Report no.
UC SESM 73-3

STRUCTURAL ENGINEERING AND STRUCTURAL MECHANICS

SMIS

SYMBOLIC MATRIX INTERPRETIVE SYSTEM

by

Edward L. Wilson

APRIL 1963
modified
JANUARY 1973

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA

SMIS

Symbolic Matrix Interpretive System

by
Edward L. Wilson
Professor of Civil Engineering

Programmed
April 1963

Modified
January 1973

Structural Engineering and Structural Mechanics
Department of Civil Engineering
University of California, Berkeley

TABLE OF CONTENTS

SUMMARY

The Symbolic Matrix Interpretive System is designed to perform a series of matrix operations under the direct control of the user. The program can be used for many problems which can be formulated as a series of matrix operations; however, several special operations have been added which allows the program to be easily used for the static and dynamic analysis of small complex structural systems. The program is written in standard FORTRAN IV and does not use an overlay system. It can be used in single precision without significant modification on most modern computers. The program is not proprietary and there is no restriction on the distribution or use of the program.

All matrices are identified by symbols which are selected by the user. The size of the matrix storage area is limited by the high speed storage capacity of the computer used (For the CDC 6400 on the Berkeley Campus this is approximately 40,000). Matrix storage on low speed storage is practically unlimited. The present dimension statements within the program limit the number of matrices in high speed storage to 100.

DATA PREPARATION

A program run consists of reading the input deck once and executing the operations designated by the control cards as they are encountered. Each matrix is referred to by its name -- one to six alphabetic or numeric characters. The name of a matrix is selected by the user when it is first loaded or generated. If an attempt is made to load or generate a matrix with the same name as a previously defined matrix, the previous matrix is destroyed <u>before</u> the operation is executed.

The input deck is composed of <u>control cards</u>, <u>data cards</u>, and <u>remark cards</u>. A control card designates the "operation" to be performed, along with its "calling sequence." The calling sequence contains the names of the matrices to be used in the operation, size of matrices and other possible numbers. Data cards follow operations which require additional information. The attached data sheet indicates the format in which cards are to be punched. Information is punched in fields widths of six. The operation symbol must begin in column one. Blanks are considered to be legitimate symbols when referring to matrices in fields F1, F2, F3 and F4.

Blanks are considered to be zero when encountered in the numerical field N1 to N4; therefore, these numbers should be packed to the right of the field. Unless a decimal point is specified in the scalar field S1 and S2, it is assumed to be at the extreme right of the field. Remark cards are used for labeling the output -- all keypunch symbols are acceptable.

INPUT FORM

6	12	18	24	30	36	42	48	54	60	66	72
OPERATION	F1	F2	F3	F4	N1	N2	N3	N4	S1	S2	

MATRIX INPUT, OUTPUT and GENERATION OPERATIONS

START This operation causes the elimination of all matrices which are in core storage. This card is not required for the first problem in the deck.

REMARK This operation causes the number of cards designated by N1, which follow the operation, to be printed as remark cards.

STOP Causes normal termination of the SMIS program. If an error is encountered in execution the remaining input data is listed until a stop card is encountered.

LOAD Matrix Load:

F1 = Name of matrix to be loaded
N1 = Number of rows in matrix
N2 = Number of columns in matrix

The elements of the matrix are punched row-wise on the data cards which must follow this control card. Normally, each element is punched in a field width of six (12F6.0). If the decimal point is omitted, it is assumed to be at the extreme right of the field. However, the standard format may be superseded by punching any desired format (enclosed in parantheses) in columns 13-30 of the control card.

PRINT Matrix Print:

F1 = Name of matrix to be printed
N1 = Number of remark cards, which follow the control card, to be printed as a label for the matrix.
Column 13 of the control card is used for carriage control -- a one in column 13 will cause the matrix to be printed on a separate input page.

When a matrix is printed, 10 columns normally appear across the page. This number can be changed by specifying in N2 the number of columns of the matrix to be printed across the page.

The normal output specification is a field width of 12 (including blank spaces) and E12.5 conversion. This can be changed by specifying the field width in F3 and the conversion specification in F4, i.e.:

F3 = 8
F4 = F8.3

WRITE * Matrix F1 is written on tape N1 in binary form as two records. The first record contains the number of rows and columns in F1. The second record contains the values of the elements in F1 packed by rows.

REWIND* Tape N1 is rewound to its load point.

READ * Data is read from tape N1 and defined as matrix F1. The data on the tape must be the result of a WRITE command (see above) or must be of the same form on the tape.

PUNCH Matrix F1 is punched onto cards in BCD form. The first card image punched is a LOAD card for the matrix as described above. The data is punched in the standard format (12F6.0) unless superseded by a user specified format punched in columns 13-30 of the control card.

PLOT A printer plot of matrix F1 is made. Matrix F1 is restricted to 9 rows or less, but may have any number of columns. The X axis is scaled from the minimum to the maximum magnitudes of the elements in F1. The elements of each row are represented on the plot by a digit designating the row. The Y axis represents the column number of the element. N1 = the number of remark cards, following the control card, which are to be printed as a label before the plot.

LPLOT Same as PLOT except the log (base 10) of each element is plotted.

OUTAPE * Matrix F1 is written on tape N1 in BCD form. The data is written in two records. The first record contains the name of the matrix, the format used (if specified in col. 13-30), and the number of rows and columns in F1. The second record contains the elements of F1 packed by rows. The standard format (12F6.0) can be superseded by a user specified format punched in columns 13-30 of the control card.

DELETE Matrix Deletion:

F1 = Name of matrix to be eliminated from core storage.

DUPL Matrix Duplication: (F2) = (F1)

Matrix F2 is defined to be identical with matrix F1.

ZERO Matrix formation:

F1 = Name of matrix to be formed
N1 = Number of rows in matrix
N2 = Number of columns in matrix
S1 = Value of all off-diagonal elements
S2 = Value of all diagonal elements

* N1 can only be the numbers 1 or 2

MATRIX OPERATIONS

ADD Matrix Addition (F1) + (F2) = (F1):

Matrix F1 is replaced by the sum of matrices F1 and F2.

SUB Matrix Subtraction (F1) - (F2) = (F1):

Matrix F1 is replaced by matrix F1 minus matrix F2.

SOLVE Solution of Linear Equations (H) (X) = (Q)

F1 = Name of matrix (H). This matrix must be non-singular.
F2 = Name of matrix (Q)

The solution matrix (X) replaces matrix (Q) and is designated by the name (Q). Matrix (H) is destroyed.

MULT Matrix Multiplication (F1) · (F2) = (F3)

The product of F1 times F2 is generated and is defined as F3.

TRANS Matrix Transpose $(F2) = (F1)^T$:

The transpose of F1 is generated and is defined as F2.

INVERT Matrix Inversion $(F1) = (F1)^{-1}$:

Matrix F1 is replaced by the inverse of itself.

SYMINV Inversion of Symmetric Matrix $(F1) = (F1)^{-1}$:

Matrix F1 is replaced by the inverse of itself.

INVEL Inversion of Each Element:
Each element in the matrix designated by F1 is replaced by the reciprocal of the element.

SQREL Square Root of Each Element:
Each element in the matrix designated by F1 is replaced by the square root of itself.

LOG The Log of each element:
Each element in the matrix designated by F1 is replaced by the natural log of the element.

SCALE Scalar Multiplication (F1) = S1 · (F1):
Each element in matrix A is replaced by S1 times the element.

MSCALE Scalar Multiplication (F1) = F2 ·(F1):
Each element in matrix F1 is replaced by F1 times the element where F1 has previously been defined by a 1 by 1 matrix.

STODG Store Row Matrix (F2) on Diagonal of (F1):
F1 = Name of square matrix.
F2 = Name of Row matrix to be stored.

REVDG Remove Row Matrix (F2) from Diagonal of (F1):
F1 = Name of square matrix.
F2 = Name of row matrix composed of the diagonal elements of (F1).

STOSM Store Submatrix (F2) in (F1) (Specified elements of F1 are replaced by elements of F2):
F1 = Name of large matrix.
F2 = Name of submatrix to be stored.
N1 = Row number in large matrix of first element of submatrix.
N2 = column number in large matrix of first element of submatrix.

ADDSM Add Submatrix (F2) to (F1):
F1 = Name of large matrix.
F2 = Name of submatrix to be added into large matrix.
N1 = Row number in large matrix of first element of submatrix.
N2 = Column number in large matrix of first element of submatrix.

RMVSM Remove Submatrix (F2) from (F1) (Specified elements of F1 are duplicated as F2):
F1 = Name of large matrix.
F2 = Name of submatrix to be removed.
N1 = Row number in large matrix of first element of submatrix.
N2 = Column number in large matrix of first element of submatrix.
N3 = Number of rows in submatrix F2.
N4 = Number of columns in submatrix F2.
Matrix F1 is not modified by this operation.

EIGEN Eigenvalues and Eigenvectors:
The eigenvalues and eigenvectors of the system $(A)\cdot(X) = \lambda\cdot (B)\cdot(X)$ are determined where (A) is a positive definite symmetrical matrix and (B) is a diagonal matrix of positive elements stored as a row Matrix.
F1 = Name of matrix (A)
F2 = Name of matrix (B) stored row-wise.
F3 = Name of matrix (X) of eigenvectors stored row-wise.
F4 = Name of row matrix of eigenvalues, λ.
N1 = The number of eigenvectors to be calculated. The ordering of eigenvectors and eigenvalues is determined by the sign of N1 as follows: if N1 is positive, eigenvalues are arranged in descending absolute values; if N1 is negative, eigenvalues are arranged in ascending absolute values.
The eigenvectors are normalized so that $(C)\cdot(B)\cdot(C)^T = (I)$. Matrices (A) and (B) are destroyed in the process. The maximum size system that may be solved is 80.

The following is a list of operations which are useful for earthquake and dynamic response analysis of structural systems:

FUNGN Function Generation:

F1 = Name of matrix which defines a function in terms of line segments X_1, Y_1, X_2, Y_2-----X_n, Y_n. The size of this matrix is nx2.

F2 = Name of matrix to be formed which is composed of the Y-ordinates at equal X-intervals of the function defined by matrix F1.

N1 = Number of Y-ordinates to be generated. The first ordinate will be equal to Y_1.

S1 = X-interval.

RESPON The following Set of Differential Equations is evaluated:

$$\ddot{\underline{X}}_t + 2\lambda\underline{\omega}\,\dot{\underline{X}}_t + \underline{\omega}^2\,\underline{X}_t = \underline{S}\,\underline{P}_t$$

F1 = Name of row matrix containing diagonal values of $\underline{\omega}$, (1xn).

F2 = Name of S matrix, (nx1).

F3 = Name of P matrix, (1xt).

F4 = Name of X matrix, (nxm).

N1 = Output interval: m, which is equal to (t-1)/N1, must be an integer.

S1 = Ratio of damping to critical damping, λ.

S2 = Integration interval which is determined by the P matrix. It should be noted that the generalized mass which is normally present in these equations is assumed to be unity.

Also, this operation may be used to evaluate a set of Differential Equations of the form

$$\ddot{\underline{X}}_t + 2\lambda\underline{\omega}\,\dot{\underline{X}}_t + \omega^2\,\underline{X}_t = \underline{S}\,\underline{P}_t$$

The calling sequence for this equation is the same as above, except for the following terms:

F2 = Name of column matrix which contains the diagonal elements of the S matrix, (nx1). S is a nxn matrix.

F3 = Name of P matrix, (nxt).

ENVEL Envelope Values of Matrix (F1):

The maximum absolute value in each row of matrix (F1) is printed along with its column number times an interval.

F1 = Name of matrix.

S1 = Interval between columns (normally the time increment).

STRUCTURAL ANALYSIS OPERATIONS

The operations given in this section are designed for the automatic formation of stiffness matrices commonly used in structural analysis. The direct stiffness method is the basis for the procedure. It is the responsibility of the user to number the joints and all displacement degrees of freedom associated with the joints of the structure. The operations are intended to be used on small structural systems less than 50 degrees of freedom. A general structural analysis program should be used for large systems. In addition, the use of the operations serve an important educational function in courses on structural analysis. The present version of SMIS contains the following operations:

LOADLM — Loads and prints a one-dimensional integer array to be used as a location matrix
F1 = Number of integer array.
N1 = Number of integers in array.
The card that follows this operation must contain the integers printed in FORMAT (12I6). This array cannot be printed with the PRINT operation since it contains integers.

ADDSTF — Adds one member stiffness matrix to the total stiffness matrix where
F1 = Name of total stiffness matrix.
F2 = Name of member stiffness matrix.
F3 = Integer array which gives the equation numbers in the total stiffness matrix where the terms of the element stiffness matrix are to be added.
(see LOADLM above)

FORCEM — Multiplication of a compacted member force-displacement transformation matrix times a displacement matrix where
F1 = Name of element force-displacement transformation matrix.
F2*= Name of integer location matrix which indicates row numbers in the displacement matrix which are to be multiplied by the columns of matrix F1.
F3 = Name of displacement matrix.
F4 = Name of element force matrix (this can now be printed or plotted).

*If the location matrix F2 contains zero terms, the corresponding operation is skipped.

FORMK Formation of a 2x2 Element Stiffness Matrix (S): Where F1 = Name of 2x2 matrix to be formed.

The card that follows this operation contains the following information (4F12.0):

Columns 1-12 Moment of Inertia of member, I.
13-24 Effective Shear Area of member, $\bar{A}$.
25-36 Length of member L.
37-48 Modulus of Elasticity, E.

The shear modulus, G, is assumed to be equal to E/2.4. The elements of this matrix are calculated from the following equations:

$$S_{11} = S_{22} = \frac{2EI}{L}\left(\frac{2 + \beta}{1 + 2\beta}\right)$$

$$S_{12} = S_{21} = \frac{2EI}{L}\left(\frac{1 - \beta}{1 + 2\beta}\right) \qquad \text{where } \beta = \frac{6EI}{L^2\bar{A}G}$$

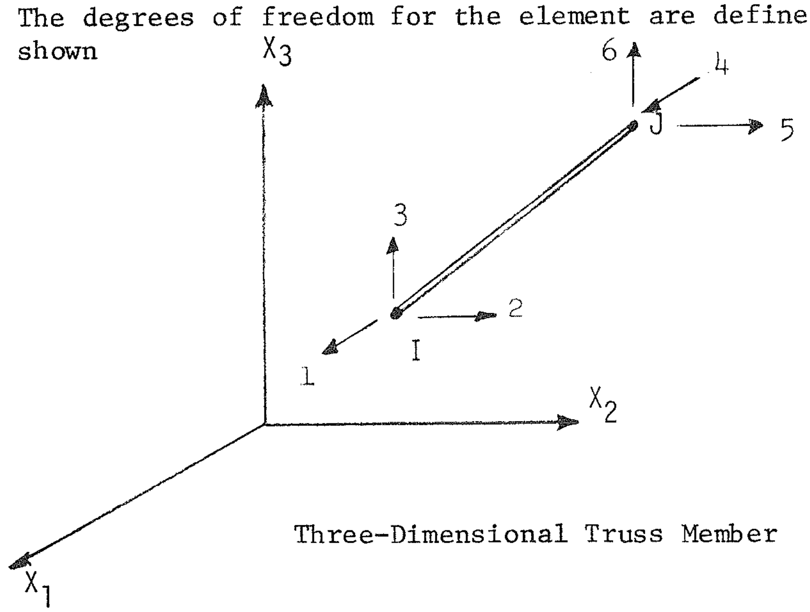
TRUSS Formation of stiffness matrix and force-displacement transformation matrix for one three-dimensional truss element where

F1 = Name of (6x6) element stiffness matrix to be formed.
F2 = Name of (1x6) force-displacement transformation matrix to be formed.
F3 = Name of matrix $\underline{X}$ of joint coordinates stored as a (Nx3) matrix where N is the number of joints in the system. (X(I,1), X(I,2) and X(I,3) are the three-ordinates of joint I).

Second Card (2I6,2F12.0)

Columns 1-6 Joint number I.
7-12 Joint number J
13-24 Modulus of Elasticity, E.
25-36 Cross-sectional Area, A.

The degrees of freedom for the element are defined in the order shown



Three-Dimensional Truss Member

SLOPED The 4x4 stiffness matrix associated with the classical slope-deflection method is formed where
F1 = the name of the 4x4 member stiffness matrix to be formed (shown below)

<u>Second card</u>

Columns	1-12	Moment of inertia, I.
	13-24	Modulus of Elasticity, E.
	25-36	Length of member, L.

For the horizontal or vertical prismatic bending members shown below the classical slope-deflection equations are:

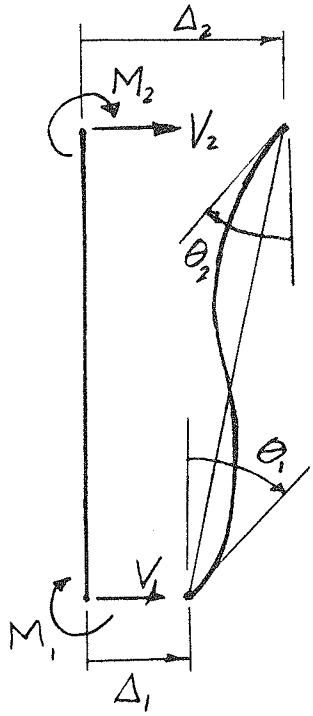
$$M_1 = \frac{4EI}{L}\theta + \frac{2EI}{L}\theta_2 + \frac{6EI}{L^2}(\Delta_1 - \Delta_2)$$

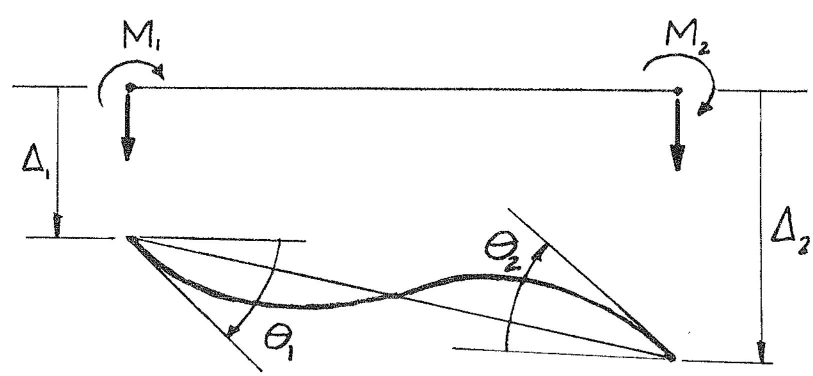
$$M_2 = \frac{2EI}{L}\theta_1 + \frac{4EI}{L}\theta_2 + \frac{6EI}{L^2}(\Delta_1 - \Delta_2)$$

$$V_1 = -V_2 = \frac{M_1 + M_2}{L}$$

Or expanded into matrix form

$$\begin{bmatrix} M_1 \\ M_2 \\ V_1 \\ V_2 \end{bmatrix} = \frac{EI}{L}\begin{bmatrix} 4 & 2 & 6/L & -6/L \\ 2 & 4 & 6/L & -6/L \\ 6/L & 6/L & 12/L^2 & -12/L^2 \\ -6/L & -6/L & -12/L^2 & 12/L^2 \end{bmatrix}\begin{bmatrix} \theta_1 \\ \theta_2 \\ \Delta_1 \\ \Delta_2 \end{bmatrix}$$





A Solution of Equations

Consider the following set of equations

$$AX=B$$

where

$$A = \begin{bmatrix} 2 & 2 & 4 & -2 \\ 1 & 3 & 2 & 1 \\ 3 & 1 & 3 & 1 \\ 1 & 3 & 4 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 10 & 6 \\ 17 & 7 \\ 18 & 8 \\ 27 & 10 \end{bmatrix}$$

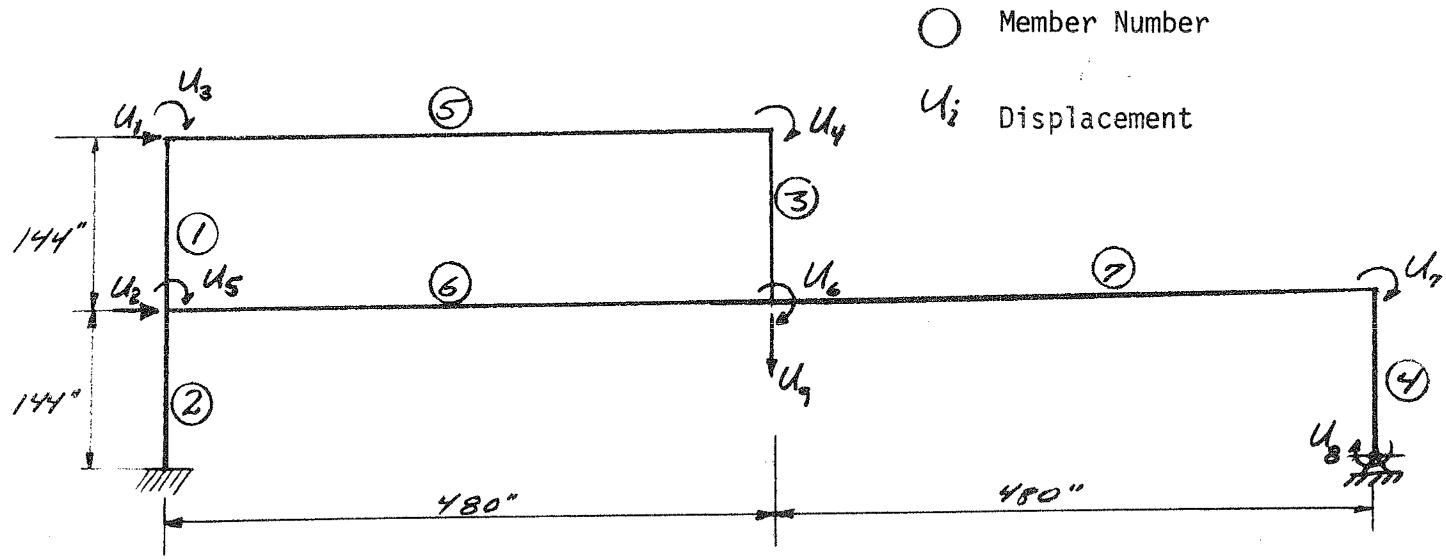
For this case the SMIS input is

```
START
LOAD        A   (4F6.0)                 4      4
    2       2      4      -2
    1       3      2       1
    3       1      3       1
    1       3      4 2.
PRINT       A
LOAD        B   (2F6.0)                 4      2
   10       6
   17       7
   18       8
   27      10
PRINT       B
SOLVE       A      B
PRINT       B
STOP
```

The program output will be

```
LOAD          A    (4F6 .0)
    4 ROWS,        4 COLUMNS

PRINT         A

            1           2           3           4
   1  2.00000E+00 2.00000E+00 4.00000E+00-2.00000E+00
   2  1.00000E+00 3.00000E+00 2.00000E+00 1.00000E+00
   3  3.00000E+00 1.00000E+00 3.00000E+00 1.00000E+00
   4  1.00000E+00 3.00000E+00 4.00000E+00 2.00000E+00

LOAD          B    (2F 6.0)
    4 ROWS,        2 COLUMNS

PRINT         B

            1           2
   1  1.00000E+01 6.00000E+00
   2  1.70000E+01 7.00000E+00
   3  1.80000E+01 8.00000E+00
   4  2.70000E+01 1.00000E+01

SOLVE         A       B

PRINT         B

            1           2
   1  1.00000E+00 1.00000E+00
   2  2.00000E+00 1.00000E+00
   3  3.00000E+00 1.00000E+00
   4  4.00000E+00 1.00000E+00
```

B. Frame Analysis by the Direct Stiffness Method



Joint Loads

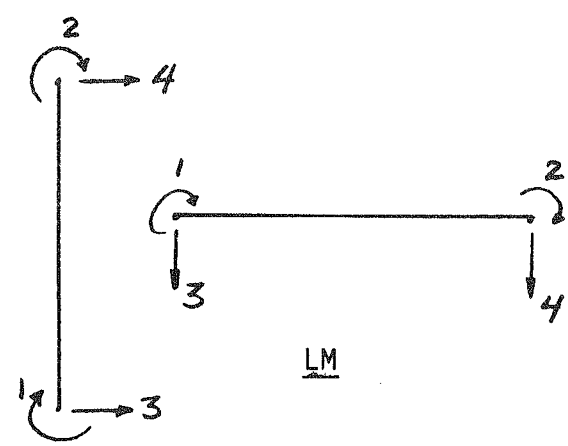
$$\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} 2. & 3. \\ 3. & 6. \end{bmatrix}$$

Member Properties

Columns	12WF58	$I = 476.1$ in^4,	$L = 144"$	$E = 30{,}000$ ksi
Beams	18WF114	$I = 2033.8$ in^4,	$L = 480"$	$E = 30{,}000$ ksi

Relationship between Member Displacements and Joint Displacements

Member \ LM	1	2	3	4
1	5	3	2	1
2	0	5	0	2
3	6	4	2	1
4	8	7	0	2
5	3	4	0	9
6	5	6	0	9
7	6	7	9	0



Frame Analysis Input

```
START
LOADLM     L1                                      4
     5     3     2     1
LOADLM     L2                                      4
     0     5     0     2
LOADLM     L3                                      4
     6     4     2     1
LOADLM     L4                                      4
     8     7     0     2
LOADLM     L5                                      4
     3     4     0     9
LOADLM     L6                                      4
     5     6     0     9
LOADLM     L7                                      4
     6     7     9     0
SLOPED     CS
 476.1          30000.      144.
PRINT      CS
SLOPED     BS
 2033.8       30000.        480.
PRINT      BS
ZERO        S                                      9     9
ADDSTF      S     CS     L1
ADDSTF      S     CS     L2
ADDSTF      S     CS     L3
ADDSTF      S     CS     L4
ADDSTF      S     BS     L5
ADDSTF      S     BS     L6
ADDSTF      S     BS     L7
LOAD        R                                      9     2
 2.    3.    3.     6.

PRINT       R
PRINT       S
SOLVE       S     R
PRINT       R
FORCEM     CS     L1     R     F
PRINT       F
FORCEM     CS     L2     R     F
PRINT       F
FORCEM     CS     L3     R     F
PRINT       F
FORCEM     CS     L4     R     F
PRINT       F
FORCEM     BS     L5     R     F
PRINT       F
FORCEM     BS     L6     R     F
PRINT       F
FORCEM     BS     L7     R     F
PRINT       F
STOP
```

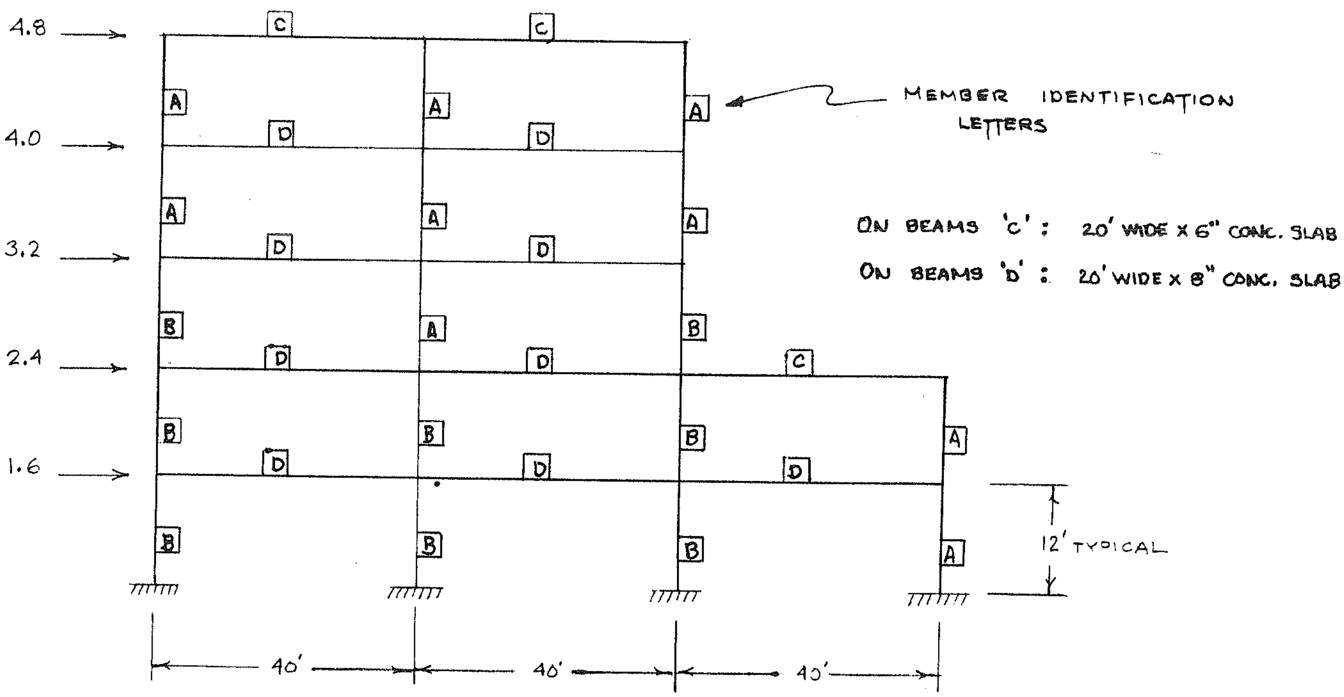
```
LOADLM     L1
    5      3      2      1

LOADLM     L2
    0      5      0      2

LOADLM     L3
    6      4      2      1

LOADLM     L4
    8      7      0      2

LOADLM     L5
    3      4      0      9

LOADLM     L6
    5      6      0      9

LOADLM     L7
    6      7      8      0

SLOPED     CS
I=  .476100E+03     E=  .300000E+05     L=  .144000E+03

PRINT      CS


            1           2           3           4
  1  3.96750E+05 1.98375E+05 4.13281E+03-4.13281E+03
  2  1.98375E+05 3.96750E+05 4.13281E+03-4.13281E+03
  3  4.13281E+03 4.13281E+03 5.74002E+01-5.74002E+01
  4 -4.13281E+03-4.13281E+03-5.74002E+01 5.74002E+01

SLOPED     BS
I=  .203380E+04     E=  .300000E+05     L=  .480000E+03

PRINT      BS


            1           2           3           4
  1  5.08450E+05 2.54225E+05 1.58891E+03-1.58891E+03
  2  2.54225E+05 5.08450E+05 1.58891E+03-1.58891E+03
  3  1.58891E+03 1.58891E+03 6.62044E+00-6.62044E+00
  4 -1.58891E+03-1.58891E+03-6.62044E+00 6.62044E+00

ZERO       S
   9 ROWS,       9 COLUMNS

ADDSTF     S      CS     L1

ADDSTF     S      CS     L2

ADDSTF     S      CS     L3

ADDSTF     S      CS     L4

ADDSTF     S      BS     L5

ADDSTF     S      BS     L6

ADDSTF     S      BS     L7

LOAD       R
   9 ROWS,      2 COLUMNS

PRINT      R


            1           2
  1  2.00000E+00 3.00000E+00
  2  3.00000E+00 6.00000E+00
  3 -0.         -0.
  4 -0.         -0.
  5 -0.         -0.
  6 -0.         -0.
  7 -0.         -0.
  8 -0.         -0.
  9 -0.         -0.
```

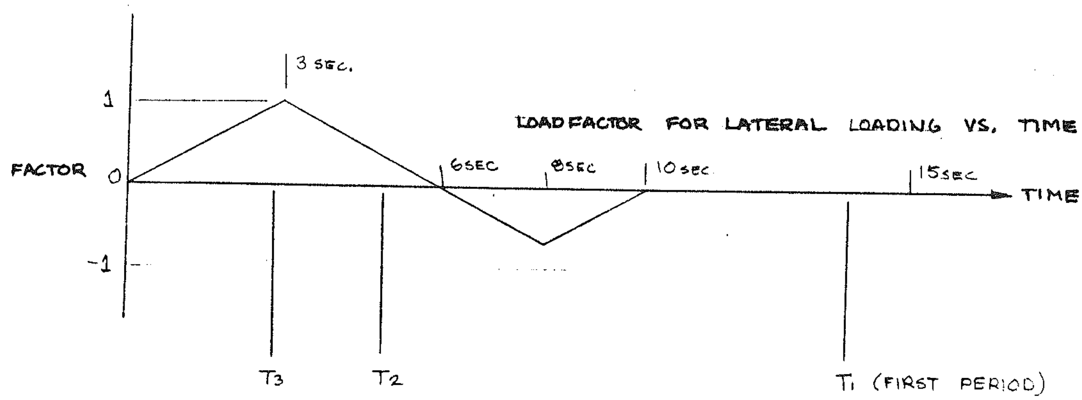
```
FORCEM     CS     L1     R      F

PRINT      F


            1           2
  1  3.34940E+01 7.92041E+01
  2 -5.10620E+01-7.36066E+01
  3 -1.22000E-01 3.53720E-02
  4  1.22000E-01-3.88720E-02

FORCEM     CS     L2     R      F

PRINT      F


            1           2
  1 -3.26029E+02-5.83945E+02
  2 -2.06934E+02-3.85023E+02
  3 -3.70113E+00-6.76367E+00
  4  3.70113E+00 6.76367E+00

FORCEM     CS     L3     R      F

PRINT      F


            1           2
  1 -1.72025E+02-2.81010E+02
  2 -9.84065E+01-1.56587E+02
  3 -1.87800E+00-3.03887E+00
  4  1.87800E+00 3.03887E+00

FORCEM     CS     L4     R      F

PRINT      F


            1           2
  1 -5.45697E-12-3.63798E-12
  2 -1.87037E+02-3.22032E+02
  3 -1.29887E+00-2.23633E+00
  4  1.29887E+00 2.23633E+00

FORCEM     BS     L5     R      F

PRINT      F


            1           2
  1  5.10620E+01 7.36066E+01
  2  9.84065E+01 1.56587E+02
  3  3.11393E-01 4.79571E-01
  4 -3.11393E-01-4.79571E-01

FORCEM     BS     L6     R      F

PRINT      F


            1           2
  1  1.73440E+02 3.05819E+02
  2  1.80771E+01 3.35147E+01
  3  3.98994E-01 7.06946E-01
  4 -3.98994E-01-7.06946E-01

FORCEM     BS     L7     R      F

PRINT      F


            1           2
  1  1.53948E+02 2.47496E+02
  2  1.87037E+02 3.22032E+02
  3  7.10386E-01 1.18652E+00
  4 -7.10386E-01-1.18652E+00
```

```
PRINT      S


            1           2           3           4           5           6           7           8           9
  1  1.14800E+02-1.14800E+02-4.13281E+03-4.13281E+03-4.13281E+03-4.13281E+03-0.         -0.         -0.
  2 -1.14800E+02 2.29601E+02 4.13281E+03 4.13281E+03 0.          4.13281E+03-4.13281E+03-4.13281E+03-0.
  3 -4.13281E+03 4.13281E+03 9.05200E+05 2.54225E+05 1.98375E+05-0.         -0.         -0.         -1.58891E+03
  4 -4.13281E+03 4.13281E+03 2.54225E+05 9.05200E+05-0.          1.98375E+05-0.         -0.         -1.58891E+03
  5 -4.13281E+03 0.          1.98375E+05-0.          1.30195E+06 2.54225E+05-0.         -0.         -1.58891E+03
  6 -4.13281E+03 4.13281E+03-0.          1.98375E+05 2.54225E+05 1.41365E+06 2.54225E+05-0.          0.
  7 -0.         -4.13281E+03-0.         -0.         -0.          2.54225E+05 9.05200E+05 1.98375E+05 1.58891E+03
  8 -0.         -4.13281E+03-0.         -0.         -0.         -0.          1.98375E+05 3.96750E+05-0.
  9 -0.         -0.         -1.58891E+03-1.58891E+03-1.58891E+03 0.          1.58891E+03-0.          1.98613E+01

SOLVE      S      R

PRINT      R


            1           2
  1  1.65592E-01 2.83733E-01
  2  1.07705E-01 1.91847E-01
  3  1.74111E-04 2.57646E-04
  4  3.60342E-04 5.84053E-04
  5  6.00354E-04 1.02796E-03
  6 -1.07687E-05-4.31579E-05
  7  1.19387E-04 2.50034E-04
  8  1.06223E-03 1.87338E-03
  9  8.12336E-02 1.29570E-01
```

C. Frame Response Problem Using Structural and Dynamic Analysis Operations

Determine natural frequencies, characteristic modes, and dynamic response to a lateral shock impulse based on superposition of first three modes with 5% of critical damping.



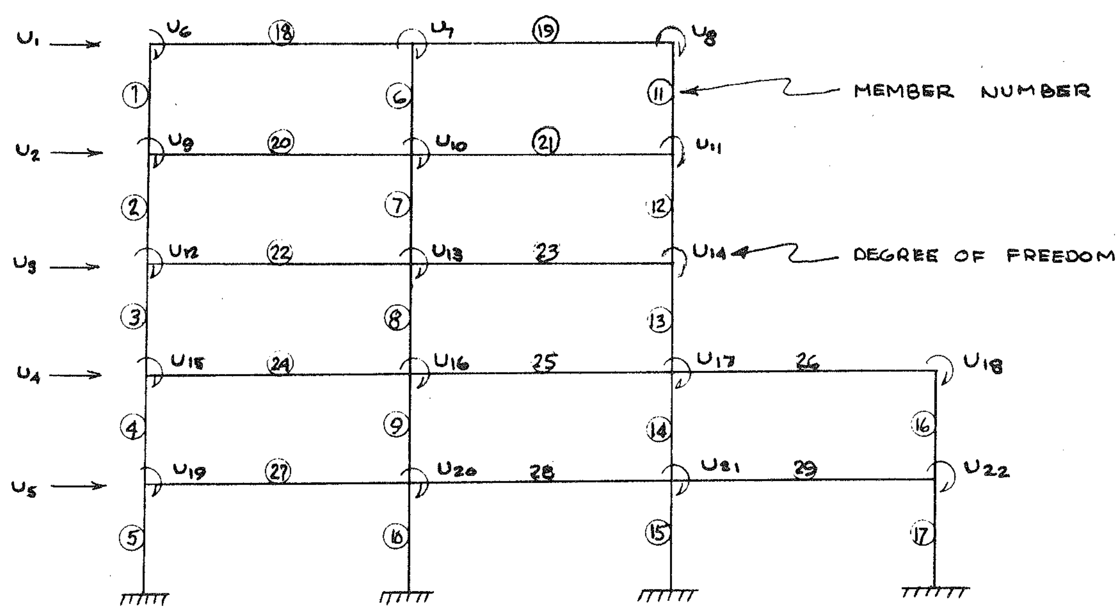


Member Properties

		Type	I(in^4)	A(in^2)	Length(in)
Columns	A	W12x120	1070	35.3	144
	B	W12x161	1540	47.4	144
Beams	C	W24x55	1340	16.2	480
	D	W24x68	1820	20.0	480

Procedure:

1. Determine structural stiffness by direct stiffness method,
2. Partition and condense stiffness matrix to desired degrees of freedom,
3. Solve eigenvalue problem for frequencies and mode shapes,
4. Separate the three primary modes, calculate generalized loads, and solve for time response of uncoupled generalized equations.
5. Superimpose modes using generalized factors and determine structural displacements and base shears.



The total structure stiffness is formed by same method as previous problem. Matrix is then condensed to degrees of freedom U_1 through U_5.

Masses (Mass quantities include beam, column, and slab mass)

$M_1 = 47.165\ k \cdot s^2/in$ (Associated with degree of freedom U_1)

$M_2 = 70.718\ k \cdot s^2/in$

$M_3 = 70.901\ k \cdot s^2/in$

$M_4 = 94.625\ k \cdot s^2/in$

$M_5 = 106.359\ k \cdot s^2/in$

The uncoupled generalized equations are evaluated using the dynamic RESPON operation. The response is calculated for a time length of 140 seconds or approximately 10 first mode periods with an integration interval of .25 seconds (approximately 1/10 of the third mode period) to insure numerical stability. Response is taken as a combination of the first three modes only.

```
LOADLML1                                4
   9      6      2      1
LOADLML2                                4
   12     9      3      2
LOADLML3                                4
   15     12     4      3
LOADLML4                                4
   19     15     5      4
LOADLML5                                4
          19            5
LOADLML6                                4
   10     7      2      1
LOADLML7                                4
   13     10     3      2
LOADLML8                                4
   16     13     4      3
LOADLML9                                4
   20     16     5      4
LOADLML10                               4
          20            5
LOADLML11                               4
   11     8      2      1
LOADLML12                               4
   14     11     3      2
LOADLML13                               4
   17     14     4      3
LOADLML14                               4
   21     17     5      4
LOADLML15                               4
          21            5
LOADLML16                               4
   22     18     5      4
LOADLML17                               4
          22            5
LOADLML18                               4
   6      7
LOADLML19                               4
   7      8
LOADLML20                               4
   9      10
LOADLML21                               4
   10     11
LOADLML22                               4
   12     13
LOADLML23                               4
   13     14
LOADLML24                               4
   15     16
LOADLML25                               4
   16     17
LOADLML26                               4
   17     18
LOADLML27                               4
   19     20
LOADLML28                               4
   20     21
LOADLML29                               4
   21     22
SLOPEDA
1070.       30000.      144.
SLOPEDB
1540.       30000.      144.
SLOPEDC
1340.       30000.      480.
SLOPEDD
1820.       30000.      480.
ZERO  S                                 22    22
ADDSTFS   A     L1
ADDSTFS   A     L2
ADDSTFS   A     L6
ADDSTFS   A     L7
ADDSTFS   A     L8
ADDSTFS   A     L11
ADDSTFS   A     L12
ADDSTFS   A     L16
ADDSTFS   A     L17
ADDSTFS   B     L3
ADDSTFS   B     L4
ADDSTFS   B     L5
ADDSTFS   B     L9
ADDSTFS   B     L10
ADDSTFS   B     L13
ADDSTFS   B     L14
ADDSTFS   B     L15
ADDSTFS   C     L18
ADDSTFS   C     L19
ADDSTFS   C     L26
ADDSTFS   D     L20
ADDSTFS   D     L21
ADDSTFS   D     L22
ADDSTFS   D     L23
ADDSTFS   D     L24
ADDSTFS   D     L25
ADDSTFS   D     L27
ADDSTFS   D     L28
ADDSTFS   D     L29
PRINT S                                 1
      STRUCTURE STIFFNESS MATRIX
RMVSM S     S11                         1     1     5     5
RMVSM S     S12                         1     6     5     17
RMVSM S     S21                         6     1     17    5
RMVSM S     S22                         6     6     17    17
DELETES
INVERTS22
MULT  S12   S22   TEMP1
MULT  TEMP1 S21   TEMP2
SUB   S11   TEMP2
DUPL  S11   S
```

```
PRINT S                                 1
      STIFFNESS MATRIX
LOAD  SMASS                             1     5
47.16570.71870.90194.62506.36
PRINT SMASS                             1
      DIAGONAL OF MASS MATRIX
DUPL  S     A
DUPL  SMASS B
EIGEN A     B     SMODE FREQ            -5
SQREL FREQ
DUPL  FREQ  PERI
INVEL PERI
SCALE PERI                                                      6.2832
PRINT PERI                              1
      PERIODS
PRINT SMODE                             1
      MODES  (EIGENVECTORS STORED ROW-WISE)
RMVSM FREQ  FREQ3                       1     1     1     3
RMVSM SMODE PHIT                        1     1     3     5
TRANS PHIT  PHI
LOAD  TEMP                              1    41
0.    .08333.16667 .25  .33333.41667.5    .58333.66667.75   .83333.91667
1.    .91667.83333.75   .66667.58333.5    .41667.33333.25   .16667.08333
0.    -.0833-.1667-.25  -.3333-.4167-.5   -.5833-.6667-.5833-.5   -.4167
-.3333-.25  -.1667-.0833
ZERO  PT                                1    561
ADDSM PT    TEMP                        1     1
LOAD  FORC                              5     1
4.8   4.0   3.2   2.4   1.6
MULT  PHIT  FORC  GENLD
RESPONFREQ3 GENLD PT    Y               7                       .05   .25
PRINT Y                                 1
GENERALIZED COORDINATE FACTORS Y,
MULT  PHI   Y     V
PRINT V                                 3
      DISPLACEMENT MATRIX
            ROWS 1 TO 5 REPRESENT DOF 1 TO 5,
            COLUMNS REPRESENT RESPONSE AT 1.75 SECOND INTERVALS,
MULT  S     V     R
ZERO  DUM                               1     5                 1.    1.
MULT  DUM   R     SHEAR
PRINT SHEAR                             1
TOTAL BASE SHEAR AT CONSECUTIVE TIME INTERVALS,
ZERO  PLOTM                             2    80
RMVSM V     D1                          1     1     1    80
RMVSM V     D5                          5     1     1    80
ADDSM PLOTM D1                          1     1
ADDSM PLOTM D5                          2     1
PLOT  PLOTM                             5
PLOT OF NODE DISPLACEMENT VS. TIME DUE TO DYNAMIC FORCING OF 10 SEC. DUR
     (Y-AXIS) TIME INTERVAL, EACH INTERVAL IS 1.75 SECONDS
     (X-AXIS) MAGNITUDE OF DISPLACEMENT (INCHES)
            1 INDICATES DISPL. OF FIRST NODE,
            2 INDICATES DISPLACEMENT OF THE FIFTH NODE,
PLOT  SHEAR                             3
PLOT OF TOTAL BASE SHEAR VS. TIME
     (Y-AXIS) TIME INTERVAL, EACH INTERVAL IS 1.75 SECONDS
     (X-AXIS) MAGNITUDE OF FORCE (KIPS)
STOP
```

```
LOADLM L1
    9     6     2     1
LOADLM L2
   12     9     3     2
LOADLM L3
   15    12     4     3
LOADLM L4
   19    15     5     4
LOADLM L5
   -0    19    -0     5
LOADLM L6
   10     7     2     1
LOADLM L7
   13    10     3     2
LOADLM L8
   16    13     4     3
LOADLM L9
   20    16     5     4
LOADLM L10
   -0    20    -0     5
LOADLM L11
   11     8     2     1
LOADLM L12
   14    11     3     2
LOADLM L13
   17    14     4     3
LOADLM L14
   21    17     5     4
LOADLM L15
   -0    21    -0     5
LOADLM L16
   22    18     5     4
LOADLM L17
   -0    22    -0     5
LOADLM L18
    6     7    -0    -0
LOADLM L19
    7     8    -0    -0
LOADLM L20
    9    10    -0    -0
LOADLM L21
   10    11    -0    -0
LOADLM L22
   12    13    -0    -0
LOADLM L23
   13    14    -0    -0
LOADLM L24
   15    16    -0    -0
LOADLM L25
   16    17    -0    -0
LOADLM L26
   17    18    -0    -0
LOADLM L27
   19    20    -0    -0
LOADLM L28
   20    21    -0    -0
LOADLM L29
   21    22    -0    -0
SLOPED A
I=  .107000E+04    E=  .300000E+05    L=  .144000E+03
SLOPED B
I=  .154000E+04    E=  .300000E+05    L=  .144000E+03
SLOPED C
I=  .134000E+04    E=  .300000E+05    L=  .480000E+03
SLOPED D
I=  .182000E+04    E=  .300000E+05    L=  .480000E+03
ZERO   S
   22 ROWS,    22 COLUMNS
ADDSTF S      A      L1
ADDSTF S      A      L2
ADDSTF S      A      L6
ADDSTF S      A      L7
ADDSTF S      A      L8
ADDSTF S      A      L11
ADDSTF S      A      L12
ADDSTF S      A      L16
ADDSTF S      A      L17
ADDSTF S      B      L3
ADDSTF S      B      L4
ADDSTF S      B      L5
ADDSTF S      B      L9
ADDSTF S      B      L10
ADDSTF S      B      L13
ADDSTF S      B      L14
ADDSTF S      B      L15
ADDSTF S      C      L18
ADDSTF S      C      L19
ADDSTF S      C      L26
ADDSTF S      D      L20
ADDSTF S      D      L21
ADDSTF S      D      L22
ADDSTF S      D      L23
ADDSTF S      D      L24
ADDSTF S      D      L25
ADDSTF S      D      L27
ADDSTF S      D      L28
ADDSTF S      D      L29
```

PRINT S

STRUCTURE STIFFNESS MATRIX

	1	2	3	4	5	6	7	8	9	10
1	3.87008E+02	-3.87008E+02	-0.	-0.	-0.	-9.28819E+03	-9.28819E+03	-9.28819E+03	-9.28819E+03	-9.28819E+03
2	-3.87008E+02	7.74016E+02	-3.87008E+02	-0.	-0.	9.28819E+03	9.28819E+03	9.28819E+03	0.	0.
3	-0.	-3.87008E+02	8.87346E+02	-5.00338E+02	-0.	-0.	-0.	-0.	9.28819E+03	9.28819E+03
4	-0.	-0.	-5.00338E+02	1.18634E+03	-6.86005E+02	-0.	-0.	-0.	-0.	-0.
5	-0.	-0.	-0.	-6.86005E+02	1.37201E+03	-0.	-0.	-0.	-0.	-0.
6	-9.28819E+03	9.28819E+03	-0.	-0.	-0.	1.22667E+06	1.67500E+05	-0.	4.45833E+05	-0.
7	-9.28819E+03	9.28819E+03	-0.	-0.	-0.	1.67500E+05	1.56167E+06	1.67500E+05	-0.	4.45833E+05
8	-9.28819E+03	9.28819E+03	-0.	-0.	-0.	-0.	1.67500E+05	1.22667E+06	-0.	-0.
9	-9.28819E+03	0.	9.28819E+03	-0.	-0.	4.45833E+05	-0.	-0.	2.23833E+06	2.27500E+05
10	-9.28819E+03	0.	9.28819E+03	-0.	-0.	-0.	4.45833E+05	-0.	2.27500E+05	2.69333E+06
11	-9.28819E+03	0.	9.28819E+03	-0.	-0.	-0.	-0.	4.45833E+05	-0.	2.27500E+05
12	-0.	-9.28819E+03	-4.07986E+03	1.33681E+04	-0.	-0.	-0.	-0.	4.45833E+05	-0.
13	-0.	-9.28819E+03	0.	9.28819E+03	-0.	-0.	-0.	-0.	-0.	4.45833E+05
14	-0.	-9.28819E+03	-4.07986E+03	1.33681E+04	-0.	-0.	-0.	-0.	-0.	-0.
15	-0.	-0.	-1.33681E+04	0.	1.33681E+04	-0.	-0.	-0.	-0.	-0.
16	-0.	-0.	-9.28819E+03	-4.07986E+03	1.33681E+04	-0.	-0.	-0.	-0.	-0.
17	-0.	-0.	-1.33681E+04	0.	1.33681E+04	-0.	-0.	-0.	-0.	-0.
18	-0.	-0.	-0.	-9.28819E+03	9.28819E+03	-0.	-0.	-0.	-0.	-0.
19	-0.	-0.	-0.	-1.33681E+04	0.	-0.	-0.	-0.	-0.	-0.
20	-0.	-0.	-0.	-1.33681E+04	0.	-0.	-0.	-0.	-0.	-0.
21	-0.	-0.	-0.	-1.33681E+04	0.	-0.	-0.	-0.	-0.	-0.
22	-0.	-0.	-0.	-9.28819E+03	0.	-0.	-0.	-0.	-0.	-0.

	11	12	13	14	15	16	17	18	19	20
1	-9.28819E+03	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.
2	0.	-9.28819E+03	-9.28819E+03	-9.28819E+03	-0.	-0.	-0.	-0.	-0.	-0.
3	9.28819E+03	-4.07986E+03	0.	-4.07986E+03	-1.33681E+04	-9.28819E+03	-1.33681E+04	-0.	-0.	-0.
4	-0.	1.33681E+04	9.28819E+03	1.33681E+04	0.	-4.07986E+03	0.	-9.28819E+03	-1.33681E+04	-1.33681E+04
5	-0.	-0.	-0.	-0.	1.33681E+04	1.33681E+04	1.33681E+04	9.28819E+03	0.	0.
6	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.
7	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.
8	4.45833E+05	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.
9	-0.	4.45833E+05	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.
10	2.27500E+05	-0.	4.45833E+05	-0.	-0.	-0.	-0.	-0.	-0.	-0.
11	2.23833E+06	-0.	-0.	4.45833E+05	-0.	-0.	-0.	-0.	-0.	-0.
12	-0.	2.63000E+06	2.27500E+05	-0.	6.41667E+05	-0.	-0.	-0.	-0.	-0.
13	-0.	2.27500E+05	2.69333E+06	2.27500E+05	-0.	4.45833E+05	-0.	-0.	-0.	-0.
14	4.45833E+05	-0.	2.27500E+05	2.63000E+06	-0.	-0.	6.41667E+05	-0.	-0.	-0.
15	-0.	6.41667E+05	-0.	-0.	3.02167E+06	2.27500E+05	-0.	-0.	6.41667E+05	-0.
16	-0.	-0.	4.45833E+05	-0.	2.27500E+05	3.08500E+06	2.27500E+05	-0.	-0.	6.41667E+05
17	-0.	-0.	-0.	6.41667E+05	-0.	2.27500E+05	3.35667E+06	1.67500E+05	-0.	-0.
18	-0.	-0.	-0.	-0.	-0.	-0.	1.67500E+05	1.22667E+06	-0.	-0.
19	-0.	-0.	-0.	-0.	6.41667E+05	-0.	-0.	-0.	3.02167E+06	2.27500E+05
20	-0.	-0.	-0.	-0.	-0.	6.41667E+05	-0.	-0.	2.27500E+05	3.47667E+06
21	-0.	-0.	-0.	-0.	-0.	-0.	6.41667E+05	-0.	-0.	2.27500E+05
22	-0.	-0.	-0.	-0.	-0.	-0.	-0.	4.45833E+05	-0.	-0.

	21	22
1	-0.	-0.
2	-0.	-0.
3	-0.	-0.
4	-1.33681E+04	-9.28819E+03
5	0.	0.
6	-0.	-0.
7	-0.	-0.
8	-0.	-0.
9	-0.	-0.
10	-0.	-0.
11	-0.	-0.
12	-0.	-0.
13	-0.	-0.
14	-0.	-0.
15	-0.	-0.
16	-0.	-0.
17	6.41667E+05	-0.
18	-0.	4.45833E+05
19	-0.	-0.
20	2.27500E+05	-0.
21	3.47667E+06	2.27500E+05
22	2.27500E+05	2.23833E+06

```
RMVSM  S        S11
 ROW NUMBER     1,  COLUMN NUMBER    1
   5 ROWS,      5 COLUMNS

RMVSM  S        S12
 ROW NUMBER     1,  COLUMN NUMBER    6
   5 ROWS,     17 COLUMNS

RMVSM  S        S21
 ROW NUMBER     6,  COLUMN NUMBER    1
  17 ROWS,      5 COLUMNS

RMVSM  S        S22
 ROW NUMBER     6,  COLUMN NUMBER    6
  17 ROWS,     17 COLUMNS

DELETE S

INVERT S22

MULT   S12     S22     TEMP1

MULT   TEMP1   S21     TEMP2

SUB    S11     TEMP2

DUPL   S11     S

PRINT  S
```

STIFFNESS MATRIX

	1	2	3	4	5
1	1.64283E+02	-2.26689E+02	7.50752E+01	-1.50488E+01	2.89828E+00
2	-2.26689E+02	5.09567E+02	-3.77493E+02	1.12331E+02	-2.15479E+01
3	7.50752E+01	-3.77493E+02	6.39766E+02	-4.47088E+02	1.33304E+02
4	-1.50488E+01	1.12331E+02	-4.47088E+02	7.92093E+02	-6.09655E+02
5	2.89828E+00	-2.15479E+01	1.33304E+02	-6.09655E+02	1.13962E+03

```
LOAD    SMASS
    1 ROWS,       5 COLUMNS

PRINT  SMASS


      DIAGONAL OF MASS MATRIX

             1            2            3            4            5
  1  4.71650E+01 7.07180E+01 7.09010E+01 9.46250E+01 1.06360E+02


DUPL    S        A

DUPL    SMASS    B

EIGEN   A        B        SMODE   FREQ
   -5

SQREL   FREQ

DUPL    FREQ     PERI

INVEL   PERI

SCALE   PERI
 SCALAR=    .6283200E+01

PRINT   PERI


      PERIODS

             1            2            3            4            5
  1  1.38556E+01 4.95476E+00 2.73890E+00 1.81520E+00 1.41420E+00


PRINT   SMODE


      MODES  (EIGENVECTORS STORED ROW-WISE)

             1            2            3            4            5
  1  8.22701E-02 7.19220E-02 5.36630E-02 3.17298E-02 1.20797E-02
  2 -7.89663E-02-2.02248E-02 4.37808E-02 6.57860E-02 3.51687E-02
  3  7.60720E-02-4.83024E-02-5.57439E-02 3.42087E-02 4.66048E-02
  4 -4.48700E-02 6.71687E-02-4.41949E-02-2.55599E-02 6.02182E-02
  5 -1.99459E-02 4.14016E-02-6.51700E-02 5.83945E-02-4.71279E-02


RMVSM  FREQ     FREQ3
 ROW NUMBER     1,   COLUMN NUMBER    1
    1 ROWS,       3 COLUMNS

RMVSM  SMODE    PHIT
 ROW NUMBER     1,   COLUMN NUMBER    1
    3 ROWS,       5 COLUMNS

TRANS  PHIT     PHI

LOAD    TEMP
    1 ROWS,      41 COLUMNS

ZERO    PT
    1 ROWS,     561 COLUMNS

ADDSM  PT       TEMP
 ROW NUMBER     1,   COLUMN NUMBER    1

LOAD    FORC
    5 ROWS,       1 COLUMNS

MULT    PHIT     FORC     GENLD

RESPON FREQ3  GENLD   PT       Y
    7       .05000       .25000

PRINT  Y


GENERALIZED COORDINATE FACTORS Y,

             1            2            3            4            5            6            7            8            9           10
  1  2.68352E-01 1.90549E+00 4.33281E+00 4.74138E+00 1.54834E+00-3.29290E+00-5.97107E+00-5.01914E+00-1.25970E+00 2.93632E+00
  2 -2.32273E-02-8.64209E-02-2.70316E-02 5.36981E-02 2.53380E-02-2.38122E-03-9.44004E-03 1.19774E-02-5.17924E-03-4.10753E-03
  3  1.85243E-02 2.81725E-02 2.90363E-03-6.54827E-03-1.97820E-02 5.62681E-03-5.60267E-04-3.21057E-03 4.02020E-03-2.36049E-03

            11           12           13           14           15           16           17           18           19           20
  1  5.12802E+00 4.21040E+00 9.46719E-01-2.61171E+00-4.40061E+00-3.52825E+00-6.97802E-01 2.31755E+00 3.77352E+00 2.95339E+00
  2  8.54226E-03-5.80409E-03-6.79888E-04 5.39067E-03-5.19995E-03 1.20965E-03 2.89074E-03-4.05399E-03 1.99774E-03 1.12940E-03
  3 -7.31169E-05 1.69597E-03-1.87143E-03 9.61170E-04 1.89987E-04-8.72093E-04 8.58176E-04-3.76265E-04-1.60191E-04 4.38619E-04

            21           22           23           24           25           26           27           28           29           30
  1  5.01056E-01-2.05208E+00-3.23335E+00-2.46939E+00-3.46661E-01 1.81337E+00 2.76343E+00 2.06226E+00 2.26527E-01-1.59942E+00
  2 -2.80992E-03 2.08749E-03 3.36105E-05-1.71396E-03 1.80008E-03-5.41053E-04-8.70314E-04 1.36272E-03-7.53025E-04-2.92754E-04
  3 -3.87447E-04 1.39199E-04 1.07076E-04-2.16436E-04 1.72030E-04-4.69855E-05-6.43324E-05 1.04993E-04-7.49843E-05 1.32002E-05

            31           32           33           34           35           36           37           38           39           40
  1 -2.36860E+00-1.72012E+00-1.34007E-01 1.40824E+00 2.02502E+00 1.43289E+00 6.36469E-02-1.23786E+00-1.73000E+00-1.19199E+00
  2  9.17435E-04-7.42650E-04 5.41480E-05 5.39294E-04-6.18422E-04 2.25707E-04 2.56543E-04-4.54923E-04 2.78718E-04 6.85960E-05
  3  3.62888E-05-5.01331E-05 3.19974E-05-1.99003E-06-1.96121E-05 2.35789E-05-1.33103E-05-1.03410E-06 1.02677E-05-1.09258E-05

            41           42           43           44           45           46           47           48           49           50
  1 -1.09834E-02 1.08641E+00 1.47687E+00 9.90179E-01-2.76248E-02-9.52086E-01-1.25986E+00-8.21291E-01 5.51411E-02 8.33206E-01
  2 -2.97185E-04 2.61658E-04-4.00236E-05-1.67681E-04 2.10925E-04-9.00494E-05-7.35689E-05 1.50816E-04-1.01629E-04-1.29028E-05
  3  5.36118E-06 1.39277E-06-5.24167E-06 4.98658E-06-2.06729E-06-1.06570E-06 2.62012E-06-2.24010E-06 7.47285E-07 6.84245E-07

            51           52           53           54           55           56           57           58           59           60
  1  1.07394E+00 6.80119E-01-7.39693E-02-7.28201E-01-9.14781E-01-5.62256E-01 8.60523E-02 6.35624E-01 7.78633E-01 4.63975E-01
  2  9.54542E-05-9.13326E-05 2.06793E-05 5.14181E-05-7.14363E-05 3.48168E-05 2.03124E-05-4.96425E-05 3.65903E-05 9.01946E-07
  3 -1.28588E-06 9.89194E-07-2.41919E-07-4.01930E-07 6.20679E-07-4.28505E-07 6.12956E-08 2.23393E-07-2.94975E-07 1.81518E-07

            61           62           63           64           65           66           67           68           69           70
  1 -9.29546E-02-5.54145E-01-6.62255E-01-3.82132E-01 9.59303E-02 4.82551E-01 5.62848E-01 3.14070E-01-9.59797E-02-4.19737E-01
  2 -3.03764E-05 3.16565E-05-9.32657E-06-1.55064E-05 2.40276E-05-1.31481E-05-5.29975E-06 1.62193E-05-1.30295E-05 8.50891E-07
  3 -4.06290E-09-1.19445E-07 1.38098E-07-7.48264E-08-9.60236E-09 6.20143E-08-6.36950E-08 2.97814E-08 9.78858E-09-3.14420E-08

            71           72           73           74           75           76           77           78           79           80
  1 -4.78001E-01-2.57352E-01 9.38963E-02 3.64709E-01 4.05634E-01 2.10690E-01-9.03050E-02-3.16567E-01-3.43956E-01-1.71899E-01
  2  9.56735E-06-1.08829E-05 3.91013E-06 4.58054E-06-8.02625E-06 4.87378E-06 1.25680E-06-5.25778E-06 4.59451E-06-6.71072E-07
  3  2.89318E-08-1.12903E-08-6.98072E-09 1.56243E-08-1.29305E-08 3.97173E-09 4.33690E-09-7.62760E-09 5.67766E-09-1.21971E-09
```

```
MULT   PHI     Y       V

PRINT  V

      DISPLACEMENT MATRIX

            ROWS 1 TO 5 REPRESENT DOF 1 TO 5,

            COLUMNS REPRESENT RESPONSE AT 1.75 SECOND INTERVALS,

            1            2            3            4            5            6            7            8            9           10
  1   2.53207E-02 1.65733E-01 3.58816E-01 3.85335E-01 1.23877E-01-2.70292E-01-4.90538E-01-4.14115E-01-1.02921E-01 2.41716E-01
  2   1.88754E-02 1.37434E-01 3.12031E-01 3.40240E-01 1.11803E-01-2.37056E-01-4.29233E-01-3.61074E-01-9.06899E-02 2.11383E-01
  3   1.23510E-02 9.69004E-02 2.31166E-01 2.57152E-01 8.53006E-02-1.77125E-01-3.20807E-01-2.68638E-01-6.80503E-02 1.57523E-01
  4   7.62041E-03 5.57393E-02 1.35800E-01 1.53751E-01 5.01187E-02-1.04447E-01-1.90101E-01-1.58578E-01-4.01733E-02 9.28178E-02
  5   3.28806E-03 2.12915E-02 5.15237E-02 5.88577E-02 1.86727E-02-3.95988E-02-7.24868E-02-6.03581E-02-1.52116E-02 3.52154E-02

           11           12           13           14           15           16           17           18           19           20
  1   4.21203E-01 3.46978E-01 7.77980E-02-2.15218E-01-3.61614E-01-2.90432E-01-5.75713E-02 1.90957E-01 3.10278E-01 2.42920E-01
  2   3.68648E-01 3.02856E-01 6.81941E-02-1.87995E-01-3.16405E-01-2.53741E-01-5.02873E-02 1.66783E-01 2.71366E-01 2.12370E-01
  3   2.75563E-01 2.25594E-01 5.08783E-02-1.39969E-01-2.36388E-01-1.89235E-01-3.73674E-02 1.24210E-01 2.02595E-01 1.58513E-01
  4   1.63270E-01 1.33271E-01 2.99304E-02-8.24813E-02-1.39966E-01-1.11901E-01-2.19216E-02 7.32558E-02 1.19859E-01 9.37998E-02
  5   6.22419E-02 5.07353E-02 1.13250E-02-3.13142E-02-5.33321E-02-4.26183E-02-8.28759E-03 2.78352E-02 4.56458E-02 3.57363E-02

           21           22           23           24           25           26           27           28           29           30
  1   4.14144E-02-1.68979E-01-2.66002E-01-2.03038E-01-2.86489E-02 1.49225E-01 2.27823E-01 1.69562E-01 1.86902E-02-1.31560E-01
  2   3.61125E-02-1.47639E-01-2.32555E-01-1.77558E-01-2.49772E-02 1.30434E-01 1.99132E-01 1.48289E-01 1.63112E-02-1.15028E-01
  3   2.67867E-02-1.10037E-01-1.73515E-01-1.32578E-01-1.85336E-02 9.72895E-02 1.48528E-01 1.10721E-01 1.21273E-02-8.58431E-02
  4   1.57003E-02-6.49699E-02-1.02587E-01-7.84733E-02-1.08752E-02 5.75005E-02 8.77822E-02 6.55282E-02 7.13556E-03-5.07680E-02
  5   5.93573E-03-2.47086E-02-3.90517E-02-2.98998E-02-4.11623E-03 2.18837E-02 3.34082E-02 2.49643E-02 2.70641E-03-1.93302E-02

           31           32           33           34           35           36           37           38           39           40
  1 -1.94935E-01-1.41460E-01-1.10267E-02 1.15813E-01 1.66646E-01 1.17868E-01 5.21497E-03-1.01803E-01-1.42349E-01-9.80717E-02
  2 -1.70375E-01-1.23697E-01-9.64073E-03 1.01273E-01 1.45657E-01 1.03050E-01 4.57307E-03-8.90205E-02-1.24431E-01-8.57314E-02
  3 -1.27068E-01-9.23364E-02-7.19065E-03 7.55939E-02 1.08642E-01 7.69015E-02 3.42746E-03-6.64473E-02-9.28252E-02-6.39623E-02
  4 -7.50936E-02-5.46296E-02-4.24737E-03 4.47185E-02 6.42120E-02 4.54808E-02 2.03592E-03-3.93071E-02-5.48738E-02-3.78175E-02
  5 -2.85780E-02-2.08070E-02-1.61537E-03 1.70300E-02 2.44389E-02 1.73179E-02 7.77238E-04-1.49691E-02-2.08876E-02-1.43970E-02

           41           42           43           44           45           46           47           48           49           50
  1 -8.79731E-04 8.93587E-02 1.21505E-01 8.14758E-02-2.28951E-03-7.83213E-02-1.03643E-01-6.75798E-02 4.54455E-03 6.85490E-02
  2 -7.84197E-04 7.81316E-02 1.06221E-01 7.12188E-02-1.99100E-03-6.84741E-02-9.06101E-02-5.90718E-02 3.96788E-03 5.99261E-02
  3 -6.02712E-04 5.83114E-02 7.92520E-02 5.31283E-02-1.47308E-03-5.10957E-02-6.76110E-02-4.40662E-02 2.95455E-03 4.47117E-02
  4 -3.67868E-04 3.44889E-02 4.68581E-02 3.14073E-02-8.62725E-04-3.02155E-02-3.99797E-02-2.60495E-02 1.74296E-03 2.64366E-02
  5 -1.42878E-04 1.31328E-02 1.78385E-02 1.19554E-02-3.26378E-04-1.15041E-02-1.52212E-02-9.91575E-03 6.62549E-04 1.00645E-02

           51           52           53           54           55           56           57           58           59           60
  1   8.83455E-02 5.59608E-02-6.08712E-03-5.99133E-02-7.52535E-02-4.62596E-02 7.07793E-03 5.22968E-02 6.40554E-02 3.81713E-02
  2   7.72380E-02 4.89173E-02-5.32043E-03-5.23747E-02-6.57915E-02-4.04392E-02 6.18864E-03 4.57164E-02 5.60002E-02 3.33700E-02
  3   5.76350E-02 3.64931E-02-3.96849E-03-3.90752E-02-4.90930E-02-3.01708E-02 4.61871E-03 3.41073E-02 4.17854E-02 2.48983E-02
  4   3.40821E-02 2.15740E-02-2.34568E-03-2.31023E-02-2.90305E-02-1.78380E-02 2.73176E-03 2.01650E-02 2.47083E-02 1.47219E-02
  5   1.29762E-02 8.21246E-03-8.92811E-04-8.79466E-03-1.10528E-02-6.79067E-03 1.04020E-03 7.67641E-03 9.40693E-03 5.60472E-03

           61           62           63           64           65           66           67           68           69           70
  1 -7.64499E-03-4.55921E-02-5.44831E-02-3.14369E-02 7.89030E-03 3.97005E-02 4.63060E-02 2.58373E-02-7.89524E-03-3.45319E-02
  2 -6.68487E-03-3.98559E-02-4.76305E-02-2.74834E-02 6.89901E-03 3.47063E-02 4.04813E-02 2.25883E-02-6.90279E-03-3.01884E-02
  3 -4.98955E-03-2.97357E-02-3.55390E-02-2.05070E-02 5.14896E-03 2.58945E-02 3.02039E-02 1.68547E-02-5.15113E-03-2.25243E-02
  4 -2.95143E-03-1.75808E-02-2.10138E-02-1.21260E-02 3.04543E-03 1.53104E-02 1.78587E-02 9.96645E-03-3.04627E-03-1.33181E-02
  5 -1.12393E-03-6.69280E-03-8.00016E-03-4.61659E-03 1.15965E-03 5.82861E-03 6.79885E-03 3.79445E-03-1.15986E-03-5.07027E-03

           71           72           73           74           75           76           77           78           79           80
  1 -3.93259E-02-2.11880E-02 7.72455E-03 3.00043E-02 3.33722E-02 1.73331E-02-7.42951E-03-2.60436E-02-2.82977E-02-1.41421E-02
  2 -3.43790E-02-1.85234E-02 6.75313E-03 2.62305E-02 2.91741E-02 1.51532E-02-6.49495E-03-2.27680E-02-2.47381E-02-1.23633E-02
  3 -2.56505E-02-1.38215E-02 5.03892E-03 1.95716E-02 2.17671E-02 1.13065E-02-4.84598E-03-1.69881E-02-1.84575E-02-9.22462E-03
  4 -1.51662E-02-8.17277E-03 2.97957E-03 1.15724E-02 1.28701E-02 6.68547E-03-2.86528E-03-1.00449E-02-1.09134E-02-5.45435E-03
  5 -5.77377E-03-3.11153E-03 1.13438E-03 4.40574E-03 4.89965E-03 2.54524E-03-1.09081E-03-3.82422E-03-4.15473E-03-2.07651E-03

MULT    S       V       R

ZERO    DUM
   1 ROWS,      5 COLUMNS

MULT    DUM     R       SHEAR

PRINT   SHEAR


TOTAL BASE SHEAR AT CONSECUTIVE TIME INTERVALS,

            1            2            3            4            5            6            7            8            9           10
  1   1.07600E+00 6.23876E+00 1.49226E+01 1.71638E+01 5.29450E+00-1.14509E+01-2.10793E+01-1.75289E+01-4.39150E+00 1.01933E+01

           11           12           13           14           15           16           17           18           19           20
  1   1.80958E+01 1.47353E+01 3.26891E+00-9.07076E+00-1.54989E+01-1.23818E+01-2.39112E+00 8.06948E+00 1.32590E+01 1.03849E+01

           21           22           23           24           25           26           27           28           29           30
  1   1.71291E+00-7.16876E+00-1.13395E+01-8.68930E+00-1.18916E+00 6.35320E+00 9.69891E+00 7.25405E+00 7.83318E-01-5.61408E+00

           31           32           33           34           35           36           37           38           39           40
  1 -8.29638E+00-6.04469E+00-4.68657E-01 4.94683E+00 7.09531E+00 5.02989E+00 2.26226E-01-4.34817E+00-6.06494E+00-4.18082E+00

           41           42           43           44           45           46           47           48           49           50
  1 -4.21872E-02 3.81443E+00 5.18015E+00 3.47147E+00-9.42695E-02-3.34103E+00-4.42038E+00-2.87917E+00 1.92156E-01 2.92269E+00

           51           52           53           54           55           56           57           58           59           60
  1   3.76850E+00 2.38467E+00-2.59222E-01-2.55384E+00-3.20989E+00-1.97192E+00 3.02126E-01 2.22910E+00 2.73186E+00 1.62761E+00

           61           62           63           64           65           66           67           68           69           70
  1 -3.26466E-01-1.94351E+00-2.32326E+00-1.34070E+00 3.36823E-01 1.69259E+00 1.97437E+00 1.10195E+00-3.36857E-01-1.47240E+00

           71           72           73           74           75           76           77           78           79           80
  1 -1.67667E+00-9.03614E-01 3.29432E-01 1.27944E+00 1.42283E+00 7.39150E-01-3.16768E-01-1.11056E+00-1.20652E+00-6.03018E-01


ZERO    PLOTM
   2 ROWS,      80 COLUMNS

RMVSM  V       D1
 ROW NUMBER    1,  COLUMN NUMBER    1
   1 ROWS,     80 COLUMNS

RMVSM  V       D5
 ROW NUMBER    5,  COLUMN NUMBER    1
   1 ROWS,     80 COLUMNS

ADDSM  PLOTM   D1
 ROW NUMBER    1,  COLUMN NUMBER    1

ADDSM  PLOTM   D5
 ROW NUMBER    2,  COLUMN NUMBER    1
```

PLOT PLOTM

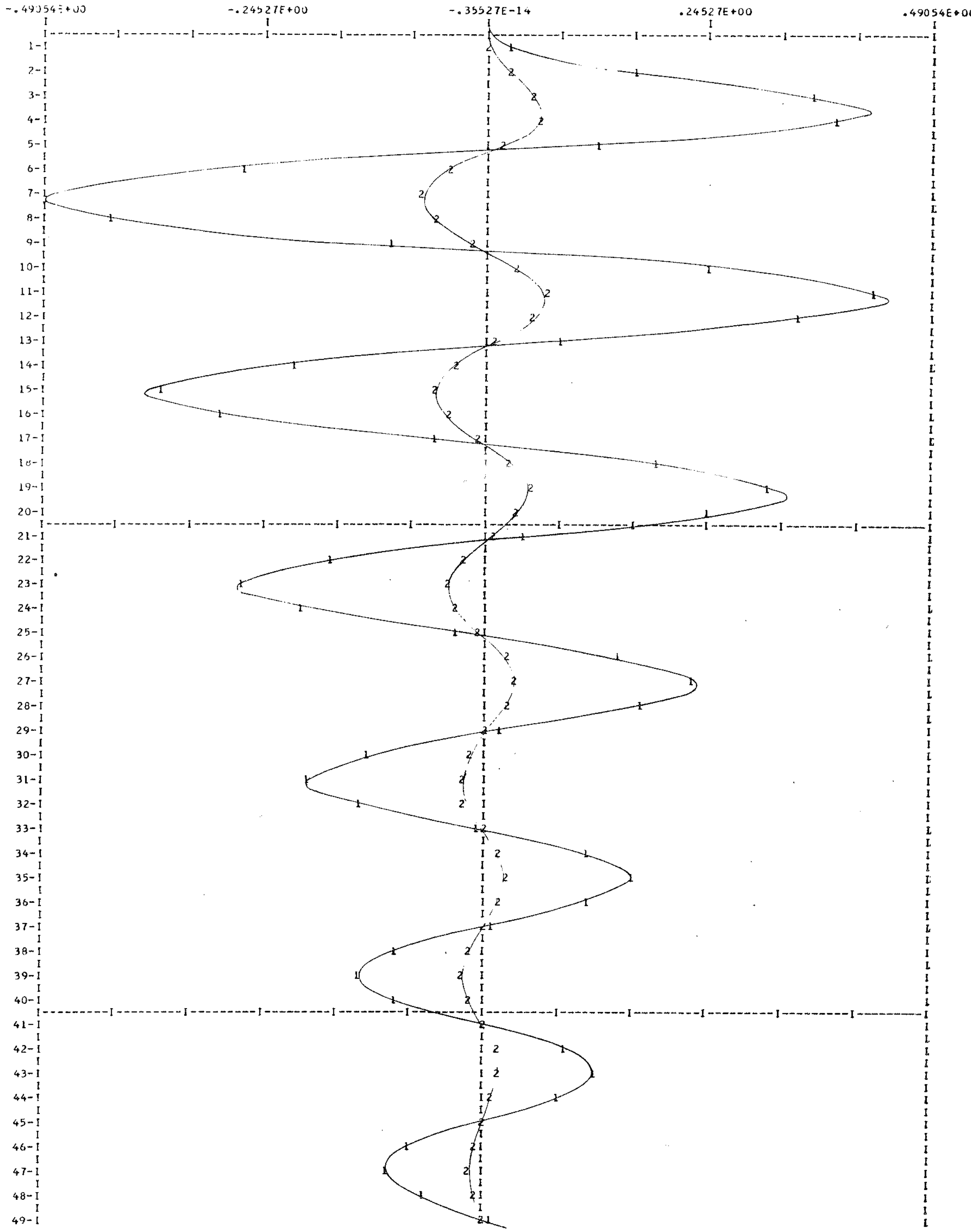
PLOT OF NODE DISPLACEMENT VS. TIME DUE TO DYNAMIC FORCING OF 10 SEC. DUR

(Y-AXIS) TIME INTERVAL, EACH INTERVAL IS 1.75 SECONDS

(X-AXIS) MAGNITUDE OF DISPLACEMENT (INCHES)

1 INDICATES DISPL. OF FIRST NODE,

2 INDICATES DISPLACEMENT OF THE FIFTH NODE,

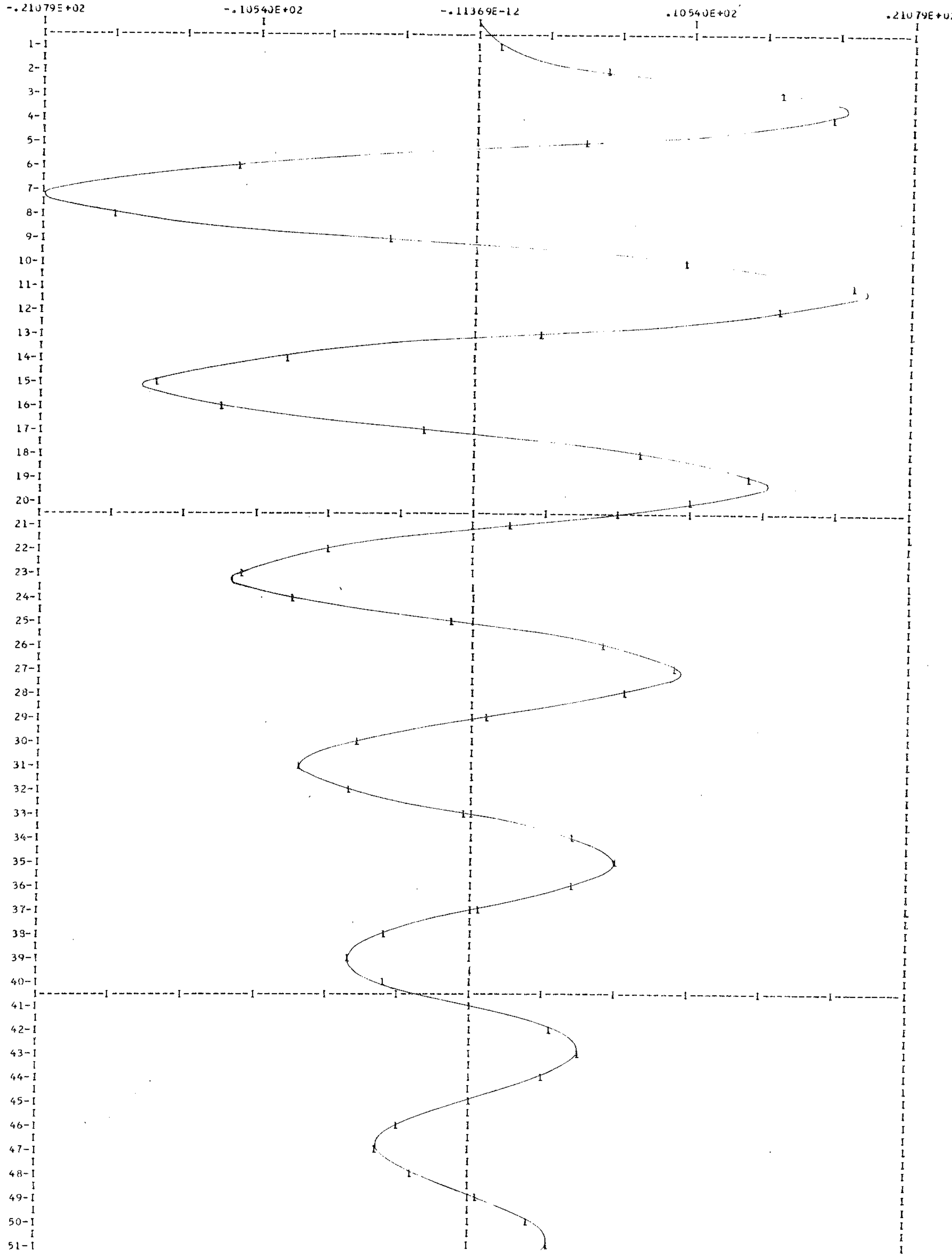


PLOT SHEAR

PLOT OF TOTAL BASE SHEAR VS. TIME

(Y-AXIS) TIME INTERVAL, EACH INTERVAL IS 1.75 SECONDS

(X-AXIS) MAGNITUDE OF FORCE (KIPS)



APPENDIX A

SMIS Fortran IV Program Listing

```
      PROGRAM SMIS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,
     1  PUNCH,TAPE7=PUNCH)
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(40000)
      COMMON/DIC/NUM,NSIZE,XNAME(100),NROW(100),NCOL(100),NN(100)
      DIMENSION OPLIST(50)
      DATA (OPLIST(I),I=1,42)/4HLOAD,5HPRINT,4HZERO,6HDELETE,4HDUPL,3HAD
     1D,3HSUB,5HSCALE,4HMULT,6HINVERT,5HTRANS,5HWRITE,4HREAD,5HSTART,6HR
     1EWIND,6HREMARK,6HSYMINV,5HSTOSM,5HRMVSM,5HADDSM,5HSQREL,5HINVEL,3H
     1LOG,5HSTODG,5HRMVDG,6HMSCALE,5HFORMK,5HEIGEN,6HRESPON,5HFUNGN,5HEN
     1VEL,4HSTOP,4HPLOT,5HPUNCH,6HOUTAPE,5HLPLOT,5HSOLVE,6HLOADLM,
     1 6HADDSTF,6HFORCEM,5HTRUSS,6HSLOPED/
      INTEGER OPER,OPLIST
C
    4 WRITE (6,99)
      NUM = 0
      NSIZE = 1
    5 READ(5,101) OPER,XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2
      IF(OPER-OPLIST(14)) 7,4,7
    7 WRITE(6,100)OPER,XA,XB,XC,XD
      DO 15 I=1,42
      II=I
      IF(OPER-OPLIST(I))15,20,15
   15 CONTINUE
      CALL ERROR(3)
   20 CALL OPGET(II)
      GO TO 5
   99 FORMAT (1H1 )
  100 FORMAT(1H05A7)
  101 FORMAT(5A6,4I6,2F6.0)
      END
```

```
      SUBROUTINE ERROR(N)
C
      COMMON/DIC/NUM,NSIZE,XNAME(100),NROW(100),NCOL(100),NN(100)
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      STAR=4HSTOP
C
      GO TO (1,2,3,4,5,6,7),N
    1 WRITE(6,101)
      GO TO 100
    2 WRITE(6,102)
      GO TO 100
    3 WRITE(6,103)
      GO TO 100
    4 WRITE(6,104)
      GO TO 100
    5 WRITE(6,105)
      GO TO 100
    6 WRITE(6,106)
      GO TO 100
    7 WRITE (6,107)
C
  100 WRITE (6,1000)
   60 READ (5,50) (TEM(I),I=1,12)
      WRITE(6,51) (TEM(I),I=1,12)
      IF (STAR.NE.TEM(1)) GO TO 60
      STOP
   50 FORMAT (12A6)
   51 FORMAT (1X 12A6)
 1000 FORMAT (1H1)
  101 FORMAT (27H MATRICES INCOMPATIBLE       )
  102 FORMAT (27H MATRIX UNDEFINED            )
  103 FORMAT (27H OPERATION UNDEFINED         )
  104 FORMAT (27H MATRIX PREVIOUSLY DEFINED   )
  105 FORMAT (27H STORAGE EXCEEDED            )
  106 FORMAT (27H OVER 100 MATRICES IN CORE   )
  107 FORMAT (27H                             )
      END
```

```
      SUBROUTINE MLIST(XX)
C
      COMMON/DIC/NUM,NSIZE,XNAME(100),NROW(100),NCOL(100),NN(100)
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
C
   30 NUM=NUM+1
      NROW(NUM)=NR
      NCOL(NUM)=NC
      XNAME(NUM)=XX
      NN(NUM)=NSIZE
      NX=NSIZE
      NSIZE=NSIZE+NR*NC
      IF(NUM-101)45,40,45
   40 CALL ERROR(6)
      GO TO 60
   45 IF(40001-NSIZE)50,60,60
   50 CALL ERROR(5)
   60 RETURN
      END
```

```
      SUBROUTINE MFIND(XX)
C
C     SUBROUTINE TO FIND LOCATION AND SIZE OF MATRIX XX
      COMMON/DIC/NUM,NSIZE,XNAME(100),NROW(100),NCOL(100),NN(100)
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      DO 20 I=1,NUM
      IF(XNAME(I)-XX)20,30,20
   20 CONTINUE
      CALL ERROR(2)
      GO TO 40
   30 NR=NROW(I)
      NC=NCOL(I)
      NX=NN(I)
   40 RETURN
      END
```

```
      SUBROUTINE OPGET(I)
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
C
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
     1 23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42),I
    1 CALL MLOAD
      GO TO 200
    2 CALL MPRINT
      GO TO 200
    3 CALL ZERO
      GO TO 200
    4 CALL DELETE(XA)
      GO TO 200
    5 CALL DUPLIC
      GO TO 200
    6 CALL ADDSUB(1.)
      GO TO 200
    7 CALL ADDSUB(-1.)
      GO TO 200
    8 CALL SCALE(1)
      GO TO 200
    9 CALL MULT
      GO TO 200
   10 CALL INVERT(1)
      GO TO 200
   11 CALL TRANS
      GO TO 200
   12 CALL TAPEOP(3)
      GO TO 200
   13 CALL TAPEOP(2)
   14 GO TO 200
   15 CALL TAPEOP(1)
      GO TO 200
   16 CALL REMARK
      GO TO 200
   17 CALL INVERT(2)
      GO TO 200
   18 CALL SUBMOP(1)
      GO TO 200
   19 CALL SUBMOP(2)
      GO TO 200
   20 CALL SUBMOP(3)
      GO TO 200
   21 CALL SCALE(4)
      GO TO 200
   22 CALL SCALE(3)
      GO TO 200
   23 CALL SCALE(5)
      GO TO 200
   24 CALL STODG
      GO TO 200
   25 CALL RMVDG
      GO TO 200
   26 CALL SCALE(2)
      GO TO 200
   27 CALL FORMK
      GO TO 200
   28 CALL EIGEN
      GO TO 200
   29 CALL RESPON
      GO TO 200
   30 CALL FUNGN
      GO TO 200
   31 CALL ENVEL
      GO TO 200
   32 STOP
      GO TO 200
   33 CALL PLOT(1)
      GO TO 200
   34 CALL TAPEOP(4)
      GO TO 200
   35 CALL TAPEOP(5)
      GO TO 200
   36 CALL PLOT(2)
      GO TO 200
   37 CALL SOLVE
      GO TO 200
   38 CALL LOADLM
      GO TO 200
   39 CALL ADDSTF
      GO TO 200
   40 CALL FORCEM
      GO TO 200
   41 CALL TRUSS
      GO TO 200
   42 CALL SLOPED
  200 RETURN
      END
```

```
      SUBROUTINE DELETE(XX)
C
      COMMON/DIC/NUM,NSIZE,XNAME(100),NROW(100),NCOL(100),NN(100)
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      IF (NUM) 10,70,10
   10 DO 20 I=1,NUM
      IF(XNAME(I)-XX)20,30,20
   20 CONTINUE
      RETURN
   30 NX=NN(I)
      NUM=NUM-1
      NS=NROW(I)*NCOL(I)
      NSIZE=NSIZE-NS
      IF (I.GT.NUM) GO TO 70
      DO 50 J=I,NUM
      XNAME(J)=XNAME(J+1)
      NROW(J)=NROW(J+1)
      NCOL(J)=NCOL(J+1)
   50 NN(J)=NN(J+1)-NS
      DO 60 J=NX,NSIZE
      I=J+NS
   60 A(J)=A(I)
   70 RETURN
      END
```

```
      SUBROUTINE REMARK
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      DO 10 I=1,NR
      READ(5,105)(TEM(I),I=1,12)
   10 WRITE(6,106)(TEM(I),I=1,12)
      RETURN
  105 FORMAT(12A6)
  106 FORMAT(1H012A6)
      END
```

```
      SUBROUTINE MLOAD
C
C     SUBROUTINE FOR MATRIX LOAD
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      WRITE(6,102)NR,NC
      CALL DELETE(XA)
      CALL MLIST(XA)
      NH=NX+NR*NC-1
      IF (XB-XC) 20,10,20
   10 READ(5,100)(A(I),I=NX,NH)
      GO TO 15
   20 READ(5,XB)(A(I),I=NX,NH)
   15 RETURN
  100 FORMAT (12F6.0)
  102 FORMAT(1I6,6H ROWS,1I6,8H COLUMNS)
      END
```

```
      SUBROUTINE MPRINT
C
C     SUBROUTINE TO PRINT A MATRIX
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      DIMENSION VF100(6), VF101(4)
      DATA (VF100(I),I=1,6)/6H(1I4,1 ,2HX,, 2H1P, 2H20, 5HE12.5, 1H) /
      DATA (VF101(I),I=1,4)/6H(1H0,2, 2H0I, 2H12, 1H) /
      DATA XBLANK/6H      /
      VF100(3) = 2H1P
      VF100(5) = 6HE12.5
      VF101(3) = 6H12
      IF((XC.EQ.XBLANK).OR.(XD.EQ.XBLANK)) GO TO 4
      VF101(3) = XC
      VF100(5) = XD
      VF100(3) = XBLANK
    4 NCC = 10
      IF (NC.NE.0) NCC=NC
      WRITE (6,103) XB
      IF(NR)10,10,5
    5 CALL REMARK
   10 CALL MFIND(XA)
      DO 50 M=1,NC,NCC
      NN=NCC-1
      MH=M+NN
      IF(MH.GT.NC) GO TO 20
      GO TO 30
   20 MH=NC
      NN=MH-M
   30 WRITE(6,VF101) (N,N=M,MH)
      DO 40 N=1,NR
      NL=NX+NC*(N-1)
      NH=NL+NN
   40 WRITE(6,VF100) N,(A(I),I=NL,NH)
   50 NX=NX+NCC
      WRITE (6,103) XB
      RETURN
* 100 FORMAT (1I4,1X,1P10E12.5)
* 101 FORMAT (1H0,10I12)
  103 FORMAT (1A6)
      END
```

```
      SUBROUTINE ZERO
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      COMMON/DIC/NUM,NSIZE,XNAME(100),NROW(100),NCOL(100),NN(100)
      WRITE(6,102)NR,NC
      CALL DELETE(XA)
      CALL MLIST(XA)
      J=NX
      DO 135 I=NX,NSIZE
      A(I)=SCAL1
      IF(J-I) 135,130,135
  130 A(I)=SCAL2
      J=J+NC+1
  135 CONTINUE
      RETURN
  102 FORMAT(1I6,6H ROWS,1I6,8H COLUMNS)
      END
```

```
      SUBROUTINE DUPLIC
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL DELETE(XB)
      CALL MFIND(XA)
      NA=NX-1
      NS=NR*NC
      CALL MLIST(XB)
      NB=NX-1
      DO 155 I=1,NS
      J=I+NB
      K=I+NA
  155 A(J)=A(K)
      RETURN
      END
```

```
      SUBROUTINE PLOT(N)
C
C     N=1  LINEAR PLOT
C     N=2  LOGARITHMIC PLOT
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      DIMENSION PNUM(10),YLINE(121),Y(5)
      DATA(PNUM(I),I=1,9)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
      DATA BLANK/1H /
      DATA AI/1HI/
      WRITE(6,2003)
      IF(NR) 1,1,5
    5 CALL REMARK
      WRITE(6,2006)
    1 IF(N.LT.2) GO TO 2
C
C     FOR LOG PLOT SET-UP DUPL MATRIX AND CONVERT TO LOG10
C
      XB=6H/$.*=(
      CALL DUPLIC
      XA=XB
C
    2 CALL MFIND(XA)
      NA=NX
      NB=NX-1+NR*NC
      IF(N.LT.2) GO TO 20
C
      DO 15 I=NA,NB
      IF(A(I).LE.0.0) GO TO 500
   15 A(I)=ALOG10(A(I))
C
C     CALCULATE RANGE NUMBERS.
C
   20 XL=A(NX)
      XS=A(NX)
      DO 100 I=NA,NB
      IF(A(I).LT.XS) XS=A(I)
      IF(A(I).GT.XL) XL=A(I)
  100 CONTINUE
C
C     SET-UP Y AXIS,CALC PLOTTING INTERVAL
C
      IF(N.GT.1) GO TO 50
C
C     AUTO-RANGING FOR LINEAR PLOT
C
      FINT=(XL-XS)/120.
      FA=XS-1.5*FINT
      IZ=(-FA/FINT)
      IF(IZ.LT.1) IZ=1
      IF(IZ.GT.121) IZ=121
      JZ=IZ-1-(((IZ-1)/10)*10)
      IF(JZ) 30,30,21
   21 IZ=IZ-JZ
      IF(JZ-5) 22,22,23
   22 IF(IZ.EQ.1) GO TO 23
      YP=(IZ-1)
      FINT=(0.0-XS)/YP
      XL=XS+120.0*FINT
      FA=XS-1.5*FINT
      GO TO 30
   23 IF(IZ.EQ.111) GO TO 22
      IZ=IZ+10
      YP=(121.0-IZ)
      FINT=XL/YP
      XS=XL-120.0*FINT
      FA=XS-1.5*FINT
C
C     SET-UP Y AXIS
C
   30 YI=30.0*FINT
      DO 31 I=1,5
   31 Y(I)=XS+(I-1)*YI
      GO TO 95
   50 CONTINUE
C
C     AUTO-RANGING FOR LOG PLOT
C
      IF(XS.LT.0.0) GO TO 60
      IXS=XS
      XS=IXS
      GO TO 70
   60 IXS=(ABS(XS-0.9999))
      XS=(-IXS)
   70 IF(XL.LT.0.0) GO TO 80
      IXL=(XL+0.9999)
      XL=IXL
      GO TO 90
   80 IXL=(ABS(XL))
      XL=(-IXL)
   90 FINT=(XL-XS)/120.0
      FA=XS-1.5*FINT
      IZ=(-FA/FINT)
      IF(IZ.LT.1) IZ=1
      IF(IZ.GT.121) IZ=121
C
C     SET-UP Y AXIS
C
      YI=30.0*FINT
      DO 91 I=1,5
      P=XS+(I-1)*YI
   91 Y(I)=10.0**P
C
C     OUTLINE GRAPH
C
   95 WRITE (6,2001) Y
      WRITE (6,2002)
      DO 40 I=1,121
   40 YLINE(I)=BLANK
      YLINE(1)=AI
      YLINE(121)=AI
      YLINE(IZ)=AI
C
```

```
      SUBROUTINE ADDSUB(XX)
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
   15 CALL MFIND(XA)
      NA=NX-1
      NNR=NR
      NNC=NC
      CALL MFIND(XB)
      IF(NR-NNR)25,20,25
   20 IF(NC-NNC)25,30,25
   25 CALL ERROR(1)
      RETURN
   30 NS=NR*NC
      DO 35 I=1,NS
      J=I+NA
      K=I+NX-1
   35 A(J)=A(J)+A(K)*XX
      RETURN
      END
```

```
      SUBROUTINE MULT
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
C
      CALL DELETE(XC)
      CALL MFIND(XA)
      NA=NX
      NRA=NR
      NCA=NC
      CALL MFIND(XB)
      NB=NX
      NCB=NC
      IF (NCA.EQ.NR) GO TO 50
      CALL ERROR(1)
      GO TO 1000
   50 NR=NRA
      CALL MLIST(XC)
      II=NX
      DO 200 I=1,NRA
      DO 200 J=1,NC
      A(II)=0.0
      JJ=NA+(I-1)*NCA
      KK=NB+J-1
      DO 100 K=1,NCA
      A(II)=A(II)+A(JJ)*A(KK)
      KK=KK+NC
  100 JJ=JJ+1
  200 II=II+1
 1000 RETURN
C
      END
```

```
      SUBROUTINE TRANS
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL DELETE(XB)
      CALL MFIND(XA)
      NA=NX
      NRA=NR
      NR=NC
      NC=NRA
      CALL MLIST(XB)
      K=NX
      DO 10 I=1,NR
      L=NA+I-1
      DO 10 J=1,NC
      A(K)=A(L)
      L=L+NR
   10 K=K+1
      RETURN
      END
```

```
      SUBROUTINE INVERT(N)
C
C     N=1 INVERSION OF GENERAL MATRIX
C     N=2 INVERSION OF SYMMETRIC MATRIX
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL MFIND(XA)
      NA=NX
      IF (NR.EQ.NC) GO TO 50
      CALL ERROR(1)
      GO TO 15
   50 GO TO (100,200) N
  100 CALL INVER(A(NA),NR,TEM(1),TEM(241))
      GO TO 15
  200 CALL SYMIN(A(NA),NR,TEM(1),TEM(241))
   15 RETURN
      END
```

```
      SUBROUTINE INVER(A,NR,W1,W3)
      DIMENSION A(1),W1(1),W3(1)
      DO 10 I=1,NR
   10 W1(I)=-I
      DO 65 I=1,NR
C
C     LOCATE LARGEST ELEMENT
C
      D=0.0
      DO 35 L=1,NR
      IF(W1(L))15,15,35
   15 J=L
      DO 30 K=1,NR
      IF(W1(K))20,20,30
   20 IF(ABS(D)-ABS(A(J)))25,25,30
   25 LD=L
      KD=K
      D=A(J)
   30 J=J+NR
   35 CONTINUE
C
C     INTERCHANGE OF COLUMNS
C
      TEMP=-W1(LD)
      W1(LD)=W1(KD)
      W1(KD)=TEMP
      L=LD
      K=KD
      DO 40 J=1,NR
      W3(J)=A(L)
      A(L)=A(K)
      A(K)=W3(J)
      L=L+NR
   40 K=K+NR
C
C     DIVIDE ROW BY LARGEST ELEMENT
C
      N=(KD-1)*NR+1
      NH=N+NR-1
      DF=1./D
      DO 45 K=N,NH
   45 A(K)=-A(K)*DF
C
C     REDUCE REMAINING ROWS AND COLUMNS
C
      L=1
      DO 60 J=1,NR
      IF(J-KD)55,50,55
   50 L=L+NR
      GO TO 60
   55 DO 58 K=N,NH
      A(L)=A(L)+W3(J)*A(K)
   58 L=L+1
   60 CONTINUE
C
C     REDUCE COLUMN
C
      W3(KD)=1.0
      J=KD
      DO 65 K=1,NR
      A(J)=W3(K)*DF
   65 J=J+NR
C
C     INTERCHANGE ROWS
C
      DO 80 I=1,NR
      L=0
   70 L=L+1
      IF(W1(L)-I)70,75,70
   75 K=(L-1)*NR+1
      J=(I-1)*NR+1
      W1(L)=W1(I)
      W1(I)=I
      DO 80 LL=1,NR
      TEMP=A(K)
      A(K)=A(J)
      A(J)=TEMP
      J=J+1
   80 K=K+1
      RETURN
      END
```

```
      SUBROUTINE SYMIN(AA,NR,W1,W3)
C
      DIMENSION AA(1),W1(1),W3(1)
      DO 35 I=1,NR
      N=(I-1)*NR
      DO 15 J=1,NR
      K=N+J
   15 W1(J)=AA(K)
      D=1./W1(I)
      DO 20 J=1,NR
   20 W3(J)=-W1(J)*D
      L=1
      DO 30 J=1,NR
      M=L
      DO 25 K=J,NR
      AA(L)=AA(L)+W1(J)*W3(K)
      AA(M)=AA(L)
      M=M+NR
   25 L=L+1
   30 L=L+J
      W3(I)=-D
      M=I
      DO 35 J=1,NR
      K=N+J
      AA(K)=W3(J)
      AA(M)=W3(J)
   35 M=M+NR
      NS=NR*NR
      DO 40 J=1,NS
   40 AA(J)=-AA(J)
      RETURN
      END
```

```
      SUBROUTINE SCALE(K)
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
C
C     K=1 SCALAR MULIPLICATION
C     K=2 B(1,1) IS THE SCALAR
C     K=3 INVERSION OF EACH ELEMENT
C     K=4 SQUARE ROOT OF EACH ELEMENT
C     K=5 LOG OF EACH ELEMENT
C
      CALL MFIND(XA)
      NA=NX
      NB=NX+NR*NC-1
      GO TO (100,200,300,400,500) K
C
  200 CALL MFIND(XB)
      SCAL1=A(NX)
  100 WRITE (6,103) SCAL1
      DO 150 I=NA,NB
  150 A(I)=SCAL1*A(I)
      GO TO 1000
C
  300 DO 350 I=NA,NB
  350 A(I)=1.0/A(I)
      GO TO 1000
C
  400 DO 450 I=NA,NB
  450 A(I)=SQRT(A(I))
      GO TO 1000
C
  500 DO 550 I=NA,NB
  550 A(I)=ALOG(A(I))
C
 1000 RETURN
  103 FORMAT (9H  SCALAR= E15.7)
      END
```

```
      SUBROUTINE RMVDG
C
C     REMOVES ROW(B) FROM DIAGONAL OF (A)
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL DELETE(XB)
      CALL MFIND(XA)
      K=NX
      NR=1
      CALL MLIST(XB)
      IL=NX
      IH=NX-1+NC
      DO 10 I=IL,IH
      A(I)=A(K)
   10 K=K+NC+1
      RETURN
      END
```

```
      SUBROUTINE STODG
C
C     STORES ROW (B) ON DIAGONAL OF (A)
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL MFIND(XA)
      K=NX
      CALL MFIND(XB)
      IL=NX
      IH=NX-1+NC
      DO 10 I=IL,IH
      A(K)=A(I)
   10 K=K+NC+1
      RETURN
      END
```

```
      SUBROUTINE SUBMOP(N)
C
C     N=1 STORE SUBMATRIX
C     N=2 REMOVE SUBMATRIX
C     N=3 ADD IN SUBMATRIX
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
C
      J=NR
      K=NC
   30 WRITE(6,104)J,K
      IF (N.NE.2) GO TO 100
      CALL DELETE(XB)
      NC=NNC
      NR=NNR
      WRITE(6,102)NR,NC
      CALL MLIST(XB)
C
  100 CALL MFIND(XA)
      NA=NX
      NRA=NR
      NCA=NC
      CALL MFIND(XB)
C
      JJ=NX
      DO 500 M=1,NR
      II=NA-1+K+(J+M-2)*NCA
      DO 500 L=1,NC
      GO TO (450,460,470) N
  450 A(II)=A(JJ)
      GO TO 480
  460 A(JJ)=A(II)
      GO TO 480
  470 A(II)=A(II)+A(JJ)
  480 JJ=JJ+1
  500 II=II+1
C
      RETURN
C
  102 FORMAT(1I6,6H ROWS,1I6,8H COLUMNS)
  104 FORMAT(12H  ROW NUMBER I4,16H,  COLUMN NUMBER I4)
      END
```

```
      SUBROUTINE EIGEN
C
C     SUBROUTINE TO SOLVE THE EQUATION (A)(X)=E(B)(X)
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL DELETE(XC)
      CALL DELETE(XD)
      M=NR
      CALL MFIND(XA)
      NA=NX
      IF (NR-NC) 10,20,10
   10 CALL ERROR(1)
   20 N=NR
      NR=1
      CALL MLIST(XD)
      CALL MLIST(XC)
      NE=NX
      CALL MFIND(XB)
      ND=NX
      NR = IABS(M)
      NB=NX
      WRITE (6,100) M
      CALL HQRW(N,N,M,A(NA),A(ND),A(NE),TEM(1),TEM(81),TEM(161),TEM(241)
     1,TEM(321),TEM(401),A(NB))
      RETURN
  100 FORMAT(I6)
      END
```

```
      SUBROUTINE HQRW(N,NM,M,G,E,V,A,B,P,W,Q,INTV,XM)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SUBROUTINE TO COMPUTE EIGENVALUES AND EIGENVECTORS OF A
C     SYMMETRIC REAL MATRIX STORED AS A TWO-DIMENSIONAL ARRAY.
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DIMENSION  G(NM,1), E(1), V(NM,1), A(1), B(1), P(1), W(1), Q(1),
     1XM(1)
      REAL LAMBDA
      LOGICAL INTV(1)
      IF (N.LE.0.OR.N.GT.NM)  GO TO 1000
      DO 2 I=1,N
    2 XM(I)=1./(SQRT(XM(I)))
      DO 3 I=1,N
      DO 3 J=1,N
    3 G(I,J)=XM(I)*G(I,J)*XM(J)
      PRECS = 1.0E-15
      BASE = 2.0
      ILIM = 972
      HOV = BASE**486
      B(1) = 0.
      SQRT2 = SQRT(2.)
      N1 = N - 1
      DO 100  I = 1,N
  100 E(I) = G(I,I)
      IF  (N-2)  900,280,110
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     TRI-DIAGONALIZE MATRIX G BY HOUSEHOLDER*S PROCEDURE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
  110 DO 250  K = 2,N1
      K1 = K - 1
      KJ = K + 1
      Y = G(K,K1)
      SUM = 0.0
      DO 120  I = KJ,N
  120 SUM = SUM + G(I,K1)**2
      IF (SUM.EQ.0.)  GO TO 230
      S = SQRT(SUM+Y**2)
      B(K) = SIGN(S,-Y)
      W(K) = SQRT(1.+ABS(Y)/S)
      X = SIGN(1./(S*W(K)),Y)
      DO 150  I = K,N
      IF (I.GT.K)  W(I) = X*G(I,K1)
      P(I) = 0.
  150 G(I,K1) = W(I)
      DO 180  I = K,N
      Y = W(I)
      IF (Y.EQ.0.)  GO TO 180
      I1 = I + 1
      DO 160  J = K,I
  160 P(J) = P(J) + Y*G(I,J)
      IF (I1.GT.N)  GO TO 180
      DO 170  J = I1,N
  170 P(J) = P(J) + Y*G(J,I)
  180 CONTINUE
  190 X = 0.
      DO 200  J = K,N
  200 X = X + W(J)*P(J)
      X = 0.5*X
      DO 210  J = K,N
  210 P(J) = X*W(J) - P(J)
      DO 220  J = K,N
      DO 220  I = J,N
  220 G(I,J) = G(I,J) + P(I)*W(J) + P(J)*W(I)
      GO TO 250
  230 G(K,K1) = SQRT2
      B(K) = -Y
      DO 240  I = KJ,N
  240 G(I,K) = -G(I,K)
  250 CONTINUE
  280 DO 290  I = 1,N
      A(I) = G(I,I)
  290 G(I,I) = E(I)
      B(N) = G(N,N1)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     GET EIGENVALUES OF TRIDIAGONAL FORM BY KAHAN-VARAH Q-R ME HOD
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      TOL = PRECS/(10.*FLOAT(N))
      BMAX = 0.
      TMAX = 0.
      NP = MINO(80,N+1)
      W(NP) = 0.0
```

```
      DO 300  I = 1,N
      BMAX = AMAX1(BMAX,ABS(B(I)))
  300 TMAX = AMAX1(BMAX,ABS(A(I)),TMAX)
      SCALE = 1.0
      IF (BMAX.EQ.0.)  GO TO 520
      DO 310  I = 1,ILIM
      IF (SCALE*TMAX.GT.HOV)  GO TO 320
  310 SCALE = SCALE*BASE
  320 DO 330  I = 1,N
      E(I) = A(I)*SCALE
  330 W(I) = (B(I)*SCALE)**2
      DELTA = TMAX*SCALE*TOL
      EPS = DELTA**2
      K=N
  350 L = K
      IF (L.LE.0)  GO TO 460
      L1 = L - 1
      DO 360  I = 1,L
      K1 = K
      K = K - 1
      IF (W(K1).LT.EPS)  GO TO 380
  360 CONTINUE
  380 IF (K1.NE.L)  GO TO 400
      W(L) = 0.
      GO TO 350
  400 T = E(L) - E(L1)
      X = W(L)
      Y = 0.5*T
      S = SQRT(X)
      IF (ABS(T).GT.DELTA)  S = (X/Y)/(1.+SQRT(1.+X/Y**2))
      E1 = E(L) + S
      E2 = E(L1)- S
      IF (K1.NE.L1)  GO TO 430
      E(L)  = E1
      E(L1) = E2
      W(L1) = 0.
      GO TO 350
  430 LAMBDA = E1
      IF (ABS(T).LT.DELTA.AND.ABS(E2).LT.ABS(E1))  LAMBDA = E2
      S = 0.
      C = 1.
      GG = E(K1)-LAMBDA
      GO TO 450
  440 C = F/T
      S = X/T
      X = GG
      GG = C*(E(K1)-LAMBDA) - S*X
      E(K) = (X-GG) + E(K1)
  450 IF (ABS(GG).LT.DELTA) GG = GG + SIGN(C*DELTA,GG)
      F = GG**2/C
      K = K1
      K1 = K + 1
      X = W(K1)
      IF(K1.GT.N) X=0.0
      T = X + F
      W(K) = S*T
      IF (K.LT.L)  GO TO 440
      E(K) = GG + LAMBDA
      GO TO 350
  460 DO 470  I = 1,N
  470 E(I) = E(I)/SCALE
      Y =  ISIGN(1,M)
      DO 500  L = 1,N1
      K = N - L
      DO 500  I = 1,K
      IF (Y*(E(I)-E(I+1)).GT.0.)  GO TO 500
      X = E(I)
      E(I) = E(I+1)
      E(I+1) = X
  500 CONTINUE
  520 IF (M.EQ.0)  GO TO 860
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     COMPUTE EIGENVECTORS BY INVERSE ITERATION
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      NVEC = IABS(M)
      IF (NVEC.GT.N)  NVEC = N
      F = SCALE/HOV
      IF (BMAX*F.LT.PRECS)  GO TO 830
      DO 530  I = 1,N
      A(I) = A(I)*F
  530 B(I) = B(I)*F
      SEP = 25.*TMAX*PRECS
      X1 = 0.
      X2 = SQRT2
      DO 800  NV = 1,NVEC
      IF(NV.EQ.1) GO TO 535
      IF(ABS(E(NV)-E(NV-1)).LT.SEP) GO TO 550
  535 DO 540 I = 1,N
  540 W(I) = 1.0
      GO TO 570
  550 DO 560  I = 1,N
      X = AMOD(X1+X2,2.0)
      X1 = X2
      X2 = X
  560 W(I) = X - 1.0
  570 EV = E(NV)*F
      X = A(1) - EV
      Y = B(2)
      J = N1
      DO 600  I = 1,N1
      C = A(I+1) - EV
      S = B(I+1)
      IF (ABS(X).GE.ABS(S))  GO TO 580
      P(I) = S
      Q(I) = C
      INTV(I) = .TRUE.
      Z = -X/S
      X = Y + Z*C
      IF (I.LT.N1)  Y = Z*B(I+2)
      GO TO 600
  580 IF (ABS(X).LT.TOL)  X = TOL
      P(I) = X
      Q(I) = Y
      INTV(I) = .FALSE.
      Z = -S/X
      X = C + Z*Y
      IF(I.LT.N1) Y=B(I+2)
  600 V(I,NV) = Z
      IF (ABS(X).LT.TOL)  X = TOL
      NITER = 0
  620 NITER = NITER + 1
      W(N) = W(N)/X
      SUM = W(N)**2
      DO 640  L = 1,N1
      I = N - L
      Y = W(I) - Q(I)*W(I+1)
      IF(I.GE.N1) GO TO 630
      IF(INTV(I)) Y = Y - B(I+2)*W(I+2)
  630 W(I) = Y/P(I)
  640 SUM = SUM + W(I)**2
      S = SQRT(SUM)
      DO 660  I = 1,N
  660 W(I) = W(I)/S
      IF (NITER.GE.2)  GO TO 760
      DO 700  I = 1,N1
      Z = V(I,NV)
      IF(INTV(I)) GO TO 680
      W(I+1) = W(I+1) + Z*W(I)
      GO TO 700
  680 Y = W(I)
      W(I) = W(I+1)
      W(I+1) = Y + Z*W(I)
  700 CONTINUE
      GO TO 620
  730 L = J
      J = J - 1
      X = 0.
      DO 740  I = L,N
  740 X = X + G(I,J)*W(I)
      DO 750  I = L,N
  750 W(I) = W(I) - X*G(I,J)
  760 IF (J.GT.1)  GO TO 730
      DO 800  I = 1,N
  800 V(I,NV) = W(I)
      DO 820  I = 1,N
      A(I) = A(I)/F
  820 B(I) = B(I)/F
      GO TO 860
  830 DO 850  NV = 1,NVEC
      DO 840  I = 1,N
  840 V(I,NV) = 0.
  850 V(NV,NV) = 1.0
  860 DO 880  I = 2,N
      K = I - 1
      DO 880  J = 1,K
  880 G(I,J) = G(J,I)
      GO TO 950
  900 V(1,1) = 1.0
      A(1) = E(1)
  950 M = IABS(M)
      DO 15 I=1,N
      DO 15 J=1,M
   15 V(I,J)=XM(I)*V(I,J)
 1000 RETURN
      END
```

```
      SUBROUTINE ENVEL
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      WRITE (6,101) XC
      CALL DELETE(XB)
      IF(NR) 10,10,5
    5 CALL REMARK
   10 CALL MFIND(XA)
      K=NX
      NNC=NC
      NC=1
      CALL MLIST(XB)
      KK=NX
      DO 30 I=1,NR
      A(KK)=0.0
      TIM=SCAL1
      DO 25 J=1,NNC
      XA=ABS(A(K))
      IF(A(KK)-XA) 15,20,20
   15 A(KK)=XA
      TMAX=TIM
   20 K=K+1
   25 TIM=TIM+SCAL1
      WRITE (6,108) I,TMAX,A(KK)
   30 KK=KK+1
      WRITE (6,101) XC
      RETURN
  101 FORMAT(1A6)
  108 FORMAT(1I6,1F15.5,1E18.7)
      END
```

```
      SUBROUTINE RESPON
C
C       RESPONSE PROGRAM
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      N=NR
      WRITE (6,107) N,SCAL1,SCAL2
      CALL MFIND(XA)
      M=NC
      CALL MFIND(XC)
      NC=NC/N
      NR=M
      CALL MLIST(XD)
      ND=NX
      CALL MFIND(XA)
      NA=NX
      CALL MFIND(XB)
      NB=NX
      CALL MFIND(XC)
      NNC=NX
      NTAG=1
      IF(NR-1) 20,10,20
   10 NTAG=0
   20 L=NC
      CALL RESPOS(A(NA),A(NB),A(NNC),A(ND),M,N,L,SCAL1,SCAL2,NTAG)
      RETURN
  107 FORMAT (I6,2F12.5)
      END
```

```
C
      SUBROUTINE RESPOS(W,XM,P,X,NM,NT,L,DAMP,DT,NTAG)
C
      DIMENSION W(1),XM(1),P(1),X(1)
C
      K=1
      C1=DT/2.
      C2=C1*DT/3.
      C3=C2*2.
      DO 160 N=1,NM
      NOUT=NT+1
      C4=W(N)**2
      C5=2.*DAMP*W(N)
      F=1.+C1*C5+C2*C4
      DISP=0.0
      VEL=0.0
      IL=L*(N-1)*NTAG+1
      ACEL=P(IL)*XM(N)
      IL=IL+1
      IH=IL+L-2
      DO 160 I=IL,IH
      A=VEL+C1*ACEL
      B=DISP+DT*VEL+C3*ACEL
      ACEL=(P(I)*XM(N)-C5*A-C4*B)/F
      VEL=A+C1*ACEL
      DISP=B+C2*ACEL
      IF (NOUT-I+IL-2) 150,150,160
  150 X(K)=DISP
      K=K+1
      NOUT=NOUT+NT
  160 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FUNGN
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      N=NR
      WRITE (6,107) N,SCAL1
      CALL DELETE(XB)
      NR=1
      NC=N
      CALL MLIST(XB)
      K=NX
      NH=NX+N-1
      CALL MFIND(XA)
      L=NX
      TIM=A(L)
      GO TO 2
    1 IF(A(L)-TIM) 2,2,6
    2 DT=A(L+2)-A(L)
      DP=A(L+3)-A(L+1)
      L=L+2
      IF(DT) 4,2,5
    4 CALL ERROR(1)
    5 SLOPE=DP/DT
    6 A(K)=A(L-1)+(TIM-A(L-2))*SLOPE
    8 TIM=TIM+SCAL1
      K=K+1
      IF(NH-K) 9,1,1
    9 RETURN
  107 FORMAT(I6,F12.5)
      END
```

```
      SUBROUTINE LOADLM
C     LOAD AND PRINT OF LOCATION MATRIX
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),LM(1)
      NC=NR
      NR=1
      CALL MLIST(XA)
      NXX=NX+NC-1
      READ  (5,1000) (LM(I),I=NX,NXX)
      WRITE (6,1000) (LM(I),I=NX,NXX)
      RETURN
 1000 FORMAT (12I6)
      END
```

```
      SUBROUTINE ADDSTF
C     ADDS ELEMENT STIFFNESS TO TOTAL STIFFNESS
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL MFIND(XA)
      NA=NX
      IF(NR.NE.NC) CALL ERROR(1)
      NRA=NR
      CALL MFIND(XB)
      NB=NX
      IF(NR.NE.NC) CALL ERROR(1)
      NRB=NR
      CALL MFIND(XC)
      IF (NR*NC.NE.NRB) CALL ERROR(1)
      CALL ADDS(A(NA),A(NB),A(NX),NRA,NRB)
      RETURN
      END
```

```
      SUBROUTINE ADDS(A,B,LM,NA,NB)
      DIMENSION A(NA,NA),B(NB,NB),LM(NB)
      DO 400 I=1,NB
      II=LM(I)
      IF(II) 400,400,100
  100 DO 300 J=1,NB
      JJ=LM(J)
      IF(JJ) 300,300,200
  200 A(II,JJ)=A(II,JJ)+B(I,J)
  300 CONTINUE
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FORCEM
C     MULTIPLICATION OF A COMPACTED MATRIX BY A FULL MATRIX
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL DELETE(XD)
      CALL MFIND(XA)
      NA=NX
      NRA=NR
      NCA=NC
      CALL MFIND(XB)
      NB=NX
      IF(NC.NE.NCA) CALL ERROR(1)
      CALL MFIND(XC)
      NZ=NX
      NRC=NR
      NCC=NC
      NR=NRA
      CALL MLIST(XD)
      ND=NX
      CALL FORCE(A(NA),A(NB),A(NZ),A(ND),NRA,NCA,NRC,NC)
      RETURN
      END
```

```
      SUBROUTINE FORCE(A,LM,D,F,NRA,NCA,NRC,NC)
      DIMENSION A(NCA,NRA),LM(NCA),D(NC,NRC),F(NC,NRA)
      DO 400 I=1,NRA
      DO 400 J=1,NC
      F(J,I)=0.0
      DO 300 K=1,NCA
      KK=LM(K)
      IF(KK) 300,300,200
  200 F(J,I)=F(J,I)+A(K,I)*D(J,KK)
  300 CONTINUE
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FORMK
C
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL DELETE(XA)
      NR=2
      NC=2
      CALL MLIST(XA)
      READ (5,100)XI,XA,XL,XE
      WRITE(6,200)XI,XA,XL,XE
      B=14.4*XI/(XA*XL**2)
      S=2.*XE*XI/((1.+2.*B)*XL)
      A(NX)=S*(2.+B)
      A(NX+1)=S*(1.-B)
      A(NX+2)=A(NX+1)
      A(NX+3)=A(NX)
      RETURN
  100 FORMAT(4F12.1)
  200 FORMAT(3H I=F12.1,5H,  A=F12.1,5H,  L=F12.1,5H,  E=F12.1)
      END
```

```
      SUBROUTINE SLOPED
C     FORMATION OF SLOPE-DEFLECTION EQUATIONS
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL DELETE(XA)
      NC=4
      NR=4
      CALL MLIST(XA)
      CALL SLOPE(A(NX))
      RETURN
      END
```

```
      SUBROUTINE SLOPE(S)
      DIMENSION S(4,4)
      READ  (5,1000) XI,E,XL
      WRITE (6,2000) XI,E,XL
      C=2.*E*XI    /XL
      S(1,1)=2.*C
      S(1,2)=C
      S(1,3)=3.*C/XL
      S(1,4)=-S(1,3)
      S(2,1)=S(1,2)
      S(2,2)=S(1,1)
      S(2,3)=S(1,3)
      S(2,4)=S(1,4)
      S(3,1)=S(1,3)
      S(3,2)=S(2,3)
      S(3,3)=2.*S(2,3)/XL
      S(3,4)=-S(3,3)
      S(4,1)=S(1,4)
      S(4,2)=S(2,4)
      S(4,3)=S(3,4)
      S(4,4)=S(3,3)
      RETURN
 1000 FORMAT (3F12.0)
 2000 FORMAT (3H I=,E13.6,4X,2HE=,E13.6,4X,2HL=E13.6)
      END
```

```
      SUBROUTINE TRUSS
C     FORMATION OF MATRICES FOR TRUSS ELEMENT
      COMMON XA,XB,XC,XD,NR,NC,NNR,NNC,SCAL1,SCAL2,NX,TEM(480),A(1)
      CALL DELETE(XA)
      CALL DELETE(XB)
      NC=6
      NR=6
      CALL MLIST(XA)
      NA=NX
      NR=1
      CALL MLIST(XB)
      NB=NX
      CALL MFIND(XC)
      CALL TRUS(A(NA),A(NB),A(NX),NR)
      RETURN
      END
```

```
      SUBROUTINE TRUS(S,SS,XYZ,NR)
      DIMENSION S(6,6),SS(6),XYZ(3,NR)
      READ (5,1000) I,J,E,A
      WRITE(6,2000) I,J,E,A
      DX=XYZ(1,J)-XYZ(1,I)
      DY=XYZ(2,J)-XYZ(2,I)
      DZ=XYZ(3,J)-XYZ(3,I)
      XL=SQRT(DX*DX+DY*DY+DZ*DZ)
      C=A*E/XL
      SS(4)=DX/XL
      SS(5)=DY/XL
      SS(6)=DZ/XL
      SS(1)=-SS(4)
      SS(2)=-SS(5)
      SS(3)=-SS(6)
      DO 200 I=1,6
      T=SS(I)*C
      DO 100 J=I,6
      S(I,J)=T*SS(J)
  100 S(J,I)=S(I,J)
  200 SS(I)=T
      RETURN
 1000 FORMAT (2I6,2F12.0)
 2000 FORMAT (3H I= I4,3H J= I4,3H E= E15.7,3H A= E15.7 )
      END
```