,K2),I=1,K1)
7 FORMAT(2014)
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
   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
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
CONTINUE

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 3 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+1,J,K)
L(N,3)=NOD(I,J+1,K)
L(N,4)=NOD(I+1,J+1,K)
N=N+1
L(N,1)=NOD(I+1,J,K)
L(N,2)=NOD(I+1,J+1,K)
L(N,3)=NOD(I+1,J,K+1)
L(N,4)=NOD(I,J+1,K)
N=N+1
L(N,1)=NOD(I+1,J,K)
L(N,2)=NOD(I+1,J+1,K)
L(N,3)=NOD(I,J,K+1)
L(N,4)=NOD(I,J+1,K)
N=N+1
L(N,1)=NOD(I+1,J,K)
L(N,2)=NOD(I+1,J,K)
L(N,3)=NOD(I+1,J+1,K)
L(N,4)=NOD(I+1,J+1,K)
N=N+1
L(N,1)=NOD(I+1,J,K)
L(N,2)=NOD(I+1,J+1,K)
L(N,3)=NOD(I+1,J+1,K)
L(N,4)=NOD(I+1,J+1,K)
GO TO 5

4 N=N+1
L(N,1)=NOD(I+1,J+1,K)
L(N,2)=NOD(I+1,J+1,K)
L(N,3)=NOD(I+1,J+1,K)
L(N,4)=NOD(I+1,J+1,K)

(33)
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 + 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 + 1, J + 1, K + 1)
L(N, 2) = NOD(I, J + 1, K)
L(N, 3) = NOD(I + 1, J, K + 1)
L(N, 4) = NOD(I + 1, J + 1, K)
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 + 1, 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, J, K)
L(N, 4) = NOD(I + 1, J, K)
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 + 1)
L(N, 3) = NOD(I + 1, J, K + 1)
L(N, 4) = NOD(I + 1, J, K)
GO TO 2
12 K = NPM(I, J) - 1
NEXP = I + J + K
NTYPE = (-1)**NEXP
IF (NTYPE) 13, 13, 14

(34)
SUBROUTINE CORSYM

SUBROUTINE CORSYM(G,N0n,K1,K2,K3,NP)
DIMENSION G(NP,3),N0n(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=N0D(I,J,K)
S(N,1)=R*SINF(F)
S(N,2)=R*COSF(F)+CENTER
G(N,3)=HLEVEL
F=F-FT
1 CONTINUE
RETURN
END

(34) SUBROUTINE CORSYM

(35)
SUBROUTINE RESMAS(NGE, GD, DM, VOL, RO, NE, NP)

DIMENSION NGE(NE, 4), GD(NP, 3), DM(NP, NP), VOL(NE), RO(NE)

DO 31 N=1, NE

12 N=NGE(N, 1)
NJ=NGE(N, 2)
NL=NGE(N, 4)

AJ=GD(NJ, 1) - GD(NJ, 1)
BJ=GD(NJ, 2) - GD(NJ, 2)
CJ=GD(NJ, 3) - GD(NJ, 3)

AK=GD(NK, 1) - GD(NK, 1)
BK=GD(NK, 2) - GD(NK, 2)
CK=GD(NK, 3) - GD(NK, 3)

AL=GD(NL, 1) - GD(NL, 1)
BL=GD(NL, 2) - GD(NL, 2)
CL=GD(NL, 3) - GD(NL, 3)

VOL(N)=AJ*BL*CK + AK*BK*AL - AK*BK*CL + AL*BL*CK - AL*BL*AJ + AJ*BL*CK)

IF (VOL(N)) 13, 14, 15

PRINT 711, N, NJ, NK

13 FORMAT (12H ELEMENT NO. I14, 31H INTERCHANGED NODAL POINT NO. S. II4)

NK=NGE(N, 2)
NJ=NGE(N, 3)

14 PRINT 713, N

15 FORMAT (21H ZERO VOLUME. EL. NO. = I14)

GO TO EXIT

6 FORMAT (56X, 514, 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

CONTINUE

RETURN

END
SUBROUTINE HYDREM

DIMENSION G(NP,3),NOD(K1,K2,K3),NPH,HX,HY,HZ

EVALUATION OF HYDRODYNAMIC LOADS

READ 21,(HYDNC(M),M=1,NPH)

PRINT 111

DO 1 K=1,K3

READ 2, ALPHA,ALOAD,NDIV

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

SN=SIN(F)

CS=COS(F)

NW=W+1

HLOAD=ALOAD*HYDNC(NW)

XLOAD=ABSF(HLOAD*SN)

YLOAD=ABSF(HLOAD*CS)

IF (K=1) 10,10,11

ZLOAD=0.0

GO TO 5

10 ZLOAD=0.0

DO TO 5

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

1 CONTINUE

RETURN

END

SUBROUTINE VCTR

DIMENSION FX(ND,MR),FY(ND,MR),FZ(ND,MR),NPN,NPP,NDIM,M

DO 1 NP=1,NPP

NPS=0

DO 2 JP=1,NP

NPS=NPS+NPN(JP)

NM=NPS-NP

DO 1 I=1,NM

HP=NPM+I

READ TAPE 23, (DUM(J),J=1,NDIM)

FX(HP,M)=DUM(1)

FY(HP,M)=DUM(2)

FZ(HP,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), DM(NP, MR), HY(NP), HZ(NP), GMASS(MR)

DO 11 J=1, NP
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)
CONTINUE
IF (NC1) 11, 11, 13

11 CONTINUE
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)
CONTINUE
14 CONTINUE

DO 21 M=1, MR
GMASS(M) = GMASS(M) + BX(M, J) * FX(J, M) + BY(M, J) * FY(J, M) + BZ(M, J) * FZ(J, M)
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
F = FREQ(M) * FREQ(M)
CF = 4.0 * C(M) * C(M) * FR
F = F + CF
CONTINUE

DO 2 N=1, NS
FF = F * F
R = R + FF
SFR(N, M) = 1.0 / (R + CF * FF)
F = F + DELTA
CONTINUE

PRINT 3
FORMAT (23H1UNIT MODAL RECEPTANCES/)
PRINT 4, (N, SFR(N, M), M=1, MR), N=1, NS)

RETURN
END

(38)
SUBROUTINE MDCYRC

SUBROUTINE MDCYRC(GMASS, SPO, SFR, APS, G, QA, CPS, VEL, NC4, MR, NP, NS, 1 DELTA)

DIMENSION GMASS(MR), SPO(NS, MR), SFR(NS, MR), APS(NS), G(NP, 3), 1 QA(MR, NP)

PI = 3.14159
CA = 10.0 * 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

FORMAT (22H MODAL POWER SPECTRUMS//)
PRINT 2
2 FORMAT (5F20.10//)
FORMAT (5F20.10//)
PRINT 3, (GMASS(M), M = 1, MR)
3 FORMAT (17H FORCE DIRECTION=112, 22H INTENSITY CONSTANT=1F5.3,
1 12H VELOCITY=1F7.1//)
PRINT 4, (N, (SPQ(N, M), M = 1, MR), N = 1, NS)
4 FORMAT (1I4, 5F20.10)
RETURN
END

SUBROUTINE DCYFRC

SUBROUTINE DCYFRC(F, VEL, G, QA, P, NC4, M, 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) * SIN(FI)
PC = PC + QA(M, J) * COS(FI)
1 CONTINUE
P = PS * PS + PC * PC
RETURN
END

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
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
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 (1I4,5F90.10)
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(13H1N0DAL POINT=1I4,11H DIRECTlON=ll2//3lH POWER SPECTRUM 0 IF 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
U=0.0
N2=NS-2
DO 2 I=3,N2,2
U=U+A(I)
2 CONTINUE
AREA=(B+4.0*C+2.0*U)*DELTA/3.0
RETURN
END
SUBROUTINE ASUMP

SUBROUTINE ASUMP(A,B,D,NS,DELTA,AREA,ARSORT)
DIMENSION A(NS),B(NS),D(NS)
F=0.0
DO 1 I=1,NS
U(I)=A(I)*F*F
F=F+DELTA
1 CONTINUE
CALL SIMPSN (ARSO,DELTA,B,NS)
ARSO=SQRTF(ARSO)
PRINT 3, ARSO, ARSORT
FORMAT (6HOARSO=1F20.3,8H ARSORT=1F20.3)
F=0.0
DO 2 I=1,NS
U(I)=B(I)*F*F
F=F+DELTA
2 CONTINUE
CALL SIMPSN (ARQT,DELTA,D,NS)
PRINT 4, ARQT
FORMAT (6HOARQT=1F20.3)
RMT=AREA+ARQT/(ARSO*ARSO)
AKSO=1.0/(RMT-1.0)
AKSORT=SORTF(AKSO)
PRINT 5, AKSO, AKSORT
FORMAT (6HOAKSO=1F20.8,8H AKSORT=1F20.8)
RETURN
END

SUBROUTINE MAGFAC

SUBROUTINE MAGFAC(ARSO,RT,SIGMA)
PRINT 1
1 FORMAT (33H0DURATION MAG.COEF. MEAN MAX.)
F=1U.0
DO 2 I=1,6
T=I*UT
DF=LOGF(ARSO*T/SIGMA)
CN=SORTF(2.0*DF)
XMAX=CN*SIGMA
PRINT 3, T,CN,XMAX
2 FORMAT (1F9.2,1F12.3,1F12.4)
RETURN
END

(46) SUBROUTINE ASUMP

(47) SUBROUTINE MAGFAC