**Westinghouse
Astronuclear
Laboratory**

SNPO-C

RECORD COPY

**WANL-TME-1940
June, 1971**

**SPRVIB
A GENERAL PURPOSE DYNAMIC
COMPUTER CODE USING
NORMAL MODE SOLUTION**

MASTER

DISTRIBUTION OF THIS DOCUMENT IS LIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

RECEIVED

1971 JUN 10 10:24

U-0
CLEVELAND, OHIO

Westinghouse Astronuclear Laboratory

Subcontract NP-1
CODE IDENT NO. 14683

WANL-TME-1940
June, 1971

NOTICE

This report contains information of a preliminary nature and was prepared primarily for internal use at the originating installation. It is subject to revision or correction and therefore does not represent a final report. It is passed to the recipient in confidence and should not be abstracted or further disclosed without the approval of the originating installation or USAEC Technical Information Center, Oak Ridge, TN 37830

SPRVIB A GENERAL PURPOSE DYNAMIC COMPUTER CODE USING NORMAL MODE SOLUTION

Approved by:

P. P. Zemanick, Manager
Structural Design

INFORMATION CATEGORY

UNCLASSIFIED

7/13/71
Authorized Classifier Date

ABSTRACT

This report describes a new general dynamic computer code called SPRVIB (Spring Vib) developed at Westinghouse Astronuclear Laboratory in support of the NERVA (Nuclear Engine for Rocket Vehicle Application) Program (Contract NP-1). Using normal mode techniques, the program computes kinematical responses of a structure caused by various combinations of harmonic and elliptic forcing functions or base excitations. Provision is made for a graphical type of force or base excitation input to the structure. Either proportional or modal damping is allowed in the structure. A description of the required input format and a listing of the program are presented along with several examples illustrating the use of the program.

SPRVIB is written in Fortran IV computer language for use on the CDC 6600 or the IBM 360/75 computers.

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1.0 INTRODUCTION	1
2.0 THEORETICAL ASPECTS	5
2.1 TOPOLOGICAL SECTION	5
2.2 FORCED MOTION	8
2.3 GENERATION OF DATA POINTS	15
3.0 EXAMPLE PROBLEMS	18
3.1 INTRODUCTION	18
3.1.1 Example Problem 1	18
3.1.2 Example Problem 2	28
3.1.3 Example Problem 3	31
4.0 PROGRAM INSTRUCTIONS	42
4.1 BASIC VARIABLES AND OPTIONS	42
4.2 DESCRIPTION OF THE SUBROUTINES	42
4.3 INPUT INSTRUCTIONS	48
APPENDIXES	
A	A-1
B	B-1
C	C-1
D	D-1

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Assumed Locally Parabolic Form of Data Point Function	10
2	Example Problem 1 Base Excited Shear Frame with Foundation	20
3	Data Card Input for Example Problem 1 Solved by Two Methods	22
4	Displacement Responses from Example 1	24
5	Velocity Responses from Example Problem 1	25
6	Acceleration Responses for Example Problem 1 Using Fourth Order Interpolation	26
7	Acceleration Responses for Example Problem 1 Using Linear Interpolation	27
8	Schematic of a Railroad Car Segment with Rolling Cargo Subjected to a Shock Load, P	29
9	Data Card Input for Example Problem 2 Solved by Two Methods	32
10	Displacement Responses of the Three Generalized Train Mass Segments for Example Problem 2	33
11	Acceleration Responses of Train Cars for Example Problem 2	34
12	Example Problem 3, Multi-story Drilling Platform Loaded by an Eccentric Load	36
13	Idealized Form of Example Problem 3	37
14	Data Card Input for Example Problem 3 Solved by Two Methods	39
15	Maximum Displacement Responses for Example Problem 3	40
16	Maximum Acceleration Responses for Example Problem 3	41
17	Flow Chart for the Modal Portion of the Solution	49
18	Flow Chart for the Response Portion of the Solution	50

1.0 INTRODUCTION

The usefulness of a dynamic computer code can be determined by examining four main features:

- 1) the method of solution
- 2) the type and generality of the forcing functions used
- 3) the flexibility of the input/output routines
- 4) size limitations of the program caused by storage and/or computational limits

The computer code, SPRVIB, should then be examined for all of these features.

SPRVIB uses a classical normal mode solution in which the individual response is computed as a sum of all the modal responses. This type of solution has two main advantages:

- 1) It completely describes the structure and often enables one to estimate the frequency dependence of the structure. This, of course, is not possible with a numerical solution.
- 2) It enables the exact solution to a linear problem to be determined in closed form if the forcing functions can be described by elementary functions. In these cases, roundoff type errors do not accumulate.

A normal mode solution also has two disadvantages:

- 1) At each time step in a transient solution to be printed out, conversion must be made from the normal coordinates to the regular coordinates of the problem.
- 2) Only proportional damping or modal damping type solutions are tractable for general programming.

The first disadvantage does not appear to be limiting in these days of the high speed computer. If the forcing functions are easily describable, such as harmonic or elliptic functions, the solution obtained is exact at each time step and the step size can be determined by convenience or plotting limitations. On the other hand, the time step size of the ordinary numerical type solutions determines the accuracy and stability limitations of the solution. Furthermore, with the possible exception of physical damping instruments, the

damping constants used in dynamic analysis are usually crude estimates of the actual damping. As such, it would appear at least as correct to use estimates of modal damping constants which often may be obtained from test results (1*). Hence, the apparent limitations are often not applicable to practical problems.

SPRVIB offers the following wide variety of possible forcing functions:

- 1) Free vibrations with arbitrary initial displacement and velocities at each generalized mass
- 2) Arbitrary harmonic forces $F_0 \sin \Omega t$ acting at each or all of the generalized mass points. Forces with different amplitudes and frequencies can be simultaneously applied at the various points.
- 3) Elliptic type forcing $F_0 \exp(\beta t)$ can act at each or all of the points similar to that described above. The parameters F_0 and β are real parameters which may have different values in as many as four consecutive time regions. Using this type of representation, a shock or explosive type of forcing can be easily described with sufficient accuracy to obtain quite accurate estimates of even velocities and accelerations. In addition, the harmonic and elliptic loads may be superimposed at any point.
- 4) Base harmonic excitation of the supports
- 5) Point force input at any mass point. In this case the force is represented as a set of data points.
- 6, 7, 8) Base motion in which the support acceleration (velocity, displacement) is described by a set of data points.

The effectiveness of the input and output features of any program greatly affect the time a user must spend in setting up a problem and the practicality of its use. A limiting feature of most dynamic computer codes available at this time is the requirement that the element matrices, such as the stiffness matrix, be generated by the direct stiffness method or some other similar method. SPRVIB will generate the element matrices for discrete mass elements connected by rigid bars, linear and rotational springs, and viscous

* The number in the text in parentheses refers to the reference number listed in an appendix to this report.

damping elements. At the Astronuclear Laboratory, SPRVIB can be linked with a finite element computer code STASYS, which can compute the mass and stiffness matrices, including shear effects for input to SPRVIB. If such a linking code is not available, the matrices must be computed separately for the more complex problems. It is in general quite simple to write an auxiliary finite element code for beam connected lumped mass models even using shear deflection. This type of model is the most common type used for overall system (not component) dynamic analysis. Such a code has also been effectively used⁽²⁾ with SPRVIB. Besides the regular input options, provision is made to input only the non-zero symmetric elements of sparse matrices. This option can be very useful for large matrices. All the input can be in a punched card format or some matrices can be read in from magnetic tape.

There are three main advantages that the code SPRVIB possesses over many of the other available computer codes. The first advantage is the wide variety of available forcing functions previously described and the ease with which the simple explicit function solutions can be combined in any problem. The second advantage is the nearly exact constrained motion type solution which can be effectively used even for the case of only scarce data points as shown by an example. The solution used is new. It interpolates new data points between the input points as required, while taking account of any smoothness, discontinuities, or output time spacing changes, approximates the data point function by a parabolic type polynomial function at each data point and uses this nearly exact approximation to obtain a solution using the closed form damped Duhamel's integral solution as the particular solution for the problem. Sample problem solutions are presented in Chapter 3 illustrating these features. The third advantage of this code is its modern and broad use of the modal damping option. In the writer's opinion, the damping in an actual problem is rarely known with any degree of precision except possibly from modal damping data. Hence, this option offers not only a simple but also a more realistic estimate of the actual damping to be used in a practical problem.

An option is available to store all the matrices and results from the topological portion of the program, including the mode shapes and frequencies, on tape so that this section of the program will not have to be redone at a later date for further load cases on the same problem. Any transient information which can be printed out can also be stored on tape for possible future plotting. Besides the regular steady state response, provision is made to output the results of each harmonic or elliptic load acting separately as well as the combined effect for each load case. In this way the transient effect of each load may be determined with a coarse time step while the effects of the combined loadings can be determined with a finer time step. One, two, or all of each of the displacements, velocities and accelerations may be output at each time step.

At the present time SPRVIB is limited to 90 degrees of freedom. The storage requirement of the program with the 90 degrees of freedom is 205,000_g.

2.0 THEORETICAL ASPECTS

The program can be separated into a topological section which computes the natural frequencies and mode shapes from the input data and a forced section which computes the response to some type of forcing.

The purpose of this section is to delineate the basic form of the solution so that the user may understand any inherent limitations of the solution and enable him to make any modifications as required for special problems.

2.1 TOPOLOGICAL SECTION

This program considers the linear damped vibration problem defined by the equation

$$M_t \ddot{X}_t + C_t \dot{X}_t + K_t X_t = A_t \quad (1)$$

where M_t , C_t , K_t are $(n+r) \times (n+r)$ matrices, representing the mass, damping, and stiffness matrices, X_t and A_t are $(n+r) \times 1$ vectors, n is the number of nodal degrees of freedom and r is number of support restraints. In order to analyze the structure it is convenient to separate Equation (1) into two sets composing the equations governing the actions of the nodal degrees of freedom and those governing the actions of the supports by partitioning the matrices of Equation (1) as shown below.

$$\begin{bmatrix} M & M_r \\ M_r^T & M_{rr} \end{bmatrix} \begin{bmatrix} \ddot{X} \\ \ddot{X}_r \end{bmatrix} + \begin{bmatrix} C & C_r \\ C_r^T & C_{rr} \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{X}_r \end{bmatrix} + \begin{bmatrix} K & K_r \\ K_r^T & K_{rr} \end{bmatrix} \begin{bmatrix} X \\ X_r \end{bmatrix} = \begin{bmatrix} A \\ A_r \end{bmatrix} \quad (2)$$

The superscript T in the above equations refers to the matrix transpose operation. If the indicated matrix multiplications are carried out, the following equations result after some rearrangement.

$$M \ddot{X} + C \dot{X} + KX = A - M_r \ddot{X}_r - C_r \dot{X}_r - K_r X_r \quad (3.a.)$$

$$M_r^T \ddot{X} + C_{rr} \dot{X}_r + K_{rr} X_r = A_r - C_r^T \dot{X} - K_r^T X - M_r^T \ddot{X} \quad (3.b.)$$

Equations (3.a.) and (3.b.) consist of n equations and r equations, respectively, which can be solved separately in most practical problems. In the case where external forces exist at the nodal degrees of freedom and the supports are fixed the last three terms on the right hand side of Equation (3.a.) vanish and the equation reduces to the classical forms. In a more general form we can set

$$P(t) = A - M_r \ddot{X}_r - C_r \dot{X}_r - K_r X_r \quad (4)$$

so that Equation (3.a.) can be written in the form

$$M \ddot{X} + C \dot{X} + K X = P \quad (5)$$

The homogeneous undamped form of Equation (4) is

$$M \ddot{X} + K X = 0 \quad (6)$$

We proceed to find the eigenvalues and eigenvectors of Equation (6) and to extend these with some damping limitations to Equation (5) in order to separate the n coupled equations in regular coordinates into independent equations in normal coordinates.

As the mass matrix is symmetric and positive definite, it can always be decomposed into a product of upper triangular matrix B and its transpose

$$M = B^T B \quad (7)$$

This, of course, is a form of the Choleski-Turing decomposition whereby the terms of the matrix B computed one at a time using the relations

$$B_{ij} = \frac{1}{B_{ii}} \left(M_{ij} - \sum_{k=1}^{i-1} B_{ki} B_{kj} \right) \quad i \geq j$$

$$= 0 \quad i < j$$

If M is diagonal so is B . Introducing Equation (7) and the variable Z defined by the equation

$$X = B^{-1} Z$$

in Equation (6), premultiplying by $(B^{-1})^T$ and assuming that $X = D e^{-i\lambda t}$, we obtain the canonical form

$$(\bar{K} - \lambda^2 I) Z = 0 \quad (8)$$

where $\bar{K} = (B^{-1})^T K B^{-1}$

The program SPRVIB computes the eigenvalues and eigenvectors of Equation (8) using Jacobi's method of successive rotations which has the advantages that it is easily programmed and that it computes the eigenvectors as well as the eigenvalues for each problem. The eigenvalues of $\bar{K}$ are the eigenvalues of the general eigenvalue problem while the matrix eigenvectors of $\bar{K}$ are related to the modal matrix of the dynamic problem by the relation

$$T_{(X)} = B^{-1} T_{(Z)} \quad (9)$$

Introducing now the normal coordinates Y defined by the relation

$$X = T_{(X)} Y \quad (10)$$

and premultiplying by $T_{(X)}^T$, Equation (4) becomes

$$T_{(X)}^T M T_{(X)} \ddot{Y} + T_{(X)}^T C T_{(X)} \dot{Y} + T_{(X)}^T K T_{(X)} Y = T_{(X)}^T P \quad (11)$$

Noting⁽³⁾ that the similarity transformations of M and K lead to diagonal matrices with a scale factor chosen to be unity and assuming that C also transforms into a diagonal matrix, we find

$$Y^{\circ\circ} + \begin{bmatrix} C_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_n \end{bmatrix} Y + \begin{bmatrix} \omega_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \omega_n^2 \end{bmatrix} Y = T_{(X)}^T P \quad (12)$$

This is the desired general form of uncoupled equations. It remains to examine the assumption that the damping matrix is similar to a diagonal matrix. Caughey⁽³⁾ has shown that a necessary and sufficient condition that C transforms into a diagonal matrix under the similarity transform is that the matrices $(M^{-1}C)$ and $(M^{-1}K)$ are commutative. A sufficient condition that the commutativity condition be satisfied is that the damping matrix be linearly proportional to the mass and stiffness matrices.

$$C = \alpha M + \beta K \quad (13)$$

Two common types of damping can be represented by the above equation:

- 1) relative damping, $\alpha = 0$, $\beta =$ any scalar constant
- 2) absolute damping, $\alpha =$ any scalar constant, $\beta = 0$

The terms of the stiffness matrix are computed from the force-deflection relations or equivalently from the strain energy expressions. On the other hand, for any structural use, excepting the use of mechanical viscous dampers, the damping terms must be estimated from the structural limits or test data. Moreover, the test data shows wide spreads dependent on the type and size of the structural elements and attachments. As the damping is generally small for most structural applications, it would appear as logical and much simpler to estimate the damping in the normal coordinates and assume that the transformed damping matrix is diagonal. At any rate, damping values are generally measured by exciting the normal modes of a structure and measuring the rate of decay of the free vibrations or by computing the modal damping constants directly from the results obtained from impedance tests (1). Hence, the use of modal damping would often appear to be consistent with the method by which the damping results were obtained.

2.2 FORCED MOTION

Equation (5) is solved for seven types of forcing functions:

Type 1 Harmonic forces at any mass point $P = P_o \sin(\Omega t)$

Type 2 Elliptic forces at any mass point

$$P = \begin{cases} P_1 e^{\beta_1 t} & 0 < t \leq T_1 \\ P_2 e^{\beta_2 t} & T_1 < t \leq T_2 \\ P_3 e^{\beta_3 t} & T_2 < t \leq T_3 \\ P_4 e^{\beta_4 t} & T_3 < t \leq T_4 \end{cases}$$

Type 3 Base harmonic motion $\ddot{X}_r = L \ddot{X}_o \sin(\Omega t)$

Type 4 Forces described by data points act at any mass point

Type 5 } Base displacement (velocity, acceleration) is described by
6 } data points
7 }

Besides the obvious constants shown above, we note that L is an r -dimensional vector usually composed of ones and zeros which denotes, up to a scale factor, the

pertinent motion at each support point and that it is used for base motions.

One great advantage here of using a normal mode solution is that a closed form solution of equation (5) is possible for each of the first three forcing functions. Hence, not only the displacements, but also the velocities and accelerations at each mass point are well known. Considering now the i th normal coordinate for the complementary solution of Equation (4) is

$$(Y_c)_i = e^{-C_i t/2} (A_i \cos \bar{W}_i t + B_i \sin \bar{W}_i t) \quad (14)$$

where C_i is the equivalent viscous damping constant in the mode, $\bar{W}_i$ is the damped natural frequency and A_i and B_i are the arbitrary constants determined by initial conditions in each mode.

The particular solution for the i th generalized coordinate caused by a load of the first three types acting at generalized mass k , are respectively:

$$(Y_i)_p = \frac{T_{ki}}{\Delta_1} (\bar{W}_i^2 + (C_i/2)^2 - \Omega^2) P_o \sin(\Omega t) - \frac{T_{ki}}{\Delta_1} C_i \Omega P_o \times$$

$$\cos(\Omega t) = (Y_{ps})_i \sin(\Omega t) + (Y_{pc})_i \cos(\Omega t) \quad (15a)$$

$$(Y_i)_p = \frac{T_{ki}}{\Delta_2} P_o e^{\beta t} = Y_{pe_i} \quad (15b)$$

$$(Y_i)_p = \frac{a \Delta_o + b C_i \Omega}{\Omega^2 \Delta_1} \sin(\Omega t) - \frac{(a C_i \Omega - b \Delta_o)}{\Omega^2 \Delta_1} \cos(\Omega t) \quad (15c)$$

$$\text{where } a = - \sum_{j=1}^n \sum_{k=1}^r T_{ji} L_k (K_{rki} - \Omega^2 M_{rki}) \overset{\infty}{X}_o$$

$$b = - \sum_{j=1}^n \sum_{k=1}^r T_{ji} L_k \Omega C_{rjk} \overset{\infty}{X}_o$$

where: $\Delta_0 = \bar{W}_i^2 - \Omega^2 + (C_i/2)^2$
 $\Delta_1 = \Delta_0^2 + C_i^2 \Omega^2$
 $\Delta_2 = \bar{W}_i^2 + \beta^2 + (C_i/2)^2 + C_i \beta$
 $\ddot{X}_0 =$ acceleration of the pertinent constrained points
 $T =$ modal matrix of eigenvectors
 K_r, C_r, M_r are matrices defined by Equation (2).

The remaining cases where explicit expressions for $P(t)$ are not available cannot be solved in closed forms. However, if we limit our consideration now* to cases where the data points are dense and the curve satisfied by these data points is locally smooth, then a very close approximation to the actual forcing function can be obtained by passing a 2nd-order parabola through each three successive data points. That is, over the interval (t_i, t_{i+1}) , the data $X(t)$ is assumed to satisfy the equation

$$X(t) = X_i + \frac{(t-t_i)}{2(t_i-t_{i-1})} (X_{i+1} - X_{i-1}) + \frac{(t-t_i)^2}{2(t_i-t_{i-1})^2} (X_{i+1} - 2X_i + X_{i-1}) \quad (16)$$

FIGURE 1

Assumed Locally Parabolic Form of Data Point Function

* Further discussion of program considerations of smoothness discontinuities and interpolation options to obtain dense data points are in Section 2.3. Also, the program can generate internally evenly spaced output points, up to output step size changes, from any input set of data points.

This condition is shown in Figure 1. An application of the Extended Law of the Mean shows that the maximum error E between Equation (16) and the true curve for abscissa spacing h is on the order of

$$E \leq \max_{C \in (t_i, t_{i+1})} \frac{(X'''(C))}{3!} h^3 \quad (17)$$

If the true curve $X(t)$ is a polynomial of second order, then of course E vanishes. Equation (17) shows that even if this condition is not satisfied, then E can be made as small as one desires by choosing h sufficiently small. The case in which the smoothness discontinuity is isolated at a point, such as known cusps in data points or sharp step function inputs, can also be solved by this method as shown in Section 3.1.3.

Introducing for simplicity the following notation into Equation (16) :

$$\begin{aligned} X_0 &= X_i \\ X_1 &= X_{i+1} - X_{i-1} \\ X_2 &= X_{i+1} - 2X_i + X_{i-1} \end{aligned}$$

where the X_i stands for the respective data point corresponding to a force, support displacement, velocity or acceleration and performing the integrations and differentiations indicated in Equation (4) then this equation can be written out explicitly for each of the four data point loads listed in the above order.

$$P(t) = T^t \left(X_0 + \frac{t-t_i}{2h} X_1 + \frac{(t-t_i)^2}{2h^2} X_2 \right) \quad (18a)$$

$$\begin{aligned} P(t) = -T^t \left(K_r L \left(X_0 + \frac{t-t_i}{2h} X_1 + \frac{(t-t_i)^2}{2h^2} X_2 \right) \right. \\ \left. + C_r L \left(\frac{X_1}{2h} + \frac{t-t_i}{h^2} X_2 \right) + M_r L \frac{X_2}{h^2} \right) \quad (18b) \end{aligned}$$

$$\begin{aligned}
P(t) = & -T^T (K_r L (E_1 + (t-t_i) X_o + \frac{(t-t_i)^2}{4h} X_1 + \frac{(t-t_i)^3}{6h^2} X_2) \\
& + G_r L (X_o + \frac{(t-t_i)}{2h} X_1 + \frac{(t-t_i)^2}{2h^2} X_2) \\
& + M_r L (\frac{X_1}{2h} + \frac{t-t_i}{h^2} X_2))
\end{aligned} \quad (18c)$$

$$\begin{aligned}
P(t) = & -T^T (K_r L (\frac{(t-t_i)^2}{2} X_o + \frac{(t-t_i)^3}{12h} X_1 + \frac{(t-t_i)^4}{24h^2} X_2 + E_2 (t-t_i) + E_1) \\
& + C_r L ((t-t_i) X_o + \frac{(t-t_i)^2}{4h} X_1 + \frac{(t-t_i)^3}{6h^2} X_2 + E_2) \\
& + M_r L (X_o + \frac{(t-t_i)}{2h} X_1 + \frac{(t-t_i)^2}{2h^2} X_2))
\end{aligned} \quad (18d)$$

where the matrices are as defined by Equations (2) and (9), E_1 is the displacement of the moveable supports and E_2 is the velocity of the moveable supports at the time $t = t_i$. The latter two quantities are, of course, the constants obtained after integrating Equation (16) when the support velocity or acceleration is input as the data points. These quantities are obtained by numerical integration using Simpson's rule after interpolating a point midway in the interval (t_i, t_{i+1}) .

Having now a form for the forcing function a particular solution to the equations incremented over each time interval can be computed using Duhamel's integral over each time interval

$$Y_{P_i} = \frac{1}{\bar{W}_i} \int_0^h P_i(\tau) e^{-\frac{C_i}{2}(t-\tau)} \sin \bar{W}_i(t-\tau) d\tau \quad (19)$$

where the $P_i(\tau)$ is obtained after performing the indicated matrix multiplications in Equation (18).

The integrations indicated in Equation (19) need be performed only for the initial time step and again for each change in output step size as the data points themselves are constants which can come outside of the separate integrals. The integrations are very lengthy and can be decomposed into integrals of the form:

$$\begin{aligned} I_k &= \int_0^h \tau^k e^{\frac{-C}{2}(t-\tau)} \sin \bar{W} (t-\tau) d\tau \\ J_k &= \int_0^h \tau^k e^{\frac{-C}{2}(t-\tau)} \cos \bar{W} (t-\tau) d\tau \end{aligned} \quad k = 0, 1, \dots, 4 \quad (20)$$

Using the fact that

$$\begin{aligned} I_0 &= \frac{\bar{W} - \bar{W} e^{-Ch/2} \cos \bar{W} h - C/2 e^{-Ch/2} \sin \bar{W} h}{\bar{W}^2 + (C/2)^2} \\ J_0 &= \frac{\bar{W} e^{-Ch/2} \sin \bar{W} h + C/2 - C/2 e^{-Ch/2} \cos \bar{W} h}{\bar{W}^2 + (C/2)^2} \end{aligned} \quad (21)$$

and the easily derived recurrence relations

$$\begin{aligned} I_k &= 1/(\bar{W}^2 + (C/2)^2) \left\{ \bar{W} h^k - k \bar{W} J_{k-1} + k C/2 I_{k-1} \right\} \\ J_k &= 1/(\bar{W}^2 + (C/2)^2) \left\{ h^k C/2 + \bar{W} k I_{k-1} - k C/2 J_{k-1} \right\} \end{aligned} \quad , k \geq 1 \quad (22)$$

then each particular solution Y_{p_i} defined by Equation (19) can be computed as an algebraic equation involving only X_0 , X_1 , X_2 , E_1 and E_2 and constants which depend on the known dynamic constants and the output spacing.

Taking into consideration now the fact that the constants A_i and B_i can as well be evaluated at each time step, then the solution can proceed stepwise with each Y_i , $\dot{Y}_i = \dot{Y}_i$, and $A_i = \ddot{Y}_i$ being evaluated at each time step using the values at the previous time step as the initial values. Hence, at the $n+1$ st time step, the i th normal coordinate satisfies the equations

$$Y_{n+1} = e^{-C/2 h} \left(Y_n \cos \bar{W} h + \left(\frac{\dot{Y}_n}{\bar{W}} + \frac{Y_n C}{2 \bar{W}} \right) \sin \bar{W} h \right) + Y_p$$

$$\dot{Y}_{n+1} = -C/2 Y_{n+1} + e^{-C/2 h} \left((\dot{Y}_n + Y_n \frac{C}{2}) \cos \bar{W} h - \bar{W} Y_n \sin \bar{W} h \right)$$

$$+ \int_0^h F(\tau) e^{-C/2(t-\tau)} \cos \bar{W} (t-\tau) d\tau \quad (23)$$

$$\ddot{Y}_{n+1} = -C \dot{Y}_{n+1} - (\bar{W}^2 + C^2/4) Y_{n+1} + P(h)$$

and the solution for point type loads can proceed sequentially for each mode and for each time step.

Returning now to the case where the forcing function is of the first three types, that is, it is known explicitly, then the constants A_i and B_i need only be computed initially or for the elliptic type loads, whenever the input constants defining the loads are changed. For these cases with a load at the k th generalized mass, then the i th arbitrary constants of Equation (14) can be computed using the notation of Equation (15) for a harmonic type input as

$$\begin{aligned}
 A_i &= Y_{oi} - Y_{pci} \\
 B_i &= \frac{Y_{oi}}{\bar{W}_i} + \frac{C_i}{2\bar{W}_i} A_i - \frac{\Omega}{\bar{W}} Y_{psi}
 \end{aligned}
 \tag{24a}$$

and for an elliptic type load

$$\begin{aligned}
 A_i &= Y_{oi} - Y_{pei} \\
 B_i &= \frac{Y_{oi}}{\bar{W}_i} + \frac{C_i}{2\bar{W}_i} A_i - \frac{\beta}{\bar{W}_i} Y_{pei}
 \end{aligned}
 \tag{24b}$$

If more than one load is acting, or if a combination of harmonic and elliptic type loads act on a system, then the last terms in each of the Equations (24) must be replaced by a sum of the corresponding terms acting at the i^{th} generalized mass for each of the loads.

This section has presented the basic equations used to compute the forced responses for a large class of inputs. The next section will consider the interpolation routines used to generate a sufficiently large set of output data points from a selected set of input points.

2.3 GENERATION OF DATA POINTS

Typically a user either has too many data points or too few data points for suitable dynamic analysis. While only engineering judgment can be used to sort out the data points in the first case, the program SPRVIB has as one feature the option to use either linear or 4th order interpolation to fill in any interval with a sufficiently dense set of data points.

The sufficiency condition should be interpreted in the light of Equation (17) which shows the approximation of the local parabolas to the true data curve is a function of the cube of output time spacing, h , while Equations (21) and (22) exhibit the dependence of the particular solution on h . Provision has been made for up to 10 changes of the output spacing to enable one to use finer time meshes in regions of rapid responses and coarser time meshes for slow transients. Provision has also been made for the user to

input data points with isolated smoothness discontinuities, in particular cusps and corners, by locating these derivative discontinuities. We proceed to describe the fourth order interpolation routines and note that the first order routines are a special case of this. All the interpolation formulas used follow from Newton's Interpolation Formula of 4th order

$$\begin{aligned}
 f(X) = & f(X_1) + (X - X_1)[X_1 \quad X_2] + (X - X_1)(X - X_2)[X_1 \quad X_2 \quad X_3] \\
 & + (X - X_1)(X - X_2)(X - X_3)[X_1 \quad X_2 \quad X_3 \quad X_4] + (X - X_1) \\
 & (X - X_2)(X - X_3)(X - X_4)[X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5]
 \end{aligned} \quad (25)$$

where the remainder has been neglected. The closed brackets stand for the respective 1st, 2nd, 3rd and 4th divided differences defined by the equations

$$\begin{aligned}
 [X_1 \quad X_2] &= \frac{Y_1}{(X_1 - X_2)} + \frac{Y_2}{(X_2 - X_1)} \\
 [X_1 \quad X_2 \quad X_3] &= \frac{[X_1 \quad X_2] - [X_2 \quad X_3]}{X_1 - X_3} = \frac{Y_1}{(X_1 - X_2)(X_1 - X_3)} \\
 &+ \frac{Y_2}{(X_2 - X_1)(X_2 - X_3)} + \frac{Y_3}{(X_3 - X_1)(X_3 - X_2)} \\
 &\vdots \\
 [X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5] &= \frac{[X_1 \quad X_2 \quad X_3 \quad X_4] - [X_2 \quad X_3 \quad X_4 \quad X_5]}{X_1 - X_5}
 \end{aligned} \quad (26)$$

The formula for central differencing about the i^{th} point can be derived from the above cases by substituting

$$\begin{aligned}X_1 &= X_i \\X_2 &= X_{i+1} \\X_3 &= X_{i-1} \\X_4 &= X_{i+2} \\X_5 &= X_{i-2}\end{aligned}\tag{27}$$

into Equations (25) and (26). The equation for first or second order forward differencing can be obtained from Equation (25) by truncating the right hand side of the equation after two or three terms, respectively. Other types of differencing can be obtained with a different choice in Equation (27).

3.0 EXAMPLE PROBLEMS

3.1 INTRODUCTION

The function of the examples in this section is to instruct the user in the basic problem formulation and input to the code while presenting him examples that are easily followed. In this way the user can gain confidence in the operation and accuracy of the code while learning to input his problems. The only difficulty, however, is the fact that if the user is to be able to easily follow the solution steps in the program and in his own computations the problem must be small in size. Yet it is generally the large problem that leads to numerical difficulties in a computer program solution, especially for a modal solution. This impasse between the goals of instruction and verification cannot be completely solved in this section.

Although some classical eigenvalue example problems exist for large matrices,⁽⁴⁾ there is a lack of suitable large scale response solutions. Hence, there does not exist a suitable standard of code verification for large size problems. The code, SPRVIB, has been successfully used for problems up to and in fact exceeding the nominal size limit of the code. The base response option has, in addition, been checked against available test results on the response of a large test stand with the NERVA engine in place due to a seismic event.⁽⁵⁾ An in-plane finite element model with 78 degrees of freedom was used in the analysis. At any rate, it should be noted that any numerical difficulties in extending the program to large size problems would probably first occur in the modal solution. A simple test of this for a practical problem is to initially run the problem without constraints in order to verify the free-free or rigid body modes to ascertain whether double precision arithmetic is required for a problem. While these difficulties have not been encountered by WANL personnel working with the CDC-6600 computer, they may occur in the solution of a large problem on a computer with a smaller number of significant digits. Hence, the solution of simple example problems appears suitable here.

3.1.1 Example Problem 1

This problem illustrates some common types of input used primarily for large sparse matrices and the base harmonic and point support motion excitations. We shall see that the input is quite simple, but that the usual caution should be exercised in interpreting the results for a problem with a limited set of data points, when the output depends on derivatives of the data

point function. The SPRVIB code has a valuable option to use 4th order interpolation to increase the effective number of data points. The use of this option is illustrated and shown to significantly improve the solution for this case.

Example Problem 1 considers the base excitation of a two story symmetrical frame with a rigid floor structure and rotationless frame-floor joints. The frame, shown in Figure 2, has 2 degrees of freedom with the mode shapes shown. We assume that the frame is massless and that the idealized structure has the following properties to some common scale (where the foundation properties are also shown).

$$\begin{aligned} M_2 &= 10 \text{ lb-sec}^2/\text{in.} & k_2 &= 10^4 \text{ lbs/in.} \\ M_1 &= 10 \text{ lb-sec}^2/\text{in.} & k_1 &= 10^4 \text{ lbs/in.} \\ M_0 &= 20 \text{ lb-sec}^2/\text{in.} & k_0 &= 9.99 \times 10^6 \text{ lbs/in.} \end{aligned}$$

A similar undamped problem is shown in Warburton ⁽⁶⁾ where some details are shown.

We consider the problem with 2% relative damping, i. e., the damping matrix $C = \beta K$, where K is the stiffness matrix and $\beta = .02$ for our problem. We recall that relative damping is similar to structural damping, but it is independent of the frequency and therefore free from the well known theoretical anomalies of structural damping. For this problem, it is assumed that the displacement of the attached foundation is measured and satisfies the equation.

$$X_0 = .01 \sin 2\pi t$$

and the responses of the frame stories are required.

We note that to specify a motion at any set of coordinates is equivalent to imposing constraints on those coordinates and, hence, from equations 2, 5 and 6, those coordinates must be removed from the nodal degrees of freedom. Furthermore, the eigenvalues and eigenvectors for this problem are those for only the nodal degrees of freedom. The program assumes that if there exists n nodal degrees of freedom for a problem, then any constrained or support

Figure 2 EXAMPLE PROBLEM 1 Base Excited Shear Frame with Foundation

coordinates have generalized numbers higher than n . A provision exists for coordinates and corresponding matrix rotation prior to the modal solution to separate the nodal degrees of freedom by matrix partition.

A typical data card input for this problem with harmonic base motion solved both by an exact closed form solution and for the case when the support displacement is described by only 17 data points per cycle is shown in Figure 3. The input cards and various options are discussed more fully in Section 4; here we wish to only indicate the salient features applied to the particular example problem.

Referring to Figure 3, we note that the choice of $\text{NOPTIO} = -3$ specifies that the mass matrix is diagonal, that only the non-zero terms of the upper triangular half of the symmetric stiffness and damping ($\text{NDP} = 1$) matrix will be input, and that the user wished to renumber his coordinates by using matrix rotation. The input data is also output for user verification purposes as shown in the printout for this example problem in Appendix B. Double coordinate rotation for this problem was chosen (from 0, 1, 2 (physical) to 3, 1, 2 (computer)) so that coordinates for the frame stories could keep their coordinate number. As there is only 1 support coordinate ($\text{NCL} = 1$) for this example problem, the choice of a scale factor of 1 for the exact solution is hardly surprising. One should note, however, that the user is free to impose motions with different amplitudes, including an amplitude of zero or out of phase in some direction, for each support coordinate. For this case, all the displacements, velocities and accelerations are printed out and also put on tape for plotting at the requested time points.

For the second load case, displacement described by only a crude mesh of data points, all too often a common problem, the fourth order interpolation option ($\text{Key (4)} = 1$) was chosen to expand the number of data points. It should be noted that the choice of only linear interpolation here would imply that the acceleration of the constrained coordinates vanishes over the regions of the interpolation. It will be shown that this assumption would lead to an "averaged" solution for the frame accelerations. Provision is also made in the code for another potential problem in the generation of additional data points by using fourth order interpolation, that is, possible smoothness limitations, such as sharp cusps, in the data point function. For this problem we have not indicated any such discontinuities ($\text{NDISC} = 0$).

Figure 3. Data Card Input for Example Problem 1 Solved by Two Methods

We note that in the exact solution, we have implicitly assumed an initial velocity $\dot{X}_0 = .01 (2\pi) \cos 2\pi 0$ and hence for comparison purposes we show one also for this case $SV = .06283$. These initial values, however, are only used in the numerical integration portion of the response that is, when the velocity or acceleration of a constrained set of points are specified.

The frame story displacements, velocities, and accelerations are plotted in Figures 4, 5, and 6, respectively, for both the exact solution (solid lines) and the data point numerical solution (dashed lines). As one can observe, the independent solutions agree very well with each other. The reader can readily carry out the details of the solutions for a problem of this size and verify that this is indeed an exact solution. The problem is also interesting in that it presents a solution with a relatively large amount of critical damping in one mode, a damping which acts as an additional forcing function to the upper stories through the coupled equations in material coordinates.

As was previously discussed, the ability of the program to generate additional data points using fourth order interpolation represents a significant advantage over other dynamic programs. This option enabled an excellent solution to be obtained for a highly damped displacement problem with a very limited set of data points.

It is interesting to note the limitations of the classical solution with linear interpolation used between the input data points. This solution would be equivalent to assuming a constant velocity and zero acceleration at the constrained coordinates over the regions of each interpolation. As the damping is very significant for this problem, this assumption leads eventually to a nearly constant forcing function over each of the regions of interpolation (6.25 sec intervals for this problem) in each of the modes. We would expect that this discontinuous forcing function would lead, after a suitable constraint velocity buildup, to a corresponding "averaged" acceleration response on the frame and, indeed, such is the case as is shown in Figure 7 in which linear interpolation had been used to generate the additional data points. Even for this extreme case of large damping, however, the displacements and velocities, which are not shown with linear interpolation input, were very closely

Figure 4. Displacement Responses from Example 1

Figure 5. Velocity Responses from Example Problem 1

Figure 6. Acceleration Responses for Example Problem 1 Using Fourth Order Interpolation

— Point Displacement
- - - Exact Solution

Figure 7. Acceleration Responses for Example Problem 1 Using Linear Interpolation

determined.

Finally we note that only a few changes; namely, load type (IL) = 7, constraint scale factors AL (1) = -39.48, and initial constraint velocity SV = -.00159*, need be made in order to use the same data cards for a constraint acceleration input. Moreover, as the other kinematical parameters are integral functions of the data points for this type of input, nearly identical results would be obtained if either linear or quartic interpolation would be used to generate additional data points.

Besides the obvious use of these options for seismic problems, it should be noted that the options discussed can easily be used for complex displacement, velocity and acceleration wave motion type problems in which the various wave fronts are constrained to arbitrarily specified motions.

Total elapsed compiling, execution, and I/O time for this problem with both load cases was 18.6 sec. CP and 16.1 sec. PP.

3.1.2 Example Problem 2

This problem considers the response of a train segment with a moveable rolling cargo when the lead car is subjected to an abrupt shock load. The problem features illustrated by this problem solution are its capability of handling non-diagonal mass matrices, the use of its modal damping option, the responses due to an elliptic shock type load, to a load described only by data points, and the option of variable output regions. A rough schematic of the system is shown in Figure 8.

The horizontal equilibrium of each of the three masses and the rotational equilibrium of the idealized roller yields the following equations

$$M_1 \ddot{X}_1 + k_1 X_1 - k_3 (X_3 - X_1) - k_2 (X_2 - X_1) - F = P(t)$$

$$M_2 \ddot{X}_2 + k_2 (X_2 - X_1) + F = 0$$

* Note the actual velocity = SV*AL(1).

Figure 8. Schematic of a Railroad Car Segment with Rolling Cargo Subjected to a Shock Load, P

$$M_3 \ddot{X}_3 + k_3 (X_3 - X_1) = 0$$

$$I_2 \ddot{X}_2 - I_2 \ddot{X}_1 = Fa^2$$

where F is the frictional force between the rotatable element and m_1 , a is the roller radius, and $P(t)$ is the external force. The latter equation can be used to eliminate F from the first three equations which can then be arranged in the form of matrix equation (1). The constants for this example problem are (to some suitable scale)

$$\begin{aligned} M_1 &= 2.00 \text{ lb-sec}^2/\text{ft} & k_1 &= 8. \text{ lbs/ft} \\ M_2 &= 4.00 \text{ lb-sec}^2/\text{ft} & k_2 &= 16. \text{ lbs/ft} \\ M_3 &= 4.00 \text{ lb-sec}^2/\text{ft} & k_3 &= 16. \text{ lbs/ft} \\ I_2/a^2 &= 2.00 \text{ lb-sec}^2/\text{ft} \end{aligned}$$

Typically on a problem of this type the damping function is not known with any degree of precision in the natural coordinates. If it is known at all, it is usually observed from modal decay data or estimated on the basis of such data. At any rate, it is usually the damping in the normal mode that can be best estimated or observed. The code SPRVIB allows direct use of such data. If $NDP < 0$, then decimals ζ_i are input for each mode and are used to compute the respective terms of the diagonal transformed damping matrix shown in equation 12 by the formula $C_i = 2 \bar{W}_i \zeta_i$, in which we have used the fact that the modal mass is chosen to be unity.

Quite often, shock type loads follow the canonical type curve shown in Figure 8 and, moreover, can be easily described by successive elliptic loads. The advantages of such an approximation lies in the ease of input for even complicated type of force plots and in the accuracy with which even the accelerations can be obtained from only a crude force approximation. The data cards for such a solution with the force described by

$$P(t) = \begin{cases} 2 * \text{EXP}(5t) & 0 \leq t < .5 \text{ sec} \\ 3616. * \text{EXP}(-10t), & .5 \leq t < 3 \text{ sec} \\ 0 & 3 \leq t \end{cases}$$

along with point force input is shown in Figure 9. As we have chosen to use quartic interpolation to generate additional input points and as the potential exists for a sharp discontinuity in the data derivatives at $t = .5 \text{ sec.}$, NDISC was set equal to 1 and the time TDISC (1) = .5 sec. was input. Actually this refinement will show a change in the outputs especially near the time of smoothness discontinuity. The problem also illustrates the use of a different output time steps in various time regions. As the solution accuracy depends on the output spacing and the accuracy of the forcing function approximation as well as the topological parameters of the problem, the option of varying output spacings allows the user to change his step size vs the force changes in order to maintain a consistent degree of solution accuracy. As many as 10 step changes and load discontinuities are allowed for each problem solution.

The detailed output from this example problem with both types of forcing functions is shown in Appendix C and plotted in Figure 10 for displacements while the train accelerations are plotted in Figure 11. We note from these figures that the agreement between the independent solutions is almost exact even for this abrupt force type of input; and, moreover, the results are quite close to those quickly obtained from an independent desk computer solution working from the modal solution, which itself agrees with a similar modal problem in Pestel and Leckie⁽⁷⁾. Total elapsed running, compiling, and I/O time on the CDC 6600 system for both cases was 19.5 sec. CP and 21.4 sec. PP.

3.1.3 Example Problem 3

In this problem we consider the response of a small idealized off shore drilling platform with an auxiliary crane excited by an eccentric harmonic machinery load. The features of the code illustrated by this example are: 1) the option to have the code form the topological matrices for simple linear and rotational spring and damper connected bodies, 2) the use of modal damping, 3) generalized harmonic force input, and 4) various output options.

[illegible]

Figure 9. Data Card Input for Example Problem 1 Solved by Two Methods

Figure 10. Displacement Responses of the Three Generalized Train Mass Segments for Example Problem 2

Figure 11. Acceleration Responses of Train Cars for Example Problem 2

The schematic for this problem is shown in Figure 12. We note that for this problem, we can neglect any transverse motions and consider the various rigid platforms as being connected by linear and rotational springs. The flexible boom is replaced by a rigid arm and an equivalent linear spring. The eccentric force can be replaced by an equivalent force and couple set as shown. The idealized problem is shown in Figure 13.

We begin by assigning any support element the designation number 0, a mass element connected to it as node 1 and then arbitrarily numbering the remaining masses 2, . . . 7. Any further support nodes are assigned sequential numbers greater than the number of mass elements (7 in this problem). The spring designation is that of the highest node point to which it is attached. For this example problem the following (suitably scaled) constants were used:

Designation	M (lb-sec ² /ft)	I 10 ³ (lb-sec ² -ft)	k 10 ⁶ lbs-ft	R 10 ⁷ lbs-ft
1	700	660	1.3	1.7
2	700	660	2.6	3.5
3	500	330	4.2	5.6
4	500	330	5.9	7.9
5	10	0	.0004	0
6	400	270	8.2	11.
7	400	270	10.	13.

For this problem, the lengths $l = 40'$, $a = 20'$, and $P = 2000 \sin(40t)$.

While only limited data is available on modal damping values for many degree of freedom frame structures, it appears logical that the percent of critical damping will decrease for the higher mode numbers ⁽⁷⁾ as the frequency increases, while the modal damping values C_i will increase. A conservative rate of decrease shown below was assumed for this problem:

Mode Number	ζ
1-5	.04
6-8	.03
9-11	.02
12-13	.01

Figure 12. Example Problem 3, Multi-story Drilling Platform Loaded by an Eccentric Load

Figure 13. Idealized Form of Example Problem 3

The maximum vertical and rotational deflections are shown in Figure 15 for the force $P(t)$ alone and for the combined loading, while the corresponding maximum vertical and rotational accelerations are shown in Figure 16. These maximum components occur at the top story for the vertical displacement and at the third story for the rotational displacement. A selected printout for both load components acting alone and for the combined effects is shown in Appendix D. For this problem with complete compile and with the I/O as indicated in Figure 14 and Appendix D, respectively, required 42.1 sec. CP time and 21.2 sec. PP time on the CDC 6600 computer.

**Astronuclear
Laboratory**

— Due to Axial Force Alone
 - - - - - Due to Both Axial Force and Moment

Figure 15. Maximum Displacement Responses for Example Problem 3

Figure 16. Maximum Acceleration Responses for Example Problem 3

4.0 PROGRAM INSTRUCTIONS

4.1 BASIC VARIABLES AND OPTIONS

The intent of this section is to list the main purposes and options for each important variable. It is hoped that this will enable the user to better follow the flow of the analysis and aid him in making any internal modifications. In the latter case, however, it should be emphasized that any changes should be carefully checked in order to eliminate any possible side effects on another portion of the program. One initial manner of checking would be to rerun the example problems shown in Chapter 3. A summary description of the parameters is given below.

A,B	Vectors of initial constants in free response equation.
AHI,AHJ:	Matrices of partial particular solutions of damped Duhamel's integral and its derivatives for polynomial functions.
AL:	Vector of constraint scale factors. Used in constraint motion solutions.
AM:	Vector of generalized mass elements.
AMR,SR,CCR:	Mass, displacement and damping coupling matrices between the constrained modes (r) and the nodal degrees of freedom (n).
BETA, EEP, TM:	Elliptic force parameter vectors for successive time regions; $F_e = EEP_i * \exp(BETA_i * T)$, $TM_{i-n} \leq T < TM_i$.
CX,CY:	Damping matrices in regular and normal coordinates (CY is also used for temporary storage of the non-diagonal mass matrix).
C:	Modal damping vector, $C = \text{DIAG}(CY)$.
CL,CR:	Vector of linear and rotational viscous damping elements (also for storage of some modal damping elements, also for storage of some modal damping input data).
DIX,DIY,(VIX,VIY):	Vectors of initial displacements (velocities) in regular and normal coordinates.

DS,DT,(VS,VT,AS,AT):	Vectors of displacements (velocities and accelerations) in regular and normal coordinates.
DYM:	$nx(n+1)/2$ dynamic vector for eigenvalue solution.
EP:	Matrix of particular solutions for various elliptic loads acting at each mass element.
E1,E2,E3:	Constraint displacement, velocity, and displacement used in point motion solution. (Note this is determined only up to a vector of scale factors).
FR, FREQ:	Vectors of natural and forcing frequencies.
H,TSTEPH:	Vector of successive output spacings H_i , $TSTEPH_i \leq T < TSTEPH_{i+1}$. Used in the point input solutions ($IL \geq 4$).
ICON:	List of rigid body mode numbers.
ID,ID1,ID2,IV,etc.: (JD,...etc)	Indicated output type and numbers for the load case (individual load).
IL:	Load type indication. IL=1 for harmonic force at any mass point; IL=2 for elliptic load; IL=3 for harmonic displacement at some or all constraint points; IL=4 for force input described by data points; IL=5(6,7) for displacement (velocity or acceleration) described by data points at any or all constraints.
IROT(IROT1,IROT2):	Indicator that rows and columns, IROT1 and IROT2 of the topological matrices must be interchanged. Activated if NOPTIO < 0.
ISS:	Binary option (1=yes) for steady state output for harmonic inputs.
ITR(NT):	Binary transient response indicator for individual load (for whole load condition with NL loads).
N:	Total number of modal degrees of freedom.
NCL:	Two uses; if NOPTIO $\neq 0$, number of degrees of freedom having imposed kinematical constraints including supports; if NOPTIO = 0, number of closed loops (close coupled) among modal degrees of freedom.
NCON:	Number of rigid body modes.

NDP:	Damping indicator. NDP = -1 for modal damping, NDP = 0 for undamped, NDP > 0 for input viscous damping.
NH:	Number of output step changes (for IL $\geq$ 4).
NL:	Number of separate loads in the load condition. (A constraint motion counts as one load).
NLC:	Number of load conditions with NL loads.
NM:	Number of translatable masses. N = NM + NROT.
NOPTIO:	<p>Various input options. NOPTIO = 0, compute topological matrices from input data. (Linear and rotational spring and damper elements only); =</p> <p>1, read in matrices, mass matrix is diagonal; =</p> <p>2, same as 1 but non-diagonal mass matrix; =</p> <p>3, read in diagonal mass matrix and other non-zero terms of UT matrices; =</p> <p>4, same as 3 but non-diagonal mass matrix; (> 5, tape input using various formats); =</p> <p>5, separate records, diagonal mass, other matrices by rows, =</p> <p>6, one record (N, MASS(IJ), J=1, N, I=1, N, S(I,J)..., CX (I,J)...), =</p> <p>7, same as 6 but only diagonal mass values used; =</p> <p>8, read modal results from previous run TAPE 10 with diagonal mass; =</p> <p>9, same as 8 but non-diagonal mass matrix; -J, set NOPTIO = J and IROT = 1 (For any value of NOPTIO $\leq$ 8, modal damping values may be read in on cards.)</p>
NROT:	Number of rotational variables if NOPTIO = 0. If NOPTIO $\neq$ 0, N = NM + NROT governs).
NS:	Number of linear springs if NOPTIO = 0. If NROT $\neq$ 0, and NOPTIO = 0, also number of rotational springs.
NT:	Binary transient indicator for whole load condition. Parameter outputs are described by indicators ID,...etc.
NTP:	<p>Tape parameter</p> <p>=0, tapes not used</p> <p>=1, use tape 10 for input (NOPTIO $\geq$ 5)</p> <p>=2, use tape 9 for output of modal and response solutions</p> <p>=-2, use tape 9 for output of response results only</p> <p>=3, use both tapes.</p>

NDISC, TDISC:	Number of smoothness discontinuities of data point input data point function located at various time points $TDISC_i$. (Used if quartic interpolation is used to generate additional data points for $IL \geq 4$).
OFFSET:	Length of rigid arm connecting possible offset mass from CM ($NOPTIO = 0$).
OMEGA:	(rad/sec) Harmonic forcing frequency for individual load. Note N harmonic loads with various frequencies can act at the N respective degrees of freedom for one load condition along with N elliptic loads.
P:	Matrix of particular solutions for undamped harmonic forcing.
PD:	Matrix similar to above containing principally damping and phase effects.
ROT:	Vector of rotational spring constants.
S:	Stiffness matrix (also used for storage of non-diagonal mass matrix after modal solution).
SR:	Coupling stiffness matrix between constrained coordinates and nodal degrees of freedom.
ST:	Vector of linear spring constants.
TM:	Vector of time intervals used for elliptic loads.
TSTEPH:	Vector of output step change times.
VECTS:	$N \times N$ General Vector of eigenvectors stored by column
Z:	Vector (with time as first element) of tape response output for each time step.
ZP,ZD,ZC,ZM:	Vectors representing the time independent matrix multiplications indicated by equations 18 for force, displacement, velocity, and acceleration, respectively.

4.2 DESCRIPTION OF THE SUBROUTINES

In this section, each of the various subroutine functions will be described. The general feature of the program that should be noted is that the main functions and options are not segmented into different subroutines as is common today. Actually, this practice is

a debugging and compiling tool, but the writer preferred the older and more centralized main program approach. Extensive useage is made of equivalence statements to make maximum use of the available storage locations.

Main SPRVIB: This centralized main program handles all the principal computations for all the options. Because it is large, some difficulty may be encountered in compiling this main program on some systems. In this case, the user can easily segment the program, for example, between the modal and the response portions of the solution. Other segments can also be used. The basic equations used in this program are detailed in Section 2.

POINTL: This subroutine, which is used for $IL = 4, \dots, 7$, generates any additional data points by linear or quadratic interpolation taking account of output time step changes and also any smoothness discontinuities in the data point function. It calls subroutines EXTRL, EXTRR, XD1,...XD4.

EXTRL, EXTRR: These subroutines, called by POINTL, compute additional fictitious data points to the left or right of a discontinuity step change or time boundaries. In general, both of these subroutines store the actual data points before they overwrite these locations with fictitious data points and return the correct data points after the critical interpolation point is passed. These subroutine calls XD1, and XD2.

XD1, XD2, XD3, XD4: These routines compute the first, second, third, and fourth central differences, respectively, of a data point function. They are chiefly called by POINTL when the quadratic interpolation option is used.

All the following subroutines are called by SPRVIB.

LOAD: This subroutine evaluates the damped form of Duhamel's integral and its time derivative for polynomial functions $(t - t_i)^j$, $j = 0, \dots, 4$.

STIFF: This subroutine computes a stiffness and damping matrix for a linear and rotational spring and damper connected body.

READA, READB: These subroutines read and print a square and diagonal matrix, respectively. Only the diagonal terms are input for the latter while the former reads in the matrix by separate rows.

TRANS: This subroutine converts the initial data in regular coordinates to normal coordinates using the relation $Z^{-1} = MZ^T$ where Z is the matrix of eigenvectors and M is the mass matrix. Note that the mass matrix must be stored until this subroutine is called, once for each load condition and once for each load, if required.

TRIANG, TRINV: These routines use Choeleski-Turing decomposition $M = B^T B$ to form the matrices B and B^{-1} .

ROTATE: This subroutine performs indicated row and column interchanges.

INPUTM: This subroutine inputs the location and value of the non-zero values in the upper triangular part of a symmetric matrix. It is especially useful in inputting sparse matrices. Numbering in the matrix is done consecutively in rows. (The numerical location of the 3rd diagonal term of an $N \times N$ matrix is $2*N + 3$.)

OUTPUT: This subroutine prints a square matrix by rows.

CHANGE: This subroutine outputs the requested values in regular coordinates using the normal mode solutions.

EIGEN: This subroutine based on one of the IBM SP routines of the same name, finds the eigenvalues and eigenvectors of a dynamic matrix using Jacobi's method of successive rotations. The final norm ANRMX in this subroutine may be adjusted for a particular computer system or for a problem in order to obtain a better eigenvector determination. Of course, there exists a tradeoff between running time and extreme accuracy. The eigenvectors in this routine are orthoganalized with respect to unity, so that transformed in the normal coordinates they satisfy the relation $Z^T M Z = I$.

The flow chart for the topological portion of the program when the modal damping option is used is shown in Figure 17. It should be noted that all subroutines return to SPRVIB. The flow chart for the response portion of the program when transient response option is chosen is shown in Figure 18.

4.3 INPUT INSTRUCTIONS

The following input instructions are illustrated by the example problems of Chapter 3.

Notes: 1) Shown for modal damping and with rotations
 2) All subroutines return to SPRVIB

Figure 17. Flow Chart for the Modal Portion of the Solution

- Notes:
- 1) Can be called for each load condition and/or each load
 - 2) Called for each load condition
 - 3) Called once for each output region
 - 4) Called for each output time

Figure 18. Flow Chart for the Response Portion of the Solution

INPUT INSTRUCTIONS

Topological Section

Case (B): Generate stiffness matrix for discrete element case (NOPTION = 0).

Case (A): (NOPTION $\neq$ 0) Read in element matrices previously computed by direct stiffness method or other linking program.

1. Card / TITLE (18A4)
2. Card 2 (.8110) NM, NS, NLC, NROT, NCL, NTP, NDP, NOPTION

Case A (NOPTION $\neq$ 0)

NM = Number of degrees of freedom (D.O.F.) of structure including supports

NS = 0 not used

NLC = Number of loading cases

NROT = 0 not used

NCL = Sum number of D.O.F. of supports

NTP = 1 use Tape 10 for input
2 use Tape 9 for output of modal and forced results
-2 use Tape 9 for output of responses only
3 use both tapes.

NDP = 0 if structure has negligible damping
1 if damping matrix is input
-1 if modal damping coefficients are input

NOPTION 0 See Case B
1 Read in diagonal M matrix, and K, C matrices by rows
2 Same as above except M non-diagonal
3 Read in diagonal M matrix and use input option to read non zero elements of K & C matrices
4 Same as above except M non-diagonal
5 Read in diagonal mass matrix, damping and stiffness matrices by rows off Tape 10
6 Same as (5) but mass matrix not diagonal

7 Same as (6) but use only the diagonal element of the mass matrix
-J., J = 1, . . . 7 Same as NOPTION = J but prepare to rotate
 NOPTION = columns and rows of the matrices
8 Read topological results off Tape 10.
9 Same as 8 but mass matrix is not diagonal.

 For NOPTION = 1, 2

3. Card(s) (6E12.4) Mass Elements
 Diagonal Mass Elements or one row at a time
4. Card(s) (6E12.4) Modal Damping Cards (If NDP < 0)
 Percent of critical damping in the mode ((NM-NCL) constants read in)
5. Card(s) Stiffness Elements (6E12.4) by rows
6. Card(s) Physical Damping Matrix (6E12.4) by rows (If NDP > 0)

 For NOPTION = 3

- 3.a. Card(s) (6E12.4) Diagonal Mass Elements
- 4.a. Same as 4., above
- 5.a. Card(s) (2I4) NC, NFIRST (For stiffness element)
 NC = Number of non-zero symmetric entries in N x N matrix
 NFIRST = nodal designation for first element (usually = 1)
 Card(s) (20I4) Upper triangular nodal designation of non-zero element ordered by
 rows (NC locations)
 Card(s) (6E12.4) Values for these non-zero elements in upper triangular symmetric
 array
- 6.a. Card(s) (2I4) NC, NFIRST (For damping elements if NDP > 0)
 Card(s) (20I4) (IN(I), I = 1, NC)
 Card(s) (6E12.4) (B(I), I = 1, NC)

53

(In preparing his input, the user should first sketch his topological model. Starting from some fixed point labeled zero, the user should arbitrarily number his discrete masses 1, . . . NM. All other points are assigned consecutive numbers higher than NM. The respective linear spring, rotational spring or damper is then assigned the largest number of the points to which it attaches. Following this scheme, some fictitious elements with zero values may be read in to simplify the internal logic.)

3. Card(s) (6E12.4) NS linear spring values in order 6 to card
4. Card(s) (6E12.4) NS linear damping values (If NDP > 0)
5. Card(s) (6E12.4) NS rotational springs (If NROT > 0)
6. Card(s) (6E12.4) N=NM+NROT modal damping constants (If NDP < 0)
7. Card(s) (2F12.5, 3I6, F12.5, I6) Mass, Cent ($\bar{I}$), JL, JC, JR, OFFSET, JO
 (NM cards) Mass is the mass of the discrete element
 CINT is the centroidal moment of inertia (=0 if mass does not rotate)
 JL, JC, JR are the ordered points to which mass is attached. When mass is attached to 3 points all indexes are used ordered in increasing number. If mass is attached to 1 point only, JL is used (JC=JR=0)
 OFFSET is the transverse distance between the C.G s of masses connected by rigid arm in case that masses can also rotate.
 JO is the mass number to which the rigid arm is directly attached.
8. Card(s) (4I10) Closed Loop Cards (If NCL > 0, NCL cards)
INDEX, NMASS, NMASSI, NSPR
INDEX = 1 for only linear springs
INDEX = 3 for both linear and rotational springs
NMASS, NMASSI are masses connected by actual closed loop $NMASS < NMASSI$
NSPR is the artificial number of attaching spring

Part II - Forced Motion Input

For each load condition a set of the following cards must appear, i.e., NLC

Subdecks of data:

1. Card General Card: (NL, NT, ID, ID1, ID2, IV, IV1, IV2, IA1, IA2,
NINC, TMAX)(1112, 16, E12.4)

NL is the number of loads in the load condition

NT is the logical transient output indicator

= 0 for no transient output, i.e., steady state output only

= 1 for transient output

ID (IV, IA) are transient output indicators for each load condition

= 0 no displacements (velocities, accelerations) are output

= 1 specified quantities are printed out

= 2 specified quantities are both printed out and put on tape

ID1, ID2 } number of the respective masses for which the relevant data is desired

IV1, IV2 } If all the velocities, for example are desired, IV1 should be set equal

IA1, IA2 } to zero. (Of course, $IV \neq 0$ in this case.)

NINC is the number of time increments desired in the output for the load condition.

TMAX is the maximum time of the output, i.e., output occurs for $0 \leq t \leq TMAX$

2. Card(s) IF { Initial Data Cards:
 NT $\neq$ 0 { Initial displacements in order 6 per card (6E12.4)
 { Initial velocities in order 6 per card (6E12.4)
 For each load (2) cards

3. Card (1) General Card (IL, ITR, ISS, JD, JD1, JD2, JV, JV1, JV2, JA, JA1, JA2, NJC, TMAX1) (1212, 16, E12.4)

IL is load type = 1 harmonic load

= 2 elliptic shock load

= 3 support harmonic acceleration

= 4 point force input

= 5 point support displacements

= 6 point support velocities

= 7 point support accelerations

ITR is transient output indicator for this load alone

= 0 no separate transient output

= 1 transient output as indicated for this load alone

ISS is an indicator for harmonic loads alone

= 0 no separate steady state output

= 1 maximum steady state output for load alone

JD,JA2 indicator as before for this load acting alone

NJC number of time increments for local transient output

TMAX1 maximum time for which transient output

4. Card Specific Card: IF IL = 1; P, OMEGA, IM, (2E12.4, 16)

P is the load amplitude

OMEGA is the harmonic load frequency

IM is the generalized mass number at which the load acts

- 4.a. Card IF IL = 2, (IM, PE1, BT1, TIM1, PE2, BT2, TIM2, PE3, BT3, TIM3, PE4, BT4, TIM4) (12, 12F6.4)

IM is the generalized mass number at which the load acts.

PE1, BT1, TIM1,, PE4, BT4, TIM4 are defined by

$$P(t) = \begin{cases} PE1 * EXP(BT1 * T), & 0 \leq t < TIM1 \\ \vdots & \vdots \\ PE4 * EXP(BT4 * T), & TIM3 \leq t < TIM4 \\ 0, & t \geq TIM4 \end{cases}$$

4. b. Card(s) (6E12.4) OMEGA, (GS(I), I=1, NCL) (IF IL = 3)

OMEGA is the frequency of support harmonic acceleration

GS(I) is the acceleration amplitude at the respective node numbered from N-NCL+1, ..., N. (This uses $G=386.4 \text{ in/sec}^2$)

4. c. For IC = (4, 5, 6 or 7)

Card (9I4, 3E12.4) NPOINTS, NDISC, NH, (KEY(I), I=1, 6) TFIRST, TLAST, TSPAC

NPOINTS is the number of data points (input points)

NDISC is the number of slope discontinuities (used when 4th order interpolation is used)

NH is the number of regions with constant output spacing

KEY(1) = $\begin{cases} -1 & \text{input evenly spaced, use as output} \\ 0 & \text{input not evenly spaced, must read in time points} \\ 1 & \text{input evenly spaced, multiple output regions} \end{cases}$

KEY(2) = $\begin{cases} 0 & \text{output solution points to be specified (NH > 1)} \\ 1 & \text{output points are at input points and midway between them} \end{cases}$

KEY(3) = is used internally for extrapolation routines

KEY(4) = $\begin{cases} 0 & \text{use linear interpolation between input data points to compute additional data points} \\ 1 & \text{use 4}^{\text{th}} \text{ order interpolation between input data points to compute additional data points} \end{cases}$

KEY(5), KEY(6) are used internally

TFIRST is the arbitrary initial time of the force reading (usually=0)

TLAST is the arbitrary time of the last data point

TSPAC is the spacing of uniform input data points (of KEY(1) ≠ 0)

Card(s) (6E12.4) DATA(I), I=1, NPOINTS

Card(s) (6E12.4) TIME(I), I=1, NPOINTS, (IF KEY(1)=0)

Card(s) (6E12.4) H(I), (TDISC(I), I=1, NDISC) (If NH=1)

or (H(I), I=1, NH), (TSTEPH(I), I=1, NSTEPH), (TDISC(I), I=1, NDISC)

(If NH ≠ 1)

$H(I)$ is the time output spacing in the I^{th} output region

$TSTEP(I)$ is the boundary time between the output regions I and $I+1$

$TDISC(I)$ is the location of the I^{th} discontinuity (used if 4th order interpolation is requested to generate additional internal data points)

Card(s) (12F6.0) (AL(I), $I=1$, NCL) SD, SV (If IL = 5, 6, or 7)

AL(I) is the scale factor (usually equal to 1 or 0) at the support $N+I$

(AL(J)=0 says the J^{th} support coordinate is not moving, while AL(J)=1 says that J^{th} support coordinate is moving with a parameter equal exactly to the input data), SD and SV are initial scaled constraint displacement and velocity.

Card (12) IM (If IL = 4) IM is generalized mass number of point load.

ADDITIONAL NOTES AND OPTIONS

1. Tape output is always in unformatted mode in binary. In this mode, one record is defined by each write statement, that is, each time increment of output is one record.
2. One can consider free vibrations by letting $NLC = 1$, $NL = 0$ or a combination of free vibrations in one load condition and forced vibrations in another condition. Initial conditions must be read in for each load condition however.
3. If only natural frequencies and modes are desired $NLC = 0$.
4. All springs and dampers are assumed to be attached to the center of mass of each element. The forces and torques must act at the center of mass.
5. A harmonic load and a shock load may act at the same generalized mass; however, two harmonic loads or two shock loads may not act at the same generalized mass.
6. Besides the output information, the list of output is presented in a table for each load as well as for each load condition.
7. In general, it is difficult to adjust the damping constants to insure proportional damping. The program assumes the damping is proportional, but also explicitly checks this assumption and prints out this check. Thus cases in which the damping is almost proportional can be solved approximately.

8. Shock type loads can be easily fitted to the elliptic type loads used in the program with different parameters in different time regions. The great advantage of this type of formulation over a more conventional pointwise time input is the accuracy with which velocities and accelerations can be obtained from even crude force approximations.
9. The program will handle a system of up to 90 degrees of freedom, 90 linear springs and 90 rotational springs without modification.

REFERENCES

1. Koeing, H. J., and D. I. Drain, "Method of Relating Modal Damping to Local Dampers in Lumped-Parameter Systems," NASA-TN-D-3637, 1967.
2. Bolton, R., "Dynamic Analysis Report Nuclear Subsystem," Westinghouse Astronuclear Laboratory report, WANL-TME-2755, 1971.
3. Caughey, T. K. and M. E. J. O'Kelly, "Effect of Damping on the Natural Frequencies of Linear Dynamic Systems," Journal of the Acoustical Society of America, November, 1961.
4. Wilkinson, J. H., "The Algebraic Eigenvalue Problem," Oxford University Press, Oxford, England, 1965.
5. Bolton, R., "ETS-1 Test Stand Dynamic Analysis for a Seismic Event," Westinghouse Astronuclear Laboratory Data Release Memorandum 51867, 1969.
6. Warburton, G. B., "The Dynamical Behavior of Structure," The Macmillan Company, New York, 1964.
7. Pestel, E. C., and F. A. Leckie, "Matrix Methods in Elastomechanics," McGraw Hill Book Company, New York, 1963.
8. Harris, C. M., and C. E. Crede, (ed.), "Shock and Vibration Handbook, McGraw Hill Book Company, New York, 1961.

APPENDIX A - Listing of SPRVIB Program and Subroutines

```

PROGRAM SPRVIB(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=PU
1NCH,TAPE10,TAPE11,TAPE9)
C      ****THIS PROGRAM WAS DEVELOPED BY R. BOLTON , WESTINGHOUSE EL. CO.****
C      ****ASTRONUCLEAR LABORATORY, LARGE PA. , IT IS NOT GUARANTEED      ****
C      ****TO BE FREE OF ALL ERRORS , HOWEVER EXTENSIVE CHECKS WERE MADE ****
C      ****OF ITS ACCURACY , IF ANY PROBLEMS DEVELOP PLEASE CONTACT AUTHOR****
000003      COMMON/AAAAA/AMM(90)
C
000003      DIMENSION DIX(90),DIY(90),VIX(90),VIY(90),PD(90,90)
000003      DIMENSION S(90,90),P(90,90),FR(90),ST(90),EP(90,90),TM(360),
1BETA(360),DYM(4095),EEP(360),LPHA(90),ROTI(90),A(90),B(90)
000003      COMMON/LIST/ AM(90),VECTS(8100)
000003      COMMON/GAMMA/Z(270),NUM
000003      DIMENSION AS(90),AT(90),VS(90),VT(90),DS(90),DT(90),FREQ(90)
000003      DIMENSION CX(90,90),C(90),CY(90,90),CL(90),CR(90)
000003      DIMENSION TITLE(20),ICCN(10)
000003      DIMENSION SR(90,9),CCR(90,9),AMR(90,9)
000003      DIMENSION PB(4002),KEY(6),H(10),AL(9),AHI(90,5),AHJ(90,5),ZP(90),
1      ZD(90),ZC(90),ZM(90) ,TSTEPH(10)
000003      EQUIVALENCE (P(1),CX(1),FB(1)),(PD(1),CY(1),AHI(1)),(EP(1),S(1),
1AHJ(1)),(DYM(1),EEP(1),ZP(1)),(DYM(361),AT(1)),
2      (DYM(451),VS(1)),(DYM(541),VT(1)),(DYM(631),DS(1)),
3      (DYM(721),DT(1)),(DYM(811),FREQ(1)),(DYM(901),DIX(1)),
4      (DYM(991),DIY(1)),(DYM(1081),VIX(1)),(DYM(1171),VIY(1))
5,(DYM(1261),BETA(1),ZC(1)),(DYM(1621),TM(1),ZD(1)),
6      (B(1),DYM(1981)),(DYM(2071),AS(1))
7,(DYM(2161),ROTI(1),ZM(1)),(DYM(2251),TITLE(1),TSTEPH(1))
8 , (DYM(2271),A(1)),(DYM(2361),C(1))
9,(DYM(2451),H(1)),(DYM(2461),AL(1)),(DYM(2471),KEY(1))
000003      9 READ(5,133) ( TITLE(I),I=1,20)
000015      WRITE(6,133) ( TITLE(I),I=1,20)
000027      READ(5,101) NM,NS,NLC,NRCT,NCL,NTP,NDP,NCPTIC
000053      WRITE(6,715)
000057      WRITE(6,101) NM,NS,NLC,NROT,NCL,NTP,NDP,NCPTIC
C      READ AND PRINT PARAMETERS. (NM) NUMBER OF MASSES, (NS) NUMBER OF
C      LINEAR SPRINGS, (NF)NUMBER OF END FIXED POINTS,(NROT) NUMBER OF
C      RCTATIONAL DEGREES OF FREEDOM , (NCL) NUMBER OF CLOSED LOOPS
C
C      IF NTP=0 NO TAPES
C      IF NTP=1 USE TAPE 10 FOR INPUT
C      IF NTP=+2 USE TAPE 5 FOR OUTPUT OF MODAL AND FORCED RESULTS

```

```

C      IF NTP=-2 USE TAPE 3 FOR OUTPUT OF RESPONSES ONLY
C      IF NTP=3  USE BOTH TAPES
C
C      NOPTION=0 COMPUTE STIFFNESS AND MASS MATRICES FROM INPUT DATA
C      NOPTION=1 READ IN STIFFNESS, DAMPING AND DIAGONAL MASS MATRICES
C      NOPTION=2 SAME AS ABOVE EXCEPT MASS MATRIX IS NOT DIAGONAL
C      NOPTION=3 READ IN DIAG MASS MATRIX AND NON-ZERO ENTRIES IN STIFF.+DAMPING
C      NOPTION=4 READ IN NON ZERO ENTRIES FOR MASS, STIFFNESS AND DAMPING MATRICES
C      NOPTION=5 TAPE INPUT SEPARATE RECORDS , DIAG MASS, STIFFNESS CONECT. ROWS
C      NOPTION=6 TAPE INPUT ONE RECCRD (N, MASS(I,J), J=1,N), I=1,N), (S(I,J),.....))
C      NOPTION=7 SAME AS ABOVE BUT ONLY DIAGONAL TERMS OF MASS MATRIX ARE USED
C      NOPTION=8 READ TGPOLCCICAL RESULTS FROM PREVIOUS RUN OFF TAPE 10
C      NOPTION=9 SAME AS ABOVE BUT MASS MATRIX IS NOT DIAGONAL
C      (FOR NOPTION.LT.8, MODAL DAMPING CONSTANTS MAY BE READ IN ON CARDS )
C      IF NOPTION=-J SET NOPTION =J AND PREPARE TO ROTATE INPUT MATRICES
C
000103      IF(NTP.EQ.(1).OR.NTP.EQ.3) REWIND 10
000115      IF(NTP.LE.-2.OR.NTP.GE.2) REWIND 9
C
000130      N=NM+NRCT
000132      IROT=0
000133      IBM=0
000134      IF(NOPTIO .LT.0) IROT=1
000136      IF(NOPTIO .LT.C) NOPTIO =-NOPTIO
000141      IF(NOPTIO .LT.8) GO TO 649
000143      NSQ=N*N
000145      IF(NOPTIO .GT.8) GO TO 662
000150      READ(10) N, (FR(I), I=1,N), (C(I), I=1,N), (VECTS(I), I=1, NSQ), (AM(I), I
1          =1,N)
000206      GO TO 663
000207      662 READ(10) N, (FR(I), I=1,N), (C(I), I=1,N), (VECTS(I), I=1, NSQ), ((S(I,J),
1          J=1,N), I=1,N)
000252      663 IF(NCL.GT.0) READ(10) ((SR(I,J), J=1, NCL), I=1,N), ((CCR(I,J), J=1,
1          NCL), I=1,N)
000307      GO TO 125
000310      649 IDP=C
C
C      MODAL DAMPING CORRECTIONS
C
000311      IDP=C
000312      IF(IDP.EQ.-1) IDP=1

```

```

000315      IF(NDP.EQ.-1)    NDP=0
C      DGES USER WISH TO INPUT CWN STIFFNESS MATRIX AND PERHAPS DAMPING MATRIX
000317      IF(NOPTIO .EQ.0) GO TO 10
000320      IF(NOPTIO .EQ.(2).OR.NOPTIO .EQ.(4)) GO TO 8
000327      IF(NOPTIO .EQ.3) GO TO 602
000330      IF(NOPTIO -5) 641,642,643
000333      642 WRITE(6,121)
000337      READ(10)(AM(I),I=1,N)
000351      WRITE(6,103)(AM(I),I=1,N)
000364      LI=N-NCL
000366      CALL READ3(LI,IDP,CL)
000371      NOPTIO =1
000372      WRITE(6,122)
000376      READ(10)((S(I,J),J=1,N),I=1,N)
000415      CALL OUTPUT(N,S)
000417      NOPTIO =1
000420      IF(NDP.LE.0) GO TO 78
000422      READ(10)((CX(I,J),J=1,N),I=1,N)
000441      WRITE(6,120)
000445      CALL OUTPUT(N,CX)
000447      GO TO 78
000450      643 READ(10) N,((CY(I,J),J=1,N),I=1,N),((S(I,J),J=1,N),I=1,N)
000504      IF(NOPTIO .NE.7) GO TO 647
000506      DO 646 I=1,N
000510      646 AM(I)= CY(I,I)
000520      WRITE(6,121)
000523      WRITE(6,103)(AM(I),I=1,N)
000536      647 WRITE(6,645)
000542      IF(NOPTIO .EQ.6) NOPTIO =2
000545      IF(NOPTIO .EQ.7) NOPTIO =1
000550      CALL OUTPUT (N,CY)
000552      IF(NCL.GT.0) GO TO 39
000555      JJJ=N-NCL
000556      JJJ1=JJJ+1
000560      DO 38 I=1,JJJ
000561      DO 38 J=JJJ1,N
000563      38 AMR(I,J-JJJ)=CY(I,J)
000600      39 CALL TRIANG(CY,CX,N)
000603      CALL TRINV(CX,CY,N)
000606      WRITE(6,122)
000612      CALL OUTPUT (N,S)
000614      IF(IDP.NE.1) GO TO 78

```

```

000616      LI=N-NCL
000620      CALL READ3(LI,1,CL)
000623      GO TO 76

C
000624      641 WRITE(6,121)
000630      CALL READ3(N,2,AM)
000633      CALL READ3(N,IDP,CL)
000636      WRITE(6,122)
000642      I=1
000643      79 CALL READA(N,I,S)
000646      IF(I.EQ.N) GO TO 7
000650      I= I+1
000651      GO TO 79
000652      7   IF(NDP.EQ.0) GO TO 73
000653      WRITE(6,120)
000657      DO 11 I=1,N
000661      CALL READA(N,I,CX)
000663      11  CONTINUE
000666      GO TO 78
000668      602 WRITE(6,121)
000672      CALL READ3(N,2,AM)
000675      IF(IDP.EQ.0) GO TO 605
000676      LI=N-NCL
000700      CALL READ3(LI,IDP,CL)
000703      GO TO 605
000704      603 CALL INPUTM(N,CY)
000706      IF(NOPTIO.EQ.4)NOPTIO =2
000711      WRITE(6,645)
000715      CALL OUTPUT (N,CY)
000717      IF(NCL.EQ.0) GO TO 35
000720      JJJ=N-NCL
000722      JJJ1=JJJ+1
000724      DO 34 I=1,JJJ
000725      DO 34 J=JJJ1,N
000727      34 AMP(I,J-JJJ)=CY(I,J)
000744      35 CALL TRIANG(CY,S,N)
000747      CALL TRINV(S,CY,N)
000752      LI=N-NCL
000754      CALL READ3(LI,IDP,CL)
000757      605 CALL INPUTM(N,S)
000761      WRITE(6,122)
000765      CALL OUTPUT (N,S)

```



```

000767      IF(NOP.EQ.0) GO TO 79
000770      CALL INPUTM(N,CX)
000772      WRITE(6,120)
000776      CALL OUTPUT (N,CX)
001000      GO TO 78
001001      8 WRITE(6,94)
001005      IF(NOPTIO .EQ.4) GO TO 603
001007      DO 6 I=1,N
001011      CALL READA(N,I,CY)
001013      WRITE(6,103)(CY(I,J),J=1,N)
001030      IF(NCL.EQ.0) GO TO 5
001031      JJJ=N-NCL+1
001033      DO 12 J=JJJ,N
001034      12 AMR(I,J+NCL-N)=CY(I,J)
001047      6 CONTINUE
001052      CALL TRIANG(CY,S,N)
001054      CALL TRINV(S,CY,N)
001057      LI=N-NCL
001061      CALL READB(LI,IDP,CL)
001064      WRITE(6,122)
001070      DO 5 I=1,N
001072      CALL READA(N,I,S)
001074      5 CONTINUE
001077      CALL OUTPUT(N,S)
001101      WRITE(6,93)
001105      CALL OUTPUT(N,CY)
001107      GO TO 7
001110      10 WRITE(6,100) NM,NS,NLC,NROT
001124      WRITE(6,114) NCL
C      READ IN SPRING CONSTANTS IN ORDER
001132      CALL STIFF(NM,NS,NROT,NCL,NOP,N, IDP,INDEX,S,ST,LPHA,ROT,CX,CL,CR)
001151      IF(INDEX.EQ.1) GO TO 9
C      CHECK FOR ROW AND COLUMN ROTATION
001153      79 IF(IRCT.EQ.0) GO TO 606
C
C      ROTATE ROWS + COLUMNS INDICATED ON DATA CARDS
001154      609 READ(5,607) IROT1,IROT2
001164      IF(IROT1.EQ.0) GO TO 606
001165      CALL ROTATE (IROT1,IROT2,S,N)
001170      607 FORMAT(3I6)
001170      IF(NOPTIO .EQ.(2)) GO TO 608
001172      ABC=AM(IRCT1)

```

```

001174      AM(IROT1)=AM(IROT2)
001176      AM(IROT2)=ABO
001200      IF(NDP.NE.0) CALL ROTATE (IROT1,IROT2,CX,N)
001204      GO TO 609
001205 608 CALL ROTATE (IROT1,IROT2,CY,N)
001210      IF(NDP.NE.0) CALL ROTATE (IROT1,IROT2,CX,N)
001214      GO TO 609
001215 606 IF(NOPTIO.EQ.(0).OR.NCL.EQ.(0)) GO TO 612
001224      NM=N-NCL
001225      N=NM
001226      DO 731 I=1,N
001227      IF(IDP.EQ.1) CR(I)=S(I,I)
001235      DO 731 J=1,NCL
001237      IF(NDP.GT.0) CCR(I,J)= CX(I,N+J)
001247      IF(IDP.GT.0) CCR(I,J)=CL(N)*S(I,N+J)
001260 731 SR(I,J)=S(I,N+J)
001273 612 IF(NOPTIO.NE.2) GO TO 95
001275      M=0
001276      DO 99 J=1,N
001277      M=M+J-1
001301      DO 99 I=1,J
001303      SUM=0.0
001304      DO 98 K=1,I
001306      DO 98 L=1,J
001307      SUM=SUM+CY(K,I)*CY(L,J)*S(K,L)
001323 98 CONTINUE
001330      K=M+I
001331 99 DYM(K)=SUM
001340      GO TO 97
001341 95 DO 77 J=1,N
001343      DO 77 JJ=1,N
001344      S(J,JJ) = S(J,JJ)/ SQRT(AM(J)*AM(JJ) )
001355 77 CONTINUE
C      CHANGE MODE OF 5 TO ONE DIMENSIONAL STORAGE IN SYMMETRIC MODE
001361      M=0
001362      DO 66 J=1,N
001363      M=M+J-1
001365      DO 66 I=1,J
001367      K=M+I
001371      DYM(K)= S(I,J)
001375 66 CONTINUE
C

```

```

C      COMPUTE EIGENVALUES AND EIGENVECTORS IN NORMAL COORDINATES
C
001401 97 CALL EIGEN(DYM,VECTS,NM,0)
C      READ OUT EIGENVALUES IN ORDER
001404 DO 88 J=1,N
001406     K1 = (J+J-J)/2 +J
001411     IF(DYM(K1))97,88,88
001413 87 WRITE(6,137) J
001421 88 FR(J)= SQRT(ABS(DYM(K1)))/6.2831953
C
C      CORRECT MODES TO REGULAR COORDINATES
C
001432 IF(NCPTIO .NE.2) GO TO 89
001434 DO 92 I=1,N
001436 KI=-N
001440 DO 92 J=1,N
001441 KI=KI+N
001443 SUM=0.0
001444 DO 91 K=1,N
001445 KJ=KI+K
001447 91 SUM=SUM+CY(I,K)*VECTS(KJ)
001456 92 S(I,J)=SUM
001466 DO 3 J=1,N
001467 IK=J*N-N
001472 DO 3 I=1,N
001473 IJ=IK+I
001475 3 VECTS(IJ)=S(I,J)
001505 CALL TRINV(CY,S,N)
001507 GO TO 90
001510 89 DO 75 I= 1,N
001512 LMJ=I-N
001514 DO 75 J=1,N
001515 LMJ=LMJ+N
001517 VECTS(LMJ) = VECTS(LMJ)/SQRT(AM(I))
001524 75 CONTINUE
001530 90 WRITE(6,107)
001534 WRITE(6,103)( FR(J),J = 1,NM)
001547 IF(NDP.EQ.0) GO TO 76
001550 WRITE(6,123)
001554 DO 85 J=1,N
001556 DO 84 L=1,N
001557 CY(J,L)=C.C

```

```

001562      DO 84 I=1,N
001564      IJ=(J-1)*N+I
001570      DO 84 K=1,N
001571      KL=(L-1)*N+K
001575      84 CY(J,L)=CY(J,L)+ CX(I,K)*VECTS(IJ)*VECTS(KL)
001615      WRITE(6,103) (CY(J,L),L=1,N)
001631      85 CONTINUE
001634      DO 86 J=1,N
001635      C(J)= CY(J,J)
001641      CRIT = C(J)/12.56637
001643      IF (FR(J).GT.(1.0E-4C))CL(J)=CRIT/FR(J)
001650      IF (FR(J).LT.ABS(CRIT)) WRITE(6,126) J
001661      FR(J)= SGRT(FR(J)*FR(J)-CRIT*CRIT)
001670      86 CONTINUE
001673      WRITE(6,111)(CL(J),J=1,N)
001705      WRITE(6,128)
001711      WRITE (6,103) (FR(J), J=1,N )

C
C      MODAL DAMPING CORRECTIONS
C

001724      76 IF (IDP.EQ.0) GO TO 82
001725      DO 91 J=1,N
001727      CRIT=CL(J)*FR(J)*AMM(J)
001732      C(J)=CRIT*12.56637
001734      81 FR(J)=SGRT(FR(J)*FR(J)-CRIT*CRIT )
001745      WRITE(6,128)
001750      WRITE(6,103)(FR(J),J=1,N)
001763      WRITE(6,241)(AMM(J),J=1,N)
001776      WRITE(6,242)(C(J),J=1,N)
002011      NDP=1
C      COMPUTE MATRIX OF SUPPORT/STRUCTURE DAMPING INTERACTIONS BY PROPORTION
002012      IF (NOPTIO .EQ.(0).OR.NCL.EQ.0) GO TO 82
002020      DO 666 I=1,N
002021      SUM=0.0
002022      IF (NOPTIO .NE.2) GO TO 669
002024      DO 668 IA=1,N
002026      IAB=IA-N
002030      DO 668 ID=1,N
002031      IDB=ID-N
002033      AMID=C.0
002034      AMAI=C.0
002035      DO 667 J=1,N

```

```

002036      AMID=AMID+S(J,I)*S(J,IO)
002045      667 AMAI=AMAI+S(J,I)*S(J,IA)
002057      DO 668 IB=1,N
002061      IDB=IDB+N
002063      IAB=IAB+N
002064      668 SUM=SUM+AMID*AMAI*VECTS(IDB)*VECTS(IAB)*C(IB)
002102      GO TO 671
002102      669 IB=I-N
002104      DO 670 J=1,N
002106      IB=IB+N
002110      670 SUM=SUM+AM(I)**2*(VECTS(IB)**2)*C(J)
002121      671 DO 666 J=1,NCL
002123      666 CCR(I,J)=SUM*SR(I,J)/CR(I)
C
002141      92 WRITE(6,109)
C      READ CUT EIGENVECTORS SEVEN COMPONENTS TO EACH LINE
002145      NCCN=0
002146      DO 60 JJ=1,NM
002150      JK=JJ*NM
002152      JM=JK+1-NM
002154      IF(FR(JJ).GT.1.0E-3) GO TO 63
002160      NCCN=NCCN+1
002161      ICON(NCCN)=JJ
002162      63 FR(JJ)=FR(JJ)*6.2831853
002165      WRITE(6,110) FR(JJ)
002172      WRITE(6,103) (VECTS(KK),KK=JM,JK)
002205      60 CONTINUE
002210      IF(NDP.NE.0) WRITE(6,124)
002214      IF(NCCN.EQ.0) GO TO 652
002215      WRITE(6,68) NCCN,(ICON(JJ),JJ=1,NCCN)
002232      DO 74 JJ=1,NCCN
002234      JJJ=N*(ICON(JJ)-1)
002240      DO 69 KK=1,N
002241      J=KK+JJJ
002243      69 VECTS(J)=0.0
002246      74 CONTINUE
C
C      FORCED MOTION
C
002251      552 IF(NTP.LE.0) GO TO 125
C
C      STORE INFORMATION ON TAPE

```

```

C
C
002253      IF(NOPTIC .EQ.2) GO TO 664
002255      NSQ=N*N
002257      WRITE(9) N,(FR(I),I=1,N),(C(I),I=1,N),(VECTS(I),I=1,NSQ),(AM(I),
1          I=1,N)
002315      GO TO 665
002316      664 WRITE(9) N,(FR(I),I=1,N),(C(I),I=1,N),(VECTS(I),I=1,NSQ),((S(I,J),
1          J=1,N),I=1,N)
002361      665 IF(NOPTIC .NE.(C).AND.NCL.GT.0) WRITE(9) ((SR(I,J),J=1,NCL),I=1,N)
1          ,((CCR(I,J),J=1,NCL),I=1,N)

C
002423      125 IF(NLC .EQ.0) GO TO 325
002424      IF(NTP.LT.C) NTP=-NTP
002427      DO 500 JIA= 1,NLC
002431      WRITE(6,202) JIA
002436      READ(5,201)NL,NT,1D,ID1,ID2,IV,IV1,IV2,IA,IA1,IA2,NINCR,TMAX
002474      IF(NT.EQ.0) GO TO 167
002475      WRITE(6,203)
002501      WRITE(6,204)
002505      IF(ID.NE.0) GO TO 140
002506      WRITE (6,209)
002512      131 WRITE (6,211)
002516      132 WRITE (6,207)
002522      IF(IV.NE.0) GO TO 149
002523      WRITE (6,209)
002527      135 WRITE (6,211)
002533      136 WRITE (6,208)
002537      IF(IA.NE.0) GO TO 157
002540      WRITE (6,209)
002544      139 WRITE (6,211)
002550      GO TO 165

C
002551      140 PRINT ENTRIES IN TABLES
002552      141 IF(ID1) 145,141,145
002556      IF(ID.NE.2)GO TO 131
002560      WRITE(6,210)
002564      GO TO 132
002565      145 WRITE(6,206)ID1,ID2
002575      IF(ID.NE.2) GO TO 131
002577      WRITE (6,212)ID1,ID2
002607      GO TO 132

```

```

002610      149  IF (IV1.GT.0) GO TO 154
002613          WRITE (6,205)
002616          IF (IV.NE.2) GO TO 135
002620          WRITE (6,210)
002624          GO TO 136
002625      154  WRITE(6,206) IV1, IV2
002635          IF (IV.NE.2) GO TO 135
002637          WRITE (6,212) IV1, IV2
002647          GO TO 136
002650      157  IF (IA1.GT.0) GO TO 162
002653          WRITE (6,205)
002656          IF (IA.NE.2) GO TO 139
002660          WRITE (6,210)
002664          GO TO 165
002665      162  WRITE (6,206) IA1, IA2
002675          IF (IA.NE.2) GO TO 139
002677          WRITE (6,212) IA1, IA2
C
C      READ IN INITIAL DATA
002707      165  READ(5,148) (DIX(J), J=1, N)
002722          READ(5,148) (VIX(J), J=1, N)
C      CONVERT INITIAL DATA TO NORMAL COORDINATES
C
002735          CALL TRANS(DIX,DIY,VIX,VIY,N,NOPTIO,S)
C      SET INITIAL MATRICES EQUAL TO ZERO
002744      167  DO 170 J=1,N
002746          IF (NDP.EQ.0) C(J)=0.0
002750          EEP(J)=0.0
002752          TM (J)=0.0
002753          BETA (J)=0.0
002755          FREQ( J)=0.0
002756          LPHA(J)=0
002757          DO 170 I=1,N
002761          P(J,I)=0.0
002764          PD(J,I)=0.0
002770          IF (NOPTIO.NE.2) EP(J,I)=0.0
002775      170  CONTINUE
C      SET INDICATOR , IF KE=2 THEN AN ELLIPTIC FORCING FUNCTION IS PRESENT
003002          KE=1
003003          IBM=C
C      DOES USER DESIRE ONLY FREE VIBRATION DATA
003004          IF (NCL.EQ.0) GO TO 613

```

```

003005      DO 514 I=1,NCL
003006      VIY(I+N)=0.0
003010      514 DIY(I+N)=0.0
003015      617 IF(NL.EQ.0) GO TO 256
003016      DO 236 J=1,NL
003020      READ(5,215)IL,ITR,ISS,JD,JD1,JD2,JV,JV1,JV2,JA,JA1,JA2,NJC,TPAX1
003057      GO TO (172,300,400,400,400,400,400,400),IL
C      ELLIPTIC SHOCK TYPE FORCING
003072      300 IF(ITR.EQ.0) GO TO 301
003073      IF(NT.NE.0) GO TO 301
003074      READ(5,117)(DIX(I),I=1,N)
003107      READ(5,117)(VIX(I),I=1,N)
003122      CALL TRANS(DIX,DIY,VIX,VIY,N,NP,TIO,S)
003131      301 READ(5,127)IM,PE1,BT1,TIM1,PE2,BT2,TIM2,PE3,BT3,TIM3,PE4,BT4,
1      TIM4
003167      WRITE(6,233)IM,PE1,BT1,
1      TIM1,PE2,BT2,TIM2,PE3,BT3,TIM3,PE4,BT4,TIM4.
003225      KE=2
003226      BETA(IM)=BT1
003230      TM(IM)=TIM1
003232      EEP(IM)=PE1
003234      I1=N+IM
003236      I2=2*N+IM
003237      I3=3*N+IM
003240      EEP(I1)=PE2
003242      EEP(I2)=PE3
003244      EEP(I3)=PE4
003246      BETA(I1)=BT2
003250      BETA(I2)=BT3
003252      BETA(I3)=BT4
003254      TM(I1)=TIM2
003256      TM(I2)=TIM3
003260      TM(I3)=TIM4
003262      TC=0.0
003263      I=IM-N
003265      DO 302 IJ=1,N
003267      I=I+N
003271      F=FR(IJ)
003272      CC=C(IJ)
003274      BE=BT1
003275      EP(IJ,IM) =VECT5(I)*PE1/(F*F+CC*CC/4. +CC*BE+BE*BE)
003310      302 CONTINUE

```



```

003313      GO TO 177
003313 400 ISM=1
003314      IF (IL.EQ.4) WRITE(6,711)
003322      IF (IL.EQ.5) WRITE(6,712)
003330      IF (IL.EQ.6) WRITE(6,713)
003336      IF (IL.EQ.7) WRITE(6,714)
003344      IF (ITR.EQ.C) GO TO 655
003345      IF (NT.NE.0) GO TO 655
003346      READ(5,148) (DIX(J),J=1,N)
003361      READ(5,143) (VIX(J),J=1,N)
003374      CALL TRANS(DIX,DIY,VIX,VIY,N,NOPTIO,S)
003403      IF (NOPTIO.NE.2) GO TO 197
003405      DO 401 I=1,N
003407      DO 401 JJ=1,N
003410 401 EP(I,JJ)=0.0
003421      GO TO 197

C
003422 655 IF (IL.GE.4) GO TO 672
003425      READ(5,117) OMEGA,(CR(I),I=1,NCL)
003441      WRITE(6,622) OMEGA
003447      DO 623 I=1,NCL
003451      NNI = N+I
003453      WRITE(6,624) CR(I),NNI
003462 623 CR(I)=CR(I)*12.*32.2/(OMEGA**2)

C
C
003470      WRITE(6,217)
003474      NNN=N+NCL
003476      DO 630 I=1,N
003500      SUM1=0.0
003501      SUM2=0.0
003502      JK=I*N-N
003504      DO 629 IJ=1,N
003506      JI=JK+IJ
003510      DO 629 K=1,NCL
003511      IF (NDP.GT.0) SUM2=SUM2-VECTS(JI)*CCR(IJ,K)*CR(K)*OMEGA
003522      IF (NOPTIO .EQ.2) SUM1=SUM1+VECTS(JI)*AMR(IJ,K)*CR(K)*(OMEGA**2)
003535 629 SUM1=SUM1-VECTS(JI)*SR(IJ,K)*CR(K)
003551      CC=C(I)
003553      F=FR(I)
003554      OMS=OMEGA*OMEGA
003556      DEM=F*F+CC*CC/4.-OMS

```

```

003563      P(I,NNN)=(SUM1*DEM+CC*OMEGA*SUM2)/(DEM**2+CC*CC*OMS)
003574      PD(I,NNN)=(SUM2*DEM-CC*OMEGA*SUM1)/(DEM**2+CC*CC*OMS)
003610      630 CONTINUE
003613          DO 654 I=1,N
003614              LK=I-N
003616              DIS1=0.0
003617              DIS2=0.0
003620              DO 628 II=1,N
003621                  LK=LK+N
003623                  DIS1=DIS1+VECTS(LK)*F(II,NNN)
003630                  IF(NDP.NE.0)DIS2= DIS2+VECTS(LK)*PD(II,NNN)
003637      628 CONTINUE
003642          DIS=SQRT(DIS1*DIS1+DIS2*DIS2)
003647          VEL=OMEGA*DIS
003650          ACC=OMEGA*VEL
003651          WRITE(6,218) I,DIS,VEL,ACC
003665      654 CONTINUE
003670          IF (ITR.NE.0)  NT=0
003672          IM=NNN
003674          FREQ(IM)=OMEGA
003676          IF(ITR.NE.0)  GO TO 556
003677          GO TO 236

```

C
C
C
C
C
C
C
C
C
C

POINT LOADING

```

003677      672 CALL PCINTL(IC,TLAST,TFIRST,PB,NH,H,TSTEP,KEY)
003707          K=1
003710          L=1
003711          T=-H(K)
003713          KEY(5)=1
003714          NT=0
003715          I=0
003716          IF(IL.NE.4)  GO TO 692
003720          READ(5,127)  IM
003725          JI=IM-N
003727      692 I=I+1

```

```

003731      IF (I.GT.N) GO TO 693
003734      DS(I)=DIY(I)
003735      VS(I)=VIY(I)
003737      AS(I)=-(FR(I)**2+C(I)**2/4.)*DS(I)-VS(I)*C(I)
003747      IF (IL.NE.4) GO TO 692
003751      JI=JI+N
003752      AS(I)=AS(I)+VECTS(JI)*PB(2)
003756      GO TO 692
003757 693 IF (IL.LE.4) GO TO 673
003762      READ(5,625) (AL(I),I=1,NCL),SD,SV
003777      WRITE(6,717)
004003      WRITE(6,103) (AL(I),I=1,NCL)
004016      WRITE(6,192) SD,SV
004026      IM= N+      NCL
004030      E1=SD
004032      E2=SV
004033      E3=SD
004034      PT=0.0
004035 673 CALL LOAD(FR,C,AHI,AHJ,NDP,H(K),N)
004044      KEY(5)=KEY(5)+1
004046 674 T=T+H(K)
004051      NUM=0
004051      IF (T.EQ.0.) GO TO 675
004052      IF (T.GT.1TLAST +1.0E-6)) GO TO 235
004057      L=L+1
004060      IF (T+H(K).GT.TSTEPH(K)+1.0E-6) GO TO 576
004066      GO TO 675
004066 676 IF (KEY(5).EQ.0) GO TO 677
004067      YS1=3.*PB(L+1)-3.*PB(L+2)+PB(L+3)
004074      YS2=3.*PB(L)- 3.*PB(L-1) +PB(L-2)
004100      AA=PB(L+1)
004102      PB(L+1)=YS2
004103      YS2=AA
004104      KEY(5)=0
004105      GO TO 575
004106 577 PB(L-1)=YS1
004110      PB(L)=YS2
004112      T=T-H(K)
004115      K=K+1
004116      KEY(5)=1
004117      L=L-1
004120      GO TO 573

```

C
C
C
C
C
C

COMPUTE ACTUAL RESPONSES.

```

004121 675 IF(IL.GE.2) GO TO 685
004124 IF(IL.GE.5)GO TO 682
004126 JI=-N+IM
004130 I=0
004131 681 I=I+1
004133 IF(I.GT.N) GO TO 699
004136 687 JI=JI+N
004140 ZP(I)=VECTS(JI)
004142 GO TO 681
004143 682 JL=-N
004145 I=0
004146 691 I=I+1
004150 IF(I.GT.N) GO TO 689
004153 ZD(I)=0.0
004154 ZC(I)=0.0
004155 ZM(I)=0.0
004156 JL=JL+N
004160 KK=0
004161 683 KK=KK+1
004163 JI=JL+ KK
004164 IF(KK.GT.N) GO TO 696
004167 M=0
004167 684 M=M+1
004171 IF(M.GT.NCL) GO TO 683
004174 ZD(I)=ZD(I)+VECTS(JI)*SR(KK,M)*AL(M)
004203 IF(NDP.NE.0) ZC(I)=ZC(I)+VECTS(JI)*CCR(KK,M)*AL(M)
004213 IF(NOPTIO .EQ.2) ZM(I)= ZM(I)+VECTS(JI)* AMR(KK,M)*AL(M)
004225 GO TO 684
004226 686 IF(IL.EQ.5) AS(I)=AS(I)-ZD(I)*PB(2)-ZC(I)*SV
004236 IF(IL.EQ.6) AS(I)=AS(I)-ZD(I)*SD-ZC(I)*PB(2)
004246 IF(IL.EQ.7) AS(I)=AS(I)-ZD(I)*SD-ZC(I)*SV-ZM(I)*PB(2)
004260 IF(IL.EQ.7) WRITE(6,117) ZD(I),ZC(I),ZM(I)
004274 GO TO 691

```

C
C
C

THIS COMPLETES THE COMPUTATION OF THE PRELIMINARY MATRICES

```

C
C      COMPUTE FORCING TERMS
004275 685 XD=PB(L)
004277      X1=(PB(L+1)-PB(L-1))/H(K)
004303      X2=(PB(L+1)-2.*PB(L)+PB(L-1))/(H(K)**2)
C      POINT LOADING
C
004312 696 IIL=IL-3
004314      I=0
004315      JI=IM-N
004317 688 I=I+1
004321      IF(I.GT.N) GO TO 689
004324      JI=JI+N
004325      GO TO(701,751,801,851),IIL
004334 701 PS=VECTS(JI)*(XD*AH1(I,1)+X1/2.*AH1(I,2)+X2/2.*AH1(I,3))
004345      PC=VECTS(JI)*(XC*AHJ(I,1)+X1/2.*AHJ(I,2)+X2/2.*AHJ(I,3))
004356      PT=VECTS(JI)*PB(L+1)
004361      GO TO 901
004361 751 PS=-ZD(I)*(XD*AH1(I,1)+X1/2.*AH1(I,2)+X2/2.*AH1(I,3))
004372      PC=-ZD(I)*(XC*AHJ(I,1)+X1/2.*AHJ(I,2)+X2/2.*AHJ(I,3))
004403      PT=-ZD(I)*PB(L+1)
004406      IF(NDP.EQ.0) GO TO 752
004407      PS=PS-ZC(I)*(X1/2.*AH1(I,1)+AH1(I,2)*X2)
004416      PC=PC-ZC(I)*(X1/2.*AHJ(I,1)+AHJ(I,2)*X2)
004424      PT=PT-ZC(I)*(X1/2.+H(K)*X2)
004433 752 IF(NOPT10.NE.2) GO TO 901
004435      PS=PS-ZM(I)*AH1(I,1)
004441      PC=PS-ZM(I)*AHJ(I,1)
004444      PT=PT-ZM(I)*X2
004447      GO TO 901
004447 801 PS=-ZD(I)*(E1*AH1(I,1)+XD*AH1(I,2)+X1/4.*AH1(I,3)+X2/6.*AH1(I,4))
004463      PC=-ZD(I)*(E1*AHJ(I,1)+XC*AHJ(I,2)+X1/4.*AHJ(I,3)+X2/6.*AHJ(I,4))
004476      PT=-ZD(I)*(E1+H(K)*XD+X1/4.*H(K)**2+X2/6.*H(K)**3)
004513      IF(NDP.EQ.0) GO TO 802
004514      PS=PS-ZC(I)*(XD*AH1(I,1)+X1/2.*AH1(I,2)+X2/2.*AH1(I,3))
004525      PC=PC-ZC(I)*(XD*AHJ(I,1)+X1/2.*AHJ(I,2)+X2/2.*AHJ(I,3))
004537      PT=PT-ZC(I)*PB(L+1)
004543 802 IF(NOPT10.NE.2) GO TO 803
004545      PS=PS-ZM(I)*(X1/2.*AH1(I,1)+X2*AH1(I,2))
004555      PC=PC-ZM(I)*(X1/2.*AHJ(I,1)+X2*AHJ(I,2))
004564      PT=PT-ZM(I)*(X1/2.+X2*H(K))
004572 803 IF(I.NE.N) GO TO 301

```

```

004574      E1=E1+H(K)/6.*(-.5*PB(L-1)+4.*PB(L)+2.5*PB(L+1))
004605      CC TC 901
004606      851 PS=-ZD(I)*(X0/2.*AHI(I,3)+X1/12.*AHI(I,4)+X2/24.*AHI(I,5)+E2*
1          AHI(I,2)+E3*AHI(I,1))
004625      PC=-ZD(I)*(X0/2.*AHJ(I,3)+X1/12.*AHJ(I,4)+X2/24.*AHJ(I,5)+E2*
1          AHJ(I,2)+E3*AHJ(I,1))
004643      IF(I.GT.1) GO TO 854
004646      DD=E3+E2*H(K)+H(K)**2/24.*(1C.*PB(L)+3.*PB(L+1)-PB(L-1))
004661      DV=E2+H(K)/6.*(4.*PB(L)+2.5*PB(L+1)-.5*PB(L-1))
004673      854 PT=-ZD(I)*DD
004676      IF(NDP.EQ.0) GO TO 352
004677      PS=PS-ZC(I)*(X0*AHI(I,2)+X1/4.*AHI(I,3)+X2/6.*AHI(I,4)+E2*AHI(I,1)
1          )
004713      PC=PC-ZC(I)*(X0*AHJ(I,2)+X1/4.*AHJ(I,3)+X2/6.*AHJ(I,4)+E2*AHJ(I,1)
1          )
004727      PT=PT-ZC(I)*DV
004732      852 IF(NCPT10.NE.2) GO TO 853
004734      PS=PS-ZM(I)*(X0*AHI(I,1)+X1/2.*AHI(I,2)+X2/2.*AHI(I,3))
004746      PC=PC-ZM(I)*(X0*AHJ(I,1)+X1/2.*AHJ(I,2)+X2/2.*AHJ(I,3))
004760      PT=PT-ZM(I)*PB(L+1)
004764      853 IF(I.NE.N) GO TO 901
004766      E3=DD
004770      E2=DV
004771      901 D=DS(I)
004773      V=VS(I)
004775      F=FR(I)
004776      CC=C(I)/2.
005000      EE=1.
005002      IF(NDP.NE.0) EE=EXP(-CC*H(K))
005010      FT=F*H(K)
005013      DS(I)=EE*(D*COS(FT)+(V/F+D*CC/F)*SIN(FT))+PS/F
005032      VS(I)=-CC * DS(I)+EE*(-F*D*SIN(FT)+(V+D*CC)*CCS(FT))+PC
005052      AS(I)=-2.*CC*VS(I)-(F*F+CC*CC)*DS(I)+PT

C
C      SET INITIAL TERMS
C
C
C
C
005062      GO TO 688

C
C      WRITE OUT SPECIFIED QUANTITIES
005063      689 WRITE(6,223)

```

```

005067      NUM=0
005070      IF(JD.EQ.2.OR.JV.EQ.2.OR.JA.EQ.2)  NUM=1
005104      IF(JD.EQ.2.OR.JV.EQ.2.OR.JA.EQ.2)  Z(1)=T
005120      IF(JD.EQ.0)  GO TO 910
005121      WRITE(6,224) T
005127      CALL CHANGE(DS,DT,N,JD,JD1,JD2)
005133      910 IF(JV.EQ.0)  GO TO 915
005134      WRITE(6,225) T
005142      CALL CHANGE(VS,VT,N,JV,JV1,JV2)
005146      915 IF(JA.EQ.0)  GO TO 920
005147      WRITE(6,226) T
005155      CALL CHANGE(AS,AT,N,JA,JA1,JA2)
005161      920 IF(NUM.GT.0) WRITE(9) (Z(LLM),LLM=1,NUM)
005175      GO TO 574

C
C      HARMONIC FORCING
005176      177 READ(5,214) PH,OMEGA,IM
005210      WRITE(6,234) PH,OMEGA,IM
005222      FREQ(IM)= OMEGA
005224      I= IM - N
005226      303 DO 304 IJ=1,N
005230      I=I+N
005232      F=FR(IJ)
005233      CC=C(IJ)
005235      DEM= F*F+CC*CC/4.-OMEGA*OMEGA
005241      P(IJ,IM)= VECTS(I)*DEM *PH/(DEM*DEM+CC*CC *OMEGA*OMEGA)
005251      PD(IJ,IM) =VECTS(I)*PH*CC*OMEGA/(DEM *DEM +CC*CC*OMEGA* OMEGA)
005264      304 CONTINUE
005267      IF (ISS.EQ.0)GO TO 177
005270      WRITE(6,215) IM
005275      WRITE(6,217)
005301      DO 176 LK=1,N
005303      DIS1=0.0
005304      DIS2=0.0
005305      I=LK-N
005306      DO 306 K=1,N
005310      I=I +N
005312      DIS1= DIS1 +VECTS(I) *P( K,IM)
005317      IF(NDP.NE.0) DIS2= DIS2 + VECTS(I) * PD(K,IM)
005326      306 CONTINUE
005331      DIS = SQRT(DIS1*DIS1 +DIS2*DIS2)
005336      VEL = OMEGA* DIS

```

```

005337      ACC = OMEGA*VEL
005340      WRITE (6,218) LK,DIS,VEL,ACC
005354 176    CONTINUE
005357 177    IF(ITR.EQ.0) GO TO 236
005360      WRITE(6,113) IM
005366      IF( IL.EQ.1) WRITE(6,220)
005374 197 IF(JD.EQ.0) GO TO 182
005375      WRITE(6,204)
005401      IF(JD1.NE.0) GO TO 191
005402      WRITE(6,205)
005406      IF(JD.EQ.2) WRITE(6,219)
005414 182 IF(JV.EQ.0) GO TO 185
005415      WRITE(6,207)
005421      IF(JV1.NE.0) GO TO 193
005422      WRITE(6,205)
005426      IF(JV.EQ.2) WRITE(6,219)
005434 185 IF(JA.EQ.0) GO TO 196
005435      WRITE(6,208)
005441      IF(JA1.NE.0) GO TO 195
005442      WRITE(6,205)
005446      IF(JA.EQ.2) WRITE(6,219)
005454      GO TO 196
005455 191 WRITE (6,206)JD1,JD2
005465      IF(JD .EQ.2) WRITE(6,219)
005473      GO TO 182
005474 193 WRITE (6,206)JV1,JV2
005504      IF(JV .EQ.2) WRITE(6,219)
005512      GO TO 185
005513 195 WRITE (6,206)JA1,JA2
005523      IF(JA .EQ.2) WRITE(6,219)
005531 196 IF(IL.GE.3) GO TO 655
005534 656 WRITE(6,223)
C      TRANSIENT MOTION
005540      DELT= TMAX1/FLOAT(NJC)
005543      NJ= NJC + 1
005544      T =- DELT
C      TIME INCREMENT
005546      DC 230L=1, NJ
005547      T=T+ DELT
005551      IF(IGM.EQ.0) GO TO 631
005552      IL=1
005553 631 NUM=C

```



```

005554      GO TO (228,330,325) , IL
005563      228  FF=FREQ(IM)
005565      DC 213LL= 1,N
005567      F= FR(LL)
005571      FT=F*T
005573      CO= COS(FT)
005575      SI= SIN(FT)
C          HARMONIC FORCING ALONE
005577      FRT= FF*T
005601      PC= PD(LL,IM)
005605      PS=P(LL,IM)
005610      SN= SIN(FRT)
005612      CS= COS(FRT)
005614      CC=C(LL)
005616      D =DIY(LL) +PC
005620      V =VIY(LL) +C(LL)*J/2.-FREQ(IM)* PS
005626      IF(NDP.NE.C) GO TO 199
005627      DS(LL)=D *CO +V/F*SI + PS * SN
005635      VS(LL)= -F*D*SI +V*CO +FF*PS * CS
005646      AS(LL)= -F*F*D*CO -V*F*SI -FF*FF *PS *SN
005657      GO TO 213
C          CHANGES CAUSED BY DAMPING
005660      199  ED=EXP(-C(LL)/2.*T)
005666      DS(LL)= (V/F*SI +D*CO) *ED + PS*SN -PC*CS
005700      VS(LL)= (V/F*SI +D*CO) *ED*(-CC/2.) +ED*(V*CO -F*D*SI) +FF*PS*CS
1          + FF*PC*SN
005724      AS(LL) = ED*(CC*CC/4.-F*F)*( D*CO +V/F*SI)+ED*CC*F*(D*SI -V/F *CO)
1          -FF*FF*PS*SN +FF*FF*PC*CS
005750      213  CONTINUE
005753      GO TO 239
C          ELLIPTIC FORCING
005753      330  IF( T.GT.C) GO TO 308
005756      DO 307 LL=1,N
005757      A(LL)= DIY(LL) -EP(LL,IM)
005764      B(LL)=(VIY(LL) +A(LL)*C(LL)/2.-BETA(IM)*EP(LL,IM))/FR(LL)
005776      307  CONTINUE
006001      308  DO 309 LL=1,N
006003      CC=C(LL)
006005      F=FR(LL)
006006      CO =COS(F*T)
006012      SI=SIN(F*T)
006016      EX=PE1*EXP(BT1*T)

```

```

006023      AA=A(LL)
006025      BB=B(LL)
006026      FREQ(N+NCL)=OMEGA
006031      BE=BT1
006033      EE=EP(LL,IM)
006037      DS(LL)=EXP(-CC/2.*T)*(AA*CC+BB*SI)+EE*EXP(BE*T)
006060      VS(LL)=EXP(-CC/2.*T)*(-AA*(CC/2.*CC+F*SI)+BB*(F*CC-CC/2.*SI))
1          +EE*BE*EXP(BE*T)
006111      AS(LL)=EXP(-CC/2.*T)*((CC*CC/4.-F*F)*(AA*CC+BB*SI)+F*CC*(AA*SI
1          -BB*CO))+BE*BE*EE*EXP(BE*T)
006142      309 CONTINUE
006145      IF(TIM1.LT.1.0E-9) GO TO 239
006147      332 IF((T+DELT/2.).LT.TIM1) GO TO 239
006154      TO=TIM1
006155      TIM1=TIM2
006156      TIM2=TIM3
006157      TIM3=TIM4
006160      TIM4=0.0
006160      PE1=PE2
006162      PE2=PE3
006163      PE3=PE4
006164      PE4=0.0
006164      BT1=BT2
006166      BT2=BT3
006167      BT3=BT4
006170      BT4=0.0
006170      BE=BT1
006172      IF(TIM1.LT.1.0E-9) PE1=0.0
006175      I=IM-N
006177      DO 311 LL=1,N
006201      F=FR(LL)
006203      I=I+N
006204      CC=C(LL)
006206      BE=BT1
006207      EP(LL,IM)=VECTS(I)*PE1/(F*F+CC*CC/4.+CC*BE+BE*BE)
006222      AA=(DS(LL)-EP(LL,IM)*EXP(BE*TO))*EXP(CC*TO/2.)/F
006241      BB=(VS(LL)-BE*EP(LL,IM)*EXP(BE*TO))*EXP(CC*TO/2.)/F
006261      A(LL)=AA*(F*COS(F*TO)-CC/2.*SIN(F*TO))-BB*SIN(F*TO)
006304      B(LL)=BB*COS(F*TO)+(CC/2.*CCS(F*TO)+F*SIN(F*TO))*AA
006331      311 CONTINUE
006333      239 IF(JD.EQ.2.0R.JV.EQ.2.0R.JA.EQ.2) Z(1)=T
006350      IF(JD.EQ.2.0R.JV.EQ.2.0R.JA.EQ.2) NUM=1

```

```

006364      IF(JD.EQ.0) GO TO 240
006365          WRITE(6,224) T
006373      CALL CHANGE(DS,DT,      N, JD,JD1,JD2)
006377  240  IF(JV.EQ.0) GO TO 245
006400          WRITE(6,225) T
006406      CALL CHANGE(VS, VT,      N, JV,JV1,JV2)
006412  245  IF( JA.EQ.0) GO TO 250
006413          WRITE (6,226) T
006421      CALL CHANGE(AS,AT,      N, JA,JA1,JA2)
006425  250  IF(NUM.GT.0) WRITE(9) (Z(LLM),LLM=1,NLM)
006441  230  CONTINUE
006444      IF (IL.NE.2) GO TO 236
006446      I=IM-N
006450      DO 235 L=1,N
006451      I=I+N
006453      F=FR(L)
006454      CC=C(L)
006456      BE=BETA(IM)
006460      PE=EEP(IM)
006461      EP(L ,IM) =VECTS(I)*PE /(F*F+CC*CC/4.+CC*BE+BE*BE)
006474  235  CONTINUE
006477  236  CONTINUE
C      ALL LOCAL OUTPUT IS COMPLETED FOR ALL LOADS
006502  231  IF(NT.EG.0) GO TO 500
C      COMBINED ACTION
C  256  TRANSIENT OUTPUT
006503  256  DELT = TMAX/FLOAT(NINCR)
006506      DO 314 J=1,N
006507      PA=0.0
006510      PB=0.0
006511      DO 312 I=1,N
006512      F=FREQ(I)
006514      PA=PA+PD(J,I)-EP(J,I)
006522      PB=PB+F *P(J,I) +BETA(I)*EP(J,I)
006531  312  CONTINUE
006533      A(J)=DIY(J)+PA
006536      B(J)=( VIY(J) +A(J)*C(J)/2.-PB )/FR(J)
006545  314  CONTINUE
006550      JJJ= NINCR + 1
006552      T= -DELT
006553      ILL=0
006554      DO 290L=1, JJJ

```

```

006555      IF ( ILL.EG.0) GO TO 320
C
C      CORRECT INITIAL CONSTANTS A(I),E(I), FOR RANGE CHANGE IN ELLIPTIC LOADS
C
006556      313 DO 317 KJ=1,ILL
006560          K=LPHA(KJ)
006562          I=K-N
006564          BE=BETA(K)
006565          PE=EEP(K)
006567          DO 317 KL=1,N
006570          I=I+N
006572          F=FR(KL)
006573          CC=C(KL)
006575          EP(KL,K ) =VECTS(I)*PE /(F*F+CC*CC/4.+CC*BE+BE*BE)
006607      317 CONTINUE
C
C
C
006614      ILL=C
006615      DO 319 KJ=1,N
006616      PA=0.0
006617      PB=0.0
006620      CC=C(KJ)
006622      EE=EXP(CC*T/2.)/FR(KJ)
006631      DO 318 KL=1,N
006632      F=FREQ(KL)
006634      PA=PA+EP(KJ,KL)*EXP(T*BETA(KL))+P(KJ,KL)*SIN(T*F) -PD(KJ,KL)*
1      COS(T*F)
006664      PB=PB+BETA(KL)*EP(KJ,KL)*EXP(T*BETA(KL))+F*(P(KJ,KL)*CCS(T*
1      F)+PD(KJ,KL)*SIN(T*F))
006715      318 CONTINUE
006717      F=FR(KJ)
006721      A(KJ)= EE*((DS(KJ)-PA)*(F*CCS(F*T)-CC/2.*SIN(F*T))-SIN(F*T)*(VS(KJ
1      )-PB))
006751      B(KJ)= EE*(( COS(F*T)*(VS(KJ)-PB))+(CC/2.*COS(F*T)+F*SIN(F*T))*{
1      DS(KJ)-PA))
007002      319 CONTINUE
C
C
007005      NE N=4*N
C
007006      320 T=T + DELT

```

```

007010      NUM = 0
007011      IF(KE.NE.2) GO TO 315
C      ELLIPTIC FORCING      CHECK FOR END OF FORCE RANGE
007013      DO 305 LL=1,N
007014      IF(TM(LL).LT.1.E-9) GO TO 305
007017      IF((T+DELT/2.).LT.TM(LL)) GO TO 305
007024      DO 258 J=1,3
007025      KJ=(J-1)*N+LL
007031      KL=J*N+LL
007033      TM(KJ)=TM(KL)
007036      EEP(KJ)=EEP(KL)
007040      BETA(KJ)= BETA(KL)
007042      258 CONTINUE
007044      TM(KL)=0.0
007046      BETA(KL)=0.0
007047      EEP(KL)=0.0
007050      ILL=ILL+1
007052      LPHA(ILL)=LL
007054      305 CONTINUE
007057      IF(ILL.EQ.0) GO TO 316
007060      T=T-DELT
007062      GO TO 313
007062      316 DO 259 LL=1,N
007064      PA=0.0
007065      PB=0.0
007066      PC=0.0
007067      DO 315 K=1,N
007070      F= FREQ(K)
007072      SN=SIN(F*T)
007076      CS=COS(F*T)
007102      EE=EXP(BETA(K)*T)
007107      PA=PA+P(LL,K)*SN -PD(LL,K)*CS+ EP(LL,K)*EE
007122      PB=PB+P(LL,K)*CS+F+ PD(LL,K)*SN*F+BETA(K)*EF(LL,K)*EE
007137      PC=PC-F*F*(P(LL,K)*SN-PD(LL,K)*CS)+BETA(K)*BETA(K)*
      1 EE*EP(LL,K)
007155      315 CONTINUE
007157      F=FR(LL)
007161      CC=C(LL)
007163      FT = F*T
007165      CO= COS(FT)
007167      SI= SIN(FT)
007171      FF=FREQ(LL)

```

```

007173      AA=A(LL)
007174      BB=B(LL)
007176      IF(NDP.NE.0) GO TO 232
007177      DS(LL)=AA*CO+BB*SI +PA
007203      VS(LL)=F* (-AA*SI+BB*CO) +PB
007211      AS(LL)=-F*F*(AA*CO+BB*SI) +PC
007220      GO TO 259

C      CHANGES CAUSED BY DAMPING
007221  232      ED=EXP (-CC/2.*T)
007227      DS(LL)=ED*(AA*CO+BB*SI) +PA
007235      VS(LL)= -CC/2.*ED*(AA*CO+BB*SI) -F*ED*(AA*SI -BB*CO) +PB
007251      AS(LL)=ED*((AA*CO+BB*SI)*(CC*CC/4.-F*F)+CC*F*(AA*SI-BB*CO)) +PC
007272  259      CONTINUE
007275      WRITE(6,223)
007300      IF(ID.EQ.2.OR.IV.EQ.2.OR.IA.EQ.2) NUM=1
007314      IF(ID.EQ.2.OR.IV.EQ.2.OR.IA.EQ.2) Z(1)=T
007330  260      IF(ID.EQ.0) GO TO 265
007331      WRITE (6,224) T
007337      CALL CHANGE(DS,DT,      N,ID,ID1,ID2)
007343  265      IF(IV.EQ.0) GO TO 270
007344      WRITE (6,225) T
007352      CALL CHANGE(VS,VT,N,IV,IV1,IV2)
007356  270 IF(IA.EQ.0) GO TO 279
007357      WRITE (6,226) T
007365      CALL CHANGE(AS,AT,      N,IA,IA1, IA2)
007371  279 IF(NUM.GT.0) WRITE(9) (Z(LLM),LLM=1,NUM)
007405  280 CONTINUE
007410  500 CONTINUE
007413      IF(NTP.NE.0) END FILE 9
007416      GO TO 325
007417      95      WRITE(6,115)
007423      68 FORMAT(15H0 THERE ARE      ,I3,43H UNRESTRAINED DEGREES OF FREEDOM
      1 AT MASSES      ,10I5)
007423      93 FORMAT(1HC,20X,46H TRANSFORMATION MATRIX...6*-1 WHERE M=B*T*B )
007423      94 FORMAT(1H1,10X,37H ROWS OF THE NONDIAGONAL MASS MATRIX      )
007423  100      FORMAT(20H0 NUMBER OF MASSES =,I2,      27H NUMBER OF LINEAR
      1SPRINGS=, I2, 30H NUMBER OF LOAD CONDITIONS= ,I2,32H NUMBER OF R
      2OTATIONAL VARIABLES=,I2)
007423      101 FORMAT(8I10)
007423      102      FORMAT (10HC      ,I2,50H SPRING CONSTANTS LISTED IN ORDER 10
      1 TO A LINE      )
007423      103 FORMAT((8E12.4)//)

```

```

007423 104  FORMAT (1H0,36X,18H IDENTIFY TOPOLOGY//)
007423 105  FORMAT(8H      MASS,7X,9H INERTIA ,6H 1 PT.,6H 2 FT.,6H 3 FT.,
      1 12H      OFFSET ,12H OFFSET MASS  )
007423 106  FORMAT(2F12.5,3I6,F12.5,I6)
007423 107  FORMAT ( 1H1 ,30X, 40HUNDAMPED NATURAL FREQUENCIES (CYC./SEC.) )
007423 108  FORMAT (1H0 ,22X, 7E12.4  )
007423 109  FORMAT ( 25X ,39H EIGENVECTORS EIGHT COMPONENTS PER LINE // )
007423 110  FORMAT(1H0,34X,53H...MODE SHAPE...CORRESPONDING TO NATURAL FRE
      1QUENCY      ,E12.4,10H RAD./SEC.  )
007423 111  FORMAT(1H0,15X,53H PCT. OF CRITICAL DAMPING IN MODE (MODAL DAMPING
      1)      /(10F12.4))
007423 112  FORMAT(1H0,I2,60H ROTATIONAL SPRING CONSTANTS LISTED IN ORDER
      18 TO A LINE  )
007423 113  FORMAT(48HTRANSIENT MOTION WITH LOAD AT GENERALIZED MASS ,I2 )
007423 114  FORMAT( 37H0      NUMBER OF INTERNAL CLOSED LOOPS=      ,I2)
007423 115  FORMAT ( 51H1 INCORRECT NEGATIVE SIGN IN INPUT CHECK RESULTS      )
007423 116  FORMAT(4I10)
007423 117  FORMAT(8E12.4)
007423 118  FORMAT(1H0,30X,36H LINEAR VISCOUS DAMPERS (LB-SEC/IN)  )
007423 119  FORMAT(1H0,30X,42H ROTATIONAL VISCOUS DAMPERS (LB-SEC-IN)  )
007423 120  FORMAT(1H0,25X, 38H DAMPING MATRIX IN REGULAR COORDINATES  )
007423 121  FORMAT(1H0,30X,22H DIAGONAL MASS MATRIX  )
007423 122  FORMAT(1H0,30X,46H STIFFNESS MATRIX      ...OUTPUT BY ROWS...  )
007423 123  FORMAT(1H0,25X,92H DAMPING MATRIX , CY , IN NORMAL COORDINATES
      1  CY = (Z**T)*CX*Z , (Z IS THE MODAL MATRIX)  )
007423 124  FORMAT(1H0,10X,45H**NOTE** THE FOLLOWING ANALYSIS ASSUMES THAT /
      1 15X, 92H(1) THE MAIN DIAGONAL TERMS OF MATRIX CY ABOVE ARE MUCH
      2 GREATER THAN THE OFF DIAGONAL TERMS  /
      3 15X, 52H(2) THE UNDAMPED FREQUENCIES ARE  SEPARATED WIDELY /
      4 15X,95H(3) PROPORTIONAL DAMPING, NO IN MODE PHASE CHANGES OF THE
      5 EIGENVECTORS DUE TO DAMPING  //)
007423 126  FORMAT(40H0**H0,H0,H0, OVERDAMPED MOTION IN  MODE ,I6  )
007423 127  FORMAT(I2,12F6.4)
007423 128  FORMAT(1H0,30X,40H DAMPED NATURAL FREQUENCIES (CYC./SEC.)  )
007423 133  FORMAT(20A4)
007423 134  FORMAT(29H0 THE COMPUTER INTERPRETS AS ,I4,21H DEGREES OF FREEDOM
      1  )
007423 137  FORMAT(32H0 NEGATIVE EIGENVALUE IN MODE      ,I4)
007423 148  FORMAT(6E12.5)
007423 192  FORMAT(101H0 YOU HAVE INPUT THE FOLLOWING INITIAL SUPPORT DISPLACE
      1MENT AND VELOCITY (USING ABOVE SCALE FACTORS) /2F12.5//)
007423 201  FORMAT(11I2,I6, E12.4)

```

```

007423 202 FORMAT(1HC,25X,12H LOAD CASE ,I2)
007423 203 FORMAT(1HC,10X,15H PRINTED OUTPUT ,20X,15H TAPE OUTPUT )
007423 204 FORMAT(2CHC DISPLACEMENTS. )
007423 205 FORMAT(20X,5H ALL )
007423 206 FORMAT(2CX,2I5)
007423 207 FORMAT(17HC VELOCITIES. )
007423 208 FORMAT(2CHC ACCELERATIONS. )
007423 209 FORMAT(20X,8H..NONE..)
007423 210 FORMAT(51X,5H ALL )
007423 211 FORMAT(49X,8H..NONE..)
007423 212 FORMAT(50X,2I5)
007423 214 FORMAT(2E12.4,I6)
007423 215 FORMAT(12I2, I6,E12.0)
007423 216 FORMAT(52HC HARMONIC STEADY STATE LOAD AT MASS,I2)
007423 217 FORMAT(30X,20H...STEADY STATE ENVELOPES... //5X,21H GENERALIZED M
1ASS ,21H MAX DISPLACEMENT ,21H MAX VELOCITY ,21
2H MAX ACCELERATION )
007423 219 FORMAT(I15,15X,E12.4,9X,E12.4,9X,E12.4 )
007423 219 FORMAT(25X,40H OUTPUT IS ALSO ON TAPE FOR LOCAL CASE )
007423 220 FORMAT(99HC **NOTE**YOU ARE ASSUMING TRANSIENT MOTION IS NOT YET
1 DAMPED OUT AND COMPLETE FORCE BUILDUP )
007423 221 FORMAT(10HC TIME ,8H DISPL.(,I2,10H) DISPL.(,I2, 9H) VEL
1.(,I2,10H) VEL.(,I2,10H) ACC.(,I2,10H) ACC.(,I2,2H) )
007423 223 FORMAT( 6HC TIME )
007423 224 FORMAT(E9.2,14H DISPLACEMENTS )
007423 225 FORMAT(E9.2,14H VELOCITIES )
007423 226 FORMAT(E9.2,14H ACCELERATIONS )
007423 229 FORMAT( 99HC **NOTE**THE SUM OF STEADY STATE MAXIMUMS GENERATES
1AN ENVELOPE**THE MOTION IS ACTUALLY TRANSIENT )
007423 233 FORMAT(1HC,10X,29H ELLIPTIC SHOCK LOAD AT MASS ,I3 /
1 15X,F12.4,5H*EXP(,F12.4,21H *T) FOR T LESS THAN,F12.6,6H(SEC.)/
2 15X,F12.4,5H*EXP(,F12.4,21H *T) FOR T LESS THAN,F12.6,6H(SEC.)/
3 15X,F12.4,5H*EXP(,F12.4,21H *T) FOR T LESS THAN,F12.6,6H(SEC.)/
4 15X,F12.4,5H*EXP(,F12.4,21H *T) FOR T LESS THAN,F12.6,6H(SEC.)/
5 /)
007423 234 FORMAT(1HC,15X,18H HARMONIC LOAD ,F12.5,5H*SIN(,F12.4,27H *T)
1 AT GENERALIZED MASS ,I3 /)
007423 241 FORMAT(15X,25H MODAL MASSES..Z**T*M*Z / (8E12.4))
007423 242 FORMAT(15X,28H IN MODE DAMPING CONSTANTS /(8E12.4))
007423 622 FORMAT(31HOBASE MOTION AT A FREQUENCY OF ,F12.5,68H RAD/SEC WAS IN
1PUT WITH THE FOLLOWING AMPLITUDES AT THE NODE POINTS )
007423 624 FORMAT(17HCINPUT G LEVEL OF , F12.5,17H G UNITS AT NODE ,I4)

```



```

007423 625 FORMAT(12F6.0)
007423 645 FORMAT( 1H0,10X,42H NONDIAGONAL MASS MATRIX ...OUTPUT BY ROWS )
007423 711 FORMAT(10X,51H INPUT DESCRIBED BY DATA POINTS, FORCE INPUT )
007423 712 FORMAT(10X,51H INPUT DESCRIBED BY DATA POINTS, SUPPORT DISPLACE. )
007423 713 FORMAT(10X,51H INPUT DESCRIBED BY DATA POINTS, SUPPORT VELOCITY )
007423 714 FORMAT(10X,51H INPUT DESCRIBED BY DATA POINTS, SUPPORT ACCELERER.)
007423 715 FORMAT(16X,48H YOU HAVE INPUT THE FOLLOWING NUMBERS OR INDICES //
      184H      D.C.F., SPRINGS, CASES , ROTATION, SUPPORTS, TAPES
      2, DAMPING, OPTION )
007423 717 FORMAT(20X,75H YOU HAVE INPUT THE FOLLOWING SCALE FACTORS AT THE
      1RESPECTIVE SUPPORTS /)
007423 325 IF(NTP.LE.1) GO TO 9
007426      END FILE 9
007430      REWIND 9
007432      GO TO 9
007433      END

```

```

000013 SUBROUTINE POINTL(IC,TLAST,TFIRST,PB,NH,H,TSTEPH,KEY)
000013 DIMENSION TDISC(11),H(10),TSTEPH(10),PB(4002)
COMMON/EXT/X(1002),Y(1002),XK(5),YK(5),KEY(6)
C NPOINTS = NUMBER OF INPUT POINTS ON DATA CARDS
C NDISC = NUMBER OF DERIVATIVE DISCONTINUITIES IN DATA (IF KEY(4)=1)
C NH = NUMBER OF DISTINCT OUTPUT REGIONS WITH UNIFORM SPACINGS
C TFIRST = ARBITRARY TIME OF FIRST DATA POINT
C TLAST = TIME OF LAST DATA POINT
C TSPAC = TIME SPACING OF DATA POINTS (IF KEY(1).NE.0)
C KEY(J) HAVE THE FOLLOWING MEANINGS
C J=1, KEY(J)=-1 INPUT EVENLY SPACED, US+ ALSO AS OUTPUT POINTS
C           0 INPUT NOT EVENLY SPACED, MUST READ IN TIME POINTS
C           1 INPUT EVENLY SPACED, MULTIPLE OUTPUT REGIONS POSSIBLE
C J=2,           0 OUTPUT SOLUTION POINTS TO BE SPECIFIED BY H REGIONS
C           1 OUTPUT POINTS ARE AT INPUT POINTS AND MIDWAY BETWEEN THEM
C J=4,           0 USE LINEAR INTERPOLATION BETWEEN INPUT DATA POINTS
C           1 USE FOURTH ORDER INTERPOLATION TO COMPUTE ADDITIONAL POINTS
C J=3,J=5,J=6, USED INTERNALLY, CAN SET =0 ON INPUT
C
000013 READ(5,9) NPOINTS,NDISC,NH      , (KEY(I),I=1,6),TFIRST,TLAST,TSPAC
000044 READ(5,10)(Y(I),I=1,NPOINTS)
000057 IF (KEY(1).EQ.0) READ(5,10) (X(I),I=1,NPOINTS)
000103 KEY(6)=0
000105 IF (KEY(1).NE.0) KEY(6)=1
000107 32 NSTEPH=NH-1
000110 IF (NH.EQ.1) READ(5,10) H(1),(TDISC(I),I=1,NDISC)
000141 TLAST=TLAST-TFIRST
000142 TSTEPH(1)=TLAST
000144 IF (NH.GT.1) READ(5,10) (H(I),I=1,NH) ,(TSTEPH(I),I=1,NSTEPH),
1 ( TDISC(I),I=1,NDISC)
000216 TSTEPH(NH)= TLAST
000220 IF (KEY(1).EQ.0) GO TO 34
000222 J=1
000223 I=1
000224 X(1)=0.0
000225 33 I=I+1
000227 IF (KEY(1).LT.0) X(I)=X(I-1)+ H(J)
000236 IF (KEY(1).GT.0) X(I)=X(I-1)+ TSPAC
000245 IF (X(I).GT.TLAST) GO TO 34
000252 IF (X(I).LE.(TSTEPH(J)+1.0E-9)) GO TO 33
000260 J=J+1

```

```

000262      GO TO 33
C
C
000262      34 IF (TFIRST.EQ.0.) GO TO 35
000263      DO 31 I=1,NPOINTS
000265      31 X(I)=X(I)-TFIRST
000272      35 WRITE(6,11) NPOINTS ,(X(I),I=1,NPOINTS)
000307      WRITE(6,12) (Y(I),I=1,NPOINTS)
000322      WRITE(6,13) NH ,(H(I),I=1,NH)
000350      WRITE(6,15) NSTEPH,(TSTEPH(I),I=1,NSTEPH)
000365      WRITE(6,14) NDISC ,(TDISC(I),I=1,NDISC)
000402      KEY(3)=3
000404      KEY(5)=2
000405      NDISC=NDISC+1
000407      TDISC(NDISC)=TLAST
000414      X(NPOINTS+1)=X(NPOINTS) +H(NH)
000421      X(NPOINTS+2)=X(NPOINTS) +H(NH)*2.
000426      IF (KEY(1).GT.0) X(NPOINTS+1)=X(NPOINTS) + TSPAC
000434      IF (KEY(1).GT.0) X(NPOINTS+2)=X(NPOINTS) + TSPAC*2.
000443      CALL EXTRR(NPOINTS,A,B)
000446      PB(1)=Y(1)-H(1)*XD1(1)
000461      PB(2)=Y(1)
C
C
C      INPUT AND OUTPUT SPACINGS ARE IDENTICAL FOR KEY(1)=-1
C
C
C
C
C
C
C
C
C      IF KEY(1).NE.1 CALL INTERPOLATION ROUTINE TO COMPLETE DATA POINTS
C
000462      IF (KEY(1).EQ.-1) GO TO 76
000465      IF (KEY(1).EQ.1) CALL EXTRL(TSPAC,1,YS1,YS2)
000476      IF (KEY(1).NE.1) CALL EXTRL(H(1),1,YS1,YS2)
000511      PB(1)=YK(2)
000512      J=1
000513      K=1
000514      I=1
000515      IC=1
000516      L=2

```

```

000517      T=0.0
C
C      BEGIN MARCH
000521      40 T=T+H(J)
000524      IC=IC+1
000526      41 L=L+1
C
000530      IF (T.GT.(TLAST+1.0E-6)) GO TO 75
C
C      CHECK TO SEE IF T IS AT ARBITRARILY NEAR A READ IN DATA POINT
000534      IF (ABS(T-X(I+1)).LT.1.0E-6) GO TO 50
C
C      CHECK TO SEE IF T IS GREATER THAN NEXT READ IN DATA POINT
C
000540      42 IF (T.GT.(X(I+1)+1.0E-6)) GO TO 60
C
C      CHECK TO SEE IF THERE IS AN OUTPUT SPACING CHANGE
C
000546      43 IF (T.GT.(TSTEPH(J)+1.0E-6)) GO TO 67
C
C      CHECK FOR EVEN SPACING SIMPLIFICATIONS
C
000554      44 IF (KEY(2).EQ.1.AND.KEY(1).EQ.1) GO TO 70
000564      IF (KEY(4).EQ.0) GO TO 45
C
C      COMPUTE GENERAL CASE
000566      PB(L)=Y(I)+(T-X(I))*XD1(I)+(T-X(I))*(T-X(I+1))*XD2(I-1)
000617      IF (I.LE.1) PB(L)=PB(L)+(T-X(I))*(T-X(I+1))*(T-X(I+2))*XD3(I-1)
1      +(T-X(I))*(T-X(I+1))*(T-X(I+2))*(T-X(I+3))*XD4(I-2)
000662      IF (I.GE.2) PB(L)=PB(L)+(T-X(I))*(T-X(I+1))*(T-X(I-1))*XD3(I-1)
1      +(T-X(I))*(T-X(I+1))*(T-X(I+2))*(T-X(I-1))*XD4(I-2)
000726      GO TO 40
C
000727      50 PB(L)=Y(I+1)
000732      T=T+H(J)
000736      IC=IC+1
000737      L=L+1
000740      60 I=I+1
000742      IF (KEY(4).EQ.0) GO TO 43
000744      IF (KEY(3).LT.3) KEY(3)=KEY(3)+1
000751      IF (KEY(3).EQ.2) GO TO 79
000754      80 IF (X(I+2).LE.(TDISC(K)+1.0E-6)) GO TO 43

```

```

000762      KEY(3)=1
000764      K=K+1
000766      CALL EXTRR(I+1,YS1,YS2)
000771      GO TO 43
000775      79 IF (KEY(1).EQ.1) CALL EXTRL(TSPAC,I,YS1,YS2)
001007      IF (KEY(1).EQ.0) CALL EXTRL(H(J),I,YS1,YS2)
001022      GO TO 43
001023      67 T=T-H(J)+H(J+1)
001027      J=J+1
001031      GO TO 44
001031      70 IF (I.LT.3) GO TO 72
001034      IF (KEY(4).EQ.0) GO TO 45
001036      PB(L)=.023459*Y(I-2)-.156250*Y(I-1)+.703125*Y(I)+.46875*Y(I+1)-
1          .039063*Y(I+2)
001055      71 L=L+1
001057      PB(L)=Y(I+1)
001062      I=I+1
001062      T=T+H(J)
001066      IC=IC+1
001067      GO TO 40
001070      72 PB(L)=Y(I)+(T-X(I))*XD1(I)
001107      GO TO 71
001110      75 PB(1)=2.*PB(2)-PB(3)
001113      PB(L)=2.*PB(L-1)-PB(L-2)
001121      78 WRITE(6,16) (PB(I),I=1,L)
001137      WRITE(6,17) IC,(KEY(I),I=1,6)
001162      RETURN

C
C      USE LINEAR EXTRAPOLATION IF KEY(4) EQUAL 1
001163      45 PB(L)=Y(I)+(T-X(I))*XD1(I)
001204      IF (KEY(2).EQ.1) GO TO 71
001210      GO TO 40

C
001210      76 DO 77 I=2,NPOINTS
001212      77 PB(I+1)=Y(I)
001217      I=NPOINTS+1
001220      PB(I+1)=PB(I)+H(NH)*XD1(NPOINTS-1)
001233      RETURN
001233      9 FORMAT(9I4,3E12.4)
001233      10 FORMAT(6E12.4)
001233      11 FORMAT(12H0 THERE ARE ,I6,72H DATA POINTS ACTING SEQUENTIALLY AT
1THE FOLLOWING RELATIVE TIME POINTS /(9E12.4))

```

C01233 12 FORMAT(1H0,17X,45H THE SEQUENTIAL VALUES OF THE DATA POINTS ARE /
 1 (8E12.4))
 C01233 13 FORMAT(1H0,17X,10H THERE ARE,16,67H DISTINCT OUTPUT REGIONS WITH R
 1ESPECTIVE OUTPUT SPACINGS EQUAL TO /(8E12.4))
 C01233 14 FORMAT(1H0,17X,10H THERE ARE,16,68H SMOOTHNESS DISCONTINUITIES OF
 1THE INPUT DATA LOCATED AT TIME POINTS / (8E12.4))
 D01233 15 FORMAT(1H0,17X, 6H THE ,16,65H CHANGES IN OUTPUT SPACINGS OCCUR A
 1T THE FOLLOWING TIME POINTS /(8E12.4))
 C01233 16 FORMAT(1H0,17X,55H CALCULATED DATA POINTS + 2 END POINTS USED AS L
 1OADING /(8E12.4))
 D01233 17 FORMAT(4X,11H THERE ARE ,16,67H CALCULATED DATA POINTS COMPUTED AC
 1CCORDING TO THE FOLLOWING KEYS //38H KEY-1- , -2-, -3-, -4-,
 2 -5-, -6- /513)
 C01233 END

```

SUBROUTINE READA(LI,I,CL)
C0C006   DIMENSION CL(90,90)
C00006   IF(I.LT.1) RETURN
C0C011   READ(5,117) (CL(I,J),J=1,LI)
0C0030   WRITE(5,103)(CL(I,J),J=1,LI)
C00051   RETURN
0C0052   103 FORMAT((7E12.4)//)
0C0052   117  FORMAT(6E12.4)
0C0052   END

```

```

SUBROUTINE READB(LI,IDP,CL)
000006 DIMENSION CL(1)
000006 IF(IDP.LT.1) RETURN
000011 READ(5,117)(CL(I),I=1,LI)
000027 IF(IDP.GT.1) GO TO 112
000034 112 WRITE(6,103) (CL(I),I=1,LI)
000052 RETURN
000053 WRITE(6,111)(CL(I),I=1,LI)
000073 RETURN
000074 103 FORMAT((7E12.4)//)
000074 111 FORMAT(1H0,15X,49H PCT. OF CRITICAL DAMPING IN MODE (MODAL DAMPING
1 / (10F12.4))
000074 117 FORMAT(6E12.4)
000074 END

```



```

SUBROUTINE STIFF(NM,NS,NFOT,NCL,NDP,N,IDP,INDEX,S,ST,LPHA,ROT,
1CX,CL,CR)
000022 COMMON/LIST/ AM(90),VECTS(8100)
000022 COMMON/GAMMA/Z(270),NUM
000022 COMMON/AAAAA/AMM(90)
000022 DIMENSION S(90,90),ST(30),LPHA(90),ROT(90),CX(90,90),CL(90),CR(90)
C THIS SUBROUTINE COMPUTES AND ASSEMBLES A STIFFNESS MATRIX FOR SPRING
C AND DAMPER CONNECTED ROTATABLE ELEMENTS WITH OFFSETS
000022 WRITE (6,102) NS
000027 READ (5,117) (ST(L),L = 1, NS)
000053 WRITE (6,103) (ST(L),L = 1, NS)
C
000077 IF (NROT) 96,14,15
000105 14 DO 15 LK=1,NS
000107 CR(LK)=0.0
000111 15 ROT(LK)=0.
000115 IF (NDP.EQ.0) GO TO 18
000116 READ (5,117) (CL(J),J=1,NS)
000136 WRITE(6,118)
000142 WRITE(6,103) (CL(J),J=1,NS)
000166 GO TO 18
C READ AND WRITE (NROT) NUMBER OF ROTATIONAL COORDINATES
000172 16 READ(5,117) ( ROT(LK),LK=1,NS)
000213 WRITE (6,112) NS
000225 WRITE (6,103) (ROT(LK),LK=1,NS)
000251 IF (NDP.EQ.0) GO TO 18
000256 READ (5,117) (CL(J),J=1,NS)
000276 WRITE(6,118)
000302 WRITE(6,103) (CL(J),J=1,NS)
000326 WRITE(6,119)
000332 READ (5,117) (CR(J),J=1,NS)
000356 WRITE(6,103) (CR(J),J=1,NS)
000402 18 DO 19 MJ=1,N
000407 LPHA(MJ)=0
000411 DO 19 MI=1,N
000413 S(MI,MJ)=0.
000417 CX(MI,MJ)=0.0
000423 19 CONTINUE
C
C MODAL DAMPING CORRECTIONS
C

```

```

000427      IF (IDP.GT.0) READ(5,117) (CL(J),J=1,N)
000455      IF (IDP.GT.0) WRITE(6,111) (CL(J),J=1,N)
C          IDENTIFY TOPOLOGY . READ IN MASSES, MAXIMUM LOADS AND NUMBERS OF
C          POINTS ATTACHING TO THE RESPECTIVE MASS
000504      WRITE (6,104)
000510      WRITE (6,105)
000514          K=NM
000521          IJK=1
000522          I=0
000523      17  I=I+1
000525          READ(5,106) AM(I),CINT,JL,JC,JR,OFFSET,J0
000547          WRITE(6,106) AM(I),CINT,JL,JC,JR,OFFSET,J0
C          CHECK WHETHER ANY 3 POINT INDICES FOR ROTATIONAL SPRINGS HAVE
C          BEEN PREVIOUSLY STORED
000572          NK=NM/2
000577          DO 21 IL=1,NK
000601          IF (LPHA(IL)) 96,22,20
000604      20  LI= LPHA(IL)/100
000611          LJ=K+1
000613          IF (LI.EQ.I) S(LJ,LI) =-ROT(LI)
000623      21  CONTINUE
C
C          COMPUTE DYNAMICAL MATRIX S=(MASS)**-1(STFS.) SEPARATING SPECIAL
C          CASES DEPENDING ON ZERO VALUES OF JL,JC, OR JR
C          FIRST SEPARATE OUT ONE POINT ATTACHMENTS
000626      22  IF(JC) 96,23,27
000630      23  S(I,I) = ST(I)
000636          IF (NDP.NE.0) CX(I,I) = CL(I)
000644          GO TO 50
C          SEPARATE TWO AND THREE POINT CASES
000645      27  IF(JR) 96,28,37
C          TWO POINT CASE
000647      28  S(I,I) = (ST(I) + ST(JC))
000660          IF (NDP.NE.0) CX(I,I) = CL(I) + CL(JC)
000670          IF (JC.GT.NM) GO TO 50
000673          S(I,JC) = -ST(JC)
000700          IF (NDP.NE.0) CX(I,JC) = -CL(JC)
000707      33  GO TO 50
C          THREE POINT CASE
000710      37  S(I,I) = (ST(I)+ST(JC)+ST(JR))
000725          IF (NDP.NE.0) CX(I,I) = CL(I) + CL(JC) + CL(JR)
000737          IF (JC.GT.NM) GO TO 50

```

```

000742      S(I,JC) = - ST(JC)
000747      IF(NDP.NE.0) CX(I,JC)=-CL(JC)
000756      IF( JR.GT.NM) GO TO 50
000761      S(I,JR) = - ST(JR)
000766      IF(NDP.NE.0) CX(I,JR)=-CL(JR)
000775      50      IF(CINT - .0000006) 53,58,52
001000      52      K=K+1
001002      AM(K) =CINT
C          FOR ROTATIONAL COORDINATES SEPARATE CASES,
001004      IF(JC) 96,53,54
C          ONE POINT CASE
001006      53      S(K,K)= +ROT(I)
001014      IF(NDP.NE.0) CX(K,K) = CR(I)
001023      GO TO 58
001024      54      IF (JR) 96,55,56
C          TWO POINT CASE
001026      55      S(K,K)= ROT(I)+ROT(JC)
001036      IF(NDP.NE.0) CX(K,K) = CR(I) + CR(JC)
001047      IF(JC.GT.NM) GO TO 57
001052      S(K,K+1) =-ROT(JC)
001057      IF(NDP.NE.0) CX(K,K+1) =-CR(JC)
001066      GO TO 58
C          THREE POINT CASE
C          TWO COMPLICATIONS, OFFSET AFFECTS DEFLECTIONS AND
C          THE EXACT INDEX FOR THIRD INERTIA IS UNKNOWN AT THIS TIME
001067      56      S(K,K)= ROT(I)+ROT(JC)+ROT(JR)
001103      IF(NDP.NE.0) CX(K,K) = CR(I) + CR(JC)+CR(JR)
001117      IF(JC.GT.NM) GO TO 57
001122      IF(ROT(JC).GT.(.000001)) GO TO 61
001126      S(K,K+1)=-ROT(JR)
001134      IF(NDP.NE.0) CX(K,K+1) =-CR(JR)
001143      GO TO 57
001144      61      S(K,K+1) =-ROT(JC)
001152      IF(NDP.NE.0) CX(K,K+1) =-CR(JC)
001161      IF(JR.GT.NM) GO TO 57
001164      IF(ROT(JR).LT.(.000001)) GO TO 57
C          STORE 3 POINT ROTATIONAL SPRING INDICES FOR FUTURE USE
001167      LPHA(IJK) =100 * JR + K
001174      IJK=IJK + 1
001175      57      IF(ABS(OFFSET).LT.(.000001)) GO TO 53
001201      S(I,K)=S(I,K)-OFFSET*ST(JC)
001211      CX(I,K)=CX(I,K)-OFFSET*CL(JC)

```

```

001221      58      CONTINUE
001221      51      IF(I.LT.NM) GO TO 17
001223          IF(NCL.EQ.0) GO TO 71
001224          DO 65 J=1,NCL
001226          READ(5,115) INDX,NMASS,NMASS1,NSPR
001241          GO TO (62,64,67) , INDX
001254      62 S(NMASS,NMASS1)= S(NMASS,NMASS1)-ST(NSPR)
001263          GO TO 65
001264      64 S(NMASS,NMASS1)=S(NMASS,NMASS1)-ROT(NSPR)
001273          GO TO 65
001274      67 GO TO 65
001275      65 CONTINUE
001300      71 NM=N
001301          NN=K
001303          WRITE(6,134) NN
001310          NCL=0
C          CONVERT FROM UPPER TRIANGULAR STORAGE TO SYMMETRIC STORAGE
001314          DO 59 I=1,N
001316          DO 59 J=1,I
001317          S( I,J) = S(J,I)
001326          CX(I,J)=CX(J,I)
001335      59 CONTINUE
001340          WRITE(6,122)
001343          DO 72 I=1,N
001350          WRITE(6,103) (S(I,J),J=1,N)
001372      72 CONTINUE
001400          IF(NDP.EQ.0) RETURN
001401          WRITE(6,120 )
001405          DO 70 I=1,N
001412          WRITE(6,103) (CX(I,J),J=1,N)
001434      70 CONTINUE
C          MULTIPLY STIFFNESS BY (MASS)**-1
C
001442          INDEX = 0
001443          RETURN
001444      96 WRITE(6,115)
001450          INDEX = 1
001452          RETURN
001452      102 FORMAT (10H0 ,I2,50H SPRING CONSTANTS LISTED IN ORDER 10
1 TO A LINE )
001452      103 FORMAT((7E12.4)//)
001452      104 FORMAT (1H0,36X,18H IDENTIFY TOPOLOGY//)

```

```

001452 105 FORMAT(8H0 MASS,7X,9H INERTIA ,6H 1 FT.,6H 2 FT.,6H 3 FT.,
      1 12H OFFSET ,12H OFFSET MASS )
001452 106 FORMAT(2E12.5,3I6,E12.5,I6)
001452 111 FORMAT(1H0,15X,49H PCT. OF CRITICAL DAMPING IN MODE (MODAL DAMPING
      1 /((10F12.4)))
001452 116 FORMAT(4I10)
001452 112 FORMAT(1H0,I2,60H ROTATIONAL SPRING CONSTANTS LISTED IN ORDER
      16 TO A LINE )
001452 115 FORMAT ( 51H1 INCORRECT NEGATIVE SIGN IN INPUT CHECK RESULTS )
001452 117 FORMAT(6E12.4)
001452 118 FORMAT(1H0,30X,36H LINEAR VISCOUS DAMPERS (LB-SEC/IN) )
001452 119 FORMAT(1H0,30X,42H ROTATIONAL VISCOUS DAMPERS (LB-SEC-IN) )
001452 120 FORMAT(1H0,25X, 36H DAMPING MATRIX IN REGULAR COORDINATES )
001452 122 FORMAT(1H0,30X,20H STIFFNESS MATRIX )
001452 134 FORMAT(29H0 THE COMPUTER INTERPRETS AS ,14,21H DEGREES OF FREEDOM
      1 )
001452 END

```

```

SUBROUTINE TRANS(DIX,DIY,VIX,VIY,N,NOPTIO,S)
COMMON/LIST/ AM(90),VECTS(8100)
DIMENSION DIY(1),VIY(1),DIX(1),VIX(1)
DIMENSION S(90,90)
DO 20 I= 1,N
  DIY( I)=0.0
  VIY( I)=0.0
20 CONTINUE
IF(NOPTIO.EQ.2) GO TO 40
JK=-N
DO 30 I= 1,N
  JK=JK+N
DO 30 J= 1,N
  JI=JK+J
  DIY( I)=DIY(I)+AM(J)*VECTS(JI)*DIX(J)
  VIY( I)=VIY(I)+AM(J)*VECTS(JI)*VIX(J )
30 CONTINUE
RETURN
40 KL=-N
DO 50 I=1,N
  KL=KL+N
DO 50 K=1,N
  KI=KL+K
DO 50 L=1,N
  DO 50 J=1,N
    A=VECTS(KI)*S(L,K)*S(L,J)
    DIY(I)=DIY(I)+A*DIX(J)
50 VIY(I)=VIY(I)+A*VIX(J)
RETURN
END

```

```

SUBROUTINE TRIANG(A,B,N)
C
C FINDS UT MATRIX B SO THAT  $A=B*I*B$ 
C
000006 DIMENSION A(90,90),B(90,90)
000006 DO 7 I=1,N
000007 DO 7 J=1,N
000010 B(I,J) =0.0
000014 IF(I.GT.J) GO TO 7
000016 B(I,J) =A(I,J)
000023 K=0
000024 3 K=K+1
000026 IF(K.GE.I) GO TO 4
000030 B(I,J)=B(I,J) - B(K,I)*B(K,J)
000042 GO TO 3
000042 4 IF(I.EQ.J) B(I,J)=SQRT(B(I,J))
000054 IF(I.NE.J) B(I,J)=B(I,J)/B(I,I)
000064 7 CONTINUE
000071 RETURN
000071 END

```

```

SUBROUTINE TRINV(A,B,N)
C
C FIND UT MATRIX B THE INVERSE OF UT MATRIX A
C
000006 DIMENSION A(90,90),B(90,90)
000006 DO 7 I=1,N
000007 DO 7 J=1,N
000010 B(I,J)=0.0
000014 IF(I.GT.J) GO TO 7
000016 IF(I.LT.J) GO TO 3
000017 B(I,I) = 1.0/A(I,I)
000025 GO TO 7
000025 3 K=0
000026 5 K=K+1
000030 IF(K.GT.(J-I)) GO TO 7
000035 B(I,J)=B(I,J)-B(I,K+I-1)*A(K+I-1,J)/A(J,J)
000052 GO TO 5
000052 7 CONTINUE
000057 RETURN
000057 END

```



```

SUBROUTINE EIGEN (A,R,N,MV)
000007   COMMON/AAAAA/AMM(90)
000007   DIMENSION A(1),R(1)
C       GENERATE IDENTITY MATRIX
000007   IF(MV -1) 10,25,10
000011   10   IQ = -N
000012       DO 20 J= 1,N
000014       IQ = IQ + N
000016       DO 20 I= 1,N
000017       IJ = IQ + I
000020       R(IJ)=0.0
000022       IF (I-J) 20,15,20
000024   15   R (IJ)= 1.0
000027   20   CONTINUE

C
C       COMPUTE INITIAL AND FINAL NORMS
C
000034   25   ANORM =0.0
000035       DO 35 I=1,N
000037       DO 35 J=1,I
000040       IF(I-J) 35,35,30
000042   30   IA=(I+I-I)/2 +J
000046       ANORM =ANORM +A(IA)*A(IA)
000052   35   CONTINUE
000057       IF (ANORM) 165,165, 40
000060   40   ANCRM= SQRT(ANCRM/FLOAT(N))
000067       ANRMX=ANORM*1.E-7/ FLOAT(N)

C
C       INITIALIZE INDICATORS AND COMPUTE THRESHOLD,THR
C
000072       IND = 0
000073   45   THR = ANORM
000075   50   L= 1
000076   55   M= L+1

C
C       COMPUTE SIN AND COS
C
000100   60   MQ = (M*M -M)/2
000103       LQ =(L*L-L)/2
000106       LM = L+ MQ
000110   62   IF(ABS(A(LM))-THR)130,65,65

```

```

000115      65      IND = 1
000116          LL = L+ LG
000120          MM = M+ MQ
000122          X  = 0.5*(A(LL)-A(MM))
000127      68      Y  = -A(LM)/ SQRT(A(LM)*A(LM)+X*X)
000142          IF (X) 70,75, 75
000144      70      Y = -Y
000145      75      SINX =Y / SQRT(2.0*(1.0 +(SQRT (1.0 -Y*Y) )))
000156          SINX2= SINX * SINX
000160      78      COSX = SQRT (1.0 -SINX2)
000164          COSX2= COSX * COS X
000165          SINCX= SINX * COSX

C
C      ROTATE L AND M  COLUMNS
C

000167          ILQ= N* (L-1)
000175          IMG= N* (M-1)
000200          DO 125 I=1,N
000201          IQ = (I*I -I)/ 2
000203          IF (I-L) 80,115,80
000205      80      IF ( I -M) 85,115,90
000210      85      IM =I +MQ
000212          GO TO 95
000213      90      IM = M + IQ
000215      95      IF (I-L) 100, 105,105
000220      100     IL = I + LQ
000222          GO TO 110
000223      105     IL = L  + IQ
000225      110     X = A(IL)*COSX -A(IM)*SINX
000233          A(IM)= A(IL)*SINX +A(IM)*COSX
000237          A(IL) = X
000241      115     IF (MV-1) 120 , 125, 120
000243      120     ILR =ILQ + I
000245          IMR =IMG + I
000246          X= R(ILR)*COSX -R(IMR)*SINX
000254          R(IMR)=R(ILR) *SINX +R(IMR)*COSX
000260          R(ILR)= X
000262      125     CONTINUE
000265          X = 2.*A(LM )* SINCX
000270          Y = A(LL)*COSX2 +A(MM)*SINX2 -X
000277          X = A(LL)*SINX2 +A(MM)*COSX2 +X
000304          A(LM) = (A(LL)-A(MM))*SINCX +A(LM)*(COSX2 - SINX2)

```

```

000315      A(LL) = Y
000317      A(MM) = X
      C      TESTS FOR COMPLETION
      C      FIRST TEST FOR M= LAST COLUMN
000322      130      IF (M-N) 135,140,135
000324      135      M=M+1
000326      GO TO 60
      C      TEST FOR L = SECOND FROM LAST CARD
000326      140      IF ( L-(N-1)) 145,150,145
000331      145      L= L +1
000333      GO TO 55
000333      150      IF (IND -1) 160,155 ,160
000335      155      IND = 0
000336      GO TO 50
      C      COMPARE THRESHOLD WITH FINAL NORM
000337      160      IF (THR-ANRMX)165, 165,45
      C      SORT NATURAL FREQUENCIES AND NATURAL MCDES
000342      165      IQ = - N
000343      DO 185 I=1,N
000345      IQ =IQ + N
000346      LL =I +(I*I-I)/2
000351      JQ= -N
000353      DO 185 J = 1,N
000354      JQ =JQ+N
000355      MM =J +(J*J-J)/2
000360      IF (A(LL)-A(MM))170, 185, 185
000365      170      X= A(LL)
000367      A( LL)= A(MM)
000372      A( MM)= X
000374      IF (MV-1)175,185,175
000376      175      DO 180 K=1,N
000400      ILR = IQ +K
000401      IMR = JQ +K
000403      X= R(ILR)
000405      R( ILR)= R(IMR)
000410      180      R( IMR)= X
000415      185      CONTINUE
      C      NORMALIZE EIGENVECTORS WITH MASS WEIGHTING
000422      DO 195 J=1,N
000423      QL= 0.0
000424      JK=J*N
000425      JM=(J-1)*N+1

```

```

000431      DO 183  I=JM,JK
000432      IF (ABS(R(I)).LT.(1.0*10.**60)) GO TO 187
000441      ABC=R(I)
000443      GO TO 189
000443 187      QL= QL+R(I)*R(I)
000446 188      CONTINUE
000451      ABC=SQRT(QL)
000452      AMM(J)=ABC
000455 189      DO 190  I= JM,JK
000462      R(I)= R(I)/ ABC
000464 190      CONTINUE
000466 195      CONTINUE
000471      RETURN
000471      END

```

```

000007      SUBROUTINE ROTATE (IJ,IK,A,N)
          DIMENSION A(90,90)
C
C      SUBROUTINE TO INTERCHANGE ROWS AND COLUMNS OF SQUARE MATRIX
C
000007      DO 11 I=1,N
000010      AA = A(IJ,I)
000014      A(IJ,I)=A(IK,I)
000021      11 A(IK,I)=AA
000025      DO 12 I=1,N
000027      BB = A(I,IJ)
000033      A(I,IJ)=A(I,IK)
000040      12 A(I,IK)=BB
000046      WRITE(6,14) IJ,IK
000055      14 FORMAT(79H0 THE FOLLOWING ROWS AND COLUMNS OF THE CORRESPONDING MA
          1TRIX WERE INTERCHANGED ,2I5)
000055      RETURN
000056      END

```

```

000005      SUBROUTINE INPUTM(IN,A)
          DIMENSION A(90,90),B(300),NCOR(300)
C          SUBROUTINE TO READ IN NON-ZERO TERMS OF SYMMETRIC MATRIX
000005      DO 17 I=1,IN
000006      DO 17 J=1,IN
000007      17 A(I,J)=0.0
000017      READ(5,10)NC,NFIRST
000026      READ(5,10) (NCOR(I),I=1,NC)
000041      10 FORMAT(20I4)
000041      WRITE(6,11)NC,NFIRST,(NCOR(I),I=1,NC)
000060      11 FORMAT(4CHONUMBER OF NON-ZERO SYMMETRIC ENTRIES IS ,I6,33H WITH T
          1HE FIRST TERM NUMBERED AS ,I5,28H AND DESIGNATED NODE NUMBERS/(
          2 20I5))
000060      READ(5,16) (B(I),I=1,NC)
000073      WRITE(6,12)(B(I),I=1,NC)
000106      12 FORMAT((7E12.4)//)
000106      DO 14 I=1,NC
000111      NCOR(I)=NCOR(I)-NFIRST+1
000115      NK=NCOR(I)/IN
000120      NJ=NCOR(I)-NK*IN
000124      IF (NJ.NE.0)GO TO 14
000125      NJ=IN
000126      NK=NK-1
000127      14 A(NK+1,NJ)=B(I)
000137      DO 15 I=1,IN
000141      DO 15 J=1,I
000142      15 A(I,J)=A(J,I)
000157      16 FORMAT(6E12.4)
000157      RETURN
000157      END

```

```

000005      SUBROUTINE OUTPUT (IN,A)
          DIMENSION  A(90,90)
C
000005      DO 10 I=1,IN
000006        WRITE (6,15)(A(I,J),J=1,IN)
000024      10 CONTINUE
000030      15 FORMAT((7E12.4)//)
000030      RETURN
000030      END

```

```

SUBROUTINE LOAD(FR,C,AHI,AHJ,NDP,H,N)
000012  DIMENSION FR(9C),C(3J),AHI(9C,5),AHJ(9C,5)
C      THIS SUBROUTINE COMPUTES DUHAMEL S INTEGRAL WITH (T-TI)**(I-1) FUNCTIONS
000012  DO 4 I=1,N
000013    F=FR(I)
000015    D=C(I)/2.
000020    X=F*F+D*D
000022    EH=EXP(-D*H)
000031    CH=COS(F*H)
000040    SH=SIN(F*H)
000047    AHI(I,1)=(F-F*EH*CH)/X
000055    AHJ(I,1)=(F*EH*SH)/X
000060    IF(NDP.EQ.0) GO TO 11
000061    AHI(I,1)= AHI(I,1)-(D*EH*SH)/X
000066    AHJ(I,1)=AHJ(I,1)-(D*EH*CH)/X      +D/X
000075  11 CONTINUE
000075    DO 5 J=2,5
000077    K=J-1
000101    Y=K
000102    AHI(I,J)=F/X*(H**K-Y*AHJ(I,J-1))
000117    AHJ(I,J)=F/X*Y*AHJ(I,J-1)
000125    IF(NDP.EQ.0) GO TO 12
000126    AHI(I,J)=AHI(I,J)-Y/X*D*AHI(I,J-1)
000136    AHJ(I,J)=AHJ(I,J)-Y/X*D*AHJ(I,J-1)+H**K*D/X
000153  12 CONTINUE
000153    5 CONTINUE
000155    4 CONTINUE
000160    RETURN
000160    END

```



```

000007      SUBROUTINE EXTRL(H,K,YS1,YS2)
000007      COMMON/EXT/X(1002),Y(1002),XK(5),YK(5),KEY(6)
000011      IF(K.LE.1) GO TO 4
000012      Y(K+1)=YS1
000014      Y(K+2)=YS2
000017      X(K-1)=X(K)-H
000023      X(K-2)=X(K)-2.*H
000025      4 DO 5 J=1,3
000032      XK(2+J)=X(K+J-1)
000037      5 YK(2+J)=Y(K+J-1)
000041      XK(2)= X(K)-H
000044      XK(1)= X(K)-2.*H
000063      YK(2)= Y(K)-H* XD1(K)+H*H*2.*XD2(K)
000103      YK(1)= Y(K)-2.*H* XD1(K) +6.*H*H *XD2(K)
000106      IF(K.LE.1) RETURN
000111      Y(K-1)= YK(2)
000113      Y(K-2)=YK(1)
000113      RETURN
000113      END

```

```

000006      SUBROUTINE EXTRR(K,YS1,YS2)
          COMMON/EXT/X(1002),Y(1002),XK(5), YK(5),KEY(6)
C
000006      YS1=Y(K+1)
000007      YS2=Y(K+2)
000010      Y(K+1)=Y(K)+(X(K+1)-X(K))*XD1(K-1)+(X(K+1)-X(K))*(X(K+1)-X(K-1))*
1  XD2(K-2)
000037      Y(K+2)=Y(K)+(X(K+2)-X(K))*XD1(K-1)+(X(K+2)-X(K))*(X(K+2)-X(K-1))*
1  XD2(K-2)
000066      WRITE(6,10) YS1,YS2,Y(K+1),Y(K+2)
000106      10 FORMAT(48X,4E12.4)
000106      RETURN
000107      END

```

```

      FUNCTION XD1(K)
C
000003      COMMON/EXT/X(1002),Y(1002),XK(5), YK(5),KEY(6)
000003      16 XD1=(Y(K+1)-Y(K))/(X(K+1)-X(K))
000011      RETURN
000011      END

```

FUNCTION XD2(K)

C

```

000003      COMMON/EXT/X(1002),Y(1002),XK(5), YK(5),KEY(6)
000003      IF(KEY(6).NE.0) GO TO 16
000004      IF (K.LE.0) GO TO 14
000006      XD2=(Y(K)-Y(K+1))/((X(K)-X(K+1))*(X(K)-X(K+2))) -(Y(K+1)-Y(K+2))/(
1          (X(K+1)-X(K+2))*(X(K)-X(K+2)))
000025      RETURN
000026  14  XD2=(YK(2)-Y(K+1))/((XK(2)-X(K+1))*(XK(2)-X(K+2)))-(Y(K+1)-Y(K+2))
1          /(X(K+1)-X(K+2))*(XK(2)-X(K+2)))
000044      RETURN
000045  16  IF(K.LE.0) GO TO 18
000047      DEM=2.*((X(K+1)-X(K))*2)
000052      XD2=(Y(K)-2.*Y(K+1)+ Y(K+2))/DEM
000060      RETURN
000061  18  DEM=2.*((X(K+2)-X(K+1))*2)
000064      XD2=(YK(2)-2.*Y(K+1)+Y(K+2))/DEM
000072      RETURN
000072      END

```

FUNCTION XD3(K)

C

```
000003      COMMON/EXT/X(1002),Y(1002),XK(5),YK(5),KEY(6)
000003      IF(KEY(6).NE.0) GO TO 16
000004      IF(K.LE.0) GO TO 14
000006      XD3=(XD2(K)-XD2(K+1))/(X(K)-X(K+3))
000022      RETURN
000022  14  XD3=(XD2(K)-XD2(K+1))/(XK(2)-X(K+3))
000037      RETURN
000037  16  IF(K.LE.0) GO TO 18
000041      DEM=6.*((X(K+1)-X(K))**3)
000044      XD3=(-Y(K)+3.*Y(K+1)-3.*Y(K+2)+Y(K+3))/DEM
000055      RETURN
000055  18  DEM=6.*((X(K+2)-X(K+1))**3)
000061      XD3=(-YK(2)+3.*Y(K+1)-3.*Y(K+2)+Y(K+3))/DEM
000070      RETURN
000071      END
```

```

      FUNCTION XD4(K)
C
000003   COMMON/EXT/X(1002),Y(1002),XK(5), YK(5),KEY(6)
000003   IF (KEY(6).NE.C) GO TO 16
000004   IF (K.EQ.0) GO TO 14
000005   IF (K.LT.0) XD4=YK(1)/((XK(1)-XK(2))*(XK(1)-XK(3))*(XK(1)-XK(4))*(XK
1(1)-XK(5)))+YK(2)/((XK(2)-XK(1))*(XK(2)-XK(3))*(XK(2)-XK(4))*(XK(2
2)-XK(5)))+YK(3)/((XK(3)-XK(1))*(XK(3)-XK(2))*(XK(3)-XK(4))*
3(XK(3)-XK(5)))+YK(4)/((XK(4)-XK(1))*(XK(4)-XK(2))*(XK(4)-XK(3)
4 )*(XK(4)-XK(5)
5))+YK(5)/((XK(5)-XK(1))*(XK(5)-XK(2))*(XK(5)-XK(3))*(XK(5)-XK(4)))
000101   IF (K.LT.0) RETURN
000104   XD4=(XD3(K)-XD3(K+1))/(X(K)-X(K+4))
000121   RETURN
000121 14 XD4=(XD3(K)-XD3(K+1))/(XK(2)-X(K+4))
000136   RETURN
000136 16 DEM=24.*((X(K+3)-X(K+2))**4)
000142   IF (K.LE.0) GO TO 18
000143   XD4=(Y(K)-4.*Y(K+1)+6.*Y(K+2)-4.*Y(K+3)+Y(K+4))/DEM
000155   RETURN
000155 18 IF (K.EQ.0) XD4=(YK(2)-4.*Y(K+1)+6.*Y(K+2)-4.*Y(K+3)+Y(K+4))/DEM
000167   IF (K.LT.0) XD4=(YK(1)-4.*YK(2)+6.*Y(K+2)-4.*Y(K+3)+Y(K+4))/DEM
000202   RETURN
000204   END

```

```

SUBROUTINE CHANGE(A,3,      V, J,JT1,JT2)
000011  COMMON/GAMMA/Z(270),NUM
000011  COMMON/LIST/ AM(90),VECTS(9100)
000011  DIMENSION A(1), B(1)
000011  IF(J.EQ.0) GO TO 60
000012  IF(JT1.GT.0) B(JT1)=0.0
000015  IF(JT2.GT.0) B(JT2)=0.0
000020  IF(JT1.NE.0) GO TO 35
C      ALL VALUES ARE NEEDED
000021  DO 30 IB=1,N
000023  B(IB)=0.0
000024  IJ=IB-N
000026  DO 30 IA=1,N
000027  IJ=IJ+N
000030  B(IB) =B(IB)+      A(IA)      *VECTS(IJ)
000036  30  CONTINUE
000042  WRITE(6,228) (B(I),I=1,N)
000061  IF(J.NE.2) GO TO 60
000067  DO 33 I=1,N
000070  NN=NUM+I
000071  33  Z(NN)=B(I)
000076  NUM=NUM+N
000077  RETURN
000100  35  IF(JT2.EQ.0) GO TO 45
000101  JK=-N
000102  DO 40 I=1,N
000104  JK=JK+N
000105  IJ=JK+JT1
000106  IK=JK+JT2
000110  B(JT1)=B(JT1)+      A(I)      *VECTS(IJ)
000116  B(JT2)=B(JT2)+      A(I)      *VECTS(IK)
000124  40  CONTINUE
000126  WRITE (6,228) B(JT1),B(JT2)
000145  IF(J.NE.2) GO TO 60
000153  N1=NUM+1
000154  N2=NUM+2
000155  Z(N1)=B(JT1)
000161  Z(N2)=B(JT2)
000164  NUM=N2
000165  RETURN
45  DO 50 I=1,N

```

```

000167      IJ = N*(I-1)+JT1
000172      B(JT1)=B(JT1)+      A(I)  *VECTS(IJ)
000177      50  CONTINUE
000201      WRITE (6,228) B(JT1)
000212      IF(J.NE.2)  GO TO 60
000220      N1=NUM+1
000221      Z(N1)=B(JT1)
000225      NUM=N1
000225      228  FORMAT(22X,7E12.4)
000225      60  RETURN
000226      END

```


Appendix B - Selected Printout of Example Problem 1

CHECK PROBLEM - DAMPED MOTION, BASE EXCITATION (UNDAMPED CASE WAR.
YOU HAVE INPUT THE FOLLOWING NUMBERS OR INDICES

D.O.F.,	SPRINGS,	CASES,	ROTATION,	SUPPORTS,	TAPES,	DAMPING,	OPTION
3	0	2	0	1	-2	1	-3

DIAGONAL MASS MATRIX

2.0000E+01 1.0000E+01 1.0000E+01

NUMBER OF NON-ZERO SYMMETRIC ENTRIES IS 6 WITH THE FIRST TERM NUMBERED AS 1 AND DESIGNATED NODE NUMBE

1 2 3 5 6 9

1.0000E+07 -1.0000E+04 0. 2.0000E+04 -1.0000E+04 1.0000E+04

STIFFNESS MATRIX ...OUTPUT BY ROWS...

1.0000E+07 -1.0000E+04 0.
-1.0000E+04 2.0000E+04 -1.0000E+04
0. -1.0000E+04 1.0000E+04

NUMBER OF NON-ZERO SYMMETRIC ENTRIES IS 6 WITH THE FIRST TERM NUMBERED AS 1 AND DESIGNATED NODE NUMBE

1 2 3 5 6 9

2.0000E+05 -2.0000E+02 0. 4.0000E+02 -2.0000E+02 2.0000E+02

DAMPING MATRIX IN REGULAR COORDINATES

2.0000E+05 -2.0000E+02 0.
-2.0000E+02 4.0000E+02 -2.0000E+02
0. -2.0000E+02 2.0000E+02

THE FOLLOWING ROWS AND COLUMNS OF THE CORRESPONDING MATRIX WERE INTERCHANGED 1 2

THE FOLLOWING ROWS AND COLUMNS OF THE CORRESPONDING MATRIX WERE INTERCHANGED 1 2

THE FOLLOWING ROWS AND COLUMNS OF THE CORRESPONDING MATRIX WERE INTERCHANGED 2 3

THE FOLLOWING ROWS AND COLUMNS OF THE CORRESPONDING MATRIX WERE INTERCHANGED 2 3

UNDAMPED NATURAL FREQUENCIES (CYC./SEC.)

3.1105E+00 8.1434E+00

DAMPING MATRIX , CY , IN NORMAL COORDINATES

CY = (Z**T)*CX*Z , (Z IS THE MODAL M

7.6393E+00 -1.7053E-13
-1.1369E-13 5.2361E+01

PCT. OF CRITICAL DAMPING IN MODE (MODAL DAMPING)

.1954 .5117

DAMPED NATURAL FREQUENCIES (CYC./SEC.)

3.0505E+00 6.9967E+00

EIGENVECTORS EIGHT COMPONENTS PER LINE

1.6625E-01 2.6900E-01 ...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 1.9167E+01 RAD./SEC.
2.6900E-01 -1.6625E-01 ...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 4.3962E+01 RAD./SEC.

NOTE THE FOLLOWING ANALYSIS ASSUMES THAT

- (1) THE MAIN DIAGONAL TERMS OF MATRIX CY ABOVE ARE MUCH GREATER THAN THE OFF DIAGONAL TERMS
- (2) THE UNDAMPED FREQUENCIES ARE SEPARATED WIDELY
- (3) PROPORTIONAL DAMPING, NC IN MODE PHASE CHANGES OF THE EIGENVECTORS DUE TO DAMPING

LOAD CASE 1

INPUT DESCRIBED BY DATA POINTS, SUPPORT DISPLACE.

DISPLACEMENTS.

ALL

OUTPUT IS ALSO ON TAPE FOR LOCAL CASE

VELOCITIES.

ALL

OUTPUT IS ALSO ON TAPE FOR LOCAL CASE

ACCELERATIONS.

ALL

OUTPUT IS ALSO ON TAPE FOR LOCAL CASE

THERE ARE 17 DATA POINTS ACTING SEQUENTIALLY AT THE FOLLOWING RELATIVE TIME POINTS

0. 6.2500E-02 1.2500E-01 1.8750E-01 2.5000E-01 3.1250E-01 3.7500E-01 4.3750E-01
5.0000E-01 5.6250E-01 6.2500E-01 6.8750E-01 7.5000E-01 8.1250E-01 8.7500E-01 9.3750E-01
1.0000E+00

THE SEQUENTIAL VALUES OF THE DATA POINTS ARE

0. 3.8268E-03 7.0711E-03 9.2388E-03 1.0000E-02 9.2388E-03 7.0711E-03 3.8269E-03
 2.6536E-08 -3.8269E-03 -7.0711E-03 -9.2388E-03 -1.0000E-02 -9.2388E-03 -7.0711E-03 -3.8268E-03
 0.

THERE ARE 1 DISTINCT OUTPUT REGIONS WITH RESPECTIVE OUTPUT SPACINGS EQUAL TO
 1.0000E-02

THE 0 CHANGES IN OUTPUT SPACINGS OCCUR AT THE FOLLOWING TIME POINTS
 1.0000E+00

THERE ARE 0 SMOOTHNESS DISCONTINUITIES OF THE INPUT DATA LOCATED AT TIME POINTS
 -0.

CALCULATED DATA POINTS + 2 END POINTS USED AS LOADING

-6.2646E-04	0.	6.2646E-04	1.2496E-03	1.8683E-03	2.4810E-03	3.0857E-03	3.6801E-03
4.2561E-03	4.8141E-03	5.3540E-03	5.8736E-03	6.3711E-03	6.8443E-03	7.2895E-03	7.7047E-03
8.0895E-03	8.4426E-03	8.7625E-03	9.0480E-03	9.2977E-03	9.5104E-03	9.6855E-03	9.8225E-03
9.9208E-03	9.9801E-03	1.0000E-02	9.9803E-03	9.9212E-03	9.8229E-03	9.6859E-03	9.5106E-03
9.2978E-03	9.0494E-03	8.7634E-03	8.4437E-03	8.0906E-03	7.7055E-03	7.2899E-03	6.8457E-03
6.3748E-03	5.8786E-03	5.3591E-03	4.8183E-03	4.2582E-03	3.6814E-03	3.0908E-03	2.4878E-03
1.8749E-03	1.2543E-03	6.2853E-04	2.6536E-08	-5.2742E-04	-1.2525E-03	-1.8728E-03	-2.4859E-03
-3.0895E-03	-3.6811E-03	-4.2575E-03	-4.8163E-03	-5.3574E-03	-5.8769E-03	-6.3735E-03	-6.8452E-03
-7.2895E-03	-7.7047E-03	-8.0895E-03	-8.4426E-03	-8.7625E-03	-9.0480E-03	-9.2977E-03	-9.5104E-03
-9.6855E-03	-9.8225E-03	-9.9208E-03	-9.9801E-03	-1.0000E-02	-9.9803E-03	-9.9212E-03	-9.8229E-03
-9.6859E-03	-9.5106E-03	-9.2978E-03	-9.0484E-03	-8.7634E-03	-8.4437E-03	-8.0906E-03	-7.7055E-03
-7.2899E-03	-6.8457E-03	-6.3748E-03	-5.8786E-03	-5.3591E-03	-4.8182E-03	-4.2591E-03	-3.6833E-03
-3.1009E-03	-2.5057E-03	-1.8980E-03	-1.2780E-03	-6.4553E-04	0.	1.4661E+79	2.9322E+79

THERE ARE 103 CALCULATED DATA POINTS COMPUTED ACCORDING TO THE FOLLOWING KEYS

KEY-1- , -2-, -3-, -4-, -5-, -6-
 1 0 0 1 2 1

YOU HAVE INPUT THE FOLLOWING SCALE FACTORS AT THE RESPECTIVE SUPPORTS

1.0000E+00

YOU HAVE INPUT THE FOLLOWING INITIAL SUPPORT DISPLACEMENT AND VELOCITY (USING ABOVE SCALE FACTORS)
 0.00000 .06283

TIME
 0. DISPLACEMENTS
 0. VELOCITIES

	C.	D.
0. ACCELERATIONS	1.2506E+00	-7.1054E-15
TIME		
1.00E-02 DISPLACEMENTS	6.3790E-05	4.5263E-06
1.00E-02 VELOCITIES	1.2787E-02	1.3862E-03
1.00E-02 ACCELERATIONS	1.2726E+00	2.8729E-01
TIME		
2.00E-02 DISPLACEMENTS	2.5341E-04	3.7960E-05
2.00E-02 VELOCITIES	2.4924E-02	5.8192E-03
2.00E-02 ACCELERATIONS	1.1432E+00	5.9754E-01
TIME		
3.00E-02 DISPLACEMENTS	5.5695E-04	1.3083E-04
3.00E-02 VELOCITIES	3.5466E-02	1.3212E-02
3.00E-02 ACCELERATIONS	9.6393E-01	8.7110E-01
TIME		
4.00E-02 DISPLACEMENTS	9.5663E-04	3.1018E-04
4.00E-02 VELOCITIES	4.4201E-02	2.2992E-02
4.00E-02 ACCELERATIONS	7.8900E-01	1.0706E+00
TIME		
5.00E-02 DISPLACEMENTS	1.4355E-03	5.9583E-04
5.00E-02 VELOCITIES	5.1328E-02	3.4321E-02
5.00E-02 ACCELERATIONS	6.4534E-01	1.1798E+00
TIME		
6.00E-02 DISPLACEMENTS	1.9790E-03	9.9870E-04
6.00E-02 VELOCITIES		

	5.7197E-02	4.6234E-02
6.00E-02 ACCELERATIONS	5.3725E-01	1.1985E+00

TIME		
7.00E-02 DISPLACEMENTS	2.5758E-03	1.5207E-03
7.00E-02 VELOCITIES	6.2014E-02	5.8014E-02
7.00E-02 ACCELERATIONS	4.3850E-01	1.1350E+00

TIME		
8.00E-02 DISPLACEMENTS	3.2166E-03	2.1557E-03
8.00E-02 VELOCITIES	6.6041E-02	6.8767E-02
8.00E-02 ACCELERATIONS	3.6838E-01	1.0064E+00

TIME		
9.00E-02 DISPLACEMENTS	3.8943E-03	2.8909E-03
9.00E-02 VELOCITIES	6.9392E-02	7.7988E-02
9.00E-02 ACCELERATIONS	3.0179E-01	8.3149E-01

TIME		
1.00E-01 DISPLACEMENTS	4.6020E-03	3.7091E-03
1.00E-01 VELOCITIES	7.2015E-02	8.5299E-02
1.00E-01 ACCELERATIONS	2.2320E-01	6.2726E-01

TIME		
1.10E-01 DISPLACEMENTS	5.3317E-03	4.5898E-03
1.10E-01 VELOCITIES	7.3775E-02	9.0482E-02
1.10E-01 ACCELERATIONS	1.2876E-01	4.0780E-01

TIME		
1.20E-01 DISPLACEMENTS	6.0741E-03	5.5113E-03
1.20E-01 VELOCITIES		

1.20E-01 ACCELERATIONS 7.4508E-02 9.3441E-02
1.8045E-02 1.8411E-01

TIME
1.30E-01 DISPLACEMENTS 6.8178E-03 6.4512E-03
1.30E-01 VELOCITIES 7.4034E-02 9.4174E-02
1.30E-01 ACCELERATIONS -1.1006E-01 -3.6200E-02

TIME
1.40E-01 DISPLACEMENTS 7.5503E-03 7.3876E-03
1.40E-01 VELOCITIES 7.2254E-02 9.2747E-02
1.40E-01 ACCELERATIONS -2.4347E-01 -2.4710E-01

TIME
1.50E-01 DISPLACEMENTS 8.2585E-03 8.2993E-03
1.50E-01 VELOCITIES 6.9177E-02 8.9283E-02
1.50E-01 ACCELERATIONS -3.7031E-01 -4.4290E-01

TIME
1.60E-01 DISPLACEMENTS 8.9297E-03 9.1670E-03
1.60E-01 VELOCITIES 6.4854E-02 8.3954E-02
1.60E-01 ACCELERATIONS -4.9060E-01 -6.1929E-01

TIME
1.70E-01 DISPLACEMENTS 9.5518E-03 9.9729E-03
1.70E-01 VELOCITIES 5.9390E-02 7.6973E-02
1.70E-01 ACCELERATIONS -5.9774E-01 -7.7275E-01

OUTPUT CONTINUES FOR THIS LOAD CASE

	:	:
	6.4291E-02	6.6350E-02
9.60E-01 ACCELERATIONS	9.6630E-02	1.0518E-01

TIME		
9.70E-01 DISPLACEMENTS	-2.1424E-03	-2.2679E-03
9.70E-01 VELOCITIES	6.5193E-02	6.7292E-02
9.70E-01 ACCELERATIONS	8.4845E-02	8.3499E-02

TIME		
9.80E-01 DISPLACEMENTS	-1.4864E-03	-1.5911E-03
9.80E-01 VELOCITIES	6.5997E-02	6.8027E-02
9.80E-01 ACCELERATIONS	7.6592E-02	6.4128E-02

TIME		
9.90E-01 DISPLACEMENTS	-8.2269E-04	-9.0794E-04
9.90E-01 VELOCITIES	6.6737E-02	5.8586E-02
9.90E-01 ACCELERATIONS	7.1635E-02	4.8275E-02

TIME		
1.00E+00 DISPLACEMENTS	-1.5180E-04	-2.1988E-04
1.00E+00 VELOCITIES	6.7436E-02	6.9007E-02
1.00E+00 ACCELERATIONS	6.8398E-02	3.6661E-02

LOAD CASE 2

DISPLACEMENTS.

ALL

OUTPUT IS ALSO ON TAPE FOR LOCAL CASE

VELOCITIES.

ALL

OUTPUT IS ALSO ON TAPE FOR LOCAL CASE

ACCELERATIONS.

ALL

OUTPUT IS ALSO ON TAPE FOR LOCAL CASE

BASE MOTION AT A FREQUENCY OF 6.28318 RAD/SEC WAS INPUT WITH THE FOLLOWING AMPLITUDES AT THE NODE POINTS

INPUT G LEVEL OF .00102 G UNITS AT NODE 3
...STEADY STATE ENVELOPES...

	GENERALIZED MASS	MAX DISPLACEMENT	MAX VELOCITY	MAX ACCELERATION
1		1.0842E-02	6.8124E-02	4.2804E-01
2		1.1281E-02	7.0878E-02	4.4534E-01

TIME	DISPLACEMENTS	VELOCITIES	ACCELERATIONS
0.	0.	0.	
0.			
0.			
1.00E-02	1.1946E-16	-7.3830E-17	
1.00E-02	1.3603E+00	1.0995E-01	
1.00E-02	6.8778E-05	9.9775E-06	
1.00E-02	1.3745E-02	2.4696E-03	
1.00E-02	1.3556E+00	3.9305E-01	
2.00E-02	2.7193E-04	5.9401E-05	
2.00E-02	2.6637E-02	7.9184E-03	
2.00E-02	1.2074E+00	6.9402E-01	
3.00E-02	5.9540E-04	1.7787E-04	
3.00E-02	3.7721E-02	1.6207E-02	
3.00E-02	1.0065E+00	9.5290E-01	
4.00E-02	1.0196E-03	3.9095E-04	
4.00E-02	4.6789E-02	2.6711E-02	
4.00E-02	8.1115E-01	1.1328E+00	
5.00E-02	1.5252E-03	7.1653E-04	
5.00E-02	5.4058E-02	3.8548E-02	
5.00E-02	6.4910E-01	1.2186E+00	

6.00E-02 DISPLACEMENTS		
6.00E-02 VELOCITIES	2.0960E-03	1.1632E-03
6.00E-02 ACCELERATIONS	5.9900E-02	5.0773E-02
7.00E-02 DISPLACEMENTS	5.2523E-01	1.2118E+00
7.00E-02 VELOCITIES	2.7196E-03	1.7304E-03
7.00E-02 ACCELERATIONS	6.4660E-02	6.2517E-02
8.00E-02 DISPLACEMENTS	4.3051E-01	1.1249E+00
8.00E-02 VELOCITIES	3.3863E-03	2.4095E-03
8.00E-02 ACCELERATIONS	6.8559E-02	7.3063E-02
9.00E-02 DISPLACEMENTS	3.5034E-01	9.7557E-01
9.00E-02 VELOCITIES	4.0881E-03	3.1859E-03
9.00E-02 ACCELERATIONS	7.1668E-02	8.1982E-02
1.00E-01 DISPLACEMENTS	2.7064E-01	7.8247E-01
1.00E-01 VELOCITIES	4.8169E-03	4.0402E-03
1.00E-01 ACCELERATIONS	7.3939E-02	8.8624E-02
1.10E-01 DISPLACEMENTS	1.8139E-01	5.6277E-01
1.10E-01 VELOCITIES	5.5637E-03	4.9508E-03
1.10E-01 ACCELERATIONS	7.5248E-02	9.3097E-02
1.20E-01 DISPLACEMENTS	7.7903E-02	3.3074E-01
1.20E-01 VELOCITIES	6.3182E-03	5.8944E-03
1.20E-01 ACCELERATIONS	7.5450E-02	9.5235E-02
1.30E-01 DISPLACEMENTS	-3.9498E-02	9.7572E-02
1.30E-01 VELOCITIES	7.0686E-03	6.8478E-03
1.30E-01 ACCELERATIONS	7.4425E-02	9.5073E-02
	-1.5677E-01	-1.2826E-01

1.40E-01 DISPLACEMENTS		
1.40E-01 VELOCITIES	7.8023E-03	7.7885E-03
1.40E-01 ACCELERATIONS	7.2103E-02	9.2716E-02
1.50E-01 DISPLACEMENTS	-2.9770E-01	-3.4038E-01
1.50E-01 VELOCITIES	8.5063E-03	8.6953E-03
1.50E-01 ACCELERATIONS	6.8481E-02	8.8327E-02
1.60E-01 DISPLACEMENTS	-4.2545E-01	-5.3394E-01
1.60E-01 VELOCITIES	9.1679E-03	9.5490E-03
1.60E-01 ACCELERATIONS	6.3626E-02	8.2112E-02
1.70E-01 DISPLACEMENTS	-5.4361E-01	-7.0511E-01
1.70E-01 VELOCITIES	9.7751E-03	1.0332E-02
1.70E-01 ACCELERATIONS	5.7659E-02	7.4310E-02
1.80E-01 DISPLACEMENTS	-6.4698E-01	-8.5090E-01
1.80E-01 VELOCITIES	1.0318E-02	1.1031E-02
1.80E-01 ACCELERATIONS	5.0748E-02	6.5187E-02
1.90E-01 DISPLACEMENTS	-7.3185E-01	-9.6897E-01
1.90E-01 VELOCITIES	1.0788E-02	1.1633E-02
1.90E-01 ACCELERATIONS	4.3091E-02	5.5028E-02
2.00E-01 DISPLACEMENTS	-7.9608E-01	-1.0578E+00
2.00E-01 VELOCITIES	1.1178E-02	1.2129E-02
2.00E-01 ACCELERATIONS	3.4898E-02	4.4132E-02
2.10E-01 DISPLACEMENTS	-8.3887E-01	-1.1165E+00
2.10E-01 VELOCITIES	1.1484E-02	1.2514E-02
2.10E-01 ACCELERATIONS	2.6383E-02	3.2798E-02
	-8.6060E-01	-1.1453E+00

OUTPUT CONTINUES FOR THIS LOAD
B-10 CASE UNTIL $t = 1.00 \text{ SEC}$

Appendix C - Selected Printout of Example Problem 2

CHECK PROBLEM** NONDIAGONAL MASS MATRIX AND ELLIPTIC LOADING **
YOU HAVE INPUT THE FOLLOWING NUMBERS OR INDICES

D.O.F.,	SPRINGS,	CASES ,	ROTATION,	SUPPORTS,	TAPES,	DAMPING,	OPTION
3	0	2	0	0	-2	-1	4

ROWS OF THE NONDIAGONAL MASS MATRIX

NUMBER OF NON-ZERO SYMMETRIC ENTRIES IS 4 WITH THE FIRST TERM NUMBERED AS 1 AND DESIGNATED NODE NUMBER

1	2	5	9
4.0000E+00	-2.0000E+00	6.0000E+00	4.0000E+00

NONDIAGONAL MASS MATRIX ...OUTPUT BY ROWS

4.0000E+00	-2.0000E+00	0.
-2.0000E+00	6.0000E+00	0.
0.	0.	4.0000E+00

PCT. OF CRITICAL DAMPING IN MODE (MODAL DAMPING

.0200	.0200	.0200
-------	-------	-------

NUMBER OF NON-ZERO SYMMETRIC ENTRIES IS 5 WITH THE FIRST TERM NUMBERED AS 1 AND DESIGNATED NODE NUMBER

1	2	3	5	9
4.0000E+01	-1.6000E+01	-1.6000E+01	1.6000E+01	1.6000E+01

STIFFNESS MATRIX ...OUTPUT BY ROWS...

4.0000E+01	-1.6000E+01	-1.6000E+01
-1.6000E+01	1.6000E+01	0.
-1.6000E+01	0.	1.6000E+01

UNDAMPED NATURAL FREQUENCIES (CYC./SEC.)
1.3140E-01 2.7865E-01 5.5711E-01

DAMPED NATURAL FREQUENCIES (CYC./SEC.)
1.3137E-01 2.7859E-01 5.5700E-01
MODAL MASSES..7**T*M*Z
1.0000E+00 1.0000E+00 1.0000E+00
IN MODE DAMPING CONSTANTS
3.3023E-02 7.0032E-02 1.4002E-01
EIGENVECTORS EIGHT COMPONENTS PER LINE

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 8.2542E-01 RAD./SEC.
2.6767E-01 3.2894E-01 3.2265E-01

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 1.7504E+00 RAD./SEC.
-7.1440E-02 2.9482E-01 -3.0571E-01

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 3.4997E+00 RAD./SEC.
4.7249E-01 6.9876E-02 -2.2900E-01

***NOTE** THE FOLLOWING ANALYSIS ASSUMES THAT
(1) THE MAIN DIAGONAL TERMS OF MATRIX CY ABOVE ARE MUCH GREATER THAN THE OFF DIAGONAL TERMS
(2) THE UNDAMPED FREQUENCIES ARE SEPARATED WIDELY
(3) PROPORTIONAL DAMPING, NO IN MODE PHASE CHANGES OF THE EIGENVECTORS DUE TO DAMPING

LOAD CASE 1
INPUT DESCRIBED BY DATA POINTS, FORCE INPUT

DISPLACEMENTS.

ALL

OUTPUT IS ALSO ON TAPE FOR LOCAL CASE

ACCELERATIONS.

1 3

OUTPUT IS ALSO ON TAPE FOR LOCAL CASE

THERE ARE 81 DATA POINTS ACTING SEQUENTIALLY AT THE FOLLOWING RELATIVE TIME POINTS

0.	5.0000E-02	1.0000E-01	1.5000E-01	2.0000E-01	2.5000E-01	3.0000E-01	3.5000E-01
4.0000E-01	4.5000E-01	5.0000E-01	5.5000E-01	6.0000E-01	6.5000E-01	7.0000E-01	7.5000E-01
8.0000E-01	8.5000E-01	9.0000E-01	9.5000E-01	1.0000E+00	1.0500E+00	1.1000E+00	1.1500E+00
1.2000E+00	1.2500E+00	1.3000E+00	1.3500E+00	1.4000E+00	1.4500E+00	1.5000E+00	1.5500E+00
1.6000E+00	1.6500E+00	1.7000E+00	1.7500E+00	1.8000E+00	1.8500E+00	1.9000E+00	1.9500E+00
2.0000E+00	2.0500E+00	2.1000E+00	2.1500E+00	2.2000E+00	2.2500E+00	2.3000E+00	2.3500E+00
2.4000E+00	2.4500E+00	2.5000E+00	2.5500E+00	2.6000E+00	2.6500E+00	2.7000E+00	2.7500E+00

2.8000E+00	2.8500E+00	2.9000E+00	2.9500E+00	3.0000E+00	3.0500E+00	3.1000E+00	3.1500E+00
3.2000E+00	3.2500E+00	3.3000E+00	3.3500E+00	3.4000E+00	3.4500E+00	3.5000E+00	3.5500E+00
3.6000E+00	3.6500E+00	3.7000E+00	3.7500E+00	3.8000E+00	3.8500E+00	3.9000E+00	3.9500E+00
4.0000E+00							

THE SEQUENTIAL VALUES OF THE DATA POINTS ARE

2.0000E+00	2.5681E+00	3.2974E+00	4.2340E+00	5.4366E+00	6.9807E+00	8.9634E+00	1.1509E+01
1.4778E+01	1.8975E+01	2.4365E+01	1.4778E+01	8.9632E+00	5.4364E+00	3.2974E+00	2.0000E+00
1.2130E+00	7.3574E-01	4.4625E-01	2.7066E-01	1.6417E-01	9.9572E-02	6.0393E-02	3.6630E-02
2.2217E-02	1.3476E-02	8.1734E-03	4.9574E-03	3.0068E-03	1.8237E-03	1.1061E-03	6.7091E-04
4.0693E-04	2.4681E-04	1.4970E-04	9.0798E-05	5.5072E-05	3.3403E-05	2.0260E-05	1.2288E-05
7.4531E-06	4.5206E-06	2.7419E-06	1.6630E-06	1.0087E-06	6.1179E-07	3.7107E-07	2.2507E-07
1.3651E-07	8.2797E-08	5.0219E-08	3.0459E-08	1.8474E-08	1.1205E-08	6.7964E-09	4.1222E-09
2.5002E-09	1.5165E-09	9.1979E-10	5.5788E-10	3.3837E-10	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.

1.2500E-02 THERE ARE 2 DISTINCT OUTPUT REGIONS WITH RESPECTIVE OUTPUT SPACINGS EQUAL TO 2.5000E-02

2.0000E+00 THE 1 CHANGES IN OUTPUT SPACINGS OCCUR AT THE FOLLOWING TIME POINTS

5.0000E-01 THERE ARE 1 SMOOTHNESS DISCONTINUITIES OF THE INPUT DATA LOCATED AT TIME POINTS

CALCULATED DATA POINTS + 2 END POINTS USED AS LOADING

1.8731E+00	2.0000E+00	2.1269E+00	2.2639E+00	2.4110E+00	2.5681E+00	2.7343E+00	2.9108E+00
3.0982E+00	3.2974E+00	3.5100E+00	3.7364E+00	3.9774E+00	4.2340E+00	4.5071E+00	4.7977E+00
5.1072E+00	5.4366E+00	5.7872E+00	6.1604E+00	6.5577E+00	6.9807E+00	7.4309E+00	7.9101E+00
8.4203E+00	8.9634E+00	9.5414E+00	1.0157E+01	1.0812E+01	1.1509E+01	1.2251E+01	1.3041E+01
1.3882E+01	1.4778E+01	1.5731E+01	1.6745E+01	1.7825E+01	1.8975E+01	2.0206E+01	2.1515E+01
2.2901E+01	2.4365E+01	2.1615E+01	1.9100E+01	1.6821E+01	1.4778E+01	1.3003E+01	1.1457E+01
1.0119E+01	8.9632E+00	7.9121E+00	6.9838E+00	6.1628E+00	5.4364E+00	4.7989E+00	4.2359E+00
3.7380E+00	3.2974E+00	2.9108E+00	2.5692E+00	2.2672E+00	2.0000E+00	1.7655E+00	1.5583E+00
1.3751E+00	1.2130E+00	1.0708E+00	9.4512E-01	8.3403E-01	7.3574E-01	6.4947E-01	5.7327E-01
5.0588E-01	4.4625E-01	3.9392E-01	3.4770E-01	3.0683E-01	2.7066E-01	2.3892E-01	2.1089E-01
1.8610E-01	1.6417E-01	1.4492E-01	1.2792E-01	1.1288E-01	9.9572E-02	8.7896E-02	7.7582E-02
6.8462E-02	6.0393E-02	5.3311E-02	4.7056E-02	4.1525E-02	3.6630E-02	3.2335E-02	2.8541E-02
2.5186E-02	2.2217E-02	1.9612E-02	1.7311E-02	1.5276E-02	1.3476E-02	1.1896E-02	1.0500E-02
9.2657E-03	8.1734E-03	7.2149E-03	6.3683E-03	5.6198E-03	4.9574E-03	4.3761E-03	3.8626E-03
3.4086E-03	3.0068E-03	2.6542E-03	2.3428E-03	2.0674E-03	1.8237E-03	1.6098E-03	1.4209E-03
1.2539E-03	1.1061E-03	9.7640E-04	8.6184E-04	7.6054E-04	6.7091E-04	5.9224E-04	5.2276E-04
4.6131E-04	4.0693E-04	3.5921E-04	3.1706E-04	2.7979E-04	2.4681E-04	2.1787E-04	1.9231E-04
1.6970E-04	1.4970E-04	1.3215E-04	1.1664E-04	1.0293E-04	9.0798E-05	8.0151E-05	7.0747E-05
6.2430E-05	5.5072E-05	4.8614E-05	4.2910E-05	3.7866E-05	3.3403E-05	2.9486E-05	2.6026E-05
2.2967E-05	2.0260E-05	1.7884E-05	1.5786E-05	1.3930E-05	1.2288E-05	1.0847E-05	9.5743E-06

8.4489E-06	7.4531E-06	5.8072E-06	4.5206E-06	3.5223E-06	2.7419E-06	2.1364E-06	1.6630E-06
1.2958E-06	1.0087E-06	7.8594E-07	6.1179E-07	4.7668E-07	3.7107E-07	2.8913E-07	2.2507E-07
1.7537E-07	1.3651E-07	1.0636E-07	8.2797E-08	6.4512E-08	5.0219E-08	3.9129E-08	3.0459E-08
2.3732E-08	1.8474E-08	1.4394E-08	1.1205E-08	8.7306E-09	6.7964E-09	5.2955E-09	4.1222E-09
3.2118E-09	2.5002E-09	1.9481E-09	1.5165E-09	1.1816E-09	9.1979E-10	7.1666E-10	5.5788E-10
4.4270E-10	3.3837E-10	1.7231E-10	0.	-3.9795E-11	0.	7.9305E-12	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.

THERE ARE 243 CALCULATED DATA POINTS COMPUTED ACCORDING TO THE FOLLOWING KEYS

KEY-1- , -2-, -3-, -4-, -5-, -6-
1 0 2 1 2 1

TIME			
0.	DISPLACEMENTS	0.	0.
0.	ACCELERATIONS	6.0000E-01	-2.6645E-15
1.25E-02	DISPLACEMENTS	4.7838E-05	1.5951E-05
1.25E-02	ACCELERATIONS	6.3675E-01	4.6865E-04
2.50E-02	DISPLACEMENTS	1.9522E-04	6.5118E-05
2.50E-02	ACCELERATIONS	6.7554E-01	1.3514E-03
3.75E-02	DISPLACEMENTS	4.4817E-04	1.4958E-04
3.75E-02	ACCELERATIONS	7.1632E-01	2.6726E-03
5.00E-02	DISPLACEMENTS	8.1308E-04	2.7155E-04
5.00E-02	ACCELERATIONS	7.5902E-01	4.4574E-03

TIME
6.25E-02 DISPLACEMENTS

6.25E-02 ACCELERATIONS 1.2966E-03 4.3338E-04 2.5631E-06
8.0327E-01 6.7316E-03

TIME
7.50E-02 DISPLACEMENTS
7.50E-02 ACCELERATIONS 1.9057E-03 6.3754E-04 5.0532E-06
8.4939E-01 9.5220E-03

TIME
8.75E-02 DISPLACEMENTS
8.75E-02 ACCELERATIONS 2.6475E-03 8.8663E-04 9.0382E-06
8.9749E-01 1.2856E-02

TIME
1.00E-01 DISPLACEMENTS
1.00E-01 ACCELERATIONS 3.5295E-03 1.1834E-03 1.5039E-05
9.4775E-01 1.6762E-02

TIME
1.12E-01 DISPLACEMENTS
1.12E-01 ACCELERATIONS 4.5597E-03 1.5307E-03 2.3667E-05
1.0006E+00 2.1270E-02

TIME
1.25E-01 DISPLACEMENTS
1.25E-01 ACCELERATIONS 5.7463E-03 1.9317E-03 3.5627E-05
1.0561E+00 2.6410E-02

TIME
1.37E-01 DISPLACEMENTS
1.37E-01 ACCELERATIONS 7.0979E-03 2.3895E-03 5.1722E-05
1.1144E+00 3.2215E-02

TIME
1.50E-01 DISPLACEMENTS
1.50E-01 ACCELERATIONS 8.6237E-03 2.9077E-03 7.2860E-05
1.1756E+00 3.8719E-02

C-6

PRINTOUT CONTINUES IN THIS FASHION

Appendix D - Selected Printout of Example Problem 3 (w/o Tape Output)

EXAMPLE PROBLEM THREE - DAMPED HARMONIC LOADS
YOU HAVE INPUT THE FOLLOWING NUMBERS OR INDICES

D.O.F.,	SPRINGS,	CASES ,	ROTATION,	SUPPORTS,	TAPES,	DAMPING,	OPTION
7	7	1	6	0	-0	-1	0

NUMBER OF MASSES = 7 NUMBER OF LINEAR SPRINGS= 7 NUMBER OF LOAD CONDITIONS= 1 NUMBER OF ROTATIONAL VARIABLES

NUMBER OF INTERNAL CLOSED LOOPS= 0

CASES = 6

7 SPRING CONSTANTS LISTED IN ORDER 10 TO A LINE

1.3000E+06 2.6000E+06 4.2000E+06 5.9000E+06 4.0000E+02 8.2000E+06 1.0000E+07

7 ROTATIONAL SPRING CONSTANTS LISTED IN ORDER 6 TO A LINE

1.7000E+07 3.5000E+07 5.6000E+07 7.9000E+07 0. 1.1000E+08 1.3000E+08

PCT. OF CRITICAL DAMPING IN MODE (MODAL DAMPING

.0400	.0400	.0400	.0400	.0400	.0300	.0300	.0300	.0200
.0200	.0100	.0100						

IDENTIFY TOPOLOGY

MASS	INERTIA	1 PT.	2 PT.	3 PT.	OFFSET	OFFSET	MASS
7.00000E+02	6.60000E+05	0	2	-0-0.		-0	
7.00000E+02	6.60000E+05	1	3	-0-0.		-0	
5.00000E+02	3.30000E+05	2	4	-0-0.		-0	
5.00000E+02	3.30000E+05	3	5	6	4.00000E+01	5	
1.00000E+01	-0.	4	-0	-0-0.		-0	
4.00000E+02	2.70000E+05	4	7	-0-0.		-0	
4.00000E+02	2.70000E+05	6	-0	-0-0.		-0	

THE COMPUTER INTERPRETS AS 13 DEGREES OF FREEDOM

STIFFNESS MATRIX

3.9000E+06	-2.6000E+06	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.
-2.6000E+06	6.8000E+06	-4.2000E+06	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.
0.	-4.2000E+06	1.0100E+07	-5.9000E+06	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.
0.	0.	-5.9000E+06	1.4100E+07	-4.0000E+02	-8.2000E+06	0.
0.	0.	0.	-1.6000E+04	0.	0.	0.

0.	0.	0.	-4.0000E+02	4.0000E+02	0.	0.
0.	0.	0.	0.	0.	0.	
0.	0.	0.	-8.2000E+06	0.	1.8200E+07	-1.0000E+07
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	-1.0000E+07	1.0000E+07
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.
5.2000E+07	-3.5000E+07	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.
-3.5000E+07	9.1000E+07	-5.6000E+07	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.
0.	-5.6000E+07	1.3500E+08	-7.9000E+07	0.	0.	
0.	0.	0.	-1.6000E+04	0.	0.	0.
0.	0.	-7.9000E+07	1.8900E+08	-1.1000E+08	0.	
0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	-1.1000E+08	2.4000E+08	-1.3000E+08	
0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	-1.3000E+08	1.3000E+08	

UNDAMPED NATURAL FREQUENCIES (CYC./SEC.)

3.4770E-01	1.0062E+00	1.1578E+00	1.8739E+00	2.6172E+00	2.9068E+00	4.1859E+00	5.8417E+00
9.2106E+00	1.5055E+01	2.1410E+01	2.9686E+01	4.1725E+01			

DAMPED NATURAL FREQUENCIES (CYC./SEC.)

3.4743E-01	1.0054E+00	1.1568E+00	1.8724E+00	2.6151E+00	2.9055E+00	4.1840E+00	5.8391E+00
9.2088E+00	1.5052E+01	2.1406E+01	2.9685E+01	4.1723E+01			

MODAL MASSES..Z**T**Z

1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00
1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00			

IN MODE DAMPING CONSTANTS

1.7477E-01	5.0578E-01	5.8195E-01	9.4194E-01	1.3155E+00	1.0958E+00	1.5781E+00	2.2023E+00
2.3149E+00	3.7836E+00	5.3810E+00	3.7305E+00	5.2433E+00			

EIGENVECTORS EIGHT COMPONENTS PER LINE

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 2.1829E+00 RAD./SEC.

9.0931E-06	1.3613E-05	1.6406E-05	1.8388E-05	2.0880E-05	1.8397E-05	1.8400E-05	4.3776E-04
6.1099E-04	6.8489E-04	7.2362E-04	7.4107E-04	7.4849E-04			

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 6.3172E+00 RAD./SEC.

1.1582E-04	1.7248E-04	2.0641E-04	2.2986E-04	3.1621E-01	2.3076E-04	2.3113E-04	-1.5263E-07
-1.1172E-07	-3.3522E-08	2.7505E-08	3.4602E-08	3.7734E-08			

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 7.2685E+00 RAD./SEC.

7.5688E-06	1.1245E-05	1.3422E-05	1.4912E-05	-4.6179E-05	1.4989E-05	1.5021E-05	-8.8613E-04
-4.3231E-04	1.2094E-04	4.8638E-04	6.7162E-04	7.5455E-04			

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 1.1765E+01 RAD./SEC.

2.0308E-06	2.9704E-06	3.4834E-06	3.8076E-06	-1.5437E-06	3.8600E-06	3.8815E-06	7.2591E-04
-8.1921E-04	-4.4640E-04	7.6383E-05	4.2007E-04	5.8993E-04			

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 1.6431E+01 RAD./SEC.

1.1537E-02	1.6465E-02	1.8774E-02	1.9987E-02	-3.4699E-03	2.0532E-02	2.0756E-02	4.8463E-07
-1.7512E-06	2.4324E-06	2.6504E-06	-2.2503E-06	-5.1333E-06			

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 1.8256E+01 RAD./SEC.

4.2434E-05	5.9840E-05	6.7288E-05	7.0688E-05	-9.6315E-06	7.3083E-05	7.4071E-05	-1.0564E-04
5.0753E-04	-1.1045E-03	-7.0821E-04	2.8511E-04	9.2809E-04			

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 2.6289E+01 RAD./SEC.

-1.1721E-05	-1.5399E-05	-1.5900E-05	-1.5325E-05	9.4058E-07	-1.6451E-05	-1.6919E-05	-1.3975E-05
1.6153E-04	-1.0457E-03	1.1201E-03	3.5109E-04	-8.0400E-04			

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 3.6688E+01 RAD./SEC.

4.0672E-06	4.6255E-06	3.9325E-06	2.9903E-06	-9.1501E-08	3.4577E-06	3.6546E-06	5.4142E-07
-1.2950E-05	1.8424E-04	-7.1283E-04	1.5240E-03	-8.4754E-04			

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 5.7861E+01 RAD./SEC.
-2.6782E-02 -1.6024E-02 -4.1936E-04 1.0809E-02 -1.3064E-04 1.6680E-02 1.9260E-02 8.5652E-12
-4.1728E-10 1.5827E-08 -1.9409E-07 3.3232E-08 -5.5816E-09

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 9.4572E+01 RAD./SEC.
2.3269E-02 -2.1149E-02 -1.7108E-02 -1.2585E-03 5.6517E-06 1.0832E-02 1.6869E-02 -1.1883E-14
1.9737E-12 -2.0491E-10 7.3069E-09 -3.7067E-10 2.1080E-11

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 1.3450E+02 RAD./SEC.
-6.0014E-03 2.0239E-02 -2.4561E-02 -1.8784E-02 4.1611E-05 6.0990E-03 2.2088E-02 -5.9008E-15
3.3269E-12 -7.0386E-10 5.2004E-08 -1.2321E-09 3.3674E-11

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 1.8652E+02 RAD./SEC.
8.3921E-04 -6.6020E-03 2.7074E-02 -2.8778E-02 3.3125E-05 -7.9147E-03 2.0208E-02 -1.1381E-14
6.9309E-13 -2.8397E-10 4.0783E-08 -4.9021E-10 6.8579E-12

...MODE SHAPE...CORRESPONDING TO NATURAL FREQUENCY 2.6215E+02 RAD./SEC.
2.5770E-05 -4.3820E-04 4.2943E-03 -1.7349E-02 1.0103E-05 3.9787E-02 -2.2745E-02 -5.0746E-15
5.0841E-14 -4.3244E-11 1.2342E-08 -7.4118E-11 5.2817E-13

NOTE THE FOLLOWING ANALYSIS ASSUMES THAT

- (1) THE MAIN DIAGONAL TERMS OF MATRIX CY ABOVE ARE MUCH GREATER THAN THE OFF DIAGONAL TERMS
- (2) THE UNDAMPED FREQUENCIES ARE SEPARATED WIDELY
- (3) PROPORTIONAL DAMPING, NO IN MODE PHASE CHANGES OF THE EIGENVECTORS DUE TO DAMPING

LOAD CASE 1

PRINTED OUTPUT

TAPE OUTPUT

DISPLACEMENTS.

ALL

..NONE..

VELOCITIES.

ALL

..NONE..

ACCELERATIONS.

ALL

..NONE..

HARMONIC LOAD 2000.00000*SIN(40.0000 *T) AT GENERALIZED MASS 3

HARMONIC STEADY STATE LOAD AT MASS 3

...STEADY STATE ENVELOPES...D-4

GENERALIZED MASS	MAX DISPLACEMENT	MAX VELOCITY	MAX ACCELERATION
1	4.0176E-04	1.6070E-02	6.4281E-01
2	4.2973E-04	1.7189E-02	6.8757E-01
3	3.3259E-04	1.3304E-02	5.3215E-01
4	5.5665E-04	2.2266E-02	8.9063E-01
5	1.4416E-05	5.7665E-04	2.3066E-02
6	6.6370E-04	2.6548E-02	1.0619E+00
7	7.0904E-04	2.8362E-02	1.1345E+00
8	8.8734E-11	3.5494E-09	1.4197E-07
9	8.4569E-10	3.3828E-08	1.3531E-06
10	7.5788E-09	3.0315E-07	1.2126E-05
11	3.6655E-08	1.4662E-06	5.8649E-05
12	2.8741E-08	1.1497E-06	4.5986E-05
13	1.3287E-08	5.3147E-07	2.1259E-05

TRANSIENT MOTION WITH LOAD AT GENERALIZED MASS 3

NOTEYOU ARE ASSUMING TRANSIENT MOTION IS NOT YET DAMPED OUT AND COMPLETE FORCE BUILDUP

DISPLACEMENTS.

5 7

ACCELERATIONS.

5 7

TIME	DISPLACEMENTS	ACCELERATIONS
0.	0.	0.
0.	-2.3170E-17	2.4853E-16
2.00E-02	4.2820E-08	1.6111E-06
2.00E-02	2.0551E-03	1.3417E-01
4.00E-02	1.3266E-06	1.2546E-04
4.00E-02	1.4543E-02	1.5081E+00
6.00E-02	8.5390E-06	6.9437E-04
6.00E-02	3.2534E-02	6.4423E-02
8.00E-02	2.9022E-05	1.3778E-03
8.00E-02	5.5205E-02	-2.7846E-01
1.00E-01	DISPLACEMENTS	

1.00E-01 ACCELERATIONS	7.1094E-05	1.8898E-03
1.20E-01 DISPLACEMENTS	6.5508E-02	-1.4894E+00
1.20E-01 ACCELERATIONS	1.3884E-04	1.8837E-03
1.40E-01 DISPLACEMENTS	5.9435E-02	-1.1972E+00
1.40E-01 ACCELERATIONS	2.2972E-04	1.4019E-03
1.60E-01 DISPLACEMENTS	3.4199E-02	-9.0683E-01
1.60E-01 ACCELERATIONS	3.3385E-04	5.3565E-04
1.80E-01 DISPLACEMENTS	-2.1785E-03	-3.7068E-01
1.80E-01 ACCELERATIONS	4.3733E-04	-4.0388E-04
2.00E-01 DISPLACEMENTS	-3.4110E-02	1.4302E+00
2.00E-01 ACCELERATIONS	5.2747E-04	-8.8870E-04
2.20E-01 DISPLACEMENTS	-5.5430E-02	9.1762E-01
2.20E-01 ACCELERATIONS	5.9607E-04	-9.6598E-04
2.40E-01 DISPLACEMENTS	-5.9361E-02	7.7513E-01
2.40E-01 ACCELERATIONS	6.4122E-04	-7.4866E-04
2.60E-01 DISPLACEMENTS	-5.4576E-02	1.0213E-01
2.60E-01 ACCELERATIONS	6.6452E-04	-5.0438E-04
2.80E-01 DISPLACEMENTS	-4.9515E-02	-6.3185E-01
2.80E-01 ACCELERATIONS	6.6781E-04	-4.7830E-04
3.00E-01 DISPLACEMENTS	-5.1186E-02	-7.9092E-01
3.00E-01 ACCELERATIONS	6.5033E-04	-7.7403E-04
3.20E-01 DISPLACEMENTS	-6.0355E-02	-4.0228E-01
3.20E-01 ACCELERATIONS	6.0879E-04	-1.1867E-03
3.40E-01 DISPLACEMENTS	-6.7980E-02	8.1028E-01

		5.4035E-04	-1.3163E-03
3.40E-01	ACCELERATIONS		
		-6.6309E-02	9.7131E-01
3.60E-01	DISPLACEMENTS		
		4.4596E-04	-1.0574E-03
3.60E-01	ACCELERATIONS		
		-4.7901E-02	1.0923E+00
3.80E-01	DISPLACEMENTS		
		3.3289E-04	-3.6635E-04
3.80E-01	ACCELERATIONS		
		-1.5975E-02	6.8049E-01
4.00E-01	DISPLACEMENTS		
		2.1353E-04	5.5108E-04
4.00E-01	ACCELERATIONS		
		1.9552E-02	-5.0134E-01
4.20E-01	DISPLACEMENTS		
		1.0172E-04	1.2888E-03
4.20E-01	ACCELERATIONS		
		4.6581E-02	-1.2929E+00
4.40E-01	DISPLACEMENTS		
		7.9539E-06	1.5359E-03
4.40E-01	ACCELERATIONS		
		5.6540E-02	-1.2497E+00
4.60E-01	DISPLACEMENTS		
		-6.3777E-05	1.2969E-03
4.60E-01	ACCELERATIONS		
		4.9745E-02	-7.0709E-01
4.80E-01	DISPLACEMENTS		
		-1.1590E-04	7.9377E-04
4.80E-01	ACCELERATIONS		
		3.4133E-02	1.9427E-02
5.00E-01	DISPLACEMENTS		
		-1.5429E-04	2.9257E-04
5.00E-01	ACCELERATIONS		
		2.0969E-02	8.3762E-01
5.20E-01	DISPLACEMENTS		
		-1.8396E-04	1.0716E-04
5.20E-01	ACCELERATIONS		
		1.7015E-02	9.3806E-01
5.40E-01	DISPLACEMENTS		
		-2.0652E-04	2.6425E-04
5.40E-01	ACCELERATIONS		
		2.1787E-02	2.5716E-01
5.60E-01	DISPLACEMENTS		
		-2.2036E-04	5.1259E-04
5.60E-01	ACCELERATIONS		
		2.7160E-02	-6.2826E-01
5.80E-01	DISPLACEMENTS		

		-2.2359E-04	5.4386E-04
5.80E-01	ACCELERATIONS		
		2.4400E-02	-8.9305E-01
6.00E-01	DISPLACEMENTS		
		-2.1750E-04	2.1513E-04
6.00E-01	ACCELERATIONS		
		8.7598E-03	-9.1520E-01
6.20E-01	DISPLACEMENTS		
		-2.0810E-04	-4.4546E-04
6.20E-01	ACCELERATIONS		
		-1.5737E-02	-2.3399E-01
6.40E-01	DISPLACEMENTS		
		-2.0511E-04	-1.1872E-03
6.40E-01	ACCELERATIONS		
		-3.8331E-02	9.1099E-01
6.60E-01	DISPLACEMENTS		
		-2.1696E-04	-1.5977E-03
6.60E-01	ACCELERATIONS		
		-4.8677E-02	1.2910E+00
6.80E-01	DISPLACEMENTS		
		-2.4763E-04	-1.4991E-03
6.80E-01	ACCELERATIONS		
		-4.0430E-02	1.2069E+00
7.00E-01	DISPLACEMENTS		
		-2.9400E-04	-9.5136E-04
7.00E-01	ACCELERATIONS		
		-1.8063E-02	4.2530E-01
7.20E-01	DISPLACEMENTS		
		-3.4744E-04	-2.2156E-04
7.20E-01	ACCELERATIONS		
		8.8408E-03	-3.6395E-01
7.40E-01	DISPLACEMENTS		
		-3.9759E-04	3.5951E-04
7.40E-01	ACCELERATIONS		
		2.8430E-02	-1.0514E+00
7.60E-01	DISPLACEMENTS		
		-4.3684E-04	5.4344E-04
7.60E-01	ACCELERATIONS		
		3.4653E-02	-1.0408E+00
7.80E-01	DISPLACEMENTS		
		-4.6257E-04	3.4725E-04
7.80E-01	ACCELERATIONS		
		3.0437E-02	-1.4013E-01
8.00E-01	DISPLACEMENTS		
		-4.7625E-04	7.0573E-05
8.00E-01	ACCELERATIONS		
		2.3255E-02	4.0063E-01
8.20E-01	DISPLACEMENTS		

		-4.8035E-04	-3.3726E-05
8.20E-01	ACCELERATIONS		
		2.3367E-02	9.3395E-01
8.40E-01	DISPLACEMENTS		
		-4.7478E-04	2.0388E-04
8.40E-01	ACCELERATIONS		
		3.3392E-02	7.3143E-01
8.60E-01	DISPLACEMENTS		
		-4.5564E-04	7.1597E-04
8.60E-01	ACCELERATIONS		
		4.9081E-02	-7.3539E-02
8.80E-01	DISPLACEMENTS		
		-4.1702E-04	1.1996E-03
8.80E-01	ACCELERATIONS		
		6.0411E-02	-8.9918E-01
9.00E-01	DISPLACEMENTS		
		-3.5473E-04	1.3289E-03
9.00E-01	ACCELERATIONS		
		5.7418E-02	-1.3529E+00
9.20E-01	DISPLACEMENTS		
		-2.7001E-04	9.5846E-04
9.20E-01	ACCELERATIONS		
		3.8179E-02	-8.5886E-01
9.40E-01	DISPLACEMENTS		
		-1.7042E-04	2.4178E-04
9.40E-01	ACCELERATIONS		
		7.7740E-03	-1.9478E-01
9.60E-01	DISPLACEMENTS		
		-6.7671E-05	-5.3515E-04
9.60E-01	ACCELERATIONS		
		-2.1299E-02	7.8787E-01
9.80E-01	DISPLACEMENTS		
		2.6967E-05	-1.0131E-03
9.80E-01	ACCELERATIONS		
		-3.8597E-02	1.3011E+00
1.00E+00	DISPLACEMENTS		
		1.0670E-04	-1.0071E-03
1.00E+00	ACCELERATIONS		
		-4.0235E-02	9.0268E-01
1.02E+00	DISPLACEMENTS		
		1.7074E-04	-6.4400E-04
1.02E+00	ACCELERATIONS		
		-3.0331E-02	2.1094E-01
1.04E+00	DISPLACEMENTS		
		2.2266E-04	-2.0826E-04
1.04E+00	ACCELERATIONS		
		-1.9521E-02	-6.2207E-01
1.06E+00	DISPLACEMENTS		

1.06E+00	ACCELRATIONS	2.6653E-04	2.9173E-06
1.08E+00	DISPLACEMENTS	-1.6223E-02	-9.0534E-01
1.08E+00	ACCELRATIONS	3.0354E-04	-1.3132E-04
1.10E+00	DISPLACEMENTS	-2.3412E-02	-6.5406E-01
1.10E+00	ACCELRATIONS	3.3101E-04	-5.0834E-04
1.12E+00	DISPLACEMENTS	-3.5399E-02	1.5887E-01
1.12E+00	ACCELRATIONS	3.4450E-04	-8.1836E-04
1.14E+00	DISPLACEMENTS	-4.2291E-02	9.7041E-01
1.14E+00	ACCELRATIONS	3.4151E-04	-7.7082E-04
1.16E+00	DISPLACEMENTS	-3.6493E-02	1.0726E+00
1.16E+00	ACCELRATIONS	3.2440E-04	-3.0599E-04
1.18E+00	DISPLACEMENTS	-1.6830E-02	6.9297E-01
1.18E+00	ACCELRATIONS	3.0076E-04	4.1619E-04
1.20E+00	DISPLACEMENTS	8.8019E-03	-1.9935E-01
1.20E+00	ACCELRATIONS	2.8045E-04	1.0598E-03
1.22E+00	DISPLACEMENTS	2.8938E-02	-1.0545E+00
1.22E+00	ACCELRATIONS	2.7122E-04	1.3054E-03
1.24E+00	DISPLACEMENTS	3.4345E-02	-1.3016E+00
1.24E+00	ACCELRATIONS	2.7517E-04	1.0510E-03
1.26E+00	DISPLACEMENTS	2.3492E-02	-8.4145E-01
1.26E+00	ACCELRATIONS	2.8820E-04	4.7780E-04
1.28E+00	DISPLACEMENTS	3.5728E-03	5.3102E-02
1.28E+00	ACCELRATIONS	3.0274E-04	-8.2625E-05
1.30E+00	DISPLACEMENTS	-1.4371E-02	7.7585E-01

		3.1190E-04	-3.4764E-04
1.30E+00	ACCELERATIONS		
		-2.1430E-02	1.0380E+00
1.32E+00	DISPLACEMENTS		
		3.1290E-04	-2.1954E-04
1.32E+00	ACCELERATIONS		
		-1.6467E-02	6.1012E-01
1.34E+00	DISPLACEMENTS		
		3.0750E-04	1.3444E-04
1.34E+00	ACCELERATIONS		
		-6.1562E-03	-2.6484E-01
1.36E+00	DISPLACEMENTS		
		2.9949E-04	3.9392E-04
1.36E+00	ACCELERATIONS		
		-3.4041E-04	-8.9378E-01
1.38E+00	DISPLACEMENTS		
		2.9094E-04	3.1289E-04
1.38E+00	ACCELERATIONS		
		-5.9556E-03	-9.6537E-01
1.40E+00	DISPLACEMENTS		
		2.7965E-04	-1.3368E-04
1.40E+00	ACCELERATIONS		
		-2.2106E-02	-4.3783E-01
1.42E+00	DISPLACEMENTS		
		2.5944E-04	-7.4135E-04
1.42E+00	ACCELERATIONS		
		-4.0405E-02	4.5610E-01
1.44E+00	DISPLACEMENTS		
		2.2336E-04	-1.1752E-03
1.44E+00	ACCELERATIONS		
		-4.9947E-02	1.1500E+00
1.46E+00	DISPLACEMENTS		
		1.6785E-04	-1.1683E-03
1.46E+00	ACCELERATIONS		
		-4.3605E-02	1.2171E+00
1.48E+00	DISPLACEMENTS		
		9.5395E-05	-7.0054E-04
1.48E+00	ACCELERATIONS		
		-2.2679E-02	5.8625E-01
1.50E+00	DISPLACEMENTS		
		1.4064E-05	-6.9927E-06
1.50E+00	ACCELERATIONS		
		3.8971E-03	-3.1666E-01
1.52E+00	DISPLACEMENTS		
		-6.5912E-05	5.6801E-04
1.52E+00	ACCELERATIONS		
		2.4516E-02	-9.9846E-01
1.54E+00	DISPLACEMENTS		

1.54E+00 ACCELERATIONS	-1.3656E-04	7.6080E-04
1.56E+00 DISPLACEMENTS	3.1437E-02	-1.1262E+00
1.56E+00 ACCELERATIONS	-1.9507E-04	5.3109E-04
1.58E+00 DISPLACEMENTS	2.5415E-02	-5.1844E-01
1.58E+00 ACCELERATIONS	-2.4359E-04	1.0093E-04
1.60E+00 DISPLACEMENTS	1.4354E-02	3.4256E-01
1.60E+00 ACCELERATIONS	-2.8618E-04	-2.0133E-04
1.62E+00 DISPLACEMENTS	8.5557E-03	9.2053E-01
1.62E+00 ACCELERATIONS	-3.2496E-04	-1.5564E-04
1.64E+00 DISPLACEMENTS	1.3999E-02	9.0327E-01
1.64E+00 ACCELERATIONS	-3.5783E-04	2.2836E-04
1.66E+00 DISPLACEMENTS	2.8453E-02	2.5682E-01
1.66E+00 ACCELERATIONS	-3.7931E-04	7.0999E-04
1.68E+00 DISPLACEMENTS	4.3108E-02	-5.9073E-01
1.68E+00 ACCELERATIONS	-3.8390E-04	9.6287E-04
1.70E+00 DISPLACEMENTS	4.7614E-02	-1.1676E+00
1.70E+00 ACCELERATIONS	-3.6995E-04	7.7494E-04
1.72E+00 DISPLACEMENTS	3.7031E-02	-1.0258E+00
1.72E+00 ACCELERATIONS	-3.4160E-04	1.9663E-04
1.74E+00 DISPLACEMENTS	1.4734E-02	-2.9308E-01
1.74E+00 ACCELERATIONS	-3.0741E-04	-4.9494E-04
1.76E+00 DISPLACEMENTS	-9.1324E-03	5.8514E-01
1.76E+00 ACCELERATIONS	-2.7654E-04	-9.6032E-04
1.78E+00 DISPLACEMENTS	-2.3309E-02	1.1874E+00

		-2.5447E-04	-9.7703E-04
1.78E+00	ACCELERATIONS		
		-2.2271E-02	1.0748E+00
1.80E+00	DISPLACEMENTS		
		-2.4088E-04	-5.8281E-04
1.80E+00	ACCELERATIONS		
		-8.8165E-03	3.7227E-01
1.82E+00	DISPLACEMENTS		
		-2.3072E-04	-4.5985E-05
1.82E+00	ACCELERATIONS		
		7.4170E-03	-5.0030E-01
1.84E+00	DISPLACEMENTS		
		-2.1786E-04	3.0305E-04
1.84E+00	ACCELERATIONS		
		1.6114E-02	-1.0077E+00
1.86E+00	DISPLACEMENTS		
		-1.9898E-04	2.7276E-04
1.86E+00	ACCELERATIONS		
		1.2463E-02	-8.4915E-01
1.88E+00	DISPLACEMENTS		
		-1.7542E-04	-8.0510E-05
1.88E+00	ACCELERATIONS		
		-1.8933E-04	-1.6444E-01
1.90E+00	DISPLACEMENTS		
		-1.5193E-04	-4.9226E-04
1.90E+00	ACCELERATIONS		
		-1.2265E-02	6.9025E-01
1.92E+00	DISPLACEMENTS		
		-1.3299E-04	-6.4388E-04
1.92E+00	ACCELERATIONS		
		-1.4423E-02	1.1039E+00
1.94E+00	DISPLACEMENTS		
		-1.1937E-04	-3.7751E-04
1.94E+00	ACCELERATIONS		
		-3.2188E-03	8.3109E-01
1.96E+00	DISPLACEMENTS		
		-1.0674E-04	2.0446E-04
1.96E+00	ACCELERATIONS		
		1.6338E-02	4.9462E-02
1.98E+00	DISPLACEMENTS		
		-8.7659E-05	8.0203E-04
1.98E+00	ACCELERATIONS		
		3.3518E-02	-8.2191E-01
2.00E+00	DISPLACEMENTS		
		-5.5593E-05	1.0876E-03
2.00E+00	ACCELERATIONS		
		3.8380E-02	-1.2335E+00

HARMONIC LOAD 40000.00000*SIN(40.0000 *T) AT GENERALIZED MASS 10

HARMONIC STEADY STATE LOAD AT MASS10
...STEADY STATE ENVELOPES...

GENERALIZED MASS	MAX	DISPLACEMENT	MAX	VELOCITY	MAX	ACCELERATION
1		1.8067E-07		7.2267E-06		2.8907E-04
2		1.9404E-07		7.7615E-06		3.1046E-04
3		1.5158E-07		6.0631E-06		2.4252E-04
4		1.0195E-07		4.0782E-06		1.6313E-04
5		6.2294E-09		2.4918E-07		9.9670E-06
6		1.2077E-07		4.8307E-06		1.9323E-04
7		1.2875E-07		5.1498E-06		2.0599E-04
8		2.2745E-07		9.0980E-06		3.6392E-04
9		6.3530E-06		2.5412E-04		1.0165E-02
10		1.0899E-04		4.3594E-03		1.7438E-01
11		3.3441E-05		1.3376E-03		5.3505E-02
12		2.6340E-05		1.0536E-03		4.2145E-02
13		1.1431E-05		4.5723E-04		1.8289E-02

TRANSIENT MOTION WITH LOAD AT GENERALIZED MASS 10

NOTEYOU ARE ASSUMING TRANSIENT MOTION IS NOT YET DAMPED OUT AND COMPLETE FORCE BUILDUP

DISPLACEMENTS.

5 7

ACCELERATIONS.

5 7

TIME			
0.	DISPLACEMENTS	0.	0.
0.	ACCELERATIONS	-1.8326E-20	4.1179E-19
2.00E-02	DISPLACEMENTS	6.7421E-11	-8.3224E-11
2.00E-02	ACCELERATIONS	1.9454E-06	-2.2471E-06
4.00E-02	DISPLACEMENTS	1.0011E-09	-1.0225E-09
4.00E-02	ACCELERATIONS	6.4507E-06	-1.5627E-06
6.00E-02	DISPLACEMENTS	4.5037E-09	-1.3496E-09
6.00E-02	ACCELERATIONS	1.0661E-05	3.6882E-05

8.00E-02 DISPLACEMENTS	1.2190E-08	1.5912E-08
8.00E-02 ACCELERATIONS	1.2610E-05	1.5826E-04
1.00E-01 DISPLACEMENTS	2.4875E-08	9.9506E-09
1.00E-01 ACCELERATIONS	1.3389E-05	3.6677E-04
1.20E-01 DISPLACEMENTS	4.3025E-08	3.3066E-07
1.20E-01 ACCELERATIONS	1.7537E-05	5.9786E-04
1.40E-01 DISPLACEMENTS	6.8505E-08	7.9778E-07
1.40E-01 ACCELERATIONS	3.1025E-05	7.3438E-04
1.60E-01 DISPLACEMENTS	1.0684E-07	1.5518E-06
1.60E-01 ACCELERATIONS	5.7743E-05	6.6874E-04
1.80E-01 DISPLACEMENTS	1.6869E-07	2.5650E-06
1.80E-01 ACCELERATIONS	9.6419E-05	3.5952E-04
2.00E-01 DISPLACEMENTS	2.6927E-07	3.7155E-06
2.00E-01 ACCELERATIONS	1.3978E-04	-1.4383E-04
2.20E-01 DISPLACEMENTS	4.2553E-07	4.8061E-06
2.20E-01 ACCELERATIONS	1.7654E-04	-7.1378E-04
2.40E-01 DISPLACEMENTS	6.5182E-07	5.6144E-06
2.40E-01 ACCELERATIONS	1.9546E-04	-1.1876E-03
2.60E-01 DISPLACEMENTS	9.5544E-07	5.9561E-06
2.60E-01 ACCELERATIONS	1.8940E-04	-1.4150E-03
2.80E-01 DISPLACEMENTS	1.3340E-06	5.7429E-06
2.80E-01 ACCELERATIONS	1.5787E-04	-1.3124E-03
3.00E-01 DISPLACEMENTS	1.7750E-06	5.0154E-06
3.00E-01 ACCELERATIONS	1.0676E-04	-8.9588E-04

3.20E-01	DISPLACEMENTS	2.2584E-06	3.9362E-06
3.20E-01	ACCELERATIONS	4.5855E-05	-2.8177E-04
3.40E-01	DISPLACEMENTS	2.7601E-06	2.7453E-06
3.40E-01	ACCELERATIONS	-1.4579E-05	3.6174E-04
3.60E-01	DISPLACEMENTS	3.2563E-06	1.6950E-06
3.60E-01	ACCELERATIONS	-6.6465E-05	8.8832E-04
3.80E-01	DISPLACEMENTS	3.7264E-06	9.9350E-07
3.80E-01	ACCELERATIONS	-1.0473E-04	1.2225E-03
4.00E-01	DISPLACEMENTS	4.1551E-06	7.7445E-07
4.00E-01	ACCELERATIONS	-1.2647E-04	1.3612E-03
4.20E-01	DISPLACEMENTS	4.5339E-06	1.0942E-06
4.20E-01	ACCELERATIONS	-1.2973E-04	1.3274E-03
4.40E-01	DISPLACEMENTS	4.8615E-06	1.9393E-06
4.40E-01	ACCELERATIONS	-1.1323E-04	1.1273E-03
4.60E-01	DISPLACEMENTS	5.1443E-06	3.2294E-06
4.60E-01	ACCELERATIONS	-7.7614E-05	7.4647E-04
4.80E-01	DISPLACEMENTS	5.3967E-06	4.8119E-06
4.80E-01	ACCELERATIONS	-2.7430E-05	1.8665E-04
5.00E-01	DISPLACEMENTS	5.6382E-06	6.4652E-06
5.00E-01	ACCELERATIONS	2.7895E-05	-4.8962E-04
5.20E-01	DISPLACEMENTS	5.8907E-06	7.9229E-06
5.20E-01	ACCELERATIONS	7.5305E-05	-1.1513E-03
5.40E-01	DISPLACEMENTS	6.1725E-06	8.9263E-06
5.40E-01	ACCELERATIONS	1.0167E-04	-1.6314E-03

5.60E-01 DISPLACEMENTS	6.4940E-06	9.2880E-06
5.60E-01 ACCELERATIONS	9.8207E-05	-1.7902E-03
5.80E-01 DISPLACEMENTS	6.8538E-06	8.9463E-06
5.80E-01 ACCELERATIONS	6.3765E-05	-1.5713E-03
6.00E-01 DISPLACEMENTS	7.2382E-06	7.9871E-06
6.00E-01 ACCELERATIONS	5.2379E-06	-1.0306E-03
6.20E-01 DISPLACEMENTS	7.6244E-06	6.6212E-06
6.20E-01 ACCELERATIONS	-6.5041E-05	-3.2432E-04
6.40E-01 DISPLACEMENTS	7.9845E-06	5.1244E-06
6.40E-01 ACCELERATIONS	-1.3376E-04	3.4556E-04
6.60E-01 DISPLACEMENTS	8.2916E-06	3.7587E-06
6.60E-01 ACCELERATIONS	-1.9085E-04	8.0757E-04
6.80E-01 DISPLACEMENTS	8.5229E-06	2.7065E-06
6.80E-01 ACCELERATIONS	-2.3161E-04	9.9137E-04
7.00E-01 DISPLACEMENTS	8.6621E-06	2.0431E-06
7.00E-01 ACCELERATIONS	-2.5598E-04	9.4868E-04
7.20E-01 DISPLACEMENTS	8.6994E-06	1.7560E-06
7.20E-01 ACCELERATIONS	-2.6585E-04	8.1035E-04
7.40E-01 DISPLACEMENTS	8.6308E-06	1.7943E-06
7.40E-01 ACCELERATIONS	-2.6226E-04	7.0166E-04
7.60E-01 DISPLACEMENTS	8.4577E-06	2.1158E-06
7.60E-01 ACCELERATIONS	-2.4416E-04	6.6763E-04
7.80E-01 DISPLACEMENTS	8.1876E-06	2.7053E-06
7.80E-01 ACCELERATIONS	-2.0964E-04	6.5685E-04

8.00E-01	DISPLACEMENTS	7.8342E-06	3.5549E-06
8.00E-01	ACCELERATIONS	-1.5846E-04	5.6872E-04
8.20E-01	DISPLACEMENTS	7.4178E-06	4.6268E-06
8.20E-01	ACCELERATIONS	-9.4298E-05	3.2960E-04
8.40E-01	DISPLACEMENTS	6.9638E-06	5.8255E-06
8.40E-01	ACCELERATIONS	-2.4990E-05	-5.5017E-05
8.60E-01	DISPLACEMENTS	6.4998E-06	7.0001E-06
8.60E-01	ACCELERATIONS	3.9503E-05	-5.0013E-04
8.80E-01	DISPLACEMENTS	6.0510E-06	7.9768E-06
8.80E-01	ACCELERATIONS	9.0334E-05	-8.8572E-04
9.00E-01	DISPLACEMENTS	5.6378E-06	8.6044E-06
9.00E-01	ACCELERATIONS	1.2244E-04	-1.1158E-03
9.20E-01	DISPLACEMENTS	5.2728E-06	8.7921E-06
9.20E-01	ACCELERATIONS	1.3525E-04	-1.1586E-03
9.40E-01	DISPLACEMENTS	4.9615E-06	8.5214E-06
9.40E-01	ACCELERATIONS	1.3130E-04	-1.0504E-03
9.60E-01	DISPLACEMENTS	4.7022E-06	7.8332E-06
9.60E-01	ACCELERATIONS	1.1379E-04	-8.6402E-04
9.80E-01	DISPLACEMENTS	4.4880E-06	6.7999E-06
9.80E-01	ACCELERATIONS	8.4758E-05	-6.5965E-04
1.00E+00	DISPLACEMENTS	4.3074E-06	5.5029E-06
1.00E+00	ACCELERATIONS	4.5053E-05	-4.4598E-04
1.02E+00	DISPLACEMENTS	4.1444E-06	4.0293E-06
1.02E+00	ACCELERATIONS	-3.9517E-06	-1.7841E-04

1.04E+00 DISPLACEMENTS	3.9798E-06	2.4881E-06
1.04E+00 ACCELERATIONS	-5.8036E-05	1.9919E-04
1.06E+00 DISPLACEMENTS	3.7920E-06	1.0306E-06
1.06E+00 ACCELERATIONS	-1.0897E-04	6.9244E-04
1.08E+00 DISPLACEMENTS	3.5611E-06	-1.4873E-07
1.08E+00 ACCELERATIONS	-1.4567E-04	1.2165E-03
1.10E+00 DISPLACEMENTS	3.2727E-06	-8.4576E-07
1.10E+00 ACCELERATIONS	-1.5761E-04	1.6122E-03
1.12E+00 DISPLACEMENTS	2.9224E-06	-9.0779E-07
1.12E+00 ACCELERATIONS	-1.3886E-04	1.7161E-03
1.14E+00 DISPLACEMENTS	2.5175E-06	-2.9609E-07
1.14E+00 ACCELERATIONS	-9.0861E-05	1.4452E-03
1.16E+00 DISPLACEMENTS	2.0769E-06	8.8251E-07
1.16E+00 ACCELERATIONS	-2.2289E-05	8.4538E-04
1.18E+00 DISPLACEMENTS	1.6277E-06	2.3935E-06
1.18E+00 ACCELERATIONS	5.3810E-05	7.6174E-05
1.20E+00 DISPLACEMENTS	1.1998E-06	3.9363E-06
1.20E+00 ACCELERATIONS	1.2424E-04	-6.5788E-04
1.22E+00 DISPLACEMENTS	8.2111E-07	5.2228E-06
1.22E+00 ACCELERATIONS	1.7918E-04	-1.1894E-03
1.24E+00 DISPLACEMENTS	5.1337E-07	6.0430E-06
1.24E+00 ACCELERATIONS	2.1352E-04	-1.4427E-03
1.26E+00 DISPLACEMENTS	2.9031E-07	6.2947E-06
1.26E+00 ACCELERATIONS	2.2607E-04	-1.4400E-03

1.28E+00 DISPLACEMENTS	1.5698E-07	5.9763E-06
1.28E+00 ACCELERATIONS	2.1751E-04	-1.2641E-03
1.30E+00 DISPLACEMENTS	1.0998E-07	5.1550E-06
1.30E+00 ACCELERATIONS	1.8895E-04	-1.0019E-03
1.32E+00 DISPLACEMENTS	1.3794E-07	3.9341E-06
1.32E+00 ACCELERATIONS	1.4185E-04	-7.0108E-04
1.34E+00 DISPLACEMENTS	2.2212E-07	2.4341E-06
1.34E+00 ACCELERATIONS	7.9280E-05	-3.6109E-04
1.36E+00 DISPLACEMENTS	3.3770E-07	7.9167E-07
1.36E+00 ACCELERATIONS	7.4156E-06	3.7101E-05
1.38E+00 DISPLACEMENTS	4.5624E-07	-8.3418E-07
1.38E+00 ACCELERATIONS	-6.4292E-05	4.8518E-04
1.40E+00 DISPLACEMENTS	5.4945E-07	-2.2664E-06
1.40E+00 ACCELERATIONS	-1.2480E-04	9.1867E-04
1.42E+00 DISPLACEMENTS	5.9343E-07	-3.3352E-06
1.42E+00 ACCELERATIONS	-1.6462E-04	1.2304E-03
1.44E+00 DISPLACEMENTS	5.7242E-07	-3.9192E-06
1.44E+00 ACCELERATIONS	-1.7894E-04	1.3221E-03
1.46E+00 DISPLACEMENTS	4.8066E-07	-3.9830E-06
1.46E+00 ACCELERATIONS	-1.6899E-04	1.1619E-03
1.48E+00 DISPLACEMENTS	3.2190E-07	-3.5886E-06
1.48E+00 ACCELERATIONS	-1.4099E-04	8.0864E-04
1.50E+00 DISPLACEMENTS	1.0707E-07	-2.8730E-06
1.50E+00 ACCELERATIONS	-1.0310E-04	3.8566E-04

1.52E+00 DISPLACEMENTS	-1.4890E-07	-2.0012E-06
1.52E+00 ACCELERATIONS	-6.2209E-05	1.8331E-05
1.54E+00 DISPLACEMENTS	-4.2979E-07	-1.1179E-06
1.54E+00 ACCELERATIONS	-2.2083E-05	-2.2549E-04
1.56E+00 DISPLACEMENTS	-7.1958E-07	-3.2087E-07
1.56E+00 ACCELERATIONS	1.6132E-05	-3.5828E-04
1.58E+00 DISPLACEMENTS	-1.0030E-06	3.3422E-07
1.58E+00 ACCELERATIONS	5.1800E-05	-4.4853E-04
1.60E+00 DISPLACEMENTS	-1.2658E-06	8.0887E-07
1.60E+00 ACCELERATIONS	8.2577E-05	-5.6679E-04
1.62E+00 DISPLACEMENTS	-1.4960E-06	1.0550E-06
1.62E+00 ACCELERATIONS	1.0369E-04	-7.3414E-04
1.64E+00 DISPLACEMENTS	-1.6852E-06	1.0074E-06
1.64E+00 ACCELERATIONS	1.0933E-04	-9.0258E-04
1.66E+00 DISPLACEMENTS	-1.8313E-06	6.0194E-07
1.66E+00 ACCELERATIONS	9.5423E-05	-9.7753E-04
1.68E+00 DISPLACEMENTS	-1.9399E-06	-1.8842E-07
1.68E+00 ACCELERATIONS	6.2191E-05	-8.7106E-04
1.70E+00 DISPLACEMENTS	-2.0241E-06	-1.3201E-06
1.70E+00 ACCELERATIONS	1.5012E-05	-5.5633E-04
1.72E+00 DISPLACEMENTS	-2.1024E-06	-2.6692E-06
1.72E+00 ACCELERATIONS	-3.7065E-05	-9.1774E-05
1.74E+00 DISPLACEMENTS	-2.1954E-06	-4.0541E-06
1.74E+00 ACCELERATIONS	-8.4373E-05	4.0083E-04

1.76F+00 DISPLACEMENTS	-2.3218F-06	-5.2821E-06
1.76F+00 ACCELERATIONS	-1.1966F-04	7.9307F-04
1.78F+00 DISPLACEMENTS	-2.4955E-06	-6.1989E-06
1.78E+00 ACCELERATIONS	-1.3959E-04	1.0081F-03
1.80F+00 DISPLACEMENTS	-2.7245F-06	-6.7184E-06
1.80F+00 ACCELERATIONS	-1.4415F-04	1.0458F-03
1.82F+00 DISPLACEMENTS	-3.0107F-06	-6.8237E-06
1.82F+00 ACCELERATIONS	-1.3467E-04	9.6221E-04
1.84F+00 DISPLACEMENTS	-3.3504E-06	-6.5461F-06
1.84E+00 ACCELERATIONS	-1.1204E-04	8.1979E-04
1.86E+00 DISPLACEMENTS	-3.7344E-06	-5.9415E-06
1.86E+00 ACCELERATIONS	-7.6256E-05	6.4469F-04
1.88E+00 DISPLACEMENTS	-4.1485F-06	-5.0807E-06
1.88F+00 ACCELERATIONS	-2.7821E-05	4.1840E-04
1.90E+00 DISPLACEMENTS	-4.5734F-06	-4.0554F-06
1.90F+00 ACCELERATIONS	3.0225E-05	1.0767E-04
1.92E+00 DISPLACEMENTS	-4.9861E-06	-2.9900E-06
1.92E+00 ACCELERATIONS	9.1067F-05	-2.9003F-04
1.94F+00 DISPLACEMENTS	-5.3626F-06	-2.0418F-06
1.94E+00 ACCELERATIONS	1.4491E-04	-7.1741E-04
1.96F+00 DISPLACEMENTS	-5.6817F-06	-1.3778E-06
1.96F+00 ACCELERATIONS	1.8180F-04	-1.0659E-03
1.98E+00 DISPLACEMENTS	-5.9289E-06	-1.1337F-06
1.98E+00 ACCELERATIONS	1.9517F-04	-1.2198E-03

2.00E+00 DISPLACEMENTS
 -6.0989E-06 -1.3687E-06
 2.00E+00 ACCELERATIONS
 1.8432E-04 -1.1138E-03

TIME		WHOLE LOAD CASE						
0.	DISPLACEMENTS	0.	0.	0.	0.	0.	0.	0.
0.	VELOCITIES	-1.9037E-16	-2.1505E-16	-1.9026E-16	-2.9532E-16	3.1630E-17	-3.2690E-16	-3.3437E-16
0.	ACCELERATIONS	3.3712E-16	-6.8531E-18	2.7605E-16	5.1061E-16	-3.1972E-17	3.1401E-16	1.5698E-16

4.00E-03	DISPLACEMENTS	2.8452E-10	9.3632E-09	1.6709E-06	1.5948E-08	4.7050E-11	9.1467E-10	1.6623E-09
4.00E-03	VELOCITIES	2.9458E-07	1.1309E-05	1.2371E-03	1.9715E-05	4.7194E-08	1.0094E-06	1.6484E-06
4.00E-03	ACCELERATIONS	2.3809E-04	1.0872E-02	5.9907E-01	1.9375E-02	3.5659E-05	9.1985E-04	1.2169E-03

8.00E-03	DISPLACEMENTS	5.9360E-09	2.6758E-07	1.2648E-05	4.7699E-07	7.7590E-10	2.7685E-08	2.5636E-08
8.00E-03	VELOCITIES	3.5346E-06	1.6121E-04	4.5156E-03	2.8823E-04	3.9791E-07	1.9111E-05	1.2623E-05
8.00E-03	ACCELERATIONS	1.8062E-03	7.6437E-02	9.9362E-01	1.3591E-01	1.5786E-04	1.1558E-02	4.6736E-03

1.20E-02	DISPLACEMENTS	4.7171E-08	1.8580E-06	3.9101E-05	3.2774E-06	4.2370E-09	2.9209E-07	1.3055E-07
1.20E-02	VELOCITIES	2.1525E-05	7.3107E-04	8.7678E-03	1.2696E-03	1.5182E-06	1.4646E-04	4.5506E-05
1.20E-02	ACCELERATIONS	8.6031E-03	2.2178E-01	1.0791E+00	3.7094E-01	4.3866E-04	6.2346E-02	1.3441E-02

TIME								
1.60E-02	DISPLACEMENTS	2.4334E-07	7.0853E-06	8.2433E-05	1.2077E-05	1.5112E-08	1.6614E-06	4.7825E-07
		7.0703E-10	5.6648E-09	3.2111E-06	1.4990E-08	1.5443E-09	1.0700E-09	
1.60E-02	VELOCITIES	8.9704E-05	2.0230E-03	1.2764E-02	3.3183E-03	4.3036E-06	6.2442E-04	1.4840E-04
		1.7577E-07	1.6247E-06	5.9081E-04	4.3291E-06	3.9066E-07	2.6607E-07	
1.60E-02	ACCELERATIONS	2.8494E-02	4.3159E-01	8.7984E-01	6.5110E-01	1.0176E-03	1.9187E-01	4.4177E-02
		3.2610E-05	3.6827E-04	7.0415E-02	9.8861E-04	7.5235E-05	4.9366E-05	
TIME								
2.00E-02	DISPLACEMENTS	9.2653E-07	1.9238E-05	1.3963E-04	3.1189E-05	4.2887E-08	6.2173E-06	1.6111E-06
		1.7158E-09	1.5835E-08	6.1741E-06	4.2176E-08	3.8266E-09	2.5968E-09	
2.00E-02	VELOCITIES	2.8056E-04	4.2037E-03	1.5598E-02	6.3740E-03	1.0276E-05	1.7967E-03	4.7745E-04
		3.4041E-07	3.6602E-06	8.9936E-04	9.8038E-06	7.8247E-07	5.1517E-07	
2.00E-02	ACCELERATIONS	7.1648E-02	6.5586E-01	5.2531E-01	8.5402E-01	2.0570E-03	4.0568E-01	1.3416E-01
		5.0296E-05	6.6811E-04	8.3431E-02	1.7989E-03	1.2329E-04	7.6055E-05	
TIME								
2.40E-02	DISPLACEMENTS	2.7991E-06	4.1818E-05	2.0525E-04	6.3729E-05	1.0434E-07	1.7309E-05	5.0309E-06
		3.5339E-09	3.6853E-08	1.0468E-05	9.8569E-08	8.1055E-09	5.3473E-09	
2.40E-02	VELOCITIES	7.0574E-04	7.2030E-03	1.6981E-02	9.9168E-03	2.1527E-05	3.9140E-03	1.3615E-03
		5.8271E-07	7.1276E-06	1.2545E-03	1.9144E-05	1.4010E-06	8.8117E-07	
2.40E-02	ACCELERATIONS	1.4674E-01	8.3031E-01	1.8076E-01	8.8473E-01	3.6680E-03	6.5131E-01	3.2799E-01
		7.1411E-05	1.0863E-03	9.3639E-02	2.9262E-03	1.8945E-04	1.0774E-04	
TIME								
2.80E-02	DISPLACEMENTS	7.0664E-06	7.7543E-05	2.7395E-04	1.1017E-04	2.2527E-07	3.8748E-05	1.3862E-05
		6.4989E-09	7.5425E-08	1.6256E-05	2.0223E-07	1.5448E-08	9.8277E-09	
2.80E-02	VELOCITIES	1.5012E-03	1.0708E-02	1.7228E-02	1.3203E-02	4.0393E-05	6.9340E-03	3.2633E-03
		9.1603E-07	1.2523E-05	1.6442E-03	3.3667E-05	2.3301E-06	1.3832E-06	
2.80E-02	ACCELERATIONS	2.5655E-01	9.0191E-01	-2.9078E-02	7.3199E-01	5.8549E-03	8.4306E-01	6.4093E-01
		9.5780E-05	1.6336E-03	1.0068E-01	4.3931E-03	2.7969E-04	1.4402E-04	
TIME								
3.20E-02	DISPLACEMENTS	1.5477E-05	1.2753E-04	3.4242E-04	1.6819E-04	4.4049E-07	7.3520E-05	3.3045E-05
		1.1001E-08	1.4031E-07	2.3651E-05	3.7665E-07	2.7307E-08	1.6618E-08	

3.20E-02 VELOCITIES

2.7966E-03	1.4250E-02	1.6982E-02	1.5629E-02	6.8993E-05	1.0505E-02	6.5838E-03
1.3531E-06	2.0377E-05	2.0553E-03	5.4751E-05	3.6808E-06	2.0390E-06	

3.20E-02 ACCELERATIONS

3.9482E-01	8.4828E-01	-6.6712E-02	4.7346E-01	8.5095E-03	9.2197E-01	1.0226E+00
1.2328E-04	2.3158E-03	1.0427E-01	6.2065E-03	4.0170E-04	1.8456E-04	

TIME

3.60E-02 DISPLACEMENTS

3.0222E-05	1.9094E-04	4.0999E-04	2.3380E-04	7.9227E-07	1.2290E-04	6.8526E-05
1.7479E-08	2.4244E-07	3.2709E-05	6.5082E-07	4.5648E-08	2.6367E-08	

3.60E-02 VELOCITIES

4.6768E-03	1.7345E-02	1.6860E-02	1.7017E-02	1.0886E-04	1.4171E-02	1.1381E-02
1.9063E-06	3.1231E-05	2.4736E-03	8.3769E-05	5.5985E-06	2.8652E-06	

3.60E-02 ACCELERATIONS

5.4564E-01	6.8379E-01	1.8649E-02	2.3395E-01	1.1461E-02	8.9918E-01	1.3562E+00
1.5384E-04	3.1336E-03	1.0425E-01	8.3570E-03	5.6486E-04	2.2922E-04	

TIME

4.00E-02 DISPLACEMENTS

5.3684E-05	2.6520E-04	4.7789E-04	3.0334E-04	1.3276E-06	1.8663E-04	1.2546E-04
2.6422E-08	3.9487E-07	4.3430E-05	1.0591E-06	7.3097E-08	3.9789E-08	

4.00E-02 VELOCITIES

7.1477E-03	1.9629E-02	1.7168E-02	1.7679E-02	1.6086E-04	1.7652E-02	1.7188E-02
2.5879E-06	4.5620E-05	2.8844E-03	1.2202E-04	8.2689E-06	3.8784E-06	

4.00E-02 ACCELERATIONS

6.8588E-01	4.5141E-01	1.3161E-01	1.2539E-01	1.4550E-02	8.4157E-01	1.5081E+00
1.8748E-04	4.0825E-03	1.0053E-01	1.0818E-02	7.8000E-04	2.7810E-04	

TIME

4.40E-02 DISPLACEMENTS

8.8071E-05	3.4666E-04	5.4782E-04	3.7513E-04	2.0958E-06	2.6386E-04	2.0617E-04
3.8370E-08	6.1279E-07	5.5754E-05	1.6409E-06	1.1311E-07	5.7666E-08	

4.40E-02 VELOCITIES

1.0115E-02	2.0939E-02	1.7833E-02	1.8265E-02	2.2534E-04	2.0943E-02	2.3098E-02
3.4104E-06	6.4053E-05	3.2729E-03	1.7067E-04	1.1924E-05	5.0962E-06	

4.40E-02 ACCELERATIONS

7.8977E-01	2.0583E-01	1.8634E-01	1.9805E-01	1.7701E-02	8.1094E-01	1.4032E+00
2.2431E-04	5.1527E-03	9.3138E-02	1.3547E-02	1.0591E-03	3.3165E-04	

TIME

4.80E-02 DISPLACEMENTS

1.3501E-04	4.3147E-04	6.2060E-04	4.5029E-04	3.1473E-06	3.5411E-04	3.0903E-04
5.3911E-08	9.1330E-07	6.9564E-05	2.4396E-06	1.7017E-07	8.0858E-08	

4.80E-02 VELOCITIES

1.3386E-02	2.1323E-02	1.8533E-02	1.9464E-02	3.0259E-04	2.4182E-02	2.8111E-02
4.3869E-06	8.6984E-05	3.6248E-03	2.3068E-04	1.6844E-05	6.5390E-06	

4.80E-02 ACCELERATIONS

8.3438E-01	-4.4211E-03	1.4917E-01	4.1926E-01	2.0950E-02	8.1055E-01	1.0741E+00
2.6454E-04	6.3292E-03	8.2213E-02	1.6485E-02	1.4149E-03	3.9081E-04	

TIME
5.20E-02 DISPLACEMENTS

1.9521E-04	5.1629E-04	6.9569E-04	5.3222E-04	4.5343E-06	4.5730E-04	4.2895E-04
7.3689E-08	1.3152E-06	8.4685E-05	3.5024E-06	2.4999E-07	1.1031E-07	

5.20E-02 VELOCITIES

1.6690E-02	2.0992E-02	1.8940E-02	2.1679E-02	3.9323E-04	2.7395E-02	3.1561E-02
5.5316E-06	1.1480E-04	3.9262E-03	3.0273E-04	2.3362E-05	8.2321E-06	

5.20E-02 ACCELERATIONS

8.0441E-01	-1.4873E-01	4.8784E-02	6.8367E-01	2.4418E-02	7.8611E-01	6.4849E-01
3.0847E-04	7.5921E-03	6.7987E-02	1.9560E-02	1.8601E-03	4.5710E-04	

TIME
5.60E-02 DISPLACEMENTS

2.6817E-04	5.9884E-04	7.7155E-04	6.2496E-04	6.3126E-06	5.7294E-04	5.5932E-04
9.8409E-08	1.8386E-06	1.0089E-04	4.8781E-06	3.5971E-07	1.4709E-07	

5.60E-02 VELOCITIES

1.9716E-02	2.0236E-02	1.8928E-02	2.4805E-02	4.9843E-04	3.0357E-02	3.3376E-02
6.8602E-06	1.4780E-04	4.1647E-03	3.8722E-04	3.1861E-05	1.0208E-05	

5.60E-02 ACCELERATIONS

6.9517E-01	-2.1714E-01	-4.8472E-02	8.5200E-01	2.8255E-02	6.7779E-01	2.7921E-01
3.5652E-04	8.9171E-03	5.0786E-02	2.2686E-02	2.4073E-03	5.3274E-04	

TIME
6.00E-02 DISPLACEMENTS

3.5215E-04	6.7800E-04	8.4673E-04	7.3104E-04	8.5435E-06	6.9931E-04	6.9437E-04
1.2884E-07	2.5048E-06	1.1790E-04	6.6167E-06	5.0809E-07	1.9241E-07	

6.00E-02 VELOCITIES

2.2155E-02	1.9348E-02	1.8638E-02	2.8209E-02	6.1988E-04	3.2688E-02	3.4006E-02
8.3899E-06	1.8618E-04	4.3291E-03	4.8416E-04	4.2771E-05	1.2510E-05	

6.00E-02 ACCELERATIONS

5.1343E-01	-2.1593E-01	-8.3235E-02	8.0915E-01	3.2544E-02	4.7416E-01	6.4460E-02
4.0918E-04	1.0276E-02	3.1028E-02	2.5769E-02	3.0675E-03	6.2071E-04	

TIME
6.40E-02 DISPLACEMENTS

4.4427E-04	7.5379E-04	9.2068E-04	8.4974E-04	1.1296E-05	8.3319E-04	8.3067E-04
1.6582E-07	3.3354E-06	1.3541E-04	8.7675E-06	7.0572E-07	2.4768E-07	

6.40E-02 VELOCITIES

2.3749E-02	1.8580E-02	1.8365E-02	3.0945E-02	7.5934E-04	3.4083E-02	3.4105E-02
1.0141E-05	2.3002E-04	4.4102E-03	5.9318E-04	5.6564E-05	1.5195E-05	

6.40E-02 ACCELERATIONS

2.7630E-01	-1.6058E-01	-4.2796E-02	5.2085E-01	3.7239E-02	2.2067E-01	7.0206E-03
4.6704E-04	1.1638E-02	9.2031E-03	2.8704E-02	3.8502E-03	7.2485E-04	

! PRINTOUT CONTINUES UNTIL T=2 SEC

Fold
or Cut

TITLE SPRVIB A General Purpose Dynamic Computer Code Using Normal Mode Solution	
	DOCUMENT NO. WANL-TME-1940	DATE June, 1971
ABSTRACT This report describes a new general dynamic computer code called SPRVIB (Spring Vib) developed at Westinghouse Astronuclear Laboratory in support of the NERVA (Nuclear Engine for Rocket Vehicle Application) Program (Contract NP-1).		
DATA ITEM NO.	SECURITY CLASSIFICATION OF DOCUMENT Unclassified	
CDRL LINE NO.	TOTAL NO. OF PAGES 163	
KEY WORDS Vibration, Modal, Kinematical-Responses, Damping, Point-Motion, Forced-Responses, Constraints, Stiffness, Natural Frequencies, Normal Mode, Interpolation, Closed-Form, Options		
AUTHOR/EDITOR R. Bolton		PREPARING DEPT. Structural Design
W Form 59943		DOCUMENT INFORMATION SUMMARY

Fold
or Cut