R[illegible]
Contract NAS8-31376

Volume IIIA

Explanations, Dense FORMA Subroutines

May 1976

Expansion and Improvement of the FORMA System for Response and Load Analysis

MARTIN MARIETTA

MCR-76-217

Contract NAS8-31376

EXPANSION AND IMPROVEMENT OF THE FORMA SYSTEM FOR RESPONSE AND LOAD ANALYSIS

Volume IIIA - Explanations, Dense FORMA Subroutines

May 1976

Author:

Richard L. Wohlen

Approved:

Richard L. Wohlen
Program Manager

Prepared for: National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama 35812

MARTIN MARIETTA CORPORATION
Denver Division
Denver, Colorado 80201

FOREWORD

This report presents results of the expansion and improvement of the FORMA system for response and load analysis. The acronym FORMA stands for FORTRAN Matrix Analysis. The study, performed from 16 May 1975 through 17 May 1976 was conducted by the Analytical Mechanics Department, Martin Marietta Corporation, Denver Division, under the contract NAS8-31376. The program was administered by the National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama under the direction of Dr. John R. Admire, Structural Dynamics Division, Systems Dynamics Laboratory.

This report is published in seven volumes:

Volume I - Programming Manual,
Volume IIA - Listings, Dense FORMA Subroutines,
Volume IIB - Listings, Sparse FORMA Subroutines,
Volume IIC - Listings, Finite Element FORMA Subroutines,
Volume IIIA - Explanations, Dense FORMA Subroutines,
Volume IIIB - Explanations, Sparse FORMA Subroutines, and
Volume IIIC - Explanations, Finite Element FORMA Subroutines.

CONTENTS

A3STRACT

This report presents techniques for the solution of structural dynamic systems on an electronic digital computer using FORMA (FORTRAN Matrix Analysis).

FORMA is a library of subroutines coded in FORTRAN IV for the efficient solution of structural dynamics problems. These subroutines are in the form of building blocks that can be put together to solve a large variety of structural dynamics problems. The obvious advantage of the building block approach is that programming and checkout time are limited to that required for putting the blocks together in the proper order.

The FORMA method has advantageous features such as:

1. subroutines in the library have been used extensively for many years and as a result are well checked out and debugged;

2. method will work on any computer with a FORTRAN IV compiler;

3. incorporation of new subroutines is no problem;

4. basic FORTRAN statements may be used to give extreme flexibility in writing a program.

Two programming techniques are used in FORMA: dense and sparse.

ACKNOWLEDGMENTS

The editor expresses his appreciation to those individuals whose assistance was necessary for the successful completion of this report. Dr. John R. Admire was instrumental in the definition of the program scope and contributed many valuable suggestions. Messrs. Carl Bodley, Wilcomb Benfield, Darrell Devers, Richard Hruda, Roger Philippus, and Herbert Wilkening, all of the Analytical Mechanics Department, Denver Division of Martin Marietta Corporation, have contributed ideas, as well as subroutines, in the formulation of the FORMA library.

The editor also expresses his appreciation to those persons who developed FORTRAN, particularly the subroutine concept of that programming tool.

LIST OF SYMBOLS

$[\ ]$	matrix
$\{\ \}$	column matrix) vector
$\{\ \}^T$	row matrix) vector
T	transpose (when symbol is a superscript)
$[\ ]_{m \times n}$	m designates the row size of matrix n designates the column size of matrix
a_{ij}	a designates an element of matrix [A] i designates the ith row of matrix [A] j designates the jth column of matrix [A]

I. INTRODUCTION

This volume presents an explanation of the function of each dense subroutine in the FORMA library. Example problems are given in some cases to clarify the operations performed by a subroutine.

II. SUBROUTINE EXPLANATIONS

The subroutines are given in alphabetical order with numbers coming before letters.

Subroutine AABB calculates the summation of two matrices, each matrix multiplied by a scalar. In matrix notation,

$$[Z]_{NRxNC} = \alpha[A]_{NRxNC} + \beta[B]_{NRxNC}$$

where

$$z_{ij} = \alpha\, a_{ij} + \beta\, b_{ij} \qquad \begin{pmatrix} i = 1, NR \\ j = 1, NC \end{pmatrix}$$

NR is the number of rows of each matrix, and

NC is the number of columns of each matrix.

The number of rows of [A] and [B] must be equal, and the number of columns of [A] and [B] must be equal.

Theorem: Matrix summation is commutative.

That is, [C] + [D] = [D] + [C].

Theorem: Matrix summation is associative.

That is, [C] + ([D] + [E]) = ([C] + [D]) + [E].

EXAMPLE

Consider input of $\alpha = 5.$, $\beta = 2.$,

$$[A]_{2\times3} = \begin{bmatrix} 7. & 2. & -3. \\ 4. & 5. & 6. \end{bmatrix}, \text{ and } [B]_{2\times3} = \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}.$$

The reader can easily verify the output to be

$$[Z]_{2\times3} = 5. \begin{bmatrix} 7. & 2. & -3. \\ 4. & 5. & 6. \end{bmatrix} - 2. \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}$$

$$= \begin{bmatrix} 21. & 26. & -33. \\ 0. & 3. & 6. \end{bmatrix}.$$

Subroutine ALOD1 takes (on option) distributed and concentrated lateral forces on a beam and replaces them with representative concentrated forces at selected points on the beam.

The x-stations of the selected points (panel points) are given in {PP}. These x-stations must be in increasing order.

The distributed lateral force, f(x), is assumed to be piecewise linear and is represented by straight line segments as shown in Figure 1.

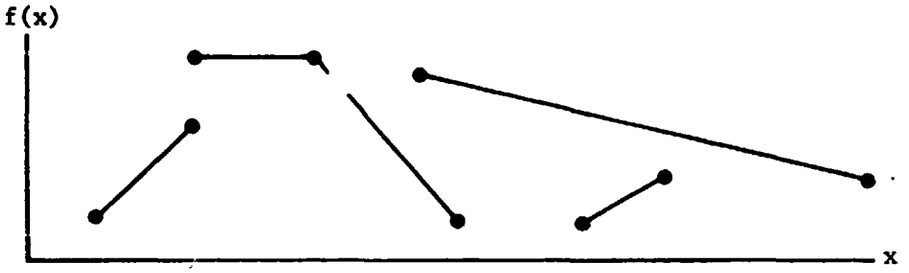


Figure 1 Distributed Lateral Force

The x-stations of the end points for the line segments giving the distributed lateral force are independent of the panel point x-stations. However, the distributed lateral force must be within the panel point limits. The line segments representing the distributed lateral force may or may not be joined and may overlap. The distributed lateral force is defined in [DIST]. Each row of [DIST] represents one nonvertical line segment. The form of each row of [DIST] is $[x_1\ x_2\ f_1\ f_2]$ where x_1, f_1 give the first end point, and x_2, f_2 give the second end point of a line segment.

The concentrated lateral forces are defined in [CONC]. Each row of [CONC] contains one concentrated lateral force, F_c, and its x location in the form $[x_c\ F_c]$. A concentrated lateral force may be outside the panel point limits.

The calculated representative concentrated forces at the selected panel points are placed in {Z}. The total lateral force and center of pressure are also calculated and printed.

DESCRIPTION OF TECHNIQUE

The replacement of distributed and concentrated lateral forces by representative concentrated forces at selected panel points is obtained using a virtual work approach. The virtual work done by the lateral forces on a beam is defined by

$$\delta W = \int_{x_S}^{x_E} f(x)\ \delta y(x)\ dx + \sum_{c} F_c\ \delta y\left(x_c\right) \tag{1}$$

where

$f(x)$ is the distributed lateral force,

F_c is a concentrated lateral force,

$\delta y(x)$ is the lateral virtual displacement measured from the body x-axis,

x is the undeformed longitudinal axis of the beam,

x_S is the starting x-station of the beam, and

x_E is the ending x-station of the beam.

The finite summation is over the number of concentrated lateral forces, F_c.

Most techniques for describing the lateral displacement (virtual or real) assume a function, $y(x)$, over the entire length of the beam. However, considerable skill is required in choosing this function. The technique used here is to represent the lateral displacement between consecutive panel points by some simple function of the panel point displacements. A *linear* displacement function will be assumed here. This is illustrated in Figure 2 where x_k and x_{k+1} are consecutive panel point stations. The region between panel points k and k+1 is referred to as bay k.

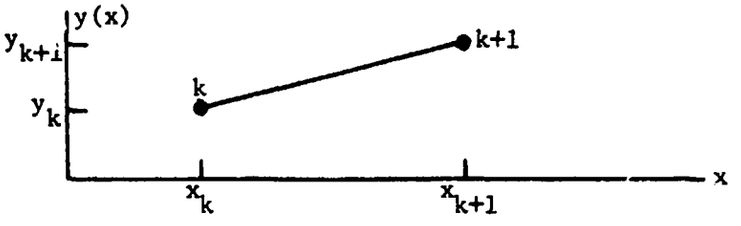


Figure 2 Linear Displacement Function

The displacement between panel points k and k+1 is given by

$$y(x) = y_k + \left(x - x_k\right)\left(y_{k+1} - y_k\right)\Big/\left(x_{k+1} - x_k\right).$$

Similarly

$$\delta y(x) = \delta y_k + \left(x - x_k\right)\left(\delta y_{k+1} - \delta y_k\right)\Big/\left(x_{k+1} - x_k\right). \tag{2}$$

The virtual work of distributed and concentrated forces will be considered separately. The distributed lateral force, f(x), is considered first. The geometry for a line segment of distributed force is shown in Figure 3.

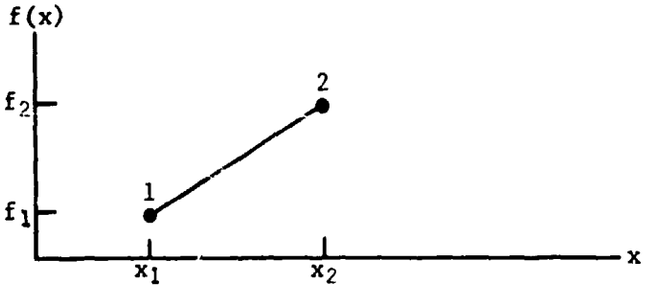


Figure 3 Line Segment Geometry

The equation for a straight line segment as shown in Figure 3 is

$$f(x) = f_1 + (x - x_1)\,(f_2 - f_1)/(x_2 - x_1). \tag{3}$$

Substituting Equations (2) and (3) into (1) gives the virtual work of the distributed lateral force represented by one line segment, i, in bay k as

$$\delta W_{i,k} = \int_{x_p}^{x_q} \left[f_p + \left(x - x_p\right)\left(f_q - f_p\right)\Big/\left(x_q - x_p\right)\right] \times$$

$$\left[\delta y_k + \left(x - x_k\right)\left(\delta y_{k+1} - \delta y_k\right)\Big/\left(x_{k+1} - x_k\right)\right] dx. \tag{4}$$

The subscripts p and q have been introduced to handle the possibility of a line segment extending past the bay limits. Thus, x_p is the greater of x_1 or x_k and x_q is the lesser of x_2 or x_{k+1}. Similarly, f_p is either f_1 or f_k, and f_q is either f_2 or f_{k+1}. The integration is continued for the line segment in adjacent bays, if necessary, until the entire line segment has been used. Performing the integration of Equation (4) yields

$$\delta W_{i,k} = \begin{bmatrix} z_k & z_{k+1} \end{bmatrix} \begin{bmatrix} \delta y_k \\ \delta y_{k+1} \end{bmatrix} \tag{5}$$

where

$$z_k = L_p \left(f_p \left\{ 3 \left(x_{k+1} - x_p \right) - L_p \right\} + f_q \left\{ 3 \left(x_{k+1} - x_p \right) - 2L_p \right\} \right) / 6L_k , \tag{5a}$$

$$z_{k+1} = L_p \left(f_p \left\{ 3 \left(x_p - x_k \right) + L_p \right\} + f_q \left\{ 3 \left(x_p - x_k \right) + 2L_p \right\} \right) / 6L_k , \tag{5b}$$

$$L_p = x_q - x_p , \text{ and} \tag{5c}$$

$$L_k = x_{k+1} - x_k . \tag{5d}$$

By definition, the coefficient of virtual displacement is the force. Thus, z_k and z_{k+1} are the representative concentrated forces at panel points k and k+1 replacing the distributed lateral force represented by one line segment i in bay k.

The concentrated lateral forces are considered last. For a force located in bay k, the displacement at the force location, $\delta y(x_c)$, is given in terms of adjacent panel point displacements $\delta y(x_k)$ and $\delta y(x_{k+1})$. Thus, from Equations (2) and (1) the virtual work of one concentrated force, F_c, in bay k is given as

$$\delta W_{c,k} = \begin{bmatrix} z_k & z_{k+1} \end{bmatrix} \begin{bmatrix} \delta y_k \\ \delta y_{k+1} \end{bmatrix} \tag{6}$$

where

$$z_k = F_c \left(x_{k+1} - x_c\right) \Big/ \left(x_{k+1} - x_k\right) \tag{6a}$$

and

$$z_{k+1} = F_c \left(x_c - x_k\right) \Big/ \left(x_{k+1} - x_k\right). \tag{6b}$$

The virtual displacement coefficients, z_k and z_{k+1}, are the representative concentrated forces at panel points k and k+1 to replace the concentrated lateral force, F_c, in bay k.

The representative concentrated panel point forces for the entire beam are finally obtained by evaluating Equations (5) for each line segment of distributed force and Equations (6) for each concentrated force. Like terms of z are then summed.

It is interesting to note that the virtual work approach with assumed linear displacement between consecutive panel points used here, gives identical results to the more common static "beaming-out" approach.

The total lateral force and center of pressure for the beam are calculated by using rigid body translation and rotation modes in the following matrix product.

$$\{Z\}^T \left[\{1\} \;\vdots\; \{PP\}\right] = \left[F_T \; P_T^o\right]$$

where

$\{1\}$ is a column of ones,

$\{PP\}$ is a column of the panel point x stations,

$\{Z\}$ is a column of the concentrated panel point forces,

F_T = total force on the beam, and

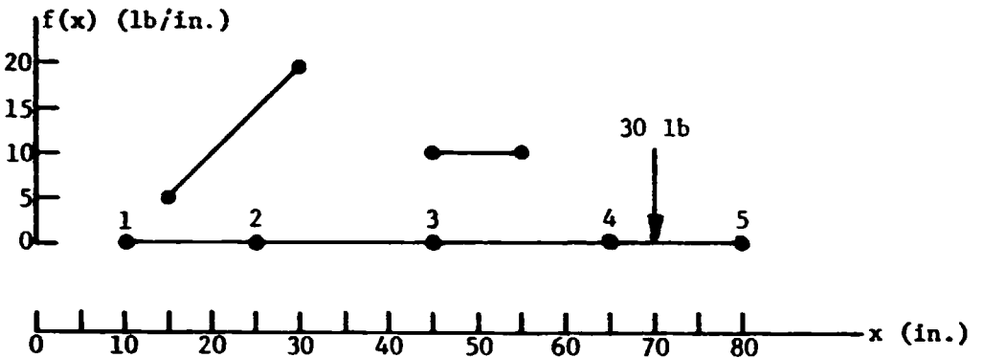
P_T^o = total moment of forces on the beam about x = 0.

From this data the center of pressure of the forces is calculated from

$$x_{cp} = P_T^o / F_T.$$

EXAMPLE

Consider a beam with distributed and concentrated lateral forces as shown in the sketch.



The panel points are denoted by the points numbered 1 thru 5. The panel point x-stations and the force data are defined in:

$$\{PP\} = \begin{bmatrix} 10. \\ 25. \\ 45. \\ 65. \\ 80. \end{bmatrix} \quad [DIST] = \begin{bmatrix} 15. & 30. & 5. & 20. \\ 45. & 55. & 10. & 10. \end{bmatrix} \text{ and } [CONC] = [70. \quad 30.].$$

Using the technique described previously and Equations (5) and (6), the reader can verify the following results.

For row 1 of [DIST]:

Bay 1: $z_1 = 27.80$ lb

$z_2 = 72.20$ lb

Bay 2: $z_2 = 76.07$ lb

$z_3 = 11.43$ lb.

For row 2 of [DIST]:

Bay 3: $z_3 = 75.00$ lb

$z_4 = 25.00$ lb.

For row 1 of [CONC]:

Bay 4: $z_4 = 20.00$ lb

$z_5 = 10.00$ lb.

The final forces at the selected panel points {PP} to replace distributed forces defined by [DIST] and concentrated forces defined by [CONC] are given by the sum of the above results as

$$\{Z\} = \begin{bmatrix} 27.80 \\ 148.27 \\ 86.43 \\ 45.00 \\ 10.00 \end{bmatrix}.$$

Subroutine ALOD2 takes (on option) distributed and concentrated axial forces on a beam and replaces them with representative concentrated forces at selected points on the beam.

The x-stations of the selected points (panel points) are given in {PP}. These x-stations must be in increasing order.

The distributed axial force, f(x), is assumed to be piecewise linear and is represented by straight line segments as shown in Figure 1.

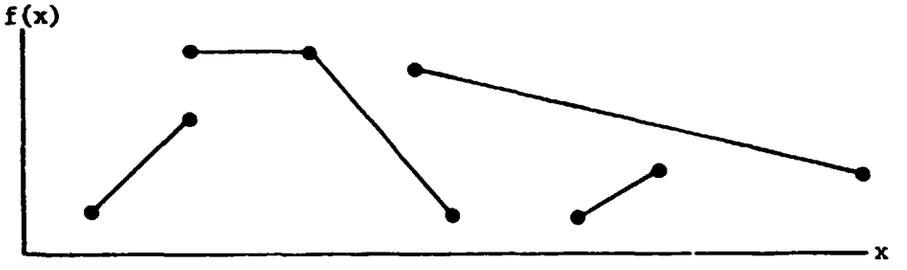


Figure 1 Distributed Axial Force

The x-stations of the end points for the line segments giving the distributed axial force are independent of the panel point x-stations. However, the distributed axial force must be within the panel point limits. The line segments representing the distributed axial force may or may not be joined and may overlap. The distributed axial force is defined in [DIST]. Each row of [DIST] represents one nonvertical line segment. The form of each row of [DIST] is $[x_1\ x_2\ f_1\ f_2]$ where x_1, f_1 give the first end point and x_2, f_2 give the second end point of a line segment.

The concentrated axial forces are defined in [CONC]. Each row of [CONC] contains one concentrated axial force, F_c, and its x location in the form $\begin{bmatrix} x_c & f_c \end{bmatrix}$. A concentrated axial force may be outside the panel point limits.

The calculated representative concentrated forces at the selected panel points are placed in {Z}. The total axial force is also calculated and printed.

DESCRIPTION OF TECHNIQUE

The replacement of distributed and concentrated axial forces by representative concentrated forces at selected panel points is obtained using a virtual work approach as follows.

The virtual work done by the axial forces on a beam is defined by

$$\delta W = \int_{x_S}^{x_E} f(x)\ \delta g(x)\ dx + \sum_c F_c\ \delta g\left(x_c\right) \tag{1}$$

where

$f(x)$ is the distributed axial force,

F_c is a concentrated axial force,

$\delta g(x)$ is the axial virtual displacement,

x is the longitudinal axis of the beam,

x_S is the starting x-station of the beam, and

x_E is the ending x-station of the beam.

The finite summation is over the number of concentrated axial forces, F_c.

Most techniques for describing the axial displacement (virtual or real) assume a function, $g(x)$, over the entire length of the beam. However, considerable skill is required in choosing this function. The technique used here is to represent the axial displacement between consecutive panel points by some simple function of the panel point displacement. A *uniform* displacement function will be assumed here. The displacement between consecutive panel points k and k+1 is assumed equal to the displacement of panel point k+1. That is,

$$g(x) = g\left(x_{k+1}\right) = g_{k+1}.$$

Similarly

$$\delta g(x) = \delta g\left(x_{k+1}\right) = \delta g_{k+1}. \tag{2}$$

The region between panel points k and k+1 is referred to as bay k.

The virtual work of distributed and concentrated forces will be considered separately. The distributed axial force, f(x), is considered first. The geometry for a line segment of distributed force is shown in Figure 2.

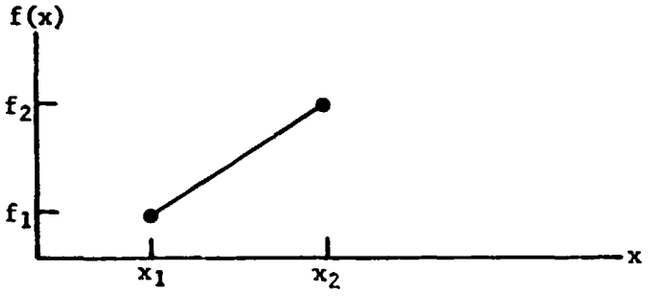


Figure 2 Line Segment Geometry

The equation for a straight line segment as shown in Figure 3 is

$$f(x) = f_1 + (x - x_1)\ (f_2 - f_1)/(x_2 - x_1). \tag{3}$$

Substituting Equations (2) and (3) into (1) gives the virtual work of the distributed axial force represented by one line segment i in bay k as

$$\delta W_{i,k} = \int_{x_p}^{x_q} \left(f_p + \left(x - x_p\right)\ \left(f_q - f_p\right) \Big/ \left(x_q - x_p\right)\right) \delta g_{k+1}\ dx. \tag{4}$$

The subscripts p and q have been introduced to handle the possibility of a line segment extending past the bay limits. Thus, x_p is the greater of x_1 or x_k and x_q is the lesser of x_2 or x_{k+1}. Similarly, f_p is either f_1 or f_k and f_q is either f_2 or f_{k+1}. The integration is continued for the line segment in adjacent bays, if necessary, until the entire line segment has been used.

Performing the integration of Equation (4) yields

$$\delta W_{i,k} = z_{k+1}\ \delta g_{k+1} \tag{5}$$

where

$$z_{k+1} = \tfrac{1}{2}\left(f_p + f_q\right)\left(x_q - x_p\right). \tag{5a}$$

By definition, the coefficient of the virtual displacement is the force. Thus, z_{k+1} is the representative concentrated force at panel point k+1 to replace the distributed axial force represented by one line segment i in bay k.

The concentrated axial forces are considered last. For a force located in bay k, the displacement at the force location is given by Equation (2) as

$$\delta g\left(x_c\right) = \delta g_{k+1}.$$

From Equation (1), the virtual work of one concentrated force, F_c, in bay k is given as

$$\delta W_{c,k} = z_{k+1}\ \delta g_{k+1} \tag{6}$$

where

$$z_{k+1} = F_c. \tag{6a}$$

The virtual displacement coefficient, z_{k+1}, is the representative concentrated force at panel point k+1 to replace the concentrated axial force, F_c, in bay k.

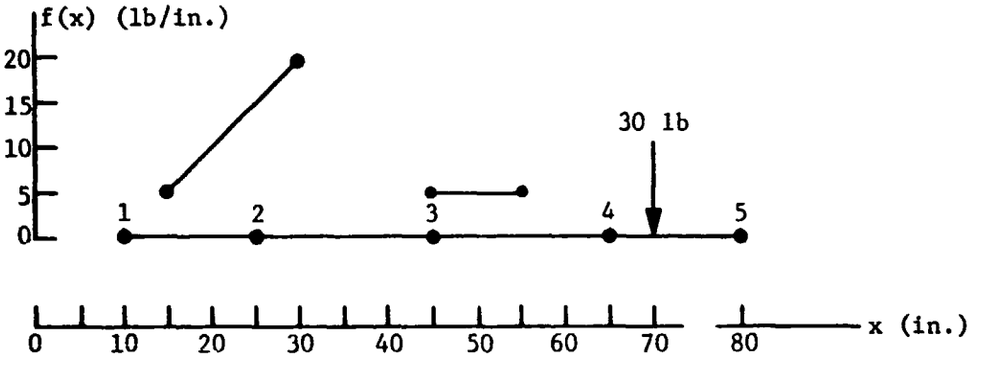
The representative concentrated panel point forces for the entire beam are finally obtained by evaluating Equation (5a) for each line segment of distributed force and Equation (6a) for each concentrated force. Like terms, z_k, are then summed.

It is interesting to note that the virtual work approach used here, with displacement in bay k assumed uniform and equal to the displacement at panel point k+1, gives identical results to the more common procedure of placing all forces in bay k at panel point k+1.

The total axial force is calculated by summing the elements of {Z}.

EXAMPLE

Consider a beam with distributed and concentrated axial forces as shown in the sketch.



The panel points are denoted by the points numbered 1 thru 5. The panel point x-stations and the force data are defined in:

$$\{PP\} = \begin{bmatrix} 10. \\ 25. \\ 45. \\ 65. \\ 80. \end{bmatrix} \quad [DIST] = \begin{bmatrix} 15. & 30. & 5. & 20. \\ 45. & 55. & 5. & 5. \end{bmatrix} \text{ and } [CONC] = [70.\ \ 30.].$$

Using the technique described previously and Equations (5a) and (6a), the reader can verify the following results.

For row 1 of [DIST]:

Bay 1: $z_2 = 100.0$ lb

Bay 2: $z_3 = 87.5$ lb.

For row 2 of [DIST]:

Bay 3: $z_4 = 50$ lb.

For row 1 of [CONC]:

Bay 4: $z_5 = 30.0$ lb.

The final forces at the selected points {PP} to replace distributed forces defined by [DIST] and concentrated forces defined by [CONC] are given by the sum of the above results as

$$\{Z\} = \begin{bmatrix} 0.0 \\ 100.00 \\ 87.5 \\ 50.0 \\ 30.0 \end{bmatrix}.$$

Subroutine ALPHAA calculates the multiplication of a matrix by a scalar. In matrix notation,

$$[Z]_{NRxNC} = \alpha[A]_{NRxNC}$$

where

$$z_{ij} = \alpha\, a_{ij} \qquad \begin{pmatrix} i = 1, NR \\ j = 1, NC \end{pmatrix}$$

NR is the number of rows of each matrix, and

NC is the number of columns of each matrix.

EXAMPLE

Consider input of $\alpha = 2.$ and

$$[A]_{2\times 3} = \begin{bmatrix} 7. & 2. & -3. \\ 4. & 5. & 6. \end{bmatrix}.$$

The reader can easily verify the output to be

$$[Z]_{2\times 3} = 2. \begin{bmatrix} 7. & 2. & -3. \\ 4. & 5. & 6. \end{bmatrix}$$

$$= \begin{bmatrix} 14. & 4. & -6. \\ 8. & 10. & 12. \end{bmatrix}.$$

Subroutine ASSEM places (assembles) a matrix [A] into a second matrix [Z] starting at a designated row, column location (IRZ, JCZ, respectively) in [Z]. The elements of [A] will replace corresponding elements in the original [Z]. Before the first use of this subroutine in forming [Z], it is important that [Z] is correctly defined. For example, if [Z] is to be originally all zeros, Subroutine ZERO could be used. This subroutine may be called repeatedly to form [Z] from the assembly of several [A] matrices. The [A] matrix must be within the row, column limits of [Z]. In subscript notation,

$$z_{ij} = a_{k\ell} \qquad \begin{pmatrix} k = 1, \text{ NRA} \\ \ell = 1, \text{ NCA} \end{pmatrix}$$

where

$$i = k + \text{IRZ} - 1$$

$$j = \ell + \text{JCZ} - 1$$

NRA is the number of rows of [A];

NCA is the number of columns of [A].

EXAMPLE

Consider a matrix defined as

$$[Z]_{3x4} = \begin{bmatrix} 1. & 0. & 0. & 0. \\ 0. & 2. & 0. & 0. \\ 0. & 0. & 0. & 0. \end{bmatrix}.$$

Matrix $[A]_{2x3} = \begin{bmatrix} 3. & 4. & 5. \\ 6. & 7. & 8. \end{bmatrix}$ is to be assembled into [Z] starting at the 2,1 location (IRZ = 2, JCZ = 1) of [Z].

The result of this operation will be

$$[Z]_{3x4} = \begin{bmatrix} 1. & 0. & 0. & 0. \\ 3. & 4. & 5. & 0. \\ 6. & 7. & 8. & 0. \end{bmatrix}.$$

Subroutine ATXBA1 calculates a special matrix product. In matrix notation,

$$[Z]_{NCB\times NCB} = ([A]^T)_{NCB\times NRB}\ [B]_{NRB\times NCB}$$

where [Z] is calculated as symmetric and placed in the same core locations as [A] to allow larger matrix sizes. NRB is the number of rows of [B], columns of $[A]^T$. NCB is the number of columns of [B], rows of $[A]^T$, and size of [Z].

In scalar summation notation,

$$z_{ij} = \sum_{k=1}^{NRB} (a^T)_{ik}\ b_{kj} = \sum_{k=1}^{NRB} a_{ki}\ b_{kj} \qquad \begin{pmatrix} i = 1, NCB \\ j = I, NCB \end{pmatrix}$$

DESCRIPTION OF TECHNIQUE

To reduce computer time and accomplish the matrix product using only two matrix core spaces, an intermediate work space vector is used. The size of this work vector determines the maximum number of columns of [B], i.e., NCB. The matrix multiplication is accomplished as follows. A single row of $[A]^T$ (columns of [A]) is multiplied times the columns of [B]. These results are stored in the work vector until all the columns of [B] have been used. Because [Z] will be symmetric, columns of [B] less than the row of $[A]^T$ need not be used. The elements in the work vector then replace the row of $[A]^T$ (column of [A]) used in the multiplication. This procedure is repeated for all rows of $[A]^T$ (columns of [A]) to calculate the lower half of the answer [Z]. The lower half is then reflected to the upper half to give the final result.

Subroutine ATXBB calculates a special matrix product. In matrix notation,

$$[Z]_{NRAT\times NCB} = ([A]^T)_{NRAT\times NRB}\ [B]_{NRB\times NCB}$$

where [Z] is placed in the same core locations as [B] to allow larger matrix sizes; NRAT is the number of rows of $[A]^T$ and [Z]; NRB is the number of rows of [B] and columns of $[A]^T$; NCB is the number of columns of [B] and [Z].

In scalar summation notation

$$z_{ij} = \sum_{k=1}^{NRB} (a^T)_{ik}\, b_{kj} = \sum_{k=1}^{NRB} a_{ki}\, b_{kj} \qquad \begin{pmatrix} i = 1, NRAT \\ j = 1, NCB \end{pmatrix}$$

DESCRIPTION OF TECHNIQUE

To reduce computer time and accomplish the matrix product using only two matrix core spaces, an intermediate work space vector is used. The size of this work vector determines the maximum number of rows of $[A]^T$, i.e., NRAT. The matrix multiplication is accomplished as follows. The rows of $[A]^T$ (columns of [A]) are multiplied times a single column of [B]. These results are stored in the work vector until all the rows of $[A]^T$ have been used. The elements in the work vector then replace the column of [B] used in the multiplication. This procedure is repeated for all columns of [B] to calculate the result [Z].

Subroutine ATXBB1 calculates a special matrix product. In matrix notation,

$$[Z]_{NRB\times NCB} = ([A]^T)_{NRB\times NRB}\ [B]_{NRB\times NCB}$$

where

[A] = a square, upper triangular matrix;

[Z] = placed in the same core locations as [B] to allow larger matrix sizes;

NRB = the number of rows of [B] and [Z] and size of [A];

NCB = the number of columns of [B] and [Z].

In scalar summation notation

$$z_{ij} = \sum_{k=1}^{NRB} (A^T)_{ik}\, b_{kj} = \sum_{k=1}^{NRB} a_{ki}\, b_{kj} \qquad \begin{pmatrix} i = 1, NRB \\ j = 1, NCB \end{pmatrix}$$

DESCRIPTION OF TECHNIQUE

To reduce computer time and accomplish the matrix product using only two matrix core spaces, an intermediate work space vector is used. The size of this work vector determines the maximum NRB. The matrix multiplication is accomplished as follows. The rows of $[A]^T$ (columns of [A]) up to the diagonal are multiplied times a single column of [B[. These results are stored in the work vector until all the rows of $[A]^T$ have been used. The elements in the work vector then replace the column of [B] used in the multiplication. This procedure is repeated for all columns of [B] to calculate the result [Z].

Subroutine ATXBB2 calculates a special matrix product. In matrix notation,

$$[Z]_{NCB\times NCB} = ([A]^T)_{NCB\times NRB}\ [B]_{NRB\times NCB}$$

where

[A[= rectangular matrix;

[B] = rectangular matrix;

[Z] = square, symmetric matrix;

NRB = number of rows of [B] and [A];

NCB = number of columns of [B] and [A]. Number of rows and columns of [Z].

The result, [Z], is formed in the same core location as [B].

The algorithm for the elements of the matrix product, [Z], is

$$z_{ij} = z_{ji} = \sum_{k=1}^{NRB} (A^T)_{ik}\, b_{kj} = \sum_{k=1}^{NRB} a_{ki}\, b_{kj} \quad \begin{pmatrix} i = 1, NCB \\ j = i, NCB \end{pmatrix}$$

DESCRIPTION OF TECHNIQUE

This subroutine is intended to be used as the second multiplication of a triple matrix product

$$[Z] = [A]^T\ [C]\ [A]$$

and matrix [B] represents the product of [C] and [A]

$$[B] = [C]\ [A].$$

Matrix [C] is assumed to be symmetric and it follows, therefore, that [Z] is also symmetric. The upper half of [Z] is calculated and reflected to form the lower half. The result, [Z], is placed in the [B] core locations.

Subroutine AXBA1 calculates a special matrix product. In matrix notation,

$$[Z]_{NRA \times NCA} = [A]_{NRA \times NCA} \; [B]_{NCA \times NCA}$$

where

[A] = rectangular matrix;

[B] = upper triangular, squ re matrix;

[Z] = rectangular matrix;

NRA = number of rows of [A] and [Z];

NCA = number cf columns of [A] and [Z];
number of rows and columns of [B].

The result, [Z], is formed in the same core locations as [A].

The algorithm for the elements of the matrix product, [Z], is

$$z_{ij} = \sum_{k=1}^{i} a_{ik} \, b_{kj} \qquad \begin{pmatrix} i = 1, \text{NRA} \\ j = 1, \text{NCA} \end{pmatrix}$$

Subroutine AXBA2 calculates a special matrix product. In matrix notation,

$$[Z]_{N\times N} = [A]_{N\times N}\,[B]_{N\times N}$$

where

[A] = square matrix;

[B] = upper triangular, square matrix;

[Z] = symmetric, square matrix;

N = number of rows and columns in [A], [B], and [Z].

The result, [Z], is formed in the same core locations as [A].

The algorithm for the elements of the matrix product, [Z], is

$$z_{ij} = z_{ji} = \sum_{k=1}^{i} a_{ik}\, b_{kj} \qquad \begin{pmatrix} i = 1, N \\ j = i, N \end{pmatrix}$$

Subroutine AXBA3 calculates a special matrix product. In matrix notation,

$$[Z]_{NRB\times NCB} = [A]_{NRB\times NRB}\ [B]_{NRB\times NCB}$$

where

[A] = upper triangular, square matrix;

[B] = rectangular matrix;

[Z] = rectangular matrix;

NRB = number of rows of [B] and [Z],
number of rows and columns of [A];

NCB = number of columns of [B] and [Z].

The result, [Z], is formed in the same core locations as [A].

The algorithm for the elements of the matrix product, [Z], is

$$z_{ij} = \sum_{k=i}^{NRB} a_{ik}\, b_{kj} \qquad \begin{pmatrix} i = 1, NRB \\ j = 1, NCB \end{pmatrix}$$

Subroutine BABT calculates a special form of the triple matrix product. In matrix notation,

$$[Z]_{NRB \times NRB} = [B]_{NRB \times NCB} \; [A]_{NCB \times NCB} \; [B]^{T}_{NCB \times NRB}$$

where

$$z_{ij} = \sum_{\ell=1}^{NCB} \sum_{k=1}^{NCB} b_{i\ell} \; a_{\ell k} \; b_{jk} \qquad \begin{pmatrix} i = 1, \; NRB \\ j = 1, \; NRB \end{pmatrix}$$

[A] = square, non-symmetric matrix.

[B] = rectangular matrix.

[Z] = square, non-symmetric matrix.

NRB = number of rows of [B] and size of [Z].

NCB = number of columns of [B] and size of [A].

Theorem: If [A] is symmetric (that is, $[A] = [A]^{T}$) then [Z] is symmetric.

Proof: $[Z]^{T} = \left([B] \; [A] \; [B]^{T}\right)^{T}$

$= \left([B]^{T}\right)^{T} \; [A]^{T} \; [B]^{T}$

$= [B] \; [A] \; [B]^{T}$

$= [Z]$.

Therefore, if [A] is non-symmetric, [Z] is non-symmetric.

DESCRIPTION OF TECHNIQUE

To reduce computer time, an intermediate work space vector is used in the subroutine. The size of this work vector determines the limitation on the number of columns of [B] and thus the size of [A] (i.e., NCB). This special triple matrix product is accomplished as follows. The rows of [A] are multiplied times a column of $[B]^{T}$. These results are stored in the work vector. Next, the rows of [B] multiply the work vector to obtain a column of the answer [Z]. This procedure is reported for all the columns

of $[B]^T$ to give the answer $[Z]$.

EXAMPLE

Consider input of

$$[A]_{2x2} = \begin{bmatrix} 13. & 14. \\ 15. & 16. \end{bmatrix} \quad \text{and} \quad [B]_{3x2} = \begin{bmatrix} 7. & 10. \\ -8. & 11. \\ 9. & 12. \end{bmatrix}$$

The reader can easily verify the output to be

$$[Z]_{3\times3} = \begin{bmatrix} 4267. & 910. & 5265. \\ 1067. & 216. & 1317. \\ 5259. & 1122. & 6489. \end{bmatrix}.$$

Subroutine BABTA calculates a special form of the triple matrix product. This subroutine is a modification of Subroutine BABT to allow larger matrix sizes by placing the answer [Z] in the same core locations as [A] . In matrix notation,

$$[Z]_{NRB \times NRB} = [B]_{NRB \times NCB} \; [A]_{NCB \times NCB} \; [B]^{T}_{NCB \times NRB}$$

where

$$z_{ij} = \sum_{\ell=1}^{NCB} \sum_{k=1}^{NCB} b_{i\ell} \; a_{\ell k} \; b_{jk} \qquad \begin{pmatrix} i = 1, \; NRB \\ j = 1, \; NRB \end{pmatrix}$$

[A] = square, non-symmetric matrix.

[B] = rectangular matrix.

[Z] = square, non-symmetric matrix.

NRB = number of rows of [B] and size of [Z].

NCB = number of columns of [B] and size of [A].

Theorem: If [A] is symmetric (that is, $[A] = [A]^T$), then [Z] is symmetric.

Proof: $[Z]^T = \left([B] \; [A] \; [B]^T\right)^T$

$= \left([B]^T\right)^T \; [A]^T \; [B]^T$

$= [B] \; [A] \; [B]^T$

$= [Z]$.

Therefore, if [A] is non-symmetric, [Z] is non-symmetric.

DESCRIPTION OF TECHNIQUE

To reduce computer time and to accomplish the special triple matrix product using only two matrix core spaces, an intermediate work space

vector is used in the subroutine. The size of this work vector determines the limitation on the number of columns of [B] (i.e., NCB). The special triple matrix product is accomplished as follows. A row of [A] is placed in the work vector which is multiplied times the columns of $[B]^T$. These results then replace the row of [A] used in the multiplication. This procedure is repeated for all rows of [A] to obtain $[AB^T]$. Next, a column of $[AB^T]$ is placed in the work vector which is multiplied by the rows of [B] to give a column of the answer [Z] . This last procedure is repeated for all columns of $[AB^T]$ to give the answer [Z] .

EXAMPLE

Consider input of

$$[A]_{2x2} = \begin{bmatrix} 13. & 14. \\ 15. & 16. \end{bmatrix} \quad \text{and} \quad [B]_{3x2} = \begin{bmatrix} 7. & 10. \\ -8. & 11. \\ 9. & 12. \end{bmatrix}$$

The reader can easily verify the output to be

$$[Z]_{3x3} = \begin{bmatrix} 4267. & 910. & 5265. \\ 1067. & 216. & 1317. \\ 5259. & 1122. & 6489. \end{bmatrix}.$$

Subroutine BTAB calculates a special triple matrix product. In matrix notation,

$$[Z]_{NCB \times NCB} = [B]^{T}_{NCB \times NRB} \; [A]_{NRB \times NRB} \; [B]_{NRB \times NCB}$$

where

$$z_{ij} = \sum_{\ell=1}^{NRB} \sum_{k=1}^{NRB} b_{\ell i} \; a_{\ell k} \; b_{kj} \qquad \begin{pmatrix} i = 1, NCB \\ j = 1, NCB \end{pmatrix}$$

[A] = square, non-symmetric matrix.

[B] = rectangular matrix

[Z] = square, non-symmetric matrix

NRB = number of rows of [B] and size of [A] .

NCB = number of columns of [B] and size of [Z] .

Theorem: If [A] is symmetric (that is, $[A] = [A]^{T}$), then [Z] is symmetric.

Proof:
$$[Z]^{T} = \left([B]^{T} \; [A] \; [B]\right)^{T}$$

$$= [B]^{T} \; [A]^{T} \; \left([B]^{T}\right)^{T}$$

$$= [B]^{T} \; [A] \; [B]$$

$$= [Z].$$

Therefore, if [A] is non-symmetric, [Z] is non-symmetric.

DESCRIPTION OF TECHNIQUE

To reduce computer time, an intermediate work space vector is used in the subroutine. The size of this work vector determines the limitation on the number of rows of [B] and thus the size of [A] (i.e., NRB). The triple matrix product is accomplished as follows. The rows of [A] are multiplied times a column of [B] . These results are stored

in the work vector. Next, the rows of $[B]^T$ (columns of $[B]$) multiply the work vector to obtain a column of the answer $[Z]$. This procedure is repeated for all the columns of $[B]$ to give the answer $[Z]$.

EXAMPLE

Consider input of

$$[A]_{2\times 2} = \begin{bmatrix} 13. & 14. \\ 15. & 16. \end{bmatrix} \text{ and } [B]_{2\times 3} = \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}.$$

The reader can easily verify the output to be

$$[Z]_{3\times 3} = \begin{bmatrix} 4267. & 910. & 5265. \\ 1067. & 216. & 1317. \\ 5259. & 1122. & 6489. \end{bmatrix}.$$

Subroutine BTABA calculates a special triple matrix product. This subroutine is a modification of Subroutine BTAB to allow larger matrix sizes by placing the answer [Z] in the same core locations as [A]. In matrix notation,

$$[Z]_{NCB \times NCB} = [B]^{T}_{NCB \times NRB} \; [A]_{NRB \times NRB} \; [B]_{NRB \times NCB}$$

where

$$z_{ij} = \sum_{\ell=1}^{NRB} \sum_{k=1}^{NRB} b_{\ell i} \, a_{\ell k} \, b_{kj} \qquad \begin{pmatrix} i = 1, \; NCB \\ j = 1, \; NCB \end{pmatrix}$$

[A] = square, non-symmetric matrix.

[B] = rectangular matrix.

[Z] = square, non-symmetric matrix.

NRB = number of rows of [B] and size of [A] .

NCB = number of columns of [B] and size of [Z] .

Theorem: If [A] is symmetric (that is, $[A] = [A]^{T}$), then [Z] is symmetric.

Proof:
$$[Z]^{T} = \left([B]^{T} \; [A] \; [B]\right)^{T}$$

$$= [B]^{T} \; [A]^{T} \; \left([B]^{T}\right)^{T}$$

$$= [B]^{T} \; [A] \; [B]$$

$$= [Z].$$

Therefore, if [A] is non-symmetric, [Z] is non-symmetric.

DESCRIPTION OF TECHNIQUE

To reduce computer time and to accomplish the triple matrix product using only two matrix core spaces, an intermediate work space vector is used in the subroutine. The size of this work vector determines the

limitation on the number of rows and columns of [B] (i.e., NRB and NCB). The triple matrix product is accomplished as follows. A row of [A] is placed in the work vector which is multiplied times the columns of [B]. These results then replace the row of [A] used in the multiplication. This procedure is repeated for all rows of [A] to obtain [AB]. Next the rows of $[B]^T$ (columns of [B]) are multiplied times a single column of [AB] just formed. These results are stored in the work vector until all the rows of $[B]^T$ have been used. The elements in the work vector then replace the column of [AB] used in the multiplication. This procedure is repeated for all columns of [AB] to give the answer [Z].

EXAMPLE

Consider input of

$$[A]_{2\times 2} = \begin{bmatrix} 13. & 14. \\ 15. & 16. \end{bmatrix} \text{ and } [B]_{2\times 3} = \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}.$$

The reader can easily verify the output to be

$$[Z]_{3\times 3} = \begin{bmatrix} 4267. & 910. & 5265. \\ 1067. & 216. & 1317. \\ 5259. & 1122. & 6489. \end{bmatrix}.$$

Subroutine BTABA2 calculates a special triple matrix product. In matrix notation,

$$[Z]_{N\times N} = [B]^{T}_{N\times N}\ [A]_{N\times N}\ [B]_{N\times N}$$

where

[A] = square, symmetric matrix;

[B] = square, upper triangular matrix;

[Z] = square, symmetric matrix. This answer is placed in the same core locations as [A] to allow larger sizes;

N = size of each matrix.

Theorem: If [A] is symmetric (that is, $[A] = [A]^{T}$), then [Z] is symmetric.

Proof:

$$\begin{aligned}[Z]^{T} &= ([B]^{T}\ [A]\ [B])^{T}\\ &= [B]^{T}\ [A]^{T}\ ([B]^{T})^{T}\\ &= [B]^{T}\ [A]\ [B]\\ &= [Z].\end{aligned}$$

DESCRIPTION OF TECHNIQUE

To reduce computer time and accomplish the triple matrix product using only two matrix core spaces, an intermediate work space vector is used. The size of this work vector determines the maximum matrix size, i.e., N. The triple matrix product is accomplished as follows. A single row of [A] is placed in the work vector and is multiplied times the columns of [B]. These results then replace the row of [A] used in the multiplication. This procedure is repeated for all the rows of [A]. Use is made of the upper triangular form of [B]. Next, the rows of $[B]^{T}$ (columns of [B]) are multiplied times a single column of [AB] just formed. These results are stored in the work vector until all of the rows of $[B]^{T}$ have been used. Again use is made of the upper triangular form of [B]. The elements in the work vector then replace the column of [AB] used in the multiplication, down to the diagonal and the corresponding row out to the diagonal. This procedure is repeated for all columns and corresponding rows of [AB] to give the answer [Z].

Example:

Consider input of

$$A_{2\times2} = \begin{bmatrix} 13. & 14. \\ 14. & 16. \end{bmatrix} \text{ and } [B]_{2\times2} = \begin{bmatrix} 7. & -8. \\ 0. & 11. \end{bmatrix}.$$

The reader can easily verify the result to be

$$[Z]_{2\times2} = \begin{bmatrix} 637. & 350. \\ 350. & 304. \end{bmatrix}$$

Subroutine COLMLT evaluates a special matrix operation by multiplying each column of a matrix [B] by a scalar. That is,

$$[Z]_{NRxNC} = \left[a_1\{b_1\}_{NRx1}\ a_2\{b_2\}_{NRx1}\ \cdots\ a_{NC}\left\{b_{NC}\right\}_{NRx1} \right]$$

where

$$z_{ij} = a_j\, b_{ij} \qquad \begin{pmatrix} i = 1, NR \\ j = 1, NC \end{pmatrix}$$

$\left\{b_j\right\}$ denotes column j of [B]. Each scalar a_j is an element of the input vector {AVEC}. NR is the number of rows in [B] and [Z], and NC is the number of columns in [B] and [Z] and the size of {AVEC}. The number of elements of {AVEC} must be equal to the number of solumns of [B].

EXAMPLE

Consider input of

$$\{AVEC\} = [2.\quad -3.\quad 4.] \text{ and } [B]_{2\times 3} = \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}.$$

The output will then be

$$[Z]_{2\times 3} = \left[2.\begin{bmatrix} 7. \\ 10. \end{bmatrix} -3.\begin{bmatrix} -8. \\ 11. \end{bmatrix} 4.\begin{bmatrix} 9. \\ 12. \end{bmatrix} \right]$$

$$= \begin{bmatrix} 14. & 24. & 36. \\ 20. & -33. & 48. \end{bmatrix}.$$

Subroutine COMENT reads input comment cards and reproduces each card in the printed output of the computer run. Each comment card may have any keypunch symbol in card columns 1 thru 78. A use of COMENT is to print an explanation of coordinates used in a computer run. Thus, this information is always retained with a run to correlate matrix location numbers with physical coordinates.

Subroutine DCOM1 decomposes [A] to form an upper triangular matrix [Z] such that $[A] = [Z]^T [Z]$. [A] must be real, square, symmetric and positive definite. The Choleski square root method is used. In matrix notation,

$$[A]_{NxN} = [Z]^T_{NxN} [Z]_{NxN}$$

where

$$[Z]_{NxN} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1N} \\ 0 & z_{22} & \cdots & z_{2N} \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & z_{NN} \end{bmatrix}.$$

N is the size of the matrices (square).

DESCRIPTION OF TECHNIQUE

To determine the elements of [Z], consider the multiplication of two matrices.

$$[A] = [B] [Z].$$

A general term of [A] is

$$a_{ij} = b_{i1} z_{1j} + b_{i2} z_{2j} + \ldots + b_{iN} z_{Nj}.$$

But

$$[B] = [Z]^T,$$

thus

$$b_{ij} = z_{ji}.$$

Therefore

$$a_{ij} = z_{1i} z_{1j} + z_{2i} z_{2j} + \ldots + z_{Ni} z_{Nj}.$$

Because

$$z_{ij} = 0 \text{ for } i > j,$$

we have

$$a_{ij} = z_{1i}\, z_{1j} + z_{2i}\, z_{2j} + \ldots + z_{ii}\, z_{ij}, \qquad (i < j)$$

and

$$a_{ii} = z_{1i}^2 + z_{2i}^2 + \ldots + z_{ii}^2.$$

From these last two equations the formulas for determining z_{ij} are obtained. That is,

$$z_{11} = \sqrt{a_{11}}$$

$$z_{1j} = a_{1j} \Big/ z_{11} \qquad (j = 2, N)$$

$$z_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} z_{ki}^2} \qquad (i = 2, N)$$

$$z_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} z_{ki}\, z_{kj} \right) \Big/ z_{ii} \qquad (j > i)$$

$$z_{ij} = 0. \qquad (i > j)$$

MISCELLANEOUS

The diagonal elements of [Z] are the square root of the determinant ratios of [A]. The determinant of [A] is the product of these determinant ratios. That is,

$$|A| = \prod_{i=1}^{N} z_{ii}^2.$$

EXAMPLE

The input matrix to be decomposed is

$$[A]_{3\times 3} = \begin{bmatrix} 9. & 15. & 12. \\ 15. & 29. & 22. \\ 12. & 22. & 53. \end{bmatrix}.$$

Using the technique described previously, the reader can verify the output to be

$$[Z]_{3\times 3} = \begin{bmatrix} 3. & 5. & 4. \\ 0. & 2. & 1. \\ 0. & 0. & 6. \end{bmatrix}.$$

REFERENCE

Faddeeva, V. N.: *Computational Methods of Linear Algebra.* Dover Publications Inc., New York, 1959.

Subroutine DIAG places the elements from a vector (row or column matrix) $\{AVEC\}$ into the corresponding diagonal locations of a square matrix [Z] and sets the off-diagonal elements of the square matrix to zero. In subscript notation,

$$z_{ii} = a_i \qquad (i = 1, N)$$

$$z_{ij} = 0. \qquad (i \neq j)$$

In matrix notation,

$$[Z]_{NxN} = \begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_N \end{bmatrix}.$$

where N is the size of [Z] (square), and the length of $\{AVEC\}$.

EXAMPLE

Consider input of

$$\{AVEC\} = \begin{bmatrix} 1. \\ 2. \\ 3. \end{bmatrix}$$

and N = 3 .

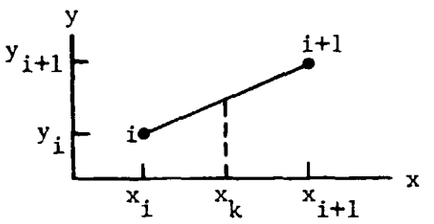
The result of this subroutine will give

$$[Z]_{3x3} = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 2. & 0. \\ 0. & 0. & 3. \end{bmatrix}.$$

Subroutine DIFF1 performs numerical differentiation assuming a linear function between known points. The x and y coordinates of the known points are given by the elements of {XA} and the corresponding elements in a column of [YA], respectively. Each column of [YA] gives the y coordinates of a different set of points. Derivatives are calculated at selected x coordinates which are given by the elements of {XZ}. These derivatives are placed in [Z]. Each column of [Z] has derivatives of the respective column of [YA]. Differentiation of an extrapolated linear function is performed when any element of {XZ} exceeds the limits of {XA}.

DERIVATION OF TECHNIQUE

Given the x,y coordinates of points i and i+1, the derivative dy/dx at x_k is to be found by assuming a linear function.



From the above sketch it is obvious that

$$\left.\frac{dy}{dx}\right|_{x_k} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} . \qquad (1)$$

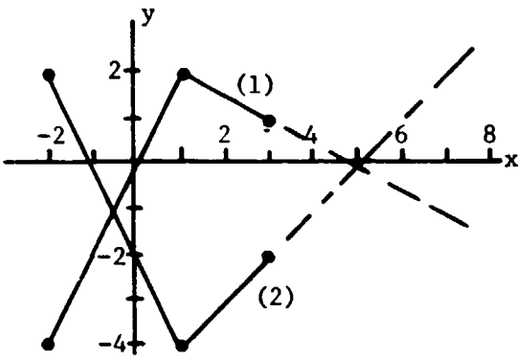
The following tabulation gives the correlation between the nomenclature of Equation (1) and that used in the Fortran coding in the subroutine.

Equation (1)	Fortran Coding*
x_i	XA(I)
x_{i+1}	XA(I+1)
y_i	YA(I) *
y_{i+1}	YA(I+1) *
x_k	XZ(K)
$dy/dx\|_{x_k}$	Z(K) *

*Subscript j, denoting different sets of points, has been omitted for clarity.

EXAMPLE

Consider the following two sets of points denoted by (1) and (2).



The coordinates of the points are given by

$$\{XA\} = \begin{bmatrix} -2. \\ 1. \\ 3. \end{bmatrix} \quad \text{and} \quad [YA] = \begin{bmatrix} -4. & 2. \\ 2. & -4. \\ 1. & -2. \end{bmatrix}.$$

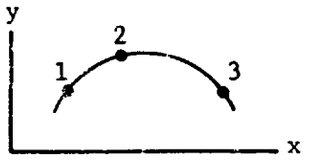
{XA} gives the x coordinates of the points in both sets (1) and (2). Column 1 of [YA] gives the y coordinates of the points in set (1) and column 2 of [YA] gives the y coordinates of the points in set (2). Derivatives are wanted at x = -1. and x = 7., that is, at $\{XZ\} = \begin{vmatrix} -1. \\ 7. \end{vmatrix}$. At x = -1., derivatives of dy/dx = 2. and dy/dx = -2. are calculated from columns 1 and 2 of [YA], respectively, using rows 1 and 2 of {XA} and [YA]. At x = 7., derivatives of dy/dx = -0.5 and dy/dx = 1. are calculated from columns 1 and 2 of [YA], respectively, using rows 2 and 3 of {XA} and [YA]. The final result is $[Z] = \begin{vmatrix} 2. & -2. \\ -0.5 & 1. \end{vmatrix}$.

To relate this example problem to a practical problem, consider {XA} to be the collocation points (panel points) of a vehicle and [YA] to be the modal displacements for two modes. Gyros are to be placed at stations x = -1. and 7. (i.e., {XZ}). The modal slopes ([Z]) are to be found at these stations.

Subroutine DIFF2 performs numerical differentiation assuming a diparabolic function between known points. A parabolic function is used where only three points are available. A diparabolic function is obtained from the weighted average of two adjacent parabolas and will be explained later. The x and y coordinates of the known points are given by the elements of {XA} and the corresponding elements in a column of [YA], respectively. Each column of [YA] gives the y coordinates of a different set of points. Derivatives are calculated at selected x coordinates which are given by the elements of {XZ}. These derivatives are placed in [Z]. Each column of [Z] has derivatives of the respective column of [YA]. Differentiation of an extrapolated parabolic function is performed when any element of {XZ} exceeds the limits of {XA}.

DERIVATION OF TECHNIQUE

The diparabolic differentiation procedure is obtained as follows. Because this procedure is dependent upon using parabolas, the parabola will be considered first. Given the x,y coordinates of points 1, 2, and 3, a parabola is to be fitted to these points.



The equation for a parabola with axis parallel to the y axis is

$$y(x) = Ax^2 + Bx + C$$

or

$$y(H) = [H^2 \quad H \quad 1] \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

where

$$H = \frac{x - x_1}{x_2 - x_1} \tag{1a}$$

or

$$H = \frac{x - x_2}{x_3 - x_2} \tag{1b}$$

is used for ease in later algebraic calculations. The coefficients a, b, and c can be determined because x (or H) and y coordinates at points 1, 2, and 3 are known. That is,

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} H_1^2 & H_1 & 1 \\ H_2^2 & H_2 & 1 \\ H_3^2 & H_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

from which

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = [\psi] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

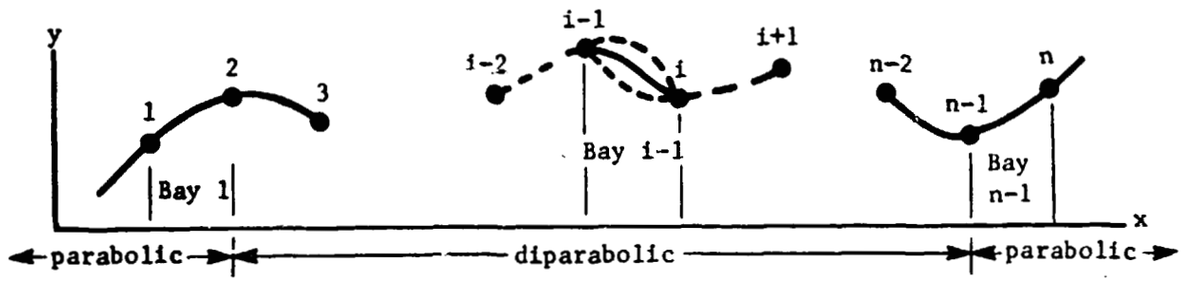
where

$$[\psi] = \frac{\begin{bmatrix} H_2-H_3 & -(H_1-H_3) & H_1-H_2 \\ -(H_2-H_3)(H_2+H_3) & (H_1-H_3)(H_1+H_3) & -(H_1-H_2)(H_1+H_2) \\ H_2H_3(H_2-H_3) & -H_1H_3(H_1-H_3) & H_1H_2(H_1-H_2) \end{bmatrix}}{H_1^2(H_2-H_3) - H_2^2(H_1-H_3) + H_3^2(H_1-H_2)} \qquad (2)$$

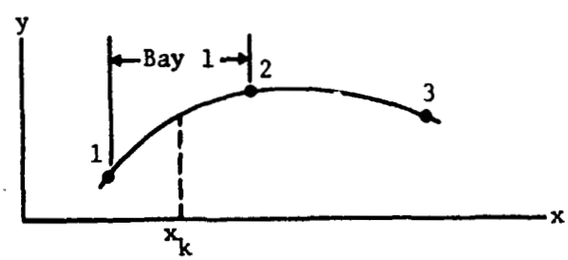
therefore

$$y(H) = [H^2 \quad H \quad 1]\,[\psi] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}. \qquad (3)$$

For a given set of points as shown below, a parabolic function is used to the left of point 1 and between points 1 and 2. Also a parabolic function is used to the right of the last point n and between points n-1 and n. Diparabolic functions are used between all other points.



Bay 1 (and to the left of point 1)



The derivative dy/dx at x_k is calculated using the chain rule for differentiation. That is,

$$\frac{dy}{dx} = \frac{dy}{dH}\,\frac{dH}{dx}$$

From Equation (3),

$$\frac{dy}{dH} = [2H \quad 1 \quad 0]\,[\psi] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

and from Equation (1a),

$$\frac{dH}{dx} = \frac{1}{x_2 - x_1}$$

therefore

$$\frac{dy}{dx} = \frac{1}{x_2 - x_1} [2H \quad 1 \quad 0] [\psi] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}. \tag{4}$$

Also, from Equation (1a),

$$H_k = (x_k - x_1)/(x_2 - x_1) \tag{4a}$$

$$H_1 = (x_1 - x_1)/(x_2 - x_1) = 0$$

$$H_2 = (x_2 - x_1)/(x_2 - x_1) = 1$$

$$H_3 = (x_3 - x_1)/(x_2 - x_1) = D. \tag{4b}$$

Using these expressions with Equations (2) and (4), the final result is obtained as

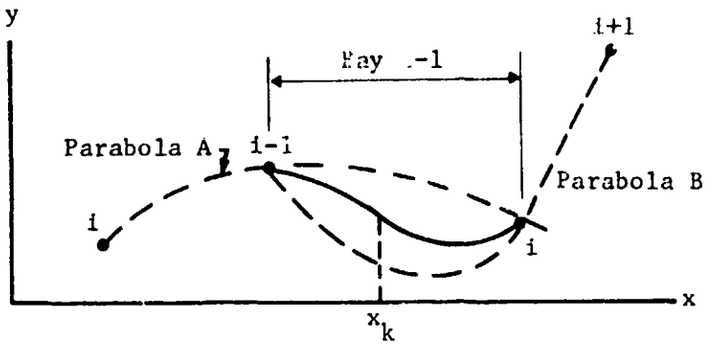
$$\left.\frac{dy}{dx}\right|_{x_k} = \frac{1}{x_2 - x_1} \left|2H_k \quad 1\right| \begin{bmatrix} 1/D & 1/(1-D) & -1/D(1-D) \\ -(1+D)/D & -D/(1-D) & 1/D(1-D) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}. \tag{4c}$$

The following table gives the correlation between the nomenclature of Equations (4a, b, c) and that used in the Fortran coding in the subroutine.

Equations (4a,b,c)	Fortran Coding*	
x_m	XA(m)	m = 1, 2, 3
y_m	YA(m) *	m = 1, 2, 3
H_k	H	
x_k	XZ(K)	
$\left.\frac{dy}{dx}\right\|_{x_k}$	Z(K) *	

*Subscript j, denoting different sets of points, has been omitted for clarity.

Interior Bay i-1



The derivative dy/dx at x_k is found as follows. A diparabolic function in bay i-1 of the above sketch is obtained as the weighted average of parabolas A and B. That is

$$y(H) = (1-H)\, y_A + H y_B \tag{5}$$

where

$$H = \frac{x - x_{i-1}}{x_i - x_{i-1}} \tag{6a}$$

as in either Equations (1a) or (1b).

For parabola A:
$$H_{i-2} = \left(x_{i-2} - x_{i-1}\right)/\left(x_i - x_{i-1}\right) = -1 \tag{6b}$$
$$H_{i-1} = \left(x_{i-2} - x_{i-1}\right)/\left(x_i - x_{i-1}\right) = 0$$
$$H_i = \left(x_i - x_{i-1}\right)/\left(x_i - x_{i-1}\right) = 1.$$

From Equations (2) and (3),

$$y_A(H) = \begin{bmatrix} H^2 & H & 1 \end{bmatrix} \left\{\psi_A\right\} \begin{bmatrix} y_{i-2} \\ v_{i-1} \\ y_i \end{bmatrix}$$

where

$$\left[\psi_A\right] = \begin{bmatrix} -1/C(1-C) & 1/C & 1/(1-C) \\ 1/C(1-C) & -(1+C)/C & -C/(1-C) \\ 0 & 1 & 0 \end{bmatrix}.$$

For parabola B: $H_{i-1} = \left(x_{i-1} - x_{i-1}\right)/\left(x_i - x_{i-1}\right) = 0$

$$H_i = \left(x_i - x_{i-1}\right)/\left(x_i - x_{i-1}\right) = 1$$

$$H_{i+1} = \left(x_{i+1} - x_{i-1}\right)/\left(x_i - x_{i-1}\right) = D. \qquad (6c)$$

From Equations (2) and (3),

$$y_B(H) = \begin{bmatrix} H^2 & H & 1 \end{bmatrix} \left[\psi_B\right] \begin{bmatrix} y_{i-1} \\ y_i \\ y_{i+1} \end{bmatrix}$$

where

$$\left[\psi_B\right] = \begin{bmatrix} 1/D & 1/(1-D) & -1/D(1-D) \\ -(1+D)/D & -D/(1-D) & 1/D(1-D) \\ 1 & 0 & 0 \end{bmatrix}.$$

Substituting these expressions for y_A and y_B into Equation (5) gives

$$y(H) = \begin{bmatrix} H^3 & H^2 & H & 1 \end{bmatrix} [\psi] \begin{bmatrix} y_{i-2} \\ y_{i-1} \\ y_i \\ y_{i+1} \end{bmatrix} \qquad (7)$$

where

$$[\psi] = \begin{bmatrix} \frac{1}{C(1-C)} & \frac{C-D}{CD} & \frac{D-C}{(1-C)(1-D)} & \frac{-1}{D(1-D)} \\ \frac{-2}{C(1-C)} & \frac{2D-C}{CD} & \frac{1-2D+C}{(1-C)(1-D)} & \frac{1}{D(1-D)} \\ \frac{1}{C(1-C)} & \frac{-(1+C)}{C} & \frac{-C}{1-C} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The derivative dy/dx at x_k is calculated using the chain rule for differentiation. That is,

$$\frac{dy}{dx} = \frac{dy}{dH}\frac{dH}{dx}.$$

From Equation (7),

$$\frac{dy}{dH} = [3H^2 \quad 2H \quad 1 \quad 0]\,[\psi] \begin{bmatrix} y_{i-2} \\ y_{i-1} \\ y_i \\ y_{i+1} \end{bmatrix}$$

and from Equation (6a)

$$\frac{dH}{dx} = \frac{1}{x_i - x_{i-1}}.$$

∴At $x = x_k$

$$\left.\frac{dy}{dx}\right|_{x_k} = \frac{1}{x_i - x_{i-1}} \begin{bmatrix} 3H_k^2 & 2H_k & 1 & 0 \end{bmatrix} [\psi] \begin{bmatrix} y_{i-2} \\ y_{i-1} \\ y_i \\ y_{i+1} \end{bmatrix} \qquad (8)$$

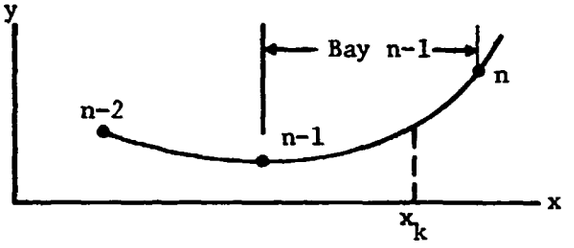
where $[\psi]$ is given above.

The following table gives the correlation between the nomenclature of Equations (6a, b, c and 8) and that used in the Fortran coding in the subroutine.

Equations (6a,b,c and 8)	Fortran Coding*	
x_{i-m}	XA(I-m)	m = 2, 1, 0, -1
y_{i-m}	YA(I-m) *	m = 2, 1, 0, -1
H_k	H	
x_k	XZ(K)	
$\left.\frac{dy}{dx}\right\|_{x_k}$	Z(K) *	

*Subscript j, denoting different sets of points, has been omitted for clarity.

Bay n-1 (and to the right of point n)



The derivative dy/dx at x_k is calculated using the chain rule for differentiation. That is,

$$\frac{dy}{dx} = \frac{dy}{dH}\frac{dH}{dx}.$$

From Equation (3)

$$\frac{dy}{dH} = [2H \quad 1 \quad 0]\,[\psi]\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

and from Equation (1b),

$$\frac{dH}{dx} = \frac{1}{x_n - x_{n-1}}$$

therefore

$$\frac{dy}{dx} = \frac{1}{x_n - x_{n-1}} [2H \quad 1 \quad 0] [C] \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}. \tag{9}$$

Also, from Equation (1b),

$$H_k = (x_k - x_{n-1})/(x_n - x_{n-1}) \tag{9a}$$

$$H_{n-2} = (x_{n-2} - x_{n-1})/(x_n - x_{n-1}) = C \tag{9b}$$

$$H_{n-1} = (x_{n-1} - x_{n-1})/(x_n - x_{n-1}) = 0$$

$$H_n = (x_n - x_{n-1})/(x_n - x_{n-1}) = 1.$$

Using these expressions with Equations (2) and (9), the final result is obtained as

$$\left.\frac{dy}{dx}\right|_{x_k} = \frac{1}{x_n - x_{n-1}} \begin{bmatrix} 2H_k & 1 \end{bmatrix} \begin{bmatrix} -1/C(1-C) & 1/C & 1/(1-C) \\ 1/C(1-C) & -(1+C)/C & -C/(1-C) \end{bmatrix} \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}. \tag{9c}$$

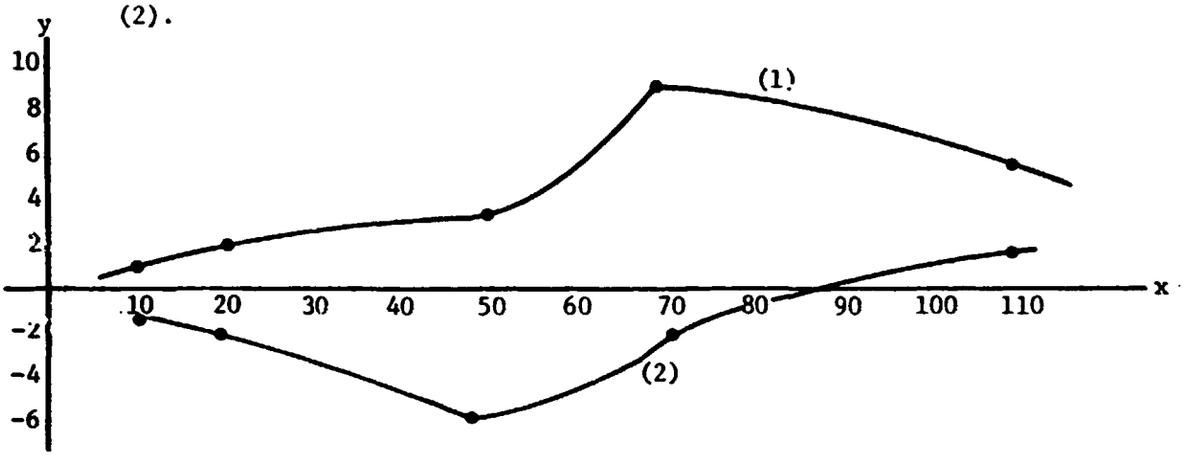
The following table gives the correlation between the nomenclature of Equations (9a, b, c) and that used in the Fortran coding in the subroutine.

Equations (9a,b,c)	Fortran Coding*	
x_{n-m}	XA(NXA-m)	m = 0, 1, 2
y_{n-m}	YA(NXA-m) *	m = 0, 1, 2
H_k	H	
x_k	XZ(K)	
$\left.\frac{dy}{dx}\right\|_{x_k}$	Z(K) *	

*Subscript j, denoting different sets of points, has been omitted for clarity.

EXAMPLE

Consider the following two sets of points denoted by (1) and (2).



The coordinates of the points are given by

$$\{XA\} = \begin{bmatrix} 10. \\ 20. \\ 50. \\ 70. \\ 110. \end{bmatrix} \quad \text{and} \quad [YA] = \begin{bmatrix} 1. & -1. \\ 2. & -2. \\ 3. & -6. \\ 9. & -2. \\ 5. & 2. \end{bmatrix}.$$

{XA} gives the x coordinates of the points in both sets (1) and (2). Column 1 of [YA] gives the y coordinates of the points in set (1) and column 2 of [YA] gives the y coordinates of the points in set (2). Derivatives are wanted at x = 5., x = 100., and x = 65., that is, at

$$\{XZ\} = \begin{bmatrix} 5. \\ 100. \\ 65. \end{bmatrix}.$$

At x = 5., derivatives of dy/dx = 0.1333 and dy/dx = -0.08333 are calculated from columns 1 and 2 of [YA], respectively, using rows 1, 2, and 3 of {XA} and [YA]. At x = 100., derivatives of dy/dx = -0.2333 and dy/dx = 0.0666 are calculated from columns 1 and 2 of [YA], respectively, using rows 3, 4, and 5 of {XA} and [YA]. At x = 65., derivatives of dy/dx = 0.3083 and dy/dx = 0.2354 are calculated from columns 1 and 2 of [YA], respectively, using

rows 2, 3, 4, and 5 of {XA} and [YA]. The result is

$$[Z] = \begin{bmatrix} 0.1333 & -0.08333 \\ -0.2333 & 0.0666 \\ 0.3083 & 0.2354 \end{bmatrix}.$$

To relate this example problem to a practical problem, consider {XA} to be the collocation points (panel points) of a vehicle and [YA] to be the modal displacements for two modes. Gyros are to be placed at stations x = 5., 100., and 65. (i.e., {XZ}). The modal slopes ([Z]) are to be found at these stations.

REFERENCES

Griffin, J.A.: "A Diparabolic Method of Four-Point Interpolation." *Journal of the Aeronautical Sciences*, Vol. 28, No.2, Readers' Forum, February 1961.

Subroutine DISA removes (disassembles) a matrix [Z] from matrix [A] starting at a designed row, column location (IRA, JCA, respectively) in [A]. The [Z] matrix must be within the row, column limits of [A]. In subscript notation,

$$z_{ij} = a_{k\ell} \qquad \begin{pmatrix} i = 1, \text{NRZ} \\ j = 1, \text{NCZ} \end{pmatrix}$$

where

$$k = i + \text{IRA} - 1$$

$$\ell = j + \text{JCA} - 1$$

NRZ is the number of rows of [Z], and

NCZ is the number of columns of [Z].

EXAMPLE

Consider a matrix defined as

$$[A]_{3x4} = \begin{bmatrix} 1. & 0. & 0. & 0. \\ 3. & 4. & 5. & 0. \\ 6. & 7. & 8. & 0. \end{bmatrix}.$$

Matrix $[Z]_{2x3}$ is to be obtained from [A] starting at the 2,1 location (IRA = 2, JCA = 1) of [A]. The result of this operation will be

$$[Z]_{2x3} = \begin{bmatrix} 3. & 4. & 5. \\ 6. & 7. & 8. \end{bmatrix}.$$

The matrix [A] remains as originally defined.

Subroutine EIGN1 calculates the eigenvalues (proper values, characteristic roots, latent roots) and eigenvectors (proper vectors, characteristic vectors, latent vectors) of a real symmetric matrix using a method of C. G. J. Jacobi. The eigenvalue problem can be expressed as

$$[A]\ [\Phi] = [\Phi]\ [\ \lambda\ ] \qquad (1)$$

where

[A] is a real symmetric matrix of order N,

$[\Phi]$ is a matrix whose columns are the eigenvectors of Equation (1),

$[\ \lambda\ ]$ is a diagonal matrix whose elements are the corresponding eigenvalues.

This method diagonalizes matrix [A] by successive plane rotations and leads simultaneously to all eigenvalues and eigenvectors. Theoretically, an infinite number of plane rotations are necessary to produce the diagonal form with all off-diagonal elements equal to zero. Practically the number of plane rotations is limited to a finite number by stopping the process when the off-diagonal elements are less than a prescribed value. This prescribed value (denoted as FOD in the subroutine) may be selected and input by the analyst or at the analyst's option calculated in the subroutine as a constant times the trace (sum of the diagonal elements) of [A].

A threshold version is used. That is, each off-diagonal element of [A] is compared in regular sequence with a threshold value and a plane rotation performed only if its magnitude exceeds the threshold value. This is done to give faster convergence to the diagonal form. Since [A] is symmetric only the upper half need be examined. The threshold value is lowered whenever there are no remaining off-diagonal elements with magnitudes larger than the threshold. The lowering of the threshold is continued until the threshold is less than a prescribed value (FOD).

The initial threshold value in the subroutine is obtained by dividing the maximum off-diagonal element of [A] by 10. Subsequent reductions in the threshold value are accomplished by dividing the threshold value by 10 also.

The threshold version is the fastest version of Jacobi's method (Reference 2).

An important feature of the Jacobi method is that equal eigenvalues and the corresponding eigenvectors are easily calculated with no change in the algorithm.

DESCRIPTION OF TECHNIQUE

Assume that an N^{th} order matrix [T] exists that is orthonormal that is, $\left([T]^T\ [T] = [I]\right)$ such that

$$[T]^T\ [A]\ [T] = \lceil\ D\ \rfloor, \tag{2}$$

where $\lceil\ D\ \rfloor$ is a diagonal matrix and [A] is the original matrix. Premultiply Equation (2) by [T] to give

$$[A]\ [T] = [T]\ \lceil\ D\ \rfloor. \tag{3}$$

Comparing Equations (1) and (3) it is seen that if the matrix [T] can be found which transforms a real symmetric matrix [A] into a diagonal matrix $\lceil\ D\ \rfloor$, then the i^{th} diagonal element of $\lceil\ D\ \rfloor$ is the i^{th} eigenvalue of [A], and the i^{th} column of [T] is the i^{th} eigenvector of [A].

The orthonormal matrix [T] is built in stepwise fashion using elementary orthonormal transformations $[T]_k$ to annihilate, in turn, selected off-diagonal elements of [A]. That is, starting with the given matrix [A], operate as follows,

$$[A] = [A]_0$$

$$[T]_1^T\ [A]_0\ [T]_1 = [A]_1$$

$$[T]_2^T\ [A]_1\ [T]_2 = [A]_2$$

$$\cdots\cdots\cdots$$

$$[T]_k^T\ [A]_{k-1}\ [T]_k = [A]_k$$

so that

$$[T]_k^T \cdots [T]_2^T [T]_1^T [A] [T]_1 [T]_2 \cdots [T]_k = [A]_k . \tag{4}$$

Comparing Equation (4) with Equation (2), the $[T]_k$ must be chosen so that the triple matrix product result $[A]_k$ converges to $[\ D\ ] = [\ \lambda\ ]$ and the continued product $[T]_1\ [T]_2 \cdots [T]_k$ converges to $[T] = [\Phi]$. The most important property of $[T]_k$ is that each triple matrix product (rotation) with $[A]_{k-1}$ in Equation (4) causes a particular off-diagonal element in $[A]_{k-1}$ to vanish. As mentioned before, an off-diagonal element of $[A]_{k-1}$ is picked for elimination if it exceeds a threshold value and is referred to as the pivot element. If the p, q element of $[A]_{k-1}$ meets the criteria, then the transformation matrix used is

$$[T]_k = \begin{array}{c} \\ 1 \\ \\ \\ p \\ \\ \\ q \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & & p & & q & \end{array} \\ \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & C & & -S & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & & S & & C & \\ & & & & & & \\ & & & & & & \end{bmatrix} \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \tag{5}$$

$$\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{bmatrix} & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & 1 \end{bmatrix}$$

which is orthonormal. For simpler notation, $C = \cos\theta$, and $S = \sin\theta$ has been used.

The value of θ is calculated as shown in the following example. A system of order 5 with pivot location p = 2, q = 4 is used, but the results may be applied to any size system with any pivot location p, q.

$$[A]_k = [T]_k^T [A]_{k-1} [T]_k$$

$$= [T]_k^T \begin{bmatrix} a_{11} & a_{1p} & a_{13} & a_{1q} & a_{15} \\ a_{p1} & a_{pp} & a_{p3} & a_{pq} & a_{p5} \\ a_{31} & a_{3p} & a_{33} & a_{3q} & a_{35} \\ a_{q1} & a_{qp} & a_{q3} & a_{qq} & a_{q5} \\ a_{51} & a_{5p} & a_{53} & a_{5q} & a_{55} \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & C & & -S & \\ & & 1 & & \\ & S & & C & \\ & & & & 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{1p}C + a_{1q}S & a_{13} & -a_{1p}S + a_{1q}C & a_{15} \\ a_{p1}C + a_{q1}S & a_{pp}C^2 + 2a_{pq}CS + a_{qq}S^2 & a_{p3}C + a_{q3}S & (a_{qq} - a_{pp})CS + a_{pq}(C^2 - S^2) & a_{p5}C + a_{q5}S \\ a_{31} & a_{3p}C + a_{3q}S & a_{33} & -a_{3p}S + a_{3q}C & a_{35} \\ -a_{p1}S + a_{q1}C & (a_{qq} - a_{pp})CS + a_{pq}(C^2 - S^2) & -a_{p3}S + a_{q3}C & a_{pp}S^2 - 2a_{pq}CS + a_{qq}C^2 & -a_{p5}S + a_{q5}C \\ a_{51} & a_{5p}C + a_{5q}S & a_{53} & -a_{5p}S + a_{5q}C & a_{55} \end{bmatrix} \quad (6)$$

where use is made of the symmetry property $a_{pq} = a_{qp}$. As can be seen, only the p, q rows and columns of $[A]_{k-1}$ are altered in calculating $[A]_k$.

Setting the pivot element p, q of $[A]_k$ to zero gives

$$a_{pq}(\cos^2\theta - \sin^2\theta) = (a_{pp} - a_{qq}) \sin\theta \cos\theta$$

from which

$$\tan\theta = \frac{2a_{pq}}{(a_{pp} - a_{qq}) \pm \sqrt{(a_{pp} - a_{qq})^2 + 4a_{pq}^2}}. \quad (7)$$

The sign used with the radical is the same as the sign of $(a_{pp} - a_{qq})$. This is done to avoid small differences of large numbers.

Using trigonometric identities gives

$$\cos\theta = 1/\sqrt{1 + \tan^2\theta} \tag{8}$$

and

$$\sin\theta = \tan\theta \cos\theta \tag{9}$$

as the values to use in Equation (6) to set $a_{pq} = 0$.

It should be obvious that even though the pivot element has been reduced to zero by one rotation, it may take on a non-zero value in some later rotation. A rotation gives a step toward the diagonal form because the sum of the squares of the off-diagonal elements of $[A]_k$ is less than that of $[A]_{k-1}$ by the amount $2a_{pq}^2$. The importance of the threshold version is re-emphasized in that small values (relatively) of the off-diagonal elements of [A] are not used for pivot elements. Proof of convergence is in the literature (References 2 and 3) and will not be given here.

As mentioned before, the continued product $[T]_1 [T]_2 \ldots [T]_k$ converges to the eigenvectors $[\Phi]$. Thus, $[\Phi]_k$ would be calculated as

$$[\Phi]_k = [\Phi]_{k-1} [T]_k$$

$$= \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} & \phi_{45} \\ \phi_{51} & \phi_{52} & \phi_{53} & \phi_{54} & \phi_{55} \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & C & & -S & \\ & & 1 & & \\ & S & & C & \\ & & & & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \phi_{11} & \phi_{12}C + \phi_{14}S & \phi_{13} & -\phi_{12}S + \phi_{14}C & \phi_{15} \\ \phi_{21} & \phi_{22}C + \phi_{24}S & \phi_{23} & -\phi_{22}S + \phi_{24}C & \phi_{25} \\ \phi_{31} & \phi_{32}C + \phi_{34}S & \phi_{33} & -\phi_{32}S + \phi_{34}C & \phi_{35} \\ \phi_{41} & \phi_{42}C + \phi_{44}S & \phi_{43} & -\phi_{42}S + \phi_{44}C & \phi_{45} \\ \phi_{51} & \phi_{52}C + \phi_{54}S & \phi_{53} & -\phi_{52}S + \phi_{54}C & \phi_{55} \end{bmatrix} \tag{10}$$

for our previous example of a system of order 5 and pivot location p = 2, q = 4. As can be seen, only the p, q columns of $[\Phi]_{R-1}$ are altered in calculating $[\Phi]_k$.

The resulting eigenvalues and eigenvectors can be checked as follows. Post multiplying Equation (1) by $[\Phi]^T$ gives

$$[A] = [\Phi] [\lambda] [\Phi]^T \tag{11}$$

because the eigenvectors are orthonormal. Therefore, to check the results, perform the triple matrix product $[\Phi] [\lambda] [\Phi]^T$ and compare with the original [A]. This operation is not performed in the subroutine.

EXAMPLE

Consider $[A] = \begin{bmatrix} 1.5 & -1/\sqrt{2} & -.5 \\ & 1.0 & -1/\sqrt{2} \\ (\text{sym}) & & 1.5 \end{bmatrix}$.

First pivot: p = 1, q = 2

$$a_{pq} = a_{12} = -1\sqrt{2}$$

$$a_{pp} = a_{11} = 1.5$$

$$a_{qq} = a_{22} = 1.0$$

From Equations (7), (8), and (9)

$$\tan \theta = -1/\sqrt{2}$$

$$\cos \theta = \sqrt{2/3}$$

$$\sin \theta = -1/\sqrt{3}$$

From Equations (6) and (10), the results of the first pivot are

$$[A]_1 = \begin{bmatrix} 2. & 0. & 0. \\ & .5 & -\sqrt{3/2} \\ (\text{sym}) & & 1.5 \end{bmatrix}$$

and $$[\Phi]_1 = \begin{bmatrix} \sqrt{2/3} & 1/\sqrt{3} & 0. \\ -1/\sqrt{3} & \sqrt{2/3} & 0. \\ 0. & 0. & 1. \end{bmatrix}.$$

Second pivot: $p = 2$, $q = 3$

$$a_{pq} = a_{23} = -\sqrt{3}/2$$

$$a_{pp} = a_{22} = .5$$

$$a_{qq} = a_{33} = 1.5$$

From Equations (7), (8), and (9)

$$\tan \theta = 1/\sqrt{3}$$

$$\cos \theta = \sqrt{3}/2$$

$$\sin \theta = .5$$

From Equations (6) and (10), the results of the second pivot are

$$[A]_2 = \begin{bmatrix} 2. & 0. & 0. \\ & 0. & 0. \\ (\text{sym}) & & 2. \end{bmatrix}$$

and

$$[\Phi]_2 = \begin{bmatrix} \sqrt{2/3} & .5 & -\frac{1}{2\sqrt{3}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0. & .5 & \frac{\sqrt{3}}{2} \end{bmatrix}.$$

The diagonal elements of the $[A]_2$ are the eigenvalues and the volumns of $[\Phi]_2$ are the corresponding eigenvectors.

REFERENCES

1. Crandall, S. H.: *Engineering Analysis.* McGraw-Hill, New York, 1956.

2. Ralston, A., Wilf, H. S.: *Mathematical Methods for Digital Computers.* John Wiley & Sons, New York 1966.

3. Wilkinson, J. H.: *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, 1965.

Subroutine EIGN1A calculates the eigenvalues (proper values, characteristic roots, latent roots) and eigenvectors (proper vectors, characteristic vectors, latent vectors) of a real symmetric matrix using a method of C. G. J. Jacobi. This subroutine is a modification of Subroutine EIGN1 and uses a convergence tolerance on λ instead of a final off-diagonal value to terminate the solution. The eigenvalue problem can be expressed as

$$[A]\ [\phi] = [\phi]\ [\lambda] \qquad [1]$$

where

[A] = a real symmetric matrix of order N;

$[\phi]$ = a matrix whose columns are the eigenvectors of Eq [1];

$[\lambda]$ = a diagonal matrix whose elements are the corresponding eigenvalues.

This method diagonalizes matrix [A] by successive plane rotations and leads simultaneously to all eigenvalues and eigenvectors. Theoretically, an infinite number of plane rotations are necessary to produce the diagonal form with all off-diagonal elements equal to zero. Practically, the number of plane rotations is limited to a finite number by stopping the process when $[\lambda_i(k) - \lambda_i(k-1]/\lambda_i(k)$, i = 1, N is less than a prescribed value (k refers to plane rotation number). This prescribed value (denoted as CTVAL in the subroutine) may be selected and input by the analyst or at the analyst's option determined in the subroutine.

A threshold version is used, that is, each off-diagonal element of [A] is compared in regular sequence with a threshold value and a plane rotation performed only if its magnitude exceeds the threshold value. This is done to give faster convergence to the diagonal form. Because [A] is symmetric, only the upper half needs be examined. The threshold value is lowered whenever there are no remaining off-diagonal elements with magnitudes larger than the threshold.

The initial threshold value in the subroutine is obtained by dividing the maximum off-diagonal element of [A] by 10. Subsequent reductions in the threshold value are accomplished by dividing the threshold value by 10 also,

The threshold version is the fastest version of Jacobi's method (Ref 2).

An important feature of the Jacobi method is that equal eigenvalues and the corresponding eigenvectors are easily calculated with no change in the algorithm.

DESCRIPTION OF TECHNIQUE

Assume that an N^{th} order matrix [T] exists that in orthonormal, that is, ($[T]^T$ [T] = [I]) such that

$$[T]^T [A] [T] = [D], \qquad [2]$$

where [D] is a diagonal matrix and [A] is the original matrix. Premultiply Eq [2] by [T] to give

$$[A] [T] = [T] [D]. \qquad [3]$$

Comparing EQ [1] and [3], it is seen that if matrix [T] can be found which transforms a real symmetric matrix [A] into a diagonal matrix [D], then the i^{th} diagonal element of [D] is the i^{th} eigenvalue of [A], and the i^{th} column of [T] is the i^{th} eigenvector of [A].

The orthonormal matrix [T] is built in stepwise fashion using elementary orthonormal transformations $[T]_k$ to annihilate, in turn, selected off-diagonal elements of [A]; that is, starting with the given matrix [A], operate as follows,

$$[A] = [A]_0$$

$$[T]_1^T [A]_0 [T]_1 = [A]_1$$

$$[T]_2^T [A]_1 [T]_2 = [A]_2$$

. . . . . .

$$[T]_k^T [A]_{k-1} [T]_k = [A]_k$$

so that

$$[T]_k^T \ldots [T]_2^T [T]_1^T [A] [T]_1 [T]_2 \ldots [T]_k = [A]_k. \qquad [4]$$

Comparing Eq [4] with Eq [2], $[T]_k$ must be chosen so that the triple matrix product result, $[A]_k$ converges to [D] = [λ] and the continued product $[T]_1 [T]_2 \ldots [T]_k$ converges to [T] = [Φ]. The most important property of $[T]_k$ is that each triple matrix product (rotation) with $[A]_{k-1}$ in Eq [4] causes a particular off-diagonal element in $[A]_{k-1}$ to vanish. As mentioned before, an off-diagonal

element of $[A]_{k-1}$ is picked for elimination if it exceeds a threshold value and is referred to as the pivot element. If the p, q element of $[A]_{k-1}$ meets the criteria, then the transformation matrix used is

$$[T]_k = \begin{array}{c} \\ 1 \\ \\ p \\ \\ q \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & & p & & q & \\ \end{array} \\ \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & C & & -S & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & S & & C & \\ & & & & & & 1 \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{bmatrix} \end{array} \qquad [5]$$

which is orthonormal. For simpler notation, C≡ cos θ, and S≡ sin θ has been used.

The value of θ is calculated as shown in the following example. A system of order 5 with pivot location p = 2, q = 4 is used, but the results may be applied to any size system with any pivot location p, q.

$$[A]_k = [T]_k^T [A]_{k-1} [T]_k$$

$$= [T]_k^T \begin{bmatrix} a_{11} & a_{1p} & a_{13} & a_{1q} & a_{15} \\ a_{p1} & a_{pp} & a_{p3} & a_{pq} & a_{p5} \\ a_{31} & a_{3p} & a_{33} & a_{3q} & a_{35} \\ a_{q1} & a_{qp} & a_{q3} & a_{qq} & a_{q5} \\ a_{51} & a_{5p} & a_{53} & a_{5q} & a_{55} \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & C & & -S & \\ & & 1 & & \\ & S & & C & \\ & & & & 1 \end{bmatrix} \qquad [6]$$

$$= \begin{bmatrix} a_{11} & a_{1p}C + a_{1q}S & a_{13} & -a_{1p}S + a_{1q}C & a_{15} \\ a_{p1}C + a_{q1}S & a_{pp}C^2 + 2a_{pq}CS + a_{qq}S^2 & a_{p3}C + a_{q3}S & (a_{qq} - a_{pp})CS + a_{pq}(C^2 - S^2) & a_{p5}C + a_{q5}S \\ a_{31} & a_{3p}C + a_{3q}S & a_{33} & -a_{3p}S + a_{3q}C & a_{35} \\ -a_{p1}S + a_{q1}C & (a_{qq} - a_{pp})CS + a_{pq}(C^2 - S^2) & -a_{p3}S + a_{q3}C & a_{pp}S^2 - 2a_{pq}CS + a_{qq}C^2 & -a_{p5}S + a_{q5}C \\ a_{51} & a_{5p}C + a_{5q}S & a_{53} & -a_{5p}S + a_{5q}C & a_{55} \end{bmatrix}$$

where use is made of the symmetry property $a_{pq} = a_{qp}$. As can be seen, only the p, q rows and columns of $[A]_{k-1}$ are altered in calculating $[A]_k$.

Setting the pivot element p, q of $[A]_k$ to zero gives

$$a_{pq} (\cos^2 \theta - \sin^2 \theta) = (a_{pp} - a_{qq}) \sin \theta \cos \theta$$

from which

$$\tan \theta = \frac{2a_{pq}}{(a_{pp} - a_{qq}) \pm \sqrt{(a_{pp} - a_{qq})^2 + 4a_{pq}^2}} \quad . \qquad [7]$$

The sign used with the radical is the same as the sign of $(a_{pp} - a_{qq})$. This is done to avoid small differences of large numbers.

Using trigonometric identities gives

$$\cos \theta = 1/\sqrt{1 + \tan^2 \theta} \qquad [8]$$

and

$$\sin \theta = \tan \theta \cos \theta \qquad [9]$$

as the values to use in Eq [6] to set $a_{pq} = 0$.

It should be obvious that even though the pivot element has been reduced to zero by one rotation, it may take on a nonzero value in some later rotation. A rotation gives a step toward the diagonal form because the sum of the squares of the off-diagonal elements of $[A]_k$ is less than that of $[A]_{k-1}$ by the amount $2a_{pq}^2$. The importance of the threshold version is reemphasized in that small values (relatively) of the off-diagonal elements of [A] are not used for pivot elements. Proof of convergence is in the literature (References 2 and 3) and will not be given here.

As mentioned before, the continued product $[T]_1 [T]_2 \ldots [T]_k$ converges to the eigenvectors $[\Phi]$. Thus, $[\Phi]_k$ would be calculated as

$$[\Phi]_k = [\Phi]_{k-1}\,[T]_k$$

$$= \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} & \phi_{45} \\ \phi_{51} & \phi_{52} & \phi_{53} & \phi_{54} & \phi_{55} \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & C & & -S & \\ & & 1 & & \\ & S & & C & \\ & & & & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \phi_{11} & \phi_{12}C + \phi_{14}S & \phi_{13} & -\phi_{12}S + \phi_{14}C & \phi_{15} \\ \phi_{21} & \phi_{22}C + \phi_{24}S & \phi_{23} & -\phi_{22}S + \phi_{24}C & \phi_{25} \\ \phi_{31} & \phi_{32}C + \phi_{34}S & \phi_{33} & -\phi_{32}S + \phi_{34}C & \phi_{35} \\ \phi_{41} & \phi_{42}C + \phi_{44}S & \phi_{43} & -\phi_{42}S + \phi_{44}C & \phi_{45} \\ \phi_{51} & \phi_{52}C + \phi_{54}S & \phi_{53} & -\phi_{52}S + \phi_{54}C & \phi_{55} \end{bmatrix} \qquad [10]$$

for our previous example of a system of order 5 and pivot location p = 2, q = 4. As can be seen, only the p, q columns of $[\Phi]_{R-1}$ are altered in calculating $[\Phi]_k$.

The resulting eigenvalues and eigenvectors can be checked as follows. Postmultiplying Eq [1] by $[\Phi]^T$ gives

$$[A] = [\Phi]\,[\lambda]\,[\Phi]^T \qquad [11]$$

because the eigenvectors are orthonormal. Therefore, to check the results, perform the triple matrix product $[\Phi]\,[\lambda]\,[\Phi]^T$ and compare with the original [A]. This operation is not performed in the subroutine.

Example

Consider $[A] = \begin{bmatrix} 1.5 & -1/\sqrt{2} & -0.5 \\ & 1.0 & -1/\sqrt{2} \\ \text{(sym)} & & 1.5 \end{bmatrix}$.

First pivot: $p = 1,\ q = 2$

$$a_{pq} = a_{12} = -1\sqrt{2}$$

$$a_{pp} = a_{11} = 1.5$$

$$a_{qq} = a_{22} = 1.0$$

From Eq [7], [8], and [9]

$$\tan\theta = -1/\sqrt{2}$$

$$\cos\theta = \sqrt{2/3}$$

$$\sin\theta = -1/\sqrt{3}$$

From Eq [6] and [10], the results of the first pivot are

$$[A]_1 = \begin{bmatrix} 2. & 0. & 0. \\ & 0.5 & -\sqrt{3/2} \\ (\text{sym}) & & 1.5 \end{bmatrix}$$

and

$$[\phi]_1 = \begin{bmatrix} \sqrt{2/3} & 1/\sqrt{3} & 0. \\ -1/\sqrt{3} & \sqrt{2/3} & 0. \\ 0. & 0. & 1. \end{bmatrix} .$$

Second pivot: $p = 2,\ ] = 3$

$$a_{pq} = a_{23} = -\sqrt{3/2}$$

$$a_{pp} = a_{22} = 0.5$$

$$a_{qq} = a_{33} = 1.5$$

From Eq [7], [8], and [9]

$$\tan\theta = 1/\sqrt{3}$$

$$\cos\theta = \sqrt{3/2}$$

$$\sin\theta = 0.5$$

From Eq [6] and [10], the results of the second pivot are

$$[A]_2 = \begin{bmatrix} 2. & 0. & 0. \\ & 0. & 0. \\ (\text{sym}) & & 2. \end{bmatrix}$$

and

$$[\Phi]_2 = \begin{bmatrix} \sqrt{2/3} & 0.5 & -\frac{1}{2\sqrt{3}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0. & 0.5 & \frac{\sqrt{3}}{2} \end{bmatrix} .$$

The diagonal elements of the $[A]_2$ are the eigenvalues and the columns of $[\Phi]_2$ are the corresponding eigenvectors.

REFERENCES

1. Crandall, S. H.: *Engineering Analysis*. McGraw-Hill, New York, 1956.

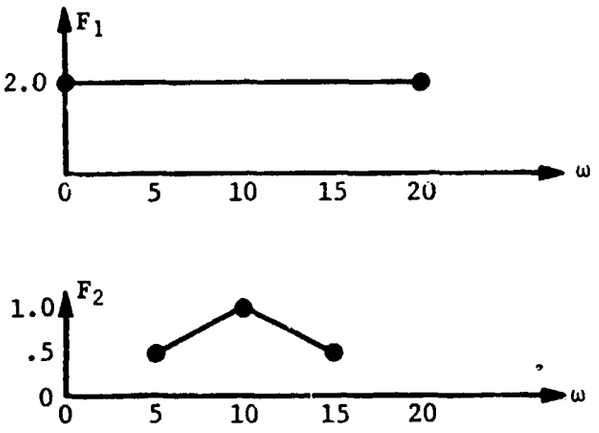
2. Ralston, A., Wilf, H. S.: *Mathematical Methods for Digital Computers*, John Wiley & Sons, New York 1966.

3. Wilkinson, J. H.: *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.

Subroutine FR1 calculates the frequency response $\{X(\omega)\}$ from the following differential equation expressed in the frequency domain:

$$(-\omega^2[A] + i\omega[B] + [C])\ \{X(\omega)\} = [D]\{F(\omega)\}. \tag{1}$$

Matrices [A], [B], [C], and [D] are input directly to this subroutine along with a set of values of ω. For a structural problem, [A] is the mass matrix, [B] the damping matrix, [C] the stiffness matrix, and [D] is the transpose of the vibration mode shapes or a unity matrix depending on whether the equations are a modal or discrete representation of the structure. The force $\{F(\omega)\}$, assumed real, is calculated internally in the subroutine using linear interpolation with [TABW] and [TABF] which are both input to the subroutine. As an illustration of the use of [TABW] and [TABF] consider the following example:



The table (A table is defined in this report as a matrix that may have incomplete column data in some rows.) giving the independent variable coordinate ω is

$$[TABW] = \begin{bmatrix} 0. & 20. & \\ 5. & 10. & 15. \end{bmatrix}.$$

The table giving the corresponding coordinates of the dependent variable F is

$$[TABF] = \begin{bmatrix} 2. & 2. & \\ .5 & 1. & .5 \end{bmatrix}.$$

The following values for $\{F(\omega)\}$ will be obtained by linear interpolation at $\omega = 7$.

$$\{F(\omega = 7)\} = \begin{bmatrix} 2. \\ .7 \end{bmatrix}.$$

DESCRIPTION OF TECHNIQUE

First, consider the inversion of a complex matrix. Let ([G] + i[H]) be the original matrix expressed as a complex matrix in terms of two real matrices [G] and [H]. The product of a matrix with its inverse (assumed to be $([\bar{G}] + i[\bar{H}])$) is, by definition, unity; thus

$$([\bar{G}] + i[\bar{H}])([G] + i[H]) = [\bar{G}][G] - [\bar{H}][H] + i([\bar{H}][G] + [\bar{G}][H]) \equiv [I] + i[0] \quad (2)$$

or, equating real and imaginary parts,

$$[\bar{G}][G] - [\bar{H}][H] = [I] \quad (2a)$$

$$[\bar{H}][G] + [\bar{G}][H] = [0]. \quad (2b)$$

Two approaches now exist for the solution. From Equation (2b),

$$\text{either } [\bar{H}] = -[\bar{G}][H][G]^{-1} \text{ (assumes [G] is nonsingular)}, \quad (3a)$$

$$\text{or} \quad [\bar{G}] = -[\bar{H}][G][H]^{-1} \text{ (assumes [H] is nonsingular)}. \quad (3b)$$

Inspection of Equation (1) shows that

$$[G] = -[C] - \omega^2[A]$$

$$\text{and } [H] = \omega[B].$$

Because many values of ω can make [G] singular (and also ill-conditioned such that the inverse is bad), it was decided to use Equation (3b) which then requires the damping matrix [B] to be nonsingular. Substituting Equation (3b) into (2a) yields

$$[\bar{H}] = -([G][H]^{-1}[G] + [H])^{-1} \quad (3c)$$

To eliminate having to do the triple matrix product $[G][H]^{-1}[G]$ for every value of ω, Equation (1) is modified so that $[H] = \omega[I]$. This is done by premultiplying by $[B]^{-1}$ so that

$$(-\omega^2[B]^{-1}[A] + i\omega[I] + [B]^{-1}[C])\{X(\omega)\} = [B]^{-1}[D]\{F(\omega)\}. \quad (1a)$$

Solving for $\{X(\omega)\}$ gives

$$\left\{X_{real}(\omega)\right\} = [\bar{G}][B]^{-1}[D]\{F(\omega)\} \tag{4a}$$

$$\text{and } \left\{X_{imag}(\omega)\right\} = [\bar{H}][B]^{-1}[D]\{F(\omega)\} \tag{4b}$$

where

$$[\bar{G}] = -\frac{1}{\omega}[\bar{H}][G] \tag{4c}$$

$$[\bar{H}] = -\omega([G]^2 + \omega^2[I])^{-1} \tag{4d}$$

$$\text{and } [G] = [B]^{-1}[C] - \omega^2[B]^{-1}[A]. \tag{4e}$$

Subroutine FRAE1 calculates frequency response additional equations. That is,

$$\{Z(\omega)\} = [A]\{X(\omega)\} \tag{1}$$

where $\{X(\omega)\}$ is the complex response previously calculated and written on tape in frequency response subroutine FR1. That tape is read in this subroutine FRAE1 to calculate $\{Z(\omega)\}$.

If $\{X(\omega)\}$ is the response calculated from a modal representation of the structure, then [A] would be the mode shapes so that Equation (1) would then give $\{Z(\omega)\}$ as the response of the discrete system. $\{Z(\omega)\}$ will also be complex. $\{Z(\omega)\}$ is printed for every ω value of $\{X(\omega)\}$ from tape NXTAPE and also written on tape NZTAPE for any subsequent use.

Subroutine INTAPE initializes a tape (disk is preferred, see writeup of Subroutine WTAPE) for the FORMA tape system by writing EOT (end of tape) at the beginning of the tape (disk). All FORMA tape subroutines recognize this EOT as being the end of written data. Each "new" tape (disk) must be initialized with this Subroutine INTAPE to make the tape (disk) compatible with the other FORMA tape subroutines (LTAPE, RTAPE, WTAPE, and UPDATE).

A "new" tape (disk) is defined as a tape (disk) for which it is desired to start writing matrix data at the front of the tape (disk). Thus, a "new" tape (disk) could be one with obsolete FORMA matrix data on it as well as one that has never been written on by the FORMA system.

As an example, pertinent statements from a program containing INTAPE could be:

```
      DATA  NIT,NOT/5,6/
1001  FORMAT (12A6)
      NRTAPE = 10
      READ (NIT,1001) IFINIT, TAPEID
      IF (IFINIT  .EQ. 6HINITIL) CALL INTAPE(NRTAPE,TAPEID)
          .
          .
          .
```

The input data (starting in card column 1) to this example program would be:

either INITILTXXXX, if the tape is to be initialized. (TXXXX represents the particular tape number used, e.g., T1234);

or NOINIT, if the tape is *not* to be initialized. The tape identification is not needed.

Subroutine INV1 inverts a matrix by the bordering method (Ref 1). In matrix notation, the inversion is represented by

$$[Z]_{N\times N} = [A]^{-1}_{N\times N}$$

where

[A] is the matrix to be inverted,

[Z] is the result, and

N is the size of the matrices.

Matrix [A] may be nonsymmetric. The determinant ratios of [A] are also calculated and printed. The determinant of [A] is the product of these determinant ratios.

An inversion check, [Z] [A], is calculated (The product should be a unity matrix.) and a summary of these results is printed.

REFERENCE

1. Bodewig, E.: *Matrix Calculus*, North Holland Publishing Company, Amsterdam, 1959.

Subroutine INV2 inverts a matrix by the rank annihilation method (Ref 1). In matrix notation, the inversion is represented by

$$[Z]_{N\times N} = [A]^{-1}_{N\times N}$$

where

[A] is the matrix to be inverted,

[Z] is the result, and

N is the size of the matrices.

Matrix [A] may be nonsymmetric.

An inversion check, [Z] [A], is calculated (The product should be a unity matrix.) and a summary of these results is printed.

REFERENCE

1. Ralston, A., and Wolf, H. S.: *Mathematical Methods for Digital Computers*. John Wiley & Sons, 1966.

Subroutine INV3 inverts a symmetric matrix using triangular decomposition and triangular inversion. In matrix notation, the inversion is represented by

$$[Z]_{N\times N} = [A]^{-1}_{N\times N} \tag{1}$$

where

[A] is the symmetric matrix to be inverted,

[Z] is the result (also symmetric), and

N is the size of the matrices.

In addition to being symmetric, matrix [A] should be positive definite. Because symmetry is assumed, only the upper half of [A] will be used to calculate [Z] which is also symmetric. The determinant ratios of [A] are also calculated and printed. The determinant of [A] is the product of these determinant ratios.

An inversion check, [Z] [A], is calculated (the product should be a unity matrix.) and a summary of these results is printed. Only the upper half of [A] is used to calculate [Z], but all of [A] (and [Z]) is used in the inversion check. Thus, errors in the inversion check can result from a nonsymmetric [A] as well as from a poor inversion.

DESCRIPTION OF TECHNIQUE

The first operation in the inversion of [A] is to decompose [A] (reference Subroutine DCOM1) into triangular factors, that is,

$$[A] = [U]^T [U] \tag{2}$$

where [U] is an upper triangular matrix. Then

$$\begin{aligned} [Z] &= [A]^{-1} \\ &= \left([U]^T [U]\right)^{-1} \\ &= [U]^{-1} \left([U]^T\right)^{-1} \\ &= [U]^{-1} \left([U]^{-1}\right)^T \end{aligned} \tag{3}$$

The inversion of the upper triangular matrix, [U], is calculated by Subroutine INV4.

Subroutine INV4 inverts an upper triangular matrix. In matrix notation the inversion is represented by

$$[Z]_{N\times N} = [A]^{-1}_{N\times N} \qquad (1)$$

where

[A] is the upper triangular matrix to be inverted,

[Z] is the result (also upper triangular), and

N is the size of the matrices.

An inversion check is not calculated.

DESCRIPTION OF TECHNIQUE

That [Z] is upper triangular if [A] is upper triangular is easily demonstrated by the following example. From [A] [Z] = [I],

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Preforming the matrix product and equating like elements gives

(3,1 element) $a_{33}\, z_{31} = 0$

Because $a_{33} \neq 0$, $\therefore z_{31} = 0$.

(3,2 element) $a_{33}\, z_{32} = 0$

Because $a_{33} \neq 0$ $\therefore z_{32} = 0$.

(2,1 element) $a_{22}\, z_{21} + a_{23}\, z_{31} = 0$

Because $z_{31} = 0$ and $a_{22} \neq 0$, $\therefore z_{21} = 0$.

The algorithm used to calculate [Z] is obtained by considering the equation [A] [Z] = [I] as shown below for size N = 4.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ 0 & z_{22} & z_{23} & z_{24} \\ 0 & 0 & z_{33} & z_{34} \\ 0 & 0 & 0 & z_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

By matrix multiplication

$$\begin{bmatrix} a_{11}\,z_{11} & a_{11}\,z_{12} + a_{12}\,z_{22} & a_{11}\,z_{13} + a_{12}\,z_{23} + a_{13}\,z_{33} & a_{11}\,z_{14} + a_{12}\,z_{24} + a_{13}\,z_{34} + a_{14}\,z_{44} \\ 0 & a_{22}\,z_{22} & a_{22}\,z_{23} + a_{23}\,z_{33} & a_{22}\,z_{24} + a_{23}\,z_{34} + a_{24}\,z_{44} \\ 0 & 0 & a_{33}\,z_{33} & a_{33}\,z_{34} + a_{34}\,z_{44} \\ 0 & 0 & 0 & a_{44}\,z_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equating elements of the matrices on the left and right side of the equal sign gives

$$z_{ii} = 1/a_{ii} \quad (i = 1,4)$$

$$z_{12} = -\,a_{12}\,z_{22}/a_{11}$$

$$= -\,z_{11}\,a_{12}\,z_{22}$$

$$z_{23} = -\,a_{23}\,z_{33}/a_{22}$$

$$= -\,z_{22}\,a_{23}\,z_{33}$$

$$z_{34} = -\,a_{34}\,z_{44}/a_{33}$$

$$= -\,z_{33}\,a_{34}\,z_{44}$$

$$z_{13} = -\,(a_{12}\,z_{23} + a_{13}\,z_{33})/a_{11}$$

$$= -\,(z_{11}\,a_{13} + z_{12}\,a_{23})\,z_{33}$$

$$z_{24} = -\,(a_{23}\,z_{34} + a_{24}\,z_{44})/a_{22}$$

$$= -\,(z_{22}\,a_{24} + z_{23}\,a_{34})\,z_{44}$$

$$z_{14} = -\,(a_{12}\,z_{24} + a_{13}\,z_{34} + a_{14}\,z_{44})/a_{11}$$

$$= -\,(z_{11}\,a_{14} + z_{12}\,a_{24} + z_{13}\,a_{34})\,z_{44}$$

From the above equations, the algorithm for the (i,j) element of [Z] for size N is given in general as

$$z_{ii} = 1/a_{ii} \qquad (i = 1,N)$$

$$z_{ij} = -z_{jj} \sum_{k=i}^{j-1} z_{ik}\, a_{kj} \qquad \begin{pmatrix} i = 1, N-1 \\ j > i \end{pmatrix}$$

Subroutine LTAPE lists the matrix headings (see Subroutine WTAPE and YWTAPE writeups) written on a FORMA tape (or disk). These matrix headings were written by Subroutines WTAPE and YWTAPE and consist of:

NO. = Matrix number on tape,

RUN NO. = Run number of problem when matrix was written on tape,

NAME = Matrix name;

NROWS = Number of rows of matrix;

NCOLS = Number of columns of matrix;

DATE = Date when matrix was written on tape;

NNZ = Number of nonzeros (used only in sparse FORMA when only nonzeros are used);

PARTITION = Partition number of sparse matrix.

Subroutine MASS1 takes (on option) distributed mass, distributed rotary inertia, and concentrated mass items of a beam and replaces them with a mass matrix. The elements of the mass matrix are representative inertial values of the beam at selected points on the beam. These elements are calculated by assuming a linear function between pairs of the selected points to describe the beam's lateral elastic deflection.

The x-stations of the selected points (panel points) are given in {PP}. These x-stations must be in increasing order.

The distributed mass, m(x), is assumed to be piecewise linear and is represented by straight line segments as shown in Figure 1. The x-stations of the end points for the line segments giving the distributed mass are independent of the panel point x-stations. However, the distributed mass must be within the panel point limits. The line segments representing the distributed mass may or may not be joined and may overlap. The distributed mass is defined in [DMASS]. Each row of [DMASS] represents one nonvertical line segment. The form of each row of [DMASS] is $[x_1\ x_2\ m_1\ m_2]$ where x_1, m_1 give the first end point and x_2, m_2 give the second end point of a line segment.

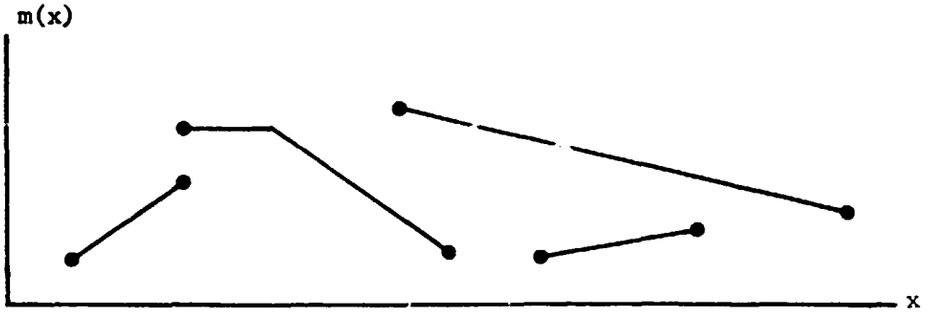


Figure 1 Distributed Mass

The distributed rotary inertia, r(x), is also assumed to be represented by straight line segments. All statements used above for distributed mass are applicable to distributed rotary inertia. The distributed rotary inertia is defined in [DRIN]. The form of each row of [DRIN] is $[x_1\ x_2\ r_1\ r_2]$. The end point stations x_1 and x_2 for the rotary inertia line segments are independent from the end point stations x_1 and x_2 for the mass line segments.

The concentrated mass items are defined in [CONC]. Each row of [CONC] represents one concentrated mass item and contains:

1) x_a, the station at which the item is attached to the beam;
2) M_c, the mass of the item;
3) x_{cg}, the center of gravity of the item; and
4) I_c^{cg}, the moment of inertia of the item about its own center of gravity.

These four elements are given in the form $\left[x_a \; M_c \; x_{cg} \; I_c^{cg}\right]$. The attachment station, x_a, of an item to the beam must be within the panel point limits.

The calculated representative inertial values at the selected panel points are placed in a mass matrix given by [Z] which is symmetric and tridiagonal. That is,

$$[Z]_{n\times n} = \begin{bmatrix} z_{1,1} & z_{1,2} & & & & & \\ z_{2,1} & z_{2,2} & z_{2,3} & & & & \\ & z_{3,2} & z_{3,3} & z_{3,4} & & & \\ & & z_{4,3} & z_{4,4} & z_{4,5} & & \\ & & & & \ddots & & \\ & & & & z_{n-1,n-2} & z_{n-1,n-1} & z_{n-1,n} \\ & & & & & z_{n,n-1} & z_{n,n} \end{bmatrix}$$

where $z_{i,j} = z_{j,i}$ and n is the number of panel points. The generalized coordinates associated with [Z] are lateral translation at each of the panel points.

The sign convention used in this paper to obtain the mass matrix [Z] is shown in Figure 2.

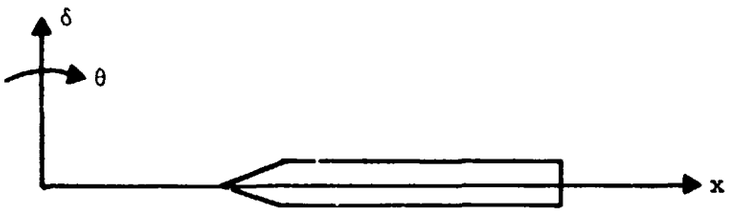


Figure 2 Sign Convention for This Paper

The beam mass matrix obtained in this paper is applicable for either the pitch, yaw, roll, or longitudinal plane with a change of variables. For the axis system shown in Figure 3, the following variables would be used:

This Paper (δ, x)	Axis System of Figure 3			
	Pitch (z, x)	Yaw (y, x)	Roll x, θ_x	Longitudinal (x)
x	x	x	x	x
δ	δ_z	δ_y	θ_x	δ_x
m(x)	m(x)	m(x)	$i_x(x)$	m(x)
r(x)	$r_y(x)$	$r_z(x)$	NA	NA
M_c	M_c	M_c	$I^o_{c,x}$	M_c
I^{cg}_c	$I^{cg}_{c,y}$	$I^{cg}_{c,z}$	NA	NA
NA = Not Applicable				

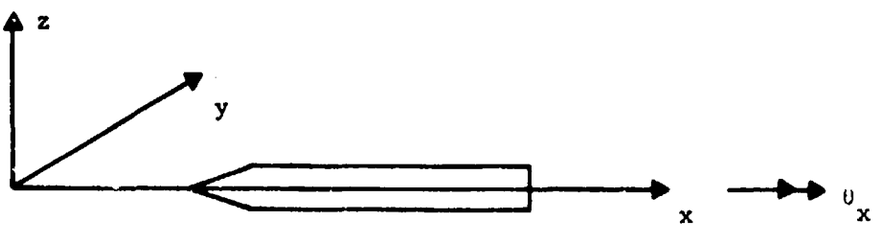


Figure 3 Beam Axis System (Right Hand)

The beam's total mass, center of gravity, and moment of inertia about the beam center of gravity are also calculated and printed.

DESCRIPTION OF TECHNIQUE

The replacement of distributed mass, distributed rotary inertia, and concentrated mass items of a beam by a mass matrix is obtained using a kinetic energy approach as follows.

For small deflections normal to the longitudinal body axis (x) of the beam shown in Figure 2, the kinetic energy is defined by

$$T = \tfrac{1}{2} \int_{x_S}^{x_E} \left[m(x)\ \dot{\delta}^2(x,t) + r(x)\ \dot{\theta}^2(x,t) \right] dx + \tfrac{1}{2} \sum_c \left[M_c\ \dot{\Delta}_o^2(t) + I_c^{cg}\ \dot{\theta}_c^2(t) \right] \qquad (1)$$

where

$m(x)$ is the distributed mass,

$r(x)$ is the distributed rotary inertia,

M_c is the mass of a concentrated item,

I_c^{cg} is the rotary inertia of a concentrated item about its own center of gravity,

$\dot{\delta}$ is the time rate of change of elastic deflection normal to the body x-axis, referred to as lateral velocity in this paper,

$\dot{\theta}$ is the angular velocity about an axis perpendicular to the paper,

$\dot{\Delta}_c$ is the velocity of a concentrated item and will be explained later,

x is the undeformed longitudinal axis of the beam,

t is time,

x_S is the starting x-station of the beam, and

x_E is the ending x-station of the beam.

The finite summation is over the number of concentrated mass items.

The technique used here to integrate Equation (1) is essentially a modification of the Rayleigh-Ritz method. In the basic Rayleigh-Ritz method the elastic deflection of a beam is represented by a summation of assumed displacement functions each weighted by a generalized coordinate representing the contribution of each of the assumed functions. Each displacement function is assumed over the entire beam length and considerable skill is required to choose these functions. The modification used here is to assume a simple displacement function between consecutive panel points and to use the panel point displacements as the generalized coordinates. A *linear* displacement function will be assumed here. This is illustrated in Figure 4 where x_k and x_{k+1} are consecutive panel point stations. The region between panel points k and k+1 is referred to as bay k. The lateral displacement between panel points k and k+1 is given by

$$\delta(x) = \delta_k + \left(x - x_k\right)\left(\delta_{k+1} - \delta_k\right)\Big/\left(x_{k+1} - x_k\right).$$

Similarly, the lateral velocity is given by

$$\dot{\delta}(x,t) = \dot{\delta}_k(t) + \left(x - x_k\right)\left[\dot{\delta}_{k+1}(t) - \dot{\delta}_k(t)\right]\Big/\left(x_{k+1} - x_k\right). \tag{2}$$

The angular velocity is obtained as the geometric derivative of the lateral velocity. That is,

$$\dot{\theta}(x,t) = -\frac{\partial\dot{\delta}(x,t)}{\partial x} = -\left[\dot{\delta}_{k+1}(t) - \dot{\delta}_k(t)\right]\Big/\left(x_{k+1} - x_k\right). \tag{3}$$

This equation for the angular velocity is slightly in error because a rotation due to shear deformation is included. A knowledge of the stiffness distribution would be necessary to determine the relative contribution of bending rotation and shear deformation to the angular velocity. This error will be small for beam type structures and will be ignored.

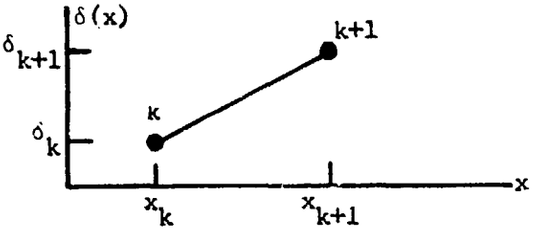


Figure 4 Linear Displacement Function

The kinetic energy of distributed mass, distributed rotary inertia, and concentrated mass items will be considered separately. The distributed mass, m(x), is considered first. The geometry for a line segment of distributed mass is shown below in Figure 5.

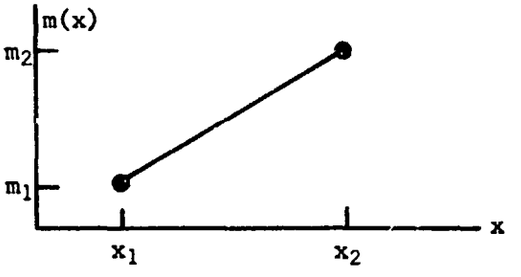


Figure 5 Line Segment Geometry

The equation for a straight line segment as shown in Figure 5 is

$$m(x) = m_1 + (x - x_1)\ (m_2 - m_1)/(x_2 - x_1). \qquad (4)$$

Substituting Equations (2) and (4) into (1) gives the kinetic energy of the distributed mass represented by one line segment i in bay k as

$$T_{i,k} = \tfrac{1}{2}\int_{x_p}^{x_q} \left[m_p + (x - x_p)\ (m_q - m_p)/(x_q - x_p)\right] \times$$

$$\left[\dot{\delta}_k(t) + (x - x_k)\ \left[\dot{\delta}_{k+1}(t) - \dot{\delta}_k(t)\right]/(x_{k+1} - x_k)\right]\ dx. \qquad (5)$$

The subscripts p and q have been introduced to handle the possibility of a line segment extending past the bay limits. Thus, x_p is the greater of x_1 or x_k and x_q is the lesser of x_2 or x_{k+1}. Similarly, m_p is either m_1 or m_k and m_q is either m_2 or m_{k+1}. The integration is continued for the line segment in adjacent bays, if necessary, until the entire line segment has been used. Performing the integration of Equation (5) yields

$$T_{i,k} = \tfrac{1}{2}\begin{bmatrix}\dot{\delta}_k(t) & \dot{\delta}_{k+1}(t)\end{bmatrix}\begin{bmatrix}z_{k,k} & z_{k,k+1}\\ (\text{sym}) & z_{k+1,k+1}\end{bmatrix}\begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\end{bmatrix} \tag{6}$$

where

$$z_{k,k} = F_1 - 2F_2 + F_3 \tag{6a}$$

$$z_{k,k+1} = F_2 - F_3 \tag{6b}$$

$$z_{k+1,k+1} = F_3 \tag{6c}$$

$$F_1 = L_p\left(m_p + m_q\right)/2 \tag{6d}$$

$$F_2 = L_p\left|\left(m_p + m_q\right)\left(H_p + H_q\right) + m_p H_p + m_q H_q\right|/6 \tag{6e}$$

$$F_3 = L_p\left|\left(m_p + m_q\right)\left(H_p + H_q\right)^2 + 2\left(m_p H_p^2 + m_q H_q^2\right)\right|/12 \tag{6f}$$

$$L_p = x_q - x_p \tag{6g}$$

$$H_p = \left(x_p - x_k\right)/L_k \tag{6h}$$

$$H_q = \left(x_q - x_k\right)/L_k, \text{ and} \tag{6i}$$

$$L_k = x_{k+1} - x_k. \tag{6j}$$

The kernel matrix in the triple matrix product of Equation (6) is the mass matrix that replaces the distributed mass represented by one line segment i in bay k.

The kinetic energy of the distributed rotary inertia, r(x), is considered next. The geometry for one rotary inertia line segment is similar to that given in Figure 5 for distributed mass. Similarly,

$$r(x) = r_1 + (x - x_1)\ (r_2 - r_1)/(x_2 - x_1). \tag{7}$$

Substituting Equations (3) and (7) into (1) gives the kinetic energy of the distributed rotary inertia represented by one line segment i in bay k as

$$T_{i,k} = \tfrac{1}{2}\int_{x_p}^{x_q} \left[r_p + \left(x - x_p\right)\ \left(r_q - r_p\right)\Big/\left(x_q - x_p\right)\right] x$$

$$\left[-\left(\dot{\delta}_{k+1}(t) - \dot{\delta}_k(t)\right)\Big/\left(x_{k+1} - x_k\right)\right] dx \tag{8}$$

The subscripts p and q have similar meaning as has been previously discussed for distributed mass.

Performing the integration of Equation (8) yields the same equation as was obtained previously for distributed mass. It is repeated here as

$$T_{i,k} = \tfrac{1}{2}\begin{bmatrix}\dot{\delta}_k(t) & \dot{\delta}_{k+1}(t)\end{bmatrix}\begin{bmatrix}z_{k,k} & z_{k,k+1}\\ (\text{sym}) & z_{k+1,k+1}\end{bmatrix}\begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\end{bmatrix}. \tag{9}$$

Now, however,

$$z_{k,k} = z_{k+1,k+1} = L_p\left(r_p + r_q\right)\Big/2L_k^2 \tag{9a}$$

and

$$z_{k,k+1} = -z_{k,k}. \tag{9b}$$

As before,

$$L_p = x_q - x_p \tag{9c}$$

and

$$L_k = x_{k+1} - x_k. \tag{9d}$$

The kernel matrix of Equation (9) is then the mass matrix that replaces the distributed rotary inertia represented by one line segment i in bay k.

The concentrated mass items are considered last. These items require some investigation because the item may be connected to the beam at an attach point a by an arm (assumed rigid) as shown in Figure 6. The square of the velocity of concentrated item c is

$$\dot{\Delta}_c^2(t) = \left[\dot{\delta}_a(t) - \left(x_c - x_a\right)\dot{\theta}_a(t)\right]^2 + \left[D\dot{\theta}_a(t)\right]^2. \qquad (10)$$

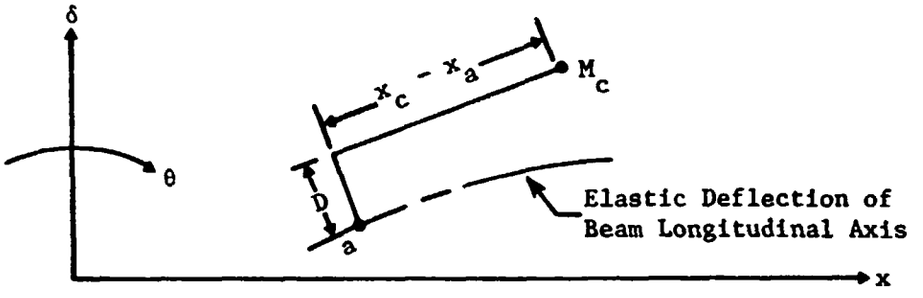


Figure 6 Concentrated Item Geometry

From Equation (1), using the velocity expression of Equation (10), the kinetic energy for a concentrated item c becomes:

$$T_c = \tfrac{1}{2}\left[M_c\,\dot{\delta}_a^2(t) - 2M_c\left(x_c - x_a\right)\dot{\delta}_a(t)\,\dot{\theta}_a(t) + M_c\left[\left(x_c - x_a\right)^2 + D^2\right]\dot{\theta}_a^2(t) + I_c^{cg}\,\dot{\theta}_a^2(t)\right] \qquad (11)$$

where use has been made of the relationship that $\dot{\theta}_c(t) = \dot{\theta}_a(t)$.

In further work, the offset D will be assumed to be zero. The reason for this is that any mass offset from the beam x-axis means that the longitudinal principal axis of the beam does not coincide with the beam x-axis. A coupled transverse-longitudinal analysis would be required to accurately describe the kinetic energy of the beam for $D \neq 0$. However, if the analyst wishes to include the effect of an offset D, inspection of Equation (11) shows that $I_c^{cg} + M_c D^2$ could be used. With the above conditions, the kinetic energy for a concentrated item can be expressed in matrix notation as

$$T_c = \tfrac{1}{2}\begin{bmatrix}\dot{\delta}_a(t) & \dot{v}_a(t)\end{bmatrix}\begin{bmatrix}M_c & -M_c(x_c - x_a) \\ -M_c(x_c - x_a) & M_c(x_c - x_a)^2 + I_c^{cg}\end{bmatrix}\begin{bmatrix}\dot{\delta}_a(t) \\ \dot{v}_a(t)\end{bmatrix} \qquad (12)$$

The lateral and rotational velocities at the attach point a are given in terms of the velocities at adjacent panel points k and k+1 from Equations (2) and (3), respectively. Using these equations the kinetic energy of one concentrated item c in bay k is finally given as

$$T_{c,k} = \tfrac{1}{2}\begin{bmatrix}\dot{\delta}_k(t) & \dot{\delta}_{k+1}(t)\end{bmatrix}\begin{bmatrix}z_{k,k} & z_{k,k+1} \\ (\text{sym}) & z_{k+1,k+1}\end{bmatrix}\begin{bmatrix}\dot{\delta}_k(t) \\ \dot{\delta}_{k+1}(t)\end{bmatrix} \qquad (13)$$

where

$$z_{k,k} = F_1 - 2F_2 + F_3, \qquad (13a)$$

$$z_{k,k+1} = F_2 - F_3, \text{ and} \qquad (13b)$$

$$z_{k+1,k+1} = F_3, \qquad (13c)$$

as were previously obtained for distributed mass. Now, however,

$$F_1 = M_c, \qquad (13d)$$

$$F_2 = M_c H_c, \qquad (13e)$$

$$F_3 = M_c H_c^2 + I_c^{cg}/L_b^2, \text{ and} \qquad (13f)$$

$$H_c = (x_c - x_k)/L_b. \qquad (13g)$$

As before,

$$L_k = x_{k+1} - x_k. \qquad (13h)$$

The terms in Equations (13a) and (13b) could be combined to give a simpler expression for $z_{k,k}$ and $z_{k,k+1}$. They are used, however, to minimize coding instructions in the subroutine. The attach point coordinate, x_a, does not appear in any of the final equations

for $z_{k,k}$, etc. It is used only to determine the bay in which the attach point is located. For an attach point coinciding with a panel point, the inertial properties could be correctly calculated using the coordinates of the panel point bay in front of the attach point or the panel point bay in back of the attach point. The results would be different, however, since the angular velocity is discontinuous at a panel point. To minimize numerical roundoff error, the attach point is assumed at the beginning of the bay.

The mass matrix for the entire beam is obtained by evaluating Equations (6) for each line segment of distributed mass, Equations (9) for each line segment of distributed rotary inertia, and Equations [13] for each concentrated mass item. Every resulting 2x2 bay mass matrix is added to previous 2x2 bay mass matrices at like panel points to form the mass matrix for the entire beam.

The total mass properties of the beam are calculated by the following triple matrix product.

$$\left[\{1\} \mid \{PP\}\right]^T [Z] \left[\{1\} \mid \{PP\}\right] = \begin{bmatrix} M_T & P_T^o \\ P_T^o & I_T^o \end{bmatrix}$$

where

$\{1\}$ is a column of ones,

$\{PP\}$ is a column of the panel point x-stations,

$[Z]$ is the mass matrix for the beam,

M_T is the beam mass,

P_T^o is the first moment of the beam about x = 0, and

I_T^o is the moment of inertia of the beam about x = 0.

From this data the center of gravity of the beam is calculated from

$$x_{cg} = P_T^o / M_T$$

and the moment of inertia of the beam about x_{cg} is calculated from

$$I_T^{cg} = I_T^o - M_T\, x_{cg}^2 .$$

SPECIAL CASES

Two cases are worthy of mention. In the first case, a *linearly* varying segment of distributed mass extending to the bay limits $\left(x_k \text{ and } x_{k+1}\right)$ has a mass matrix (from Equations (6) of:

$$\frac{x_{k+1} - x_k}{12} \begin{bmatrix} 3m_k + m_{k+1} & m_k + m_{k+1} \\ (\text{sym}) & m_k + 3m_{k+1} \end{bmatrix}.$$

In the second case, a *uniform* $\left(\text{i.e., } m_k = m_{k+1} = m\right)$ segment of distributed mass extending to the bay limits $\left(x_k \text{ and } x_{k+1}\right)$ has a mass matrix of:

$$\frac{m\left(x_{k+1} - x_k\right)}{6} \begin{bmatrix} 2 & 1 \\ (\text{sym}) & 2 \end{bmatrix}.$$

Subroutine MASS2 takes (on option) distributed mass, distributed rotary inertia, and concentrated mass items of a beam and replaces them with a mass matrix. The elements of the mass matrix are representative inertial values of the beam at selected points on the beam. These elements are calculated by assuming a cubic function between pairs of the selected points to describe the beam's lateral elastic deflection. The x-stations of the selected points (panel points) are given in {PP}. These x-stations must be in increasing order.

The distributed mass, m(x), is assumed to be piecewise linear and is represented by straight line segments as shown in Figure 1. The x-stations of the end points for the line segments giving the distributed mass are independent of the panel point x-stations. However, the distributed mass must be within the panel point limits. The line segments representing the distributed mass may or may not be joined and may overlap. The distributed mass is defined in [DMASS]. Each row of [DMASS] represents one nonvertical line segment. The form of each row of [DMASS] is $[x_1\ x_2\ m_1\ m_2]$ where x_1, m_1 give the first end point and x_2, m_2 give the second end point of a line segment.

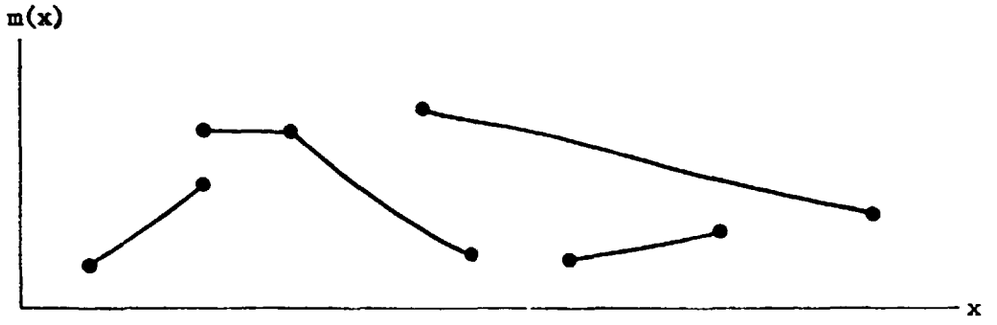


Figure 1 Distributed Mass

The distributed rotary inertia, r(x), is also assumed to be represented by straight line segments. All statements used above for distributed mass are applicable to distributed rotary inertia. The distributed rotary inertia is defined in [DRIN]. The form of each row of [DRIN] is $[x_1\ x_2\ r_1\ r_2]$. The end point stations x_1 and x_2 for the rotary inertia line segments are independent from the end point stations x_1 and x_2 for the mass line segments.

The concentrated mass items are defined in [CONC]. Each row of [CONC] represents one concentrated mass item and contains:

1) x_a, the station at which the item is attached to the beam;
2) M_c, the mass of the item;
3) x_{cg}, the center of gravity of the item;
4) I_c^{cg}, the moment of inertia of the item about its own center of gravity.

These four elements are given in the form $\left[x_a \; M_c \; x_{cg} \; I_c^{cg}\right]$. The attachment station, x_a, of an item to the beam must be within the panel point limits.

The calculated representative inertial values at the selected panel points are placed in a mass matrix given by [Z]. The form of [Z] is

$$[Z]_{2n\times 2n} = \left[\begin{array}{c|c} \left[Z_{\delta,\delta}\right] & \left[Z_{\delta,\theta}\right] \\ \hline \left[Z_{\theta,\delta}\right] & \left[Z_{\theta,\theta}\right] \end{array}\right]$$

where n is the number of panel points. Matrix [Z] is symmetric, i.e., $z_{ij} = z_{ji}$. The partition form of [Z] results from the two generalized coordinates (lateral translation, δ, and rotation, θ) at each panel point arranged with all translation coordinates first followed by all rotation coordinates. Each partition of [Z] is square and tri-diagonal, for example,

$$\left[Z_{\delta,\delta}\right]_{n\times n} = \begin{bmatrix} z_{1,1} & z_{1,2} & & & & & \\ z_{2,1} & z_{2,2} & z_{2,3} & & & & \\ & z_{3,2} & z_{3,3} & z_{3,4} & & & \\ & & & \ddots & & & \\ & & & & & & \\ & & & z_{n-1,n-2} & z_{n-1,n-1} & z_{n-1,n} & \\ & & & & z_{n,n-1} & z_{n,n} & \end{bmatrix}.$$

The sign convention used in this paper to obtain the mass matrix [Z] is shown in Figure 2.

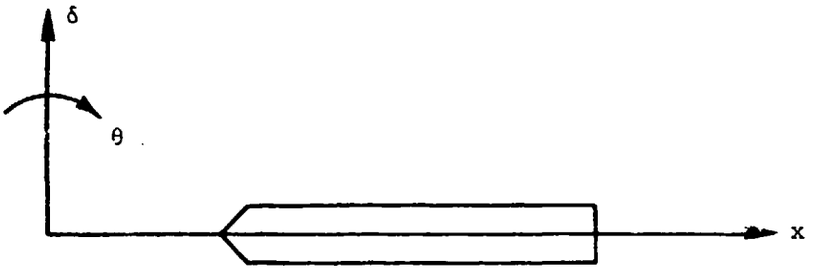


Figure 2 Sign Convention for This Paper

The beam mass matrix obtained in this paper is applicable for either the pitch or yaw plane with a change of variables and possible sign changes. For the axis system shown in Figure 3, the following variables would be used. Note that in order to have a right-hand system such as that shown in Figure 3, the sign of the $\left[Z_{\delta,\theta}\right]$ and $\left[Z_{\theta,\delta}\right]$ partitions of [Z] from this subroutine would have to be changed for the yaw plane.

This Paper (δ, x, θ)	Axis System of Figure 3	
	Pitch (z, x, θ_y)	Yaw (y, x, θ_z)
x	x	x
δ	δ_z	δ_y
θ	θ_y	$-\theta_z$
m(x)	m(x)	m(x)
r(x)	$r_y(x)$	$r_z(x)$
M_c	M_c	M_c
I_c^{cg}	$I_{c,y}^{cg}$	$I_{c,z}^{cg}$

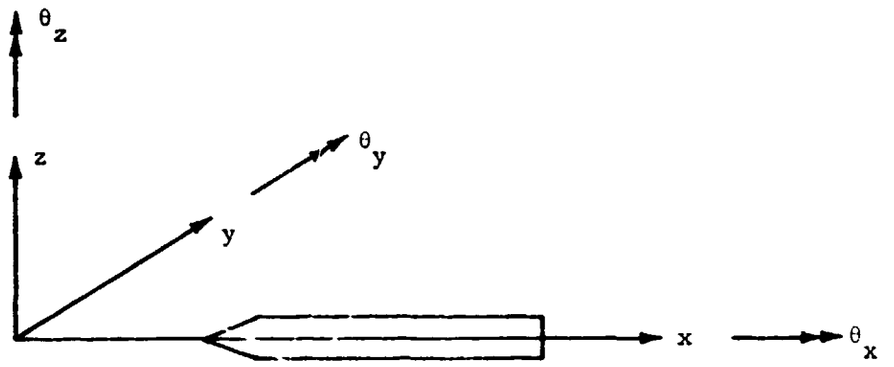


Figure 3 Beam Axis System (Right Hand)

The beam's total mass, center of gravity, and moment of inertia about the beam's center of gravity are also calculated and printed.

DESCRIPTION OF TECHNIQUE

The replacement of distributed mass, distributed rotary inertia, and concentrated mass items of a beam by a mass matrix is obtained using a kinetic energy approach as follows.

For small deflections normal to the longitudinal body axis (x) of the beam shown in Figure 2, the kinetic energy is defined by

$$T = \tfrac{1}{2} \int_{x_S}^{x_E} \left[m(x)\ \dot{\delta}^2(x,t) + r(x)\ \dot{\theta}^2(x,t) \right] dx + \tfrac{1}{2} \sum_c \left[M_c\ \dot{\Delta}_c^2(t) + I_c^{cg}\ \dot{\theta}_c^2(t) \right] \qquad (1)$$

where

$m(x)$ is the distributed mass,

$r(x)$ is the distributed rotary inertia,

M_c is the mass of a concentrated item,

I_c^{cg} is the rotary inertia of a concentrated item about its own center of gravity,

$\dot{\delta}$ is the time rate of change of elastic deflection normal to the body x-axis, and is referred to as lateral velocity in this paper,

$\dot{\theta}$ is the angular velocity about an axis perpendicular to the paper,

$\dot{\Delta}_c$ is the velocity of a concentrated item and will be explained later,

x is the undeformed longitudinal axis of the beam,

t is time,

x_S is the starting x-station of the beam, and

x_E is the ending x-station of the beam.

The finite summation is over the number of concentrated mass items.

The technique used here to integrate Equation (1) is essentially a modification of the Rayleigh-Ritz method. In the basic Rayleigh-Ritz method the elastic deflection of a beam is represented by a summation of assumed displacement functions each weighted by a generalized coordinate representing the contribution of each of the assumed functions. Each displacement function is assumed over the entire beam length and considerable skill is required to choose these functions. The modification used here is to assume a simple displacement function between consecutive panel points and to use the panel point displacements as the generalized coordinates. A *cubic* displacement function will be assumed here. This is illustrated in Figure 4 where x_k and x_{k+1} are consecutive panel point stations. The region between panel points k and k+1 is referred to as bay k.

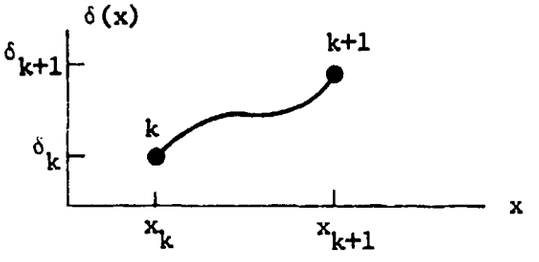


Figure 4 Cubic Displacement Function

The cubic displacement function in bay k is determined as follows. The lateral displacement is given by

$$\delta(x) = A\left(x - x_k\right)^3 + B\left(x - x_k\right)^2 + c\left(x - x_k\right) + D$$

or

$$\delta(H) = aH^3 + bH^2 + cH + d$$

$$= \begin{bmatrix} H^3 & H^2 & H & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \tag{2}$$

where

$$H = \left(x - x_k\right)/L_k \tag{2a}$$

with

$$L_k = x_{k+1} - x_k. \tag{2b}$$

H is a local nondimensional bay coordinate to aid in the solution of Equation (1). The angular displacement is obtained as the geometric derivative of the lateral displacement. That is,

$$\theta(x) = -\frac{\partial \delta(x)}{\partial x}$$

or

$$\theta(H) = -\frac{d\ \delta(H)}{L_k\ dH}$$

$$= \frac{1}{L_k}\begin{bmatrix} -3H^2 & -2H & -1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}. \tag{3}$$

The coefficients a, b, c, d are determined as follows from Equations (2) and (3).

$$\begin{bmatrix} \delta_k \\ \delta_{k+1} \\ L_k\ \theta_k \\ L_k\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} H_k^3 & H_k^2 & H_k & 1 \\ H_{k+1}^3 & H_{k+1}^2 & H_{k+1} & 1 \\ -3H_k^2 & -2H_k & -1 & 0 \\ -3H_{k+1}^2 & -2H_{k+1} & -1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}.$$

From which

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = [\psi] \begin{bmatrix} \delta_k \\ \delta_{k+1} \\ \theta_k \\ \theta_{k+1} \end{bmatrix}$$

where

$$[\psi] = \begin{bmatrix} 2 & -2 & -L_k & -L_k \\ -3 & 3 & 2L_k & L_k \\ 0 & 0 & -L_k & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Thus the lateral and angular displacement in bay k is determined in terms of the adjacent panel point lateral and angular displacements. Similarly, the lateral and angular velocity in bay k is given by

$$\dot{\delta}(H,t) = [H^3 \quad H^2 \quad H \quad 1]\ [\psi] \begin{bmatrix} \dot{\delta}_k(t) \\ \dot{\delta}_{k+1}(t) \\ \dot{\theta}_k(t) \\ \dot{\theta}_{k+1}(t) \end{bmatrix} \qquad (4)$$

and

$$\dot{\theta}(H, t) = \frac{1}{L_k} [-3H^2 \quad -2H \quad -1 \quad 0] [\psi] \begin{bmatrix} \dot{\delta}_k(t) \\ \dot{\delta}_{k+1}(t) \\ \dot{\theta}_k(t) \\ \dot{\theta}_{k+1}(t) \end{bmatrix}. \tag{5}$$

The previous expression for the angular velocity is slightly in error because a rotation due to shear deformation is included. A knowledge of the stiffness distribution would be necessary to determine the relative contribution of bending rotation and shear deformation to the angular velocity. This error will be small for beam-type structures and will be ignored.

The kinetic energy of distributed mass, distributed rotary inertia, and concentrated mass items will be considered separately. The distributed mass, m(x), is considered first. The geometry for a line segment of distributed mass is shown in Figure 5.

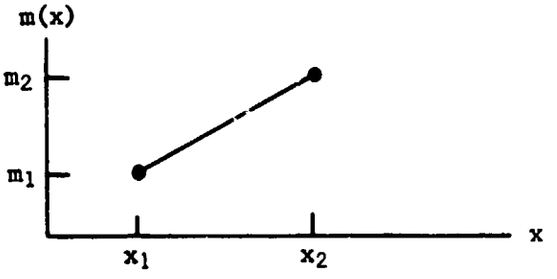


Figure 5 Line Segment Geometry

The equation for a straight line segment as shown in Figure 5 is

$$m(x) = m_1 + (x - x_1)\ (m_2 - m_1)/(x_2 - x_1),$$

or

$$m(H) = WH + m_1 - WH_1 \tag{6}$$

by using the nondimensional coordinate of Equation (2a) and

$$W = (m_2 - m_1)/(H_2 - H_1). \tag{7}$$

Modifying Equation (1) to use the nondimensional coordinate of Equation (2a) and substituting Equations (4) and (6) gives the kinetic energy of the distributed mass represented by one line segment i in bay k as

$$T_{i,k} = \tfrac{1}{2}\begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\\ \dot{\theta}_k(t)\\ \dot{\theta}_{k+1}(t)\end{bmatrix}^T [\nu]^T \left(L_k \int_{H_p}^{H_q} \begin{bmatrix} H^6 & H^5 & H^4 & H^3\\ H^5 & H^4 & H^3 & H^2\\ H^4 & H^3 & H^2 & H\\ H^3 & H^2 & H & 1\end{bmatrix} \left| WH + m_p - WH_p \right| dH \right) [\nu] \begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\\ \dot{\theta}_k(t)\\ \dot{\theta}_{k+1}(t)\end{bmatrix} \qquad (8)$$

The subscripts p and q have been introduced to handle the possibility of a line segment extending past the bay limits. Thus, x_p (or H_p) is the greater of x_1 or x_k, and x_q (or H_q) is the lesser of x_2 or x_{k+1}. Similarly, m_p is either m_1 or m_k and m_q is either m_2 or m_{k+1}. The integration is continued for the line segment in adjacent bays, if necessary until the entire line segment has been used. Performing the integration of Equation (8) yields

$$T_{i,k} = \tfrac{1}{2}\begin{bmatrix}\dot{\delta}_k(t) & \dot{\delta}_{k+1}(t) & \dot{\theta}_k(t) & \dot{\theta}_{k+1}(t)\end{bmatrix} \begin{bmatrix} z_{k,k} & z_{k,k+1} & z_{k,k+n} & z_{k,k+n+1}\\ & z_{k+1,k+1} & z_{k+1,k+n} & z_{k+1,k+n+1}\\ & & z_{k+n,k+n} & z_{k+n,k+n+1}\\ \text{(sym)} & & & z_{k+n+1,k+n+1}\end{bmatrix} \begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\\ \dot{\theta}_k(t)\\ \dot{\theta}_{k+1}(t)\end{bmatrix} \qquad (9)$$

where

$$z_{k,k} = F_1 - 6F_3 + 4F_4 + P_1 \qquad (9a)$$

$$z_{k,k+1} = 3F_3 - 2F_4 - P_1 \qquad (9b)$$

$$z_{k,k+n} = (-F_2 + 2F_3 - F_4 - P_2)\, L_k \qquad (9c)$$

$$z_{k,k+n+1} = (F_3 - F_4 - P_3)\, L_k \qquad (9d)$$

$$z_{k+1,k+1} = P_1 \qquad (9e)$$

$$z_{k+1,k+n} = P_2\, L_k \qquad (9f)$$

$$z_{k+1,k+n+1} = P_3\, L_k \qquad (9g)$$

$$z_{k+n,k+n} = (2F_5 - 4F_6 + F_7 + F_8 - 4F_9 + 4F_{10})\ L_k^2 \qquad (9h)$$

$$z_{k+n,k+n+1} = (F_5 - 3F_6 + F_7 - F_9 + 2F_{10})\ L_k^2 \qquad (9i)$$

$$z_{k+n+1,k+r+1} = (-2F_6 + F_7 + F_{10})\ L_k \qquad (9j)$$

$$P_1 = -12F_6 + 4F_7 + 9F_{10} \qquad (9k)$$

$$P_2 = 2F_5 - 7F_6 + 2F_7 - 3F_9 + 6F_{10} \qquad (9\ell)$$

$$P_3 = -5F_6 + 2F_7 + 3F_{10} \qquad (9m)$$

$$F_j = L_k\ E_j \quad (j = 1, 7) \qquad (9n)$$

$$F_8 = L_k\ E_3 \qquad (9p)$$

$$F_9 = L_k\ E_4 \qquad (9q)$$

$$F_{10} = L_k\ E_5 \qquad (9r)$$

$$E_j = W\left(H_q^{j+1} - H_p^{j+1}\right)/(j + 1) + \left(m_p - WH_p\right)\left(H_q^j - H_p^j\right)/j \qquad (9s)$$

$$W = \left(m_q - m_p\right)\left(H_q - H_p\right) \qquad (9t)$$

$$H_p = \left(x_p - x_k\right)/L_k \qquad (9u)$$

$$H_q = \left(x_q - x_k\right)/L_k \qquad (9v)$$

$$L_k = x_{k+1} - x_k, \text{ and} \qquad (9w)$$

n is the number of panel points.

It is possible to simplify these equations for z since $F_8 = F_3$, $F_9 = F_4$, and $F_{10} = F_5$. This was not done so that these same equations for z could also be used for distributed rotary inertia and concentrated mass items and thus save coding instructions in the subroutine. The kernel matrix in the triple matrix product of Equation (9) is the mass matrix which replaces the distributed mass represented by one line segment i in bay k.

The kinetic energy of the distributed rotary inertia, r(x), is considered next. The geometry for one rotary inertia line segment is similar to that given in Figure 5 for distributed mass. Similarly,

$$r(H) = WH + r_1 - WH_1 \tag{10}$$

using the nondimensional coordinate of Equation (2a) and

$$W = (r_2 - r_1)/(H_2 - H_1). \tag{11}$$

Modifying Equation (1) to use the nondimensional coordinate of Equation (2a) and substituting Equations (5) and (10) gives the kinetic energy of the distributed rotary inertia represented by one line segment i in bay k as

$$T_{i,k} = \tfrac{1}{2}\begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\\ \dot{\theta}_k(t)\\ \dot{\theta}_{k+1}(t)\end{bmatrix}^T [\psi]^T \left(\frac{1}{L_k} \int_{H_p}^{H_q} \begin{bmatrix} 9H^2 & 6H^3 & 3H^2 & 0\\ 6H^3 & 4H^2 & 2H & 0\\ 3H^2 & 2H & 1 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} \left[WH + r_p - WH_p\right] dH \right) [\psi] \begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\\ \dot{\theta}_k(t)\\ \dot{\theta}_{k+1}(t)\end{bmatrix} \tag{12}$$

The subscripts p and q have similar meaning, as has been previously discussed, for distributed mass. Performing the integration of Equation (12) yields results identical to Equations (9) thru (9m) and similar to Equations (9s) thru (9w) as obtained previously for distributed mass. These equations will not be repeated here. The differences from the distributed mass equations are:

$$F_j = 0 \quad (j = 1, 4) \tag{13a}$$

$$F_5 = 3E_3/L_k \tag{13b}$$

$$F_6 = 6E_4/L_k \tag{13c}$$

$$F_7 = 9E_5/L_k \tag{13d}$$

$$F_8 = E_1/L_k \tag{13e}$$

$$F_9 = 2E_2/L_k, \text{ and} \tag{13f}$$

$$F_{10} = 4E_3/L_k. \tag{13g}$$

The kernel matrix in the triple matrix product of Equation (9) is then the mass matrix which replaces the distributed rotary inertia represented by one line segment i in bay k.

The concentrated mass items are considered last. These items require some investigation because the item may be connected to the beam at an attach point a by an arm (assumed rigid) as shown in Figure 6.

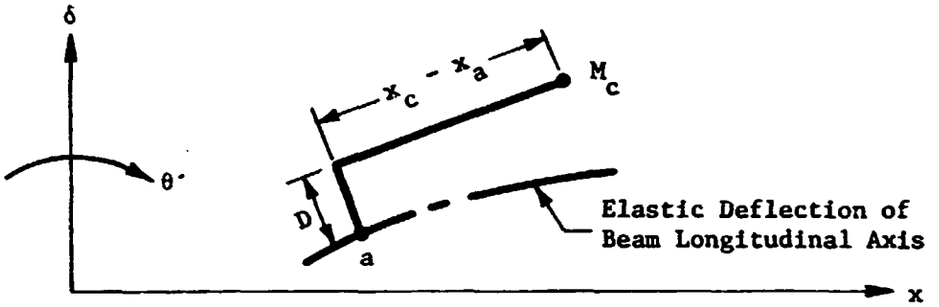


Figure 6 Concentrated Item Geometry

The square of the velocity of concentrated item c is

$$\dot{\Delta}_c^2(t) = \left[\dot{\delta}_a(t) - \left(x_c - x_a\right)\dot{\theta}_a(t)\right]^2 + \left[D\dot{\theta}_a(t)\right]^2 \qquad (14)$$

From Equation (1), using the velocity expression of Equation (14), the kinetic energy for a concentrated item c becomes:

$$T_c = \tfrac{1}{2}\left\{M_c\,\dot{\delta}_a^2(t) - 2M_c\left(x_c - x_a\right)\dot{\delta}_a(t)\,\dot{\theta}_a(t) + M_c\left[\left(x_c - x_a\right)^2 + D^2\right]\dot{\theta}_a^2(t) + I_c^{cg}\,\dot{\theta}_a^2(t)\right\} \qquad (15)$$

where use has been made of the relationship that $\dot{\theta}_c(t) = \dot{\theta}_a(t)$.

In further work, the offset D will be assumed to be zero. The reason for this is that any mass offset from the beam x-axis means that the longitudinal principal axis of the beam does not coincide with the beam x-axis. A coupled transverse-longitudinal analysis would be required to accurately describe the kinetic energy of the beam for $D \neq 0$. However, if the analyst wishes to include the effect of an offset D, inspection of Equation (15) shows that $I_c^{cg} + M_c D^2$ could be used. With the above conditions, the kinetic energy for a concentrated item can be expressed in matrix notation as

$$T_c = \tfrac{1}{2}\begin{bmatrix}\dot{\delta}_a(t) & \dot{\theta}_a(t)\end{bmatrix}\begin{bmatrix}M_c & -M_c\left(x_c - x_a\right) \\ -M_c\left(x_c - x_a\right) & M_c\left(x_c - x_a\right)^2 + I_c^{cg}\end{bmatrix}\begin{bmatrix}\dot{\delta}_a(t) \\ \dot{\theta}_a(t)\end{bmatrix}. \qquad (16)$$

The lateral and rotational velocities at the attach point a are given in terms of the velocities at adjacent panel points k and k+1 from Equations (4) and (5), respectively. Using these equations and modifying Equation (16) to use the nondimensional coordinate of Equation (2a), the kinetic energy of one concentrated item c in bay k is identical to Equations (9) thru (9m) as obtained previously for distributed mass. These equations will not be repeated here. The differences from the distributed mass equations are:

$$F_1 = M_c \tag{17a}$$

$$F_2 = M_c H_c \tag{17b}$$

$$F_3 = H_a M_c \left(2H_c - H_a\right) \tag{17c}$$

$$F_4 = H_a^2 M_c \left(3H_c - 2H_a\right) \tag{17d}$$

$$F_5 = H_a^2 \left[M_c H_c \left(3H_c - 2H_a\right) + 3I_c^{cg}/L_k^2\right] \tag{17e}$$

$$F_6 = H_a^3 \left[M_c \left(3H_c - 2H_a\right) \left(2H_c - H_a\right) + 6I_c^{cg}/L_k^2\right] \tag{17f}$$

$$F_7 = H_a^4 \left[M_c \left(3H_c - 2H_a\right)^2 + 9I_c^{cg}/L_k^2\right] \tag{17g}$$

$$F_8 = M_c H_c^2 + I_c^{cg}/L_k^2 \tag{17h}$$

$$F_9 = H_a \left[M_c H_c \left(2H_c - H_a\right) + 2I_c^{cg}/L^2\right] \tag{17i}$$

$$F_{10} = H_a^2 \left[M_c \left(2H_c - H_a\right)^2 + 4I_c^{cg}/L^2\right] \tag{17j}$$

$$H_a = \left(x_a - x_k\right)/L_k \tag{17k}$$

$$H_c = \left(x_c - x_k\right)/L_k, \text{ and} \tag{17l}$$

$$L_k = x_{k+1} - x_k. \tag{17m}$$

The mass matrix for the entire beam is obtained by evaluating Equations (9) thru (9w) for each line segment of distributed mass, Equations (9) thru (9m), (9s) thru (9w), and (13a) thru (13g) for each line segment of distributed rotary inertia, and Equations (9) thru (9m) and (17a) thru (17m) for each concentrated mass item. Every resulting 4x4 bay mass matrix is added to previous 4x4 bay mass matrices at like panel points to form the mass matrix for the entire beam.

The total mass properties of the beam are calculated by the following triple matrix product.

$$\left[\begin{array}{c|c} \{1\} & \{PP\} \\ \hline \{0\} & \{-1\} \end{array}\right]^T \left[\begin{array}{c|c} [Z_{\delta\delta}] & [Z_{\delta\theta}] \\ \hline [Z_{\theta\delta}] & [Z_{\theta\theta}] \end{array}\right] \left[\begin{array}{c|c} \{1\} & \{PP\} \\ \hline \{0\} & \{-1\} \end{array}\right] = \left[\begin{array}{c|c} M_T & P_T^o \\ \hline P_T^o & I_T^o \end{array}\right]$$

where

$\{1\}$ is a column of ones,

$\{0\}$ is a column of zeroes,

$\{PP\}$ is a column of the panel point x-stations,

$[Z_{\delta\delta}]$ etc, are partitions of the mass matrix $[Z]$,

M_T is the beam mass,

P_T^o is the first moment of the beam about $x = 0$, and

I_T^o is the moment of inertia of the beam about $x = 0$.

From this data, the center of gravity of the beam is calculated from

$$x_{cg} = P_T^o / M_T$$

and the moment of inertia of the beam about x_{cg} is calculated from

$$I_T^{cg} = I_T^o - M_T x_{cg}^2 .$$

SPECIAL CASES

Two cases are worthy of mention. In the first case, a *linearly* varying segment of distributed mass extending to the bay limits $(x_k$ and $x_{k+1})$ has a mass matrix [from Equation (9)] of:

$$\frac{L_k}{840}\begin{bmatrix} 240m_k + 72m_{k+1} & 54m_k + 54m_{k+1} & -(30m_k + 14m_{k+1})L_k & (14m_k + 12m_{k+1})L_k \\ & 72m_k + 240m_{k+1} & -(12m_k + 14m_{k+1})L_k & (14m_k + 30m_{k+1})L_k \\ & & (5m_k + 3m_{k+1})L_k^2 & -(3m_k + 3m_{k+1})L_k^2 \\ & (\text{sym}) & & (3m_k + 5m_{k+1})L_k^2 \end{bmatrix}$$

where

$$L_k = x_{k+1} - x_k.$$

In the second case, a *uniform* $(\text{i.e., } m_k = m_{k+1} = m)$ segment of distributed mass extending to the bay limits $(x_k$ and $x_{k+1})$ has a mass matrix of

$$\frac{mL_k}{420}\begin{bmatrix} 156 & 54 & -22L_k & 13L_k \\ & 156 & -13L_k & 22L_k \\ & & 4L_k^2 & -3L_k^2 \\ & (\text{sym}) & & 4L_k^2 \end{bmatrix}$$

where

$$L_k = x_{k+1} - x_k.$$

This last result agrees with Archer's consistent mass matrix for a uniform (mass and stiffness) beam segment given in Reference 1.

REFERENCES

1. John S. Archer: *Consistent Mass Matrix for Distributed Mass Systems*. Proceedings of the American Society of Civil Engineers, Journal of the Structural Division, August 1963.

Subroutine MASS2A takes distributed mass of fluid in a container and the slosh mass of the same fluid and replaces them with a mass matrix. The mass matrix includes coupling between these two types of masses. The elements in the mass matrix for the distributed mass and the coupling terms are representative inertial values at selected points on the beam. These elements are calculated by assuming (1) a cubic function between pairs of the selected points to describe the container's lateral elastic deflection, and (2) a uniform motion of the fluid slosh relative to the container's deflected longitudinal axis.

The x stations of the selected points (panel points) are given in {PP}. These x stations must be in increasing order.

The distributed mass, m(x), is assumed to be piecewise linear and is represented by straight line segments as shown in Figure 1.

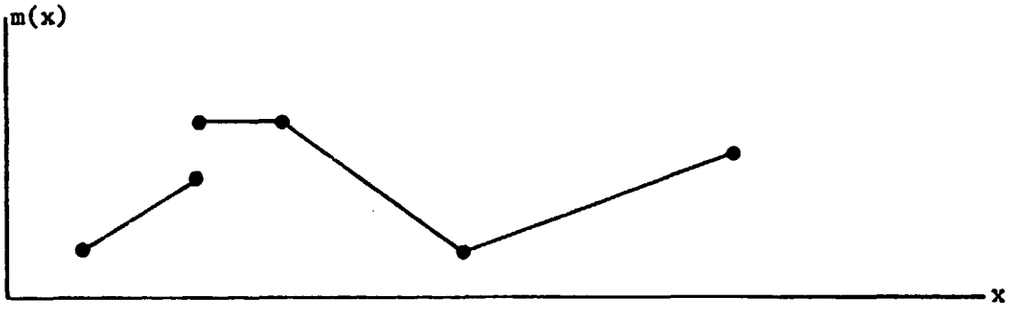


Figure 1 Distributed Mass

The x stations of the end points for the line segments giving the distributed mass are independent of the panel point x stations. However, the distributed mass must be within the panel point limits. The line segments representing the distributed mass may or may not be joined; however, there must not be any x voids or overlaps. The distributed mass is defined in [DMASS]. Each row of [DMASS] represents one nonvertical line segment. The form of each row of [DMASS] is $[x_1\ x_2\ m_1\ m_2]$ where x_1, m_1 give the first end point and x_2, m_2 give the second end point of a line segment. The x_2 of row 1 of [DMASS] must be equal to x_1 of row 2, etc.

To aid the analyst, the fluid level (FLEVEL) is input so that [DMASS] can describe the complete container fluid. Only the fluid from the fluid level to the container bottom will be used in calculating the mass matrix. Another feature is that [DMASS]

can be used to define the cross-sectional area of the complete container and the input scalar CONVRT used to define the fluid density.

It is assumed that there is no distributed rotary inertia for a fluid as there is for a structure.

The slosh mass (SMASS) of the fluid is a value that is separately obtained either from test or mathematical modeling techniques.

The calculated inertial terms are placed in a mass matrix given by [Z]. The form of [Z] is

$$[Z]_{2n+1,2n+1} = \left[\begin{array}{c|c|c} \left[Z_{\delta,\delta}\right] & \left[Z_{\delta,\theta}\right] & \left\{Z_{\delta,s}\right\} \\ \hline \left[Z_{\theta,\delta}\right] & \left[Z_{\theta,\theta}\right] & \left\{Z_{\theta,s}\right\} \\ \hline \left\{Z_{s,\delta}\right\}^T & \left\{Z_{s,\theta}\right\}^T & Z_{s,s} \end{array}\right]$$

where n is the number of panel points. Matrix [Z] is symmetric, i.e., $z_{ij} = z_{ji}$. The partition form of [Z] results from the two generalized coordinates (lateral translation δ and rotation θ) at each panel point arranged with all translation coordinates first followed by all rotation coordinates. These panel point coordeinates are then followed by the single fluid slosh coordinate. Each panel point partition of [Z] is square and tri-diagonal, for example,

$$\left|Z_{\delta,\delta}\right|_{n\times n} = \begin{bmatrix} z_{1,1} & z_{1,2} & & & & & \\ z_{2,1} & z_{2,2} & z_{2,3} & & & & \\ & z_{3,2} & z_{3,3} & z_{3,4} & & & \\ & & & \ddots & & & \\ & & & & z_{n-1,n-2} & z_{n-1,n-1} & z_{n-1,n} \\ & & & & & z_{n,n-1} & z_{n,n} \end{bmatrix}.$$

The element $Z_{s,s}$ is the input slosh mass (SMASS).

The sign convention used in this paper to obtain the mass matrix [Z] is shown in Figure 2.

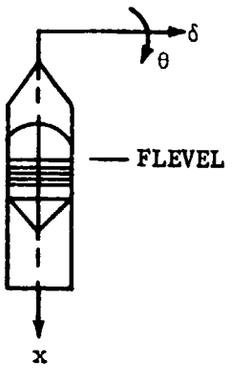


Figure 2 Sign Convention for This Paper

The mass matrix obtained in this paper is applicable for either the pitch or yaw plane with a change of variables and possible sign changes. For the axis system shown in Figure 3, the following variables would be used.

This Paper (δ,x,θ)	Axis System of Figure 3	
	Pitch (z,x,θ_y)	Yaw (y,x,θ_z)
x	x	x
δ	δ_z	δ_y
θ	θ_y	$-\theta_z$
m(x)	m(x)	m(x)

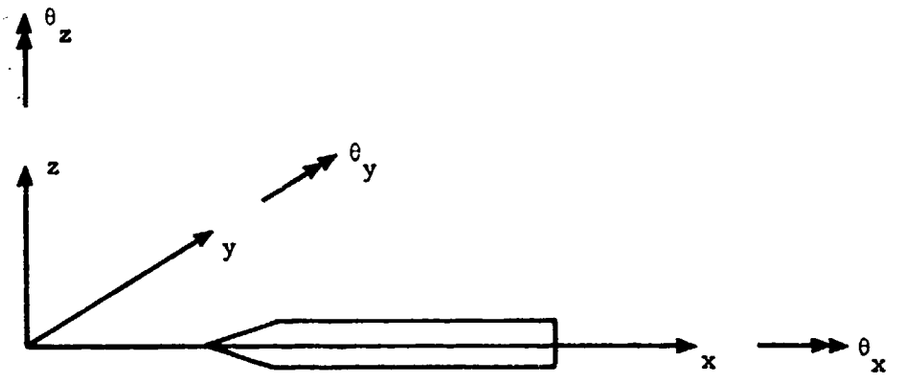


Figure 3 Beam Axis System (Right Hand)

Note that in order to have a right-hand system such as that shown in Figure 3, the sign of the $\left[Z_{\delta,\theta}\right]$, $\left[Z_{\theta,\delta}\right]$, $\left\{Z_{\theta,s}\right\}$, and $\left\{Z_{s,\theta}\right\}^T$ partitions of [Z] from this subroutine would have to be changed for the yaw plane.

The fluid's total mass, center of gravity, and moment of inertia about the fluid's center of gravity are also calculated and printed.

DESCRIPTION OF TECHNIQUE

The replacement of the distributed mass and the slosh mass of fluid in a container by a mass matrix is obtained using a kinetic energy approach as follows.

For small deflections normal to the longitudinal body axis (x) of a container, the kinetic energy of the fluid in the container shown in Figure 2 is defined by

$$T = \frac{1}{2}\int_{x_S}^{x_E} \left[m_i(x)\ \dot{\delta}_c^2(x,t) + m_s(x)\ \left(\dot{\delta}_c(x,t) + \dot{\delta}_s(x,t)\right)^2 \right] dx \qquad (1)$$

where

$m_i(x)$ is the distributed inert mass of the fluid,

$m_s(x)$ is the distributed slosh mass of the fluid,

$\dot{\delta}_c(x,t)$ is the time rate of change of the elastic deflection of the container's longitudinal axis normal to the body x-axis,

$\dot{\delta}_s(x,t)$ is the time rate of change of the fluid sloshing motion relative to the container's deflected longitudinal axis,

x is the undeformed longitudinal axis of the container,

t is time,

x_S is the starting x-station of the fluid, and

x_E is the ending x-station of the fluid.

The distributed mass, m(x), is the sum of the distributed inert mass and the distributed slosh mass, that is,

$$m(x) = m_i(x) + m_s(x).$$

Define

$$R = \frac{SMASS}{M_{TOT}}$$

where SMASS is the slosh mass and is input to the subroutine,

and M_{TOT} is the fluid total mass and is calculated in the subroutine.

Assume that the distributed slosh mass and distributed mass vary in the same ratio as the total respective masses, that is,

$$\frac{m_s(x)}{m(x)} = R.$$

Then Equation (1) can be written as

$$T = \frac{1}{2}\int_{x_S}^{x_E} m(x) \left[(1 - R)\ \dot{\delta}_c^2(x,t) + R\left(\dot{\delta}_c(x,t) + \dot{\delta}_s(x,t)\right)^2\right] dx. \qquad (1a)$$

The technique used here to describe the elastic deflection the container is essentially a modification of the Rayleigh-Ritz method. In the basic Rayleigh-Ritz method the elastic deflection of the container is represented by a summation of assumed displacement functions each weighted by a generalized coordinate representing the contribution of each of the assumed functions. Each displacement function is assumed over the entire container length and considerable skill is required to choose these functions. The modification used here is to assume a simple displacement function between consecutive panel points and to use the panel point displacements as the generalized coordinates. A *cubic* displacement function will be assumed here. This is illustrated in Figure 4 where x_k and x_{k+1} are consecutive panel point stations. The region between panel points k and k+1 is referred to as bay k.

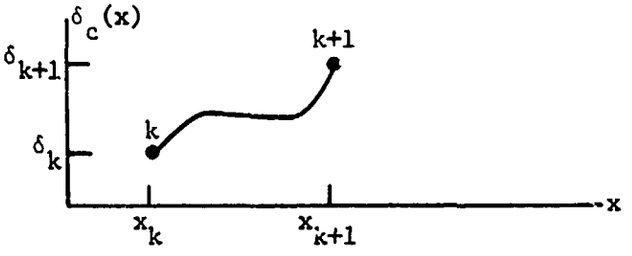


Figure 4 Cubic Displacement Function

The cubic displacement function in bay k is determined as follows. The lateral displacement is given by

$$\delta_c(x) = A(x - x_k)^3 + B(x - x_k)^2 + C(x - x_k) + D$$

or

$$\delta_c(H) = a\,H^3 + b\,H^2 + c\,H + d$$

$$= [H^3\ H^2\ H\ 1] \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \tag{2}$$

where

$$H = (x - x_k)/L_k \tag{2a}$$

with

$$L_k = x_{k+1} - x_k. \tag{2b}$$

H is a local nondimensional bay coordinate to aid in the solution of Equation (1). The angular displacement is obtained as the geometric derivative of the lateral displacement. That is,

$$\theta_c(x) = -\frac{\partial\,\delta_c(x)}{\partial x}$$

or

$$\theta_c(H) = -\frac{d\,\delta_c(H)}{L_k\,dH}$$

$$= \frac{1}{L_k} [-3H^2\ -2H\ -1\ 0] \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}. \tag{3}$$

The coefficients a, b, c, d are determined from Equations (2) and (3) as follows:

$$\begin{bmatrix} \delta_k \\ \delta_{k+1} \\ L_k\ \theta_k \\ L_k\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} H_k^3 & H_k^2 & H_k & 1 \\ H_{k+1}^3 & H_{k+1}^2 & H_{k+1} & 1 \\ -3H_k^2 & -2H_k & -1 & 0 \\ -3H_{k+1}^2 & -2H_{k+1} & -1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

From which

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = [\psi] \begin{bmatrix} \delta_k \\ \delta_{k+1} \\ \theta_k \\ \theta_{k+1} \end{bmatrix}$$

where

$$[\psi] = \begin{bmatrix} 2 & -2 & -L_k & -L_k \\ -3 & 3 & 2L_k & L_k \\ 0 & 0 & -L_k & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Thus, the lateral displacement in bay k is determined in terms of the adjacent panel point lateral and angular displacements. Similarly, the lateral velocity in bay k is given by

$$\dot{\delta}_c(H,t) = [H^3\ H^2\ H\ 1]\ [\psi] \begin{bmatrix} \dot{\delta}_k(t) \\ \dot{\delta}_{k+1}(t) \\ \dot{\theta}_k(t) \\ \dot{\theta}_{k+1}(t) \end{bmatrix}. \qquad (4)$$

The fluid sloshing motion, $\delta_s(x,t)$, relative to the container's deflected longitudinal axis is given as the product of a displacement function, $f_s(x)$, and the slosh amplitude, $\Delta_s(t)$. A *uniform* displacement function, $f_s(x) = G$, will be assumed here. The value of G will be determined later. The time derivative of the sloshing motion is then

$$\dot{\delta}_s(x,t) = G\,\dot{\Delta}_s(t). \tag{5}$$

Combining Equations (4) and (5) gives the total velocity of the fluid as

$$\dot{\delta}_c(H,t) + \dot{\delta}_s(H,t) = [H^3\ H^2\ H\ 1\ G]\,[\bar{\psi}] \begin{bmatrix} \dot{\delta}_k(t) \\ \dot{\delta}_{k+1}(t) \\ \dot{\theta}_k(t) \\ \dot{\theta}_{k+1}(t) \\ \dot{\Delta}_s(t) \end{bmatrix} \tag{6}$$

where

$$[\bar{\psi}] = \begin{bmatrix} 2 & -2 & -L_k & -L_k & 0 \\ -3 & 3 & 2L_k & L_k & 0 \\ 0 & 0 & -L_k & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The distributed mass of the fluid will be considered next. The geometry for a line segment of distributed mass is shown below in Figure 5.

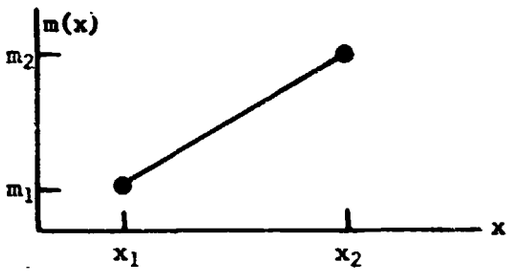


Figure 5 Line Segment Geometry

The equation for a straight line segment as shown in Figure 5 is

$$m(x) = m_1 + (x - x_1)\ (m_2 - m_1)/(x_2 - x_1)$$

or

$$m(H) = WH + m_1 - WH_1 \tag{7}$$

by using the nondimensional coordinate of Equation (2a) and

$$W = (m_2 - m_1)/(H_2 - H_1). \tag{8}$$

Modifying Equation (1a) to use the nondimensional coordinate of Equation (2a) and substituting Equations (4), (6), and (7) gives the kinetic energy of the distributed mass of fluid represented by one line segment i in bay k as

$$T_{i,k} = \frac{1}{2}\begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\\ \dot{\vartheta}_k(t)\\ \dot{\vartheta}_{k+1}(t)\\ \dot{\Delta}_s(t)\end{bmatrix}^T [\bar{\Psi}]^T \left(L_k \int_{H_p}^{H_q} \begin{bmatrix} H^6 & H^5 & H^4 & H^3 & H^3GR \\ H^5 & H^4 & H^3 & H^2 & H^2GR \\ H^4 & H^3 & H^2 & H & HGR \\ H^3 & H^2 & H & 1 & GR \\ H^3GR & H^2GR & HGR & GR & G^2R \end{bmatrix} (WH + m_p - WH_p)\,dH \right) [\bar{\Psi}] \begin{bmatrix}\dot{\delta}_k(t)\\ \dot{\delta}_{k+1}(t)\\ \dot{\vartheta}_k(t)\\ \dot{\vartheta}_{k+1}(t)\\ \dot{\Delta}_s(t)\end{bmatrix} \tag{9}$$

The subscripts p and q have been introduced to handle the possibility of a line segment extending past the bay limits. Thus, x_p $(\text{or } H_p)$ is the greater of x_1 or x_k, and x_q $(\text{or } H_q)$ is the lesser of x_2 or x_{k+1}. Similarly, m_p is either m_1 or m_k, and m_q is either m_2 or m_{k+1}. The integration is continued for the line segment in adjacent bays, if necessary, until the entire line segment has been used. Performing the integration of Equation (9) yields

$$T_{i,k} = \frac{1}{2}\begin{bmatrix} \dot{\delta}_k(t) \\ \dot{\delta}_{k+1}(t) \\ \dot{\xi}_k(t) \\ \dot{\xi}_{k+1}(t) \\ \dot{\lambda}_s(t) \end{bmatrix}\begin{bmatrix} z_{k,k} & z_{k,k+1} & z_{k,k+n} & z_{k,k+n+1} & z_{k,2n+1} \\ & z_{k+1,k+1} & z_{k+1,k+n} & z_{k+1,k+n+1} & z_{k+1,2n+1} \\ & & z_{k+n,k+n} & z_{k+n,k+n+1} & z_{k+n,2n+1} \\ & & & z_{k+n+1,k+n+1} & z_{k+n+1,2n+1} \\ & (\text{sym}) & & & z_{2n+1,2n+1} \end{bmatrix}\begin{bmatrix} \dot{\delta}_k(t) \\ \dot{\delta}_{k+1}(t) \\ \dot{\xi}_k(t) \\ \dot{\theta}_{k+1}(t) \\ \dot{\lambda}_s(t) \end{bmatrix} \tag{10}$$

where

$$z_{k,k} = F_1 - 6F_3 + 4F_4 + P_1 \tag{10a}$$

$$z_{k,k+1} = 3F_3 - 2F_4 - P_1 \tag{10b}$$

$$z_{k,k+n} = \left(-F_2 + 2F_3 - F_4 - P_2\right)L_k \tag{10c}$$

$$z_{k,k+n+1} = \left(F_3 - F_4 - P_3\right)L_k \tag{10d}$$

$$z_{k+1,k+1} = P_1 \tag{10e}$$

$$z_{k+1,k+n} = P_2 L_k \tag{10f}$$

$$z_{k+1,k+n+1} = P_3 L_k \tag{10g}$$

$$z_{k+n,k+n} = \left(F_3 - 4F_4 + 6F_5 - 4F_6 + F_7\right)L_k^2 \tag{10h}$$

$$z_{k+n,k+n+1} = \left(-F_4 + 3F_5 - 3F_6 + F_7\right)L_k^2 \tag{10i}$$

$$z_{k+n+1,k+n+1} = \left(F_5 - 2F_6 + F_7\right)L_k^2 \tag{10j}$$

$$z_{k,2n+1} = \left(F_1 - 3F_3 + 2F_4\right)GR \tag{10k}$$

$$z_{k+1,2n+1} = \left(3F_3 - 2F_4\right)GR \tag{10ℓ}$$

$$z_{k+n,2n+1} = \left(-F_2 + 2F_3 - F_4\right)L_k \, GR \tag{10m}$$

$$z_{k+n+1,2n+1} = \left(F_3 - F_4\right)L_k \, GR \tag{10n}$$

$$z_{2n+1,2n+1} = F_1 \, G^2R \tag{10p}$$

$$P_1 = 9F_5 - 12F_6 + 4F_7 \tag{10q}$$

$$P_2 = -3F_4 + 8F_5 - 7F_6 + 2F_7 \tag{10r}$$

$$P_3 = 3F_5 - 5F_6 + 2F_7 \tag{10s}$$

$$F_j = \left(W\left(H_q^{j+1} - H_p^{j+1}\right)\Big/(j+1) + \left(m_p - WH_p\right)\left(H_q^j - H_p^j\right)\Big/j\right)L_k \tag{10t}$$

$$W = \left(m_q - m_p\right)\Big/\left(H_q - H_p\right) \tag{10u}$$

$$H_p = \left(x_p - x_k\right)\Big/L_k \tag{10v}$$

$$H_q = \left(x_q - x_k\right)\Big/L_k \tag{10w}$$

$$L_k = x_{k+1} - x_k, \text{ and} \tag{10x}$$

n is the number of panel points.

G is determined as follows. Because $z_{2n+1,2n+1}$ is defined to be equal to the slosh mass, SMASS,

$$G = \sqrt{\frac{SMASS}{R\sum F_1}}$$

where the summation is over the number of segments giving the distributed mass of the fluid. From Equation (10t),

$$F_1 = \frac{1}{2}\left(x_q - x_p\right)\left(V_p + V_q\right)$$

which is the fluid mass in the inteval x_p to x_q. The summation of F_1 is then the fluid total mass, M_{TOT}. R was previously defined as the ratio $SMASS/M_{TOT}$. Therefore

$$G = 1. \tag{11}$$

The kernel matrix in the triple matrix product of Equation (10) is the mass matrix that replaces the distributed mass of fluid represented by one line segment i in bay k.

The mass matrix for the entire fluid is obtained by evaluating Equations (10) for each line segment of distributed mass. Every resulting 5x5 bay mass matrix is added to previous 5x5 bay mass matrices at like panel points to form the mass matrix for the entire fluid.

The technique just described was developed by Mr. Carl Bodley and Mr. Herbert Wilkening.

The total mass properties of the fluid are calculated by the following triple matrix product:

$$\left[\begin{array}{c|c} \{1\} & \{PP\} \\ \hline \{0\} & \{-1\} \end{array}\right]^T \left[\begin{array}{c|c} [Z_{\delta\delta}] & [Z_{\delta\theta}] \\ \hline [Z_{\theta\delta}] & [Z_{\theta\theta}] \end{array}\right] \left[\begin{array}{c|c} \{1\} & \{PP\} \\ \hline \{0\} & \{-1\} \end{array}\right] = \left[\begin{array}{cc} M_T & P_T^o \\ P_T^o & I_T^o \end{array}\right]$$

where

$\{1\}$ is a column of ones,

$\{0\}$ is a column of zeroes,

$\{PP\}$ is a column of the panel point x stations,

$[Z_{\delta\delta}]$ etc., are partition of the mass matrix $[Z]$,

M_T is the fluid mass,

P_T^o is the first moment of the luid about x = 0, and

I_T^o is the moment of inertia of the fluid about x = 0.

From this data the center of gravity of the fluid is calculated from

$$x_{cg} = P_T^o / M_T$$

and the moment of inertia of the fluid about x_{cg} is calculated from

$$I_T^{cg} = I_T^o - M_T x_{cg}^2 .$$

Subroutine MODE1 calculates the mode shapes and natural frequencies of a structure using its mass and stiffness matrices. The method of Jacobi is used in the solution of the general eigenvalue problem

$$[A]^{-1}_{N\times N}\ [S]_{N\times N}\ [\Phi]_{N\times N} = [\Phi]_{N\times N}\ [\ \omega^2\ ]_{N\times N} \qquad (1)$$

where the mass matrix is given by [A], the stiffness matrix by [S], and the size of the matrices by N. The mass and stiffness matrices must both be real and symmetric. The mass matrix must also be positive definite. The i,i element of the calculated diagonal matrix $[\ \omega^2\ ]$ is the i^{th} circular frequency squared. The corresponding mode shape is the i^{th} column of $[\phi]$. Because Jacobi's method is used, all mode shapes and frequencies are calculated. The circular frequency squared, ω^2, the circular frequency, ω, and the frequency, $f = \omega/2\pi$, are output from the subroutine in increasing order of magnitude in {W2}, {W}, and {FREQ}, respectively. The mode shapes are altered so that the first element in each column is positive and normalized (automatically due to Jacobi's method) so that $\{\phi\}_i^T\ [A]\ \{\phi\}_i = 1$. Orthogonality checks $[\phi]^T\ [A]\ [\phi]$ and $[\phi]^T\ [S]\ [\phi]$ are calculated and a summary of these results is printed (on option).

DESCRIPTION OF TECHNIQUE

Consider a discrete coordinate model of a structural system having N degrees of freedom. The undamped, free equations of motion for small vibrations about a position of stable equilibrium can be expressed in matrix notation as

$$[A]_{N\times N}\ \ \{\ddot{h}(t)\}_{N\times 1} + [S]_{N\times N}\ \ \{h(t)\}_{N\times 1} = \{0\}_{N\times 1}. \qquad (2)$$

As before, [A] is the mass matrix and [S] is the stiffness matrix. {h(t)} is a column of discrete coordinate displacements. The time dependence in Equation (2) can be removed by the transformation $\{h(t)\} = \alpha\{\phi\}\ e^{j\omega t}$ because the solutions of Equation (2) are assumed to be harmonic in time. Thus, for a nontrivial solution, Equation (2) becomes

$$(-\omega^2[A] + [S])\ \{\phi\} = \{0\}. \qquad (3)$$

Equation (3) is recognized as a general eigenvalue problem of order N with eigenvector $\{\phi\}$ (mode shape) and eigenvalue ω^2 (circular frequency squared). There are N values of ω^2 and $\{\phi\}$ so that the expansion of Equation (3) to include all solutions can be expressed as

$$[A]\left[\{\phi\}_1 \mid \{\phi\}_2 \mid \cdots \mid \{\phi\}_N\right]\begin{bmatrix}\omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_N^2\end{bmatrix} = [S]\left[\{\phi\}_1 \mid \{\phi\}_2 \mid \cdots \mid \{\phi\}_N\right] \tag{4}$$

or

$$[A] \quad [\phi] \quad \lceil \omega^2 \rfloor = [S] \quad [\phi]. \tag{4a}$$

Premultiplying by $[A]^{-1}$ gives

$$[A]^{-1} \quad [S] \quad [\phi] = [\phi] \quad \lceil \omega^2 \rfloor \tag{5}$$

as the eigenvalue problem to be solved in this paper. Jacobi's method cannot be used directly on Equation (5) because even though [A] and [S] are symmetric, their product in general is not.

The first operation in the solution of Equation (5) is to decompose (reference subroutine DCOM1) the mass matrix into triangular factors, that is,

$$[A] = [U]^T [U] \tag{6}$$

where [U] is an upper triangular matrix. Using $[A]^{-1} = \left([U]^T [U]\right)^{-1} = [U]^{-1}\left([U]^T\right)^{-1}$ and premultiplying Equation [5] by [U] gives

$$\left([U]^T\right)^{-1} [S] [\phi] = [U] [\phi] \lceil \omega^2 \rfloor. \tag{7}$$

Define

$$[\bar{\Phi}] = [U]\ [\Phi] \tag{8}$$

from which

$$[\Phi] = [U]^{-1}\ [\bar{\Phi}]. \tag{9}$$

Using the fact that $\left([U]^T\right)^{-1} = \left([U]^{-1}\right)^T$ we get

$$[D]\ [\bar{\Phi}] = [\bar{\Phi}]\ \lceil \omega^2 \rfloor \tag{10}$$

where

$$[D] = \left([U]^{-1}\right)^T\ [S]\ [U]^{-1} \tag{10a}$$

and is referred to as the dynamic matrix in the subroutine. Because the stiffness matrix [S] is symmetric, [D] is symmetric. Equation (10) is the eigenvalue problem from which the eigenvalues $\lceil \omega^2 \rfloor$ and eigenvectors $[\bar{\Phi}]$ are calculated by Jacobi's method (reference Subroutine EIGN1). The product of equation (9) to obtain the mode shapes $[\Phi]$ is accomplished by using $[U]^{-1}$ as the **initial eigenvectors in Subroutine EIGN1, thus eliminating a matrix multiplication.**

Two orthogonality checks of the mode shapes are calculated in the subroutine as follows. First, a summary of the results of $[\Phi]^T\ [A]\ [\Phi]$ are printed. This calculation checks the orthogonality of the mode shapes on the mass matrix $\left(\text{i.e., } \{\phi\}_i^T\ [A]\ \{\phi\}_j = 0\right)$ and the normalization $\left(\text{i.e., } \{\phi\}_i^T\ [A]\ \{\phi\}_i = 1\right)$. Next, the results of $[\Phi]^T\ [S]\ [\Phi]$ are compared with $\lceil \omega^2 \rfloor$. (They should be equal.)

This check results from premultiplying Equation (4a) by $[\Phi]^T$ and assuming $[\Phi]^T\ [A]\ [\Phi] = [I]$. Only the upper half of [A] and [S] are used in the calculation of the mode shapes and frequencies but all of [A] and [S] are used in the orthogonality checks. Thus, errors in the checks can result from nonsymmetric [A] or [S].

Subroutine MODE1A calculates the mode shapes and natural frequencies of a structure using its mass and stiffness matrices. The method of Jacobi is used in the solution of the general eigenvalue problem

$$\left(c[A]_{N\times N} + [S]_{N\times N}\right)^{-1} [A]_{N\times N} [\Phi]_{N\times N} = [\Phi]_{N\times N} \left[\frac{1}{c + \omega^2}\right]_{N\times N} \tag{1}$$

where the mass matrix is given by [A], the stiffness matrix by [S], the size of the matrices by N, and c is a constant automatically calculated in the subroutine. The mass and stiffness matrices must both be real and symmetric. The mass matrix must also be positive definite. The i,i element of the calculated diagonal matrix $\left[\frac{1}{c + \omega^2}\right]$ contains the i^{th} circular frequency squared. The corresponding mode shape is the i^{th} column of $[\Phi]$. Because Jacobi's method is used, all mode shapes and frequencies are calculated. The circular frequency squared, ω^2, the circular frequency, ω, and the frequency, $f = \omega/2\pi$, are output from the subroutine in increasing order of magnitude in {W2}, {W}, and {FREQ}, respectively. The mode shapes are altered so that the first element in each column is positive and normalized (automatically due to Jacobi's method) so that $\{\phi\}_i^T [A] \{\phi\}_i = 1$. Orthogonality checks $[\Phi]^T [A] [\Phi]$ and $[\Phi]^T [S] [\Phi]$ are calculated and a summary of these results is printed **(on option)**.

DESCRIPTION OF TECHNIQUE

Consider a discrete coordinate model of a structural system having N degrees of freedom. The undamped, free equations of motion for small vibrations about a position of stable equilibrium can be expressed in matrix notation as

$$[A]_{N\times N} \{\ddot{h}(t)\}_{N\times 1} + [S]_{N\times N} \{h(t)\}_{N\times 1} = \{0\}_{N\times 1}. \tag{2}$$

As before, [A] is the mass matrix and [S] is the stiffness matrix. {h(t)} is a column of discrete coordinate displacements. The time dependence in Equation (2) can be removed by the transformation $\{h(t)\} = \alpha\{\phi\} e^{j\omega t}$ because the solutions of Equation (2) are assumed to be harmonic in time. Thus, for a nontrivial solution, Equation (2) becomes

$$\left(-\omega^2[A] + [S]\right) \{\phi\} = \{0\}. \tag{3}$$

Equation (3) is recognized as a general eigenvalue problem of order N with eigenvector $\{\phi\}$ (mode shape) and eigenvalue ω^2 (circular frequency squared). There are N values of ω^2 and $\{\phi\}$ so that the expansion of Equation (3) to include all solutions can be expressed as

$$[A]\left[\{\phi\}_1 \mid \{\phi\}_2 \mid \cdots \mid \{\phi\}_N\right]\begin{bmatrix}\omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_N^2\end{bmatrix} = [S]\left[\{\phi\}_1 \mid \{\phi\}_2 \mid \cdots \mid \{\phi\}_N\right] \tag{4}$$

or

$$[A]\,[\phi]\,[\omega^2] = [S]\,[\phi]. \tag{4a}$$

Adding $c[A]\,[\phi]$, where c is a constant, to both sides of the equation results in

$$(c[A] + [S])^{-1}\,[A]\,[\phi] = [\phi]\left[\frac{1}{c + \omega^2}\right] \tag{5}$$

as the eigenvalue problem to be solved in this paper. The constant, c, is calculated from

$$c = \frac{\text{Norm (S)}}{\text{Norm (A)}}$$

where the Norm of a matrix is defined as the sum of the absolute values of the elements of the matrix. Jacobi's method cannot be used directly on Equation (5) because even though [A] and [S] are symmetric, their product in general is not.

The first operation in the solution of Equation (5) is to decompose (reference Subroutine DCOM1) into triangular factors, that is,

$$c\,[A] + [S] = [U]^T\,[U] \tag{6}$$

where [U] is an upper triangular matrix.

Using $(c\,[A] + [S])^{-1} = [U]^{-1}\,([U]^T)^{-1}$, premultiplying Equation (5) by [U], and using a superscript on $[\phi]$ gives $([U]^T)^{-1}\,[A]\,[\hat{\phi}] = [U]\,[\hat{\phi}]\left[\frac{1}{c+\omega^2}\right]$. (7)

Next, we define

$$[\bar{\Phi}] = [U]\,[\hat{\Phi}] \tag{8}$$

from which

$$[\hat{\Phi}] = [U]^{-1}\,[\bar{\Phi}]. \tag{9}$$

Using the fact $([U]^T)^{-1} = ([U]^{-1})^T$ we get

$$[D]\,[\bar{\Phi}] = [\bar{\Phi}]\,[\lambda] \tag{10}$$

where

$$[D] = ([U]^{-1})^T\,[A]\,[U]^{-1} \tag{10a}$$

and

$$[\lambda] = \left[\frac{1}{c + \omega^2}\right]. \tag{10b}$$

[D] is referred to as the dynamic matrix in the subroutine. Because the mass matrix [A] is symmetric, [D] is symmetric. Equation (10) is the eigenvalue problem from which the eigenvalues $[\lambda]$ and eigenvectors $[\bar{\Phi}]$ are calculated by Jacobi's method (reference Subroutine EIGN1). The product of Equation (9) to obtain the mode shapes $[\hat{\Phi}]$ is accomplished by using $[U]^{-1}$ as the initial eigenvectors in Subroutine EIGN1, thus eliminating a matrix multiplication. The mode shapes $[\hat{\Phi}]$ must be renormalized because where $[\Phi]^T\,[A]\,[\Phi] = [I]$ is wanted, we have $[\hat{\Phi}]^T\,[A]\,[\hat{\Phi}] = [\lambda]$. This last relation is obtained from Equations (10), (10a), and (8) and using the orthonormal properties of $[\bar{\Phi}]$, that is, $[\bar{\Phi}]^{-1} = [\bar{\Phi}]^T$. The final renormalized mode i is obtained from

$$\{\Phi\}_i = \frac{\{\hat{\Phi}\}_i}{\sqrt{\lambda_i}}. \tag{11}$$

The i^{th} circular frequency squared, ω_i^2, is obtained from Equation (10b) as

$$\omega_i^2 = \frac{1}{\lambda_i} - c. \tag{12}$$

Two orthogonality checks of the mode shapes are calculated (on option) in the subroutine as follows. First a summary of the results of $[\Phi]^T\,[A]\,[\Phi]$ are printed. This calculation checks

the orthogonality of the mode shapes on the mass matrix (i.e. $\{\phi\}_i^T [A] \{\phi\}_j = 0$) and the normalization (i.e. $\{\phi\}_i^T [A] \{\phi\}_i = 1$). Next, the results of $[\Phi]^T [S] [\Phi]$ are compared with $[\omega^2]$. (They should be equal.) This check results from premultiplying Equation (4a) by $[\Phi]^T$ and assuming $[\Phi]^T [A] [\Phi] = [I]$. Only the upper half of [A] and [S] are used in the calculation of the mode shapes and frequencies, but all of [A] and [S] are used in the orthogonality checks. Thus, errors in the checks can result from nonsymmetric [A] or [S].

Subroutine MODE1B calculates the mode shapes and natural frequencies of a structure using its mass and flexibility matrices. The method of Jacobi is used in the solution of the general eigenvalue problem

$$[E]_{N\times N}\,[A]_{N\times N}\,[\Phi]_{N\times N} = [\Phi]_{N\times N}\left[\frac{1}{\omega^2}\right]_{N\times N} \tag{1}$$

where the mass matrix is given by [A], the flexibility matrix by [E], and the size of the matrices by N. The mass and flexibility matrices must both be real and symmetric. The mass matrix must also be positive definite. The i,i element of the calculated diagonal matrix $\left[\frac{1}{\omega^2}\right]$ is the reciprocal of the i^{th} circular frequency squared. The corresponding mode shape is the i^{th} column of [Φ]. Because Jacobi's method is used, all mode shapes and frequencies are calculated. The circular frequency squared, ω^2, the circular frequency, ω, and the frequency, f = ω/2π, are output from the subroutine in increasing order of magnitude in {W2}, {W}, and {FREQ}, respectively. Rigid body frequencies and mode shapes will be in the last positions of {W2}, {W}, {FREQ}, and [Φ]. This is because $\frac{1}{\omega^2}$ will be calculated as zero (or a small number due to computer roundoff error) corresponding to the rigid body modes. The mode shapes are altered so that the first element in each column is positive and normalized (automatically due to Jacobi's method) so that $\{\phi\}_i^T\,[A]\,\{\phi\}_i = 1$. Orthogonality checks $[\Phi]^T\,[A]\,[\Phi]$ and $[\Phi]^T\,[A]\,[E]\,[A]\,[\Phi]$ are calculated and a summary of these results is printed (on option).

DESCRIPTION OF TECHNIQUE

Consider a discrete coordinate model of a structural system having N degrees of freedom. The undamped, free equations of motion for small vibrations about a position of stable equilibrium can be expressed in matrix notation as

$$[A]_{N\times N}\,\{\ddot{h}(t)\}_{N\times 1} + [S]_{N\times N}\,\{h(t)\}_{N\times 1} = \{0\}_{N\times 1}. \tag{2}$$

As before, [A] is the mass matrix, [S] is the stiffness matrix, and {h(t)} is a column of discrete coordinate displacements. The time dependence in Equation (2) can be removed by the transformation $\{h(t)\} = \alpha\{\phi\}\,e^{j\omega t}$ because the solutions of Equation (2) are assumed to be harmonic in time. Thus, for a nontrivial solution, Equation (2) becomes

$$(-\omega^2[A] + [S])\ \{\phi\} = \{0\}. \tag{3}$$

Equation (3) is recognized as a general eigenvalue problem of order N with eigenvector $\{\phi\}$ (mode shape) and eigenvalue ω^2 (circular frequency squared). There are N values of ω^2 and $\{\phi\}$ so that the expansion of Equation (3) to include all solutions can be expressed as

$$[A]\ \left[\{\phi\}_1 \mid \{\phi\}_2 \mid \cdots \mid \{\phi\}_N\right] \begin{bmatrix} \omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_N^2 \end{bmatrix} = [S]\ \left[\{\phi\}_1 \mid \{\phi\}_2 \mid \cdots \mid \{\phi\}_N\right] \tag{4}$$

or

$$[A]\ [\Phi]\ \lceil \omega^2 \rfloor = [S]\ [\Phi]. \tag{4a}$$

Assume the structure is grounded so that the stiffness matrix is nonsingular and its inverse exists. The inverse of the stiffness matrix is defined to be the flexibility matrix, that is,

$$[E] = [S]^{-1} \tag{5}$$

Using this relationship in Equation (4a) and postmultiplying by $\lceil \omega^2 \rfloor^{-1}$ gives

$$[E]\ [A]\ [\Phi] = [\Phi] \left\lceil \frac{1}{\omega^2} \right\rfloor \tag{6}$$

as the eigenvalue problem to be solved in this paper. If the structure is not grounded, the stiffness matrix, [S], is singular, and Equation (5) is not applicable. The flexibility matrix, [E], can still be obtained by other techniques (either directly or by operation on the stiffness matrix) so that Equation (6) is still valid. These techniques will not be discussed here. Jacobi's method cannot be used directly on Equation (6) because even though [E] and [A] are symmetric, their product in general is not.

The first operation in the solution of Equation (6) is to decompose (reference Subroutine DCOM1) the mass matrix into triangular factors, that is

$$[A] = [U]^T [U] \tag{7}$$

where [U] is an upper triangular matrix. Next, we define

$$[\bar{\Phi}] = [U] [\Phi] \tag{8}$$

from which

$$[\Phi] = [U]^{-1} [\bar{\Phi}]. \tag{9}$$

Using these three equations in Equation (6) and premultiplying by [U] gives

$$[D] [\bar{\Phi}] = [\bar{\Phi}] \lceil \lambda \rfloor \tag{10}$$

where

$$[D] = [U] [E] [U]^T \tag{10a}$$

and

$$\lceil \lambda \rfloor = \left\lceil \frac{1}{\omega^2} \right\rfloor . \tag{10b}$$

[D] is referred to as the dynamic matrix in the subroutine. Because the flexibility matrix [E] is symmetric, [D] is symmetric. Equation (10) is the eigenvalue problem from which the eigenvalue $\lceil \lambda \rfloor$ and eigenvectors $[\bar{\Phi}]$ are calculated by Jacobi's method (reference Subroutine EIGN1). The product of equation (9) to obtain the mode shapes $[\Phi]$ is accomplished by using $[U]^{-1}$ as the initial eigenvector in Subroutine EIGN1, thus eliminating a matrix multiplication. The i^{th} circular frequency squared ω_i^2, is obtained from Equation (10b) as

$$\omega_i^2 = \frac{1}{\lambda_i} . \tag{11}$$

Two orthogonality checks of the mode shapes are calculated (on option) in the subroutine as follows. First, a summary of the results of $[\Phi]^T [A] [\Phi]$ are printed. This calculation checks the orthogonality of the mode shapes on the mass matrix (i.e. $\{\phi\}_i^T [A] \{\phi\}_j = 0$)

and the normalization $\left(\text{i.e } \{\phi\}_i^T [A] \{\phi\}_i = 1\right)$. Next the results of $[\Phi]^T [A] [E] [A] [\Phi]$ are compared with $\lceil -1/\omega^2 \rfloor$. (They should be equal.) This check results from premultiplying Equation (6) by $[\Phi]^T [A]$ and assuming $[\Phi]^T [A] [\Phi] = [1]$. Only the upper half of [A] and [E] are used in the calculation of the mode shapes and frequencies but all of [A] and [E] are used in the orthogonality checks. Thus errors in the checks can result from nonsymmetric [A] or [E].

Subroutine MODE1X calculates the mode shapes and natural frequencies of a structure using its mass and stiffness matrices. This subroutine is a modification of Subroutine MODE1 to allow a nonpositive definite mass matrix, remove orthogonality checks, use ω^2 convergence tolerance, ω, f not calculated, and removal of making first element of each mode positive. The method of Jacobi is used in the solution of the general eigenvalue problem

$$[A]^{-1}_{N\times N}\ [S]_{N\times N}\ [\phi]_{N\times N} = [\phi]_{N\times N}\ [\omega^2]_{N\times N} \qquad [1]$$

where the mass matrix is given by [A], the stiffness matrix by [S], and the size of the matrices by N. The mass and stiffness matrices must both be real and symmetric. The i,i element of the calculated diagonal matrix $[\omega^2]$ is the i^{th} circular frequency squared. The corresponding mode shape is the i^{th} column of $[\phi]$. Because Jacobi's method is used, all mode shapes and frequencies are calculated. The circular frequency squared, ω^2, is output from the subroutine in increasing order of magnitude in {W2}. The mode shapes are normalized (automatically due to Jacobi's method) so that $\{\phi\}_i^T\ [A]\ \{\phi\}_i = 1$.

DESCRIPTION OF TECHNIQUE

Consider a discrete coordinate model of a structural system having N degrees of freedom. The undamped, free equations of motion for small vibrations about a position of stable equilibrium can be expressed in matrix notation as

$$[A]_{N\times N}\ \ddot{\{h(t)\}}_{N\times 1} + [S]_{N\times N}\ \{h(t)\}_{N\times 1} = \{0\}_{N\times 1}. \qquad [2]$$

As before, [A] is the mass matrix and [S] is the stiffness matrix. {h(t)} is a column of discrete coordinate displacements. The time dependence in Eq [2] can be removed by the transformation $\{h(t)\} = \alpha\{\phi\}\ e^{j\omega t}$ because the solutions of Equation (2) are assumed to be harmonic in time. Thus, for a nontrivial solution, Eq [2] becomes

$$(-\omega^2[A] + [S])\ \{\phi\} = \{0\}\ . \qquad [3]$$

Equation [3] is recognized as a general eigenvalue problem of order N with eigenvector {ϕ} (mode shape) and eigenvalue ω^2 (circular frequency squared). There are N values of ω^2 and {ϕ} so that the expansion of Eq [3] to include all solutions can be expressed as

$$[A]\ [\{\phi\}_1 \mid \{\phi\}_2 \mid \cdots \mid \{\phi\}_N] \begin{bmatrix} \omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_N^2 \end{bmatrix} = [S]\ [\{\phi\}_1 \mid \{\phi\}_2 \mid \cdots \mid \{\phi\}_N] \qquad [4]$$

or

$$[A]\ [\Phi]\ [\omega^2] = [S]\ [\Phi]. \qquad [4a]$$

Premultiplying by $[A]^{-1}$ gives

$$[A]^{-1}\ [S]\ [\Phi] = [\Phi]\ [\omega^2] \qquad [5]$$

as the eigenvalue problem to be solved in this paper. Jacobi's method cannot be used directly on Eq [5] because even though [A] and [S] are symmetric, their product in general is not.

The first operation in the solution of Eq [5] is to decompose the mass matrix into triangular factors, that is,

$$[A] = [U]^T\ [U] \qquad [6]$$

where [U] is an upper triangular matrix. Modified statements from Subroutines DCOM1 are used, that is the square-root of absolute values are used. This allows for a nonpositive definite mass matrix. Using $[A]^{-1} = ([U]^T\ [U])^{-1} = [U]^{-1}\ ([U]^T)^{-1}$ and premultiplying Eq [5] by [U] gives

$$([U]^T)^{-1}\ [S]\ [\Phi] = [U]\ [\Phi]\ [\omega^2]. \qquad [7]$$

Define

$$[\bar{\Phi}] = [U]\ [\Phi] \qquad [8]$$

from which

$$[\Phi] = [U]^{-1}\ [\bar{\Phi}]. \qquad [9]$$

Using the fact that $([U]^T)^{-1} = ([U]^{-1})^T$ we get

$$[D]\ [\bar{\Phi}] = [\bar{\Phi}]\ [\omega^2] \qquad [10]$$

where

$$[D] = ([U]^{-1})^T\ [S]\ [U]^{-1} \qquad [10a]$$

and is referred to as the dynamic matrix in the subroutine, Because the stiffness matrix [S] is symmetric, [D] is symmetric. Eq [10] is the eigenvalue problem from which the eigenvalues $[\omega^2]$ and eigenvectors $[\bar{\phi}]$ are calculated by Jacobi's method (reference Subroutine EIGN1A). The product of Eq [9] to obtain the mode shapes $[\Phi]$ is accomplished by Using $[U]^{-1}$ as the initial eigenvectors in Subroutine EIGN1A, thus eliminating a matrix multiplication.

Subroutine MULT calculates the product of two matrices. In matrix notation,

$$[Z]_{NRA \times NCB} = [A]_{NRA \times NRB} \, [B]_{NRB \times NCB}$$

where

$$z_{ij} = \sum_{k=1}^{NRB} a_{ik} \, b_{kj} \qquad \begin{pmatrix} i = 1, \, NRA \\ j = 1, \, NCB \end{pmatrix}$$

NRA is the number of rows of [A] and [Z]. NRB is the number of rows of [B] and the number of columns of [A]. NCB is the number of columns of [B] and [Z]. The number of columns of [A] must be equal to the number of rows of [B].

Theorem: Multiplication of matrices is not commutative in general. That is,

$$[A][B] \neq [B][A]$$

for any values of a_{ij} and b_{ij}.

Theorem: Multiplication of matrices is associative. That is,

$$[A] \left([B][C]\right) = \left([A][B]\right) [C].$$

Theorem: Multiplication of matrices is distributive. That is,

$$[A] \left([B] + [C]\right) = [A][B] + [A][C].$$

EXAMPLE

Consider input of $[A]_{1 \times 2} = [1. \quad 2.]$ and $[B]_{2 \times 3} = \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}$.

The reader can easily verify the output to be

$$[Z]_{1 \times 3} = [1. \quad 2.] \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}$$

$$= [27. \quad 14. \quad 33.].$$

Subroutine MULTA calculates the product of two matrices. This subroutine is a modification of Subroutine MULT to allow larger matrix sizes by placing the answer [Z] in the same core locations as [A]. In matrix notation,

$$[Z]_{NRAxNCB} = [A]_{NRAxNRB}\ [B]_{NRBxNCB}$$

where

$$z_{ij} = \sum_{k=1}^{NRB} a_{ik}\ b_{kj} \qquad \begin{pmatrix} i = 1, NRA \\ j = 1, NCB \end{pmatrix}$$

NRA is the number of rows of [A] and [Z]. NRB is the number of rows of [B] and the number of columns of [A]. NCB is the number of columns of [B] and [Z]. The number of columns of [A] must be equal to the number of rows of [B].

Theorem: Multiplication of matrices is not commutative in general. That is,

$$[A][B] \neq [B][A]$$

for any values of a_{ij} and b_{ij}.

Theorem: Multiplication of matrices is associative. That is,

$$[A]\ \big([B][C]\big) = \big([A][B]\big)\ [C].$$

Theorem: Multiplication of matrices is distributive. That is,

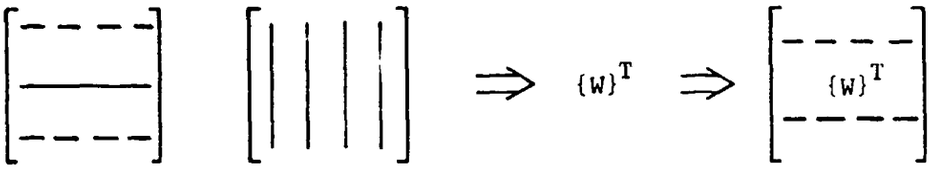
$$[A]\ \big([B] + [C]\big) = [A][B] + [A][C].$$

DESCRIPTION OF TECHNIQUE

To accomplish matrix multiplication using only two matrix core spaces, an intermediate work space vector is used in the subroutine. The size of this work vector determines the limitation on the number of columns of [B] (i.e., NCB). The matrix multiplication is accomplished as follows. A single row of [A] is multiplied times the columns of [B]. These results are stored in the work vector until all the columns of [B] have been used.

The elements in the work vector then replace the row of [A] used in the multiplication. This procedure is repeated for all the rows of [A]. Schematically, the procedure can be represented by

$$[A_0] \qquad [B] \qquad \{\text{work vector}\}^T \qquad [A_1]$$



where $[A_0]$ is the original [A], $[A_1] = [Z]$ is the answer.

EXAMPLE

Consider input of $[A]_{1\times2} = [1. \quad 2.]$ and $[B]_{2\times3} = \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}$.

The reader can easily verify the output to be

$$[Z]_{1\times3} = [1. \quad 2.] \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}$$

$$= [27. \quad 14. \quad 33.].$$

Subroutine MULTB calculates the product of two matrices. This subroutine is a modification of Subroutine MULT to allow larger matrix sizes by placing the answer [Z] in the same core locations as [B]. In matrix notation,

$$[Z]_{NRAxNCB} = [A]_{NRAxNRB}\ [B]_{NRBxNCB}$$

where

$$z_{ij} = \sum_{k=1}^{NRB} a_{ik}\ b_{kj} \qquad \begin{pmatrix} i = 1, NRA \\ j = 1, NCB \end{pmatrix}$$

NRA is the number of rows of [A] and [Z]. NRB is the number of rows of [B] and the number of columns of [A]. NCB is the number of columns of [B] and [Z]. The number of columns of [A] must be equal to the number of rows of [B].

Theorem: Multiplication of matrices is not commutative in general. That is,

$$[A][B] \neq [B][A]$$

for any values of a_{ij} and b_{ij}.

Theorem: Multiplication of matrices is associative. That is,

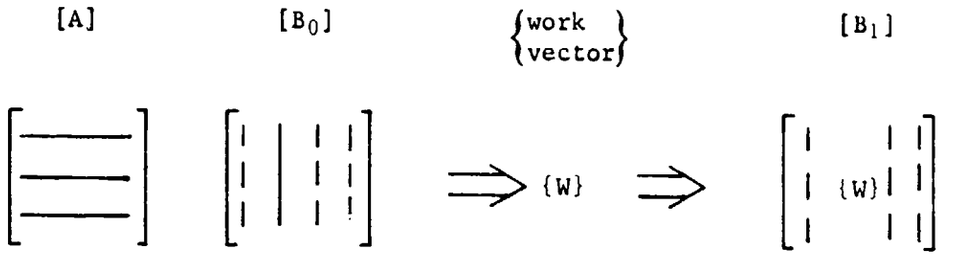
$$[A]\ \big([B][C]\big) = \big([A][B]\big)\ [C].$$

Theorem: Multiplication of matrices is distributive. That is,

$$[A]\ \big([B] + [C]\big) = [A][B] + [A][C].$$

DESCRIPTION OF TECHNIQUE

To accomplish matrix multiplication using only two matrix core spaces, an intermediate work space vector is used in the subroutine. The size of this work vector determines the limitation on the number of rows of [A] (i.e., NRA). The matrix multiplication is accomplished as follows. The rows of [A] are multiplied times a single column of [B]. These results are stored in the work vector until all the rows of [A] have been used. The elements in the work vector then replace the column of [B] used in the multiplication. This procedure is repeated for all the columns of [B]. Schematically, this procedure can be represented by



where $[B_0]$ is the original [B], $[B_1]$ = [Z] is the answer.

EXAMPLE

Consider input of $[A]_{1\times2}$ = [1. 2.] and $[B]_{2\times3} = \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}$.
The reader can easily verify the output to be

$$[Z]_{1\times3} = [1. \quad 2.] \begin{bmatrix} 7. & -8. & 9. \\ 10. & 11. & 12. \end{bmatrix}$$

$$= [27. \quad 14. \quad 33.].$$

Subroutine ONES generates a matrix with each element equal to one. That is,

$$z_{ij} = 1. \qquad \begin{pmatrix} i = 1, NR \\ j = 1, NC \end{pmatrix}$$

In matrix notation,

$$[Z]_{NRxNC} = \begin{bmatrix} 1. & 1. & \cdots & 1. \\ 1. & 1. & & \vdots \\ \vdots & & & \\ 1. & . & . & 1. \end{bmatrix}$$

where NR is the number of rows of [Z]; NC is the number of columns of [Z].

Subroutine ONRBM calculates orthonormal rigid body modes $[\Phi]_{N\times NRBM}$ from any rigid body modes $[\psi]_{N\times NRBM}$ and mass matrix $[A]_{N\times N}$, where N is the number of degrees of freedom and NRBM is the number of rigid body modes. The orthonormal rigid body modes are orthogonal such that $\{\phi\}_i^T[A]\{\phi\}_j = 0$ $(i\neq j)$ where $\{\phi\}$ is one rigid body mode (a column) of $[\Phi]$ and i, j refer to the column number of $[\Phi]$. The orthonormal rigid body modes are normalized such that $\{\phi\}_i^T[A]\{\phi\}_i = 1$.

DESCRIPTION OF TECHNIQUE

Perform the triple matrix product

$$[\psi]^T[A][\psi] = [B]. \tag{1}$$

Calculate the eigenvalues $[\lambda]$ and eigenvectors [E] of [B]. By definition,

$$[E]^T[B][E] = [\lambda] \tag{2}$$

$$\text{or } [B] = [E]^{-T}[\lambda][E]^{-1}$$

$$= [E]^{-T}[\sqrt{\lambda}][\sqrt{\lambda}][E]^{-1} \tag{3}$$

Substitution of Equation (3) into Equation (1), premultiplying by $[E]^T$, and postmultiplying by [E] yields

$$[E]^T[\psi]^T[A][\psi][E] = [\sqrt{\lambda}][\sqrt{\lambda}]$$

$$\text{or } \left[\frac{1}{\sqrt{\lambda}}\right][E]^T[\psi]^T[A][\psi][E]\left[\frac{1}{\sqrt{\lambda}}\right] = [I].$$

The orthonormal rigid body modes are then defined as

$$[\Phi] = [\psi][E]\left[\frac{1}{\sqrt{\lambda}}\right].$$

Subroutine PAGEHD brings up a new page and prints a heading at the top of the page. This heading consists of:

1) Run number;

2) Date;

3) Page number - Initialized as zero in subroutine START and incremented by one each time Subroutine PAGEHD is entered;

4) User's name;

5) Title Card 1;

6) Title Card 2.

Each of the above items was obtained in Subroutine START and transferred by a COMMON block labeled LSTART.

Subroutine PLOT1 plots points on a graph with linear x and y axes. The x, y coordinates of the points are supplied to the subroutine in {XVEC} and [YMAT], respectively. Each column of [YMAT] is considered a separate curve with a maximum of three curves allowed. On option (IFCURV = 1), the points are connected by straight line segments. All curves will have the same y scale.

Subroutine PLOT2 plots points on a semilog or log-log graph, and connects the points with straight line segments. The input value of IPLOT determines the type of axes. IPLOT = 1 gives a log x axis and a linear y axis, IPLOT = 2 gives a linear x axis and a log y axis, IPLOT = 3 gives both log x, y axes. The x, y coordinates of the points are supplied to the subroutine in {XVEC} and [YMAT], respectively. Each column of [YMAT] is considered a separate curve with a maximum of ten curves allowed. All curves will have the same y scale.

Subroutine PLOT3 plots points and connects the points with straight line segments to produce (on option) a perspective view or a stereo pair of a three-dimensional object. The x, y, z coordinates of the i^{th} point of the object comprise the i^{th} row of the input matrix [CLOC]. The points to be connected are specified in [MLOC]. Interpolation is done if a point falls outside of the *cone of vision* (usually about 60 deg) of the viewer.

Subroutine PLOTSS selects the scale and calculates the top and bottom values of a 10-step linear scale from the imput values of the maximum and minimum values to be plotted. Each step (1/10 of the total scale) will have a value of .5, 1, or 2.

Subroutine PSD1 calculates power spectral density response and variance due to a power spectral density forcing function. The basic equation is given in the frequency domain as

$$\left(-\omega^2 \left[M\right] + i\,\omega \left[D\right] + \left[K\right]\right) \left\{q(\omega)\right\} = \left\{F\right\} \quad u(\omega) \qquad (1)$$

where M is the mass matrix

D is the damping matrix

K is the stiffness matrix

F is the force distribution

u is the force variation

The response $\{q(\omega)\}$ could be obtained from

$$\left\{q(\omega)\right\} = \left(-\omega^2 \left[M\right] + i\,\omega \left[D\right] + \left[K\right]\right)^{-1} \left\{F\right\} u(\omega) \qquad (2)$$

The disadvantage of this approach is that a complex matrix inversion must be performed for each ω used. Because each matrix inversion requires much computer time, the approach of equation (2) is not used in this subroutine PSD1. Rather, the following approach used.

The second order differential equation can be put into an equivalent first order differential equation given as

$$\begin{bmatrix} \{\ddot{q}(t)\} \\ \{\dot{q}(t)\} \end{bmatrix} = \left[\begin{array}{c|c} -\left[M\right]^{-1}\left[D\right] & -\left[M\right]^{-1}\left[K\right] \\ \hline \left[I\right] & \left[0\right] \end{array}\right] \begin{bmatrix} \{\dot{q}(t)\} \\ \{q(t)\} \end{bmatrix} + \left[\begin{array}{c} \left[M\right]^{-1}\{F\} \\ \hline \{0\} \end{array}\right] u(t) \qquad (3)$$

The response is expressed in terms of numerator and denominator containing products of first and second order polynomials permitting rapid evaluation at many ω. Calculation of the complex roots uses considerable computer time but this technique is still faster than the technique of equation (2) when solutions at many ω are sought.

$$\text{Define } \left[A^*\right] = \left[\begin{array}{c|c} -\left[M\right]^{-1}\left[D\right] & -\left[M\right]^{-1}\left[K\right] \\ \hline \left[I\right] & \left[0\right] \end{array}\right] \qquad (3a)$$

$$\{Z(t)\} = \begin{bmatrix} \{\dot{q}(t)\} \\ \{q(t)\} \end{bmatrix} \tag{3b}$$

$$\{b\} = \begin{bmatrix} [M]^{-1}\{F\} \\ \{0\} \end{bmatrix} \tag{3c}$$

to give $\{\dot{Z}(t)\} = [A^*]\ \{Z(t)\} + \{b\}\ u(t)$ (4)

The system characteristic matrix $[A^*]$ provides the basis for evaluating the resonant characteristics (natural frequencies) of the system described by matrices $[M]$, $[D]$, and $[K]$.

Taking the Laplace transform of equation (4) gives

$$\left[[Is] - [A^*]\right]\ \{Z(s)\} = \{b\}\ u(s) \tag{5}$$

Employ Cramer's Rule to evaluate the pth element of $\{Z(s)\}$ due to an input u, that is

$$\frac{Z_p(s)}{u(s)} = \frac{\text{Aug}\ |Is - A^*|}{|Is - A^*|} \tag{6}$$

where Aug $|Is - A^*|$ is accomplished by placing $\{b\}$ into column p of $|Is - A^*|$.

It is desired to evaluate both the numerator and denominator roots of equation (6). The Q-R algorithm (Reference J.G.F. Francis, "The QR-Transformation - A Unitary Analogue to the LR-Transformation", The Computer Journal, Volume 4, October 1961, (Part 1) and Volume 5, January 1962 (Part 2).) is a useful tool to extract the roots of the complex system and is used here.

The denominator root extraction is straight forward in that we wish to find $p_1, p_2, p_3, \ldots p_n$ from an expression of the form

$$D(s) = \det\left([I]s - [A^*]\right) \tag{7}$$

such that

$$D(s) = (s - p_1)\ (s - p_2)\ \ldots\ (s - p_N)$$

$$= \prod_{i=1}^{N} (s - p_i)\ . \qquad (8)$$

This evaluation is made by extracting the characteristic roots of the matrix $[A^*]$. In general, these roots will be complex because $[A^*]$ is not symmetric.

The numerator root extraction is more complicated in that the augmented matrix can very easily have a zero pivot element from $\{b\}$. Thus, the null p_i's are eliminated from the expression giving the characteristic polynomial of order n less than N. It is a rather common occurence for the number of zeroes (order of the numerator N(s)) to be less than the number of poles (order of the denominator D(s)). The numerator expression N(s) can be written as

$$N_s = k_R \prod_{i=1}^{n} (s - s_i) \qquad (9)$$

where the numerator root gain is

$$k_R = (-1)^{N-n} \prod_{i=1}^{n} p_i \ \det \left|\tilde{A}^* - \tilde{I}x\right| \qquad (10)$$

$\tilde{I}$ is the identity matrix of size N with a null diagonal element. $\tilde{A}^*$is the A^* matrix with a negative column of $\{b\}$ in the pth column; and the Bode gain for the numerator is

$$k_B = k_R\ (-1)^m \prod_{i=1}^{m} s_i \qquad \text{where } m \leq n. \qquad (11)$$

Transfer function poles, zeros, and root gain can be converted to the standard Bode form for frequency response by combining time constants (τ), damping (ζ) and resonant frequencies (ω) as

$$TF = k_B \frac{s^r \prod_{i=1}^{N_1} (1 + \tau_i s) \prod_{i=1}^{N_2} \left(1 + \frac{2\zeta_i s}{\omega_i} + \frac{s^2}{\omega_i^2}\right)}{\prod_{j=1}^{M_1} (1 + \tau_j s) \prod_{j=1}^{M_2} \left(1 + \frac{2\zeta_j s}{\omega_j} + \frac{s^2}{\omega_j^2}\right)} \qquad (12)$$

where the composite Bode gain is

$$k_B = k_R \frac{\prod_{i=1}^{n} z_i}{\prod_{j=1}^{m} p_j} \qquad (13)$$

z_i is a root of the numerator and, p_j is a root of the denominator.

The frequency response is then calculated by substituting $j\omega$ for s and evaluating the transfer function expression (that is $Z_p(s)$) at various ω. In this subroutine the first and last ω and the number of ω wanted is specified by the user.

The additional equations are obtained from

$$\{L(t)\} = [AA] \{\ddot{q}(t)\} + [BB] \{\dot{q}(t)\} + [CC] \{q(t)\} + \{DD\}$$

$$\text{or} \quad \{L(\omega)\} = \left(-\omega^2 [AA] + i\omega [BB] + [CC]\right) \left(\{q_R(\omega)\} + i \{q_I(\omega)\}\right) + \{DD\}$$

which can be split into real and imaginary parts as

$$\{L_R(\omega)\} = \left(-\omega^2 [AA] + [CC]\right) \{q_R(\omega)\} - [BB] \{q_I(\omega)\} + \{DD\}$$

$$\{L_I(\omega)\} = \left(-\omega^2 [AA] + [CC]\right) \{q_I(\omega)\} + [BB] \{q_R(\omega)\}$$

Note that $\{q(\omega)\}$ (real, imaginary) is the second half of $\{Z(\omega)\}$, see equation (3b).

Although the modal response is calculated for each modal coordinate in turn for many frequencies, the need in the calculation of the additional equations is for all of the modal coordinate responses at each frequency in turn. To avoid extensive tape reading, it was decided to place the modal responses into core for the additional equations calculation. Although this limits the number of frequencies that can be used, the loss in accuracy in calculating the variance should be minor.

A constant step size between a specified minimum and maximum frequency is used. Because the area under the PSD-frequency curve (variance) is the final result, using a constant frequency step size gives a reasonable result even though some individual peaks may be missed.

The PSD of the additional equations (at one station here) is given by

$$L(\omega) = H^2(\omega)\ u(\omega)$$

where $H^2(\omega) = L_R^2(\omega) + L_I^2(\omega)$

and $u(\omega)$ is the PSD-ω variation of the forcing function $\{F\}$. The PSD-ω variation is given in the input by TABF, TABW.

The variance of the additional equation is

$$\bar{L}^2 = \frac{1}{2\pi} \int_{\omega_1}^{\omega_e} L(\omega)\ d\omega$$

The square root of the variance (or mean square response) is the standard deviation. Printout (on option) in the subroutine includes:

(1) $- A^* - \equiv \left[A^*\right] = \left[\begin{array}{c|c} -M^{-1} D & -M^{-1} K \\ \hline I & 0 \end{array}\right]$

(2) BCOL $\equiv \{b\}$

(3) R AR gives the real and imaginary parts of the roots of $\left[A^*\right]$.

RART gives the real and imaginary parts of the roots of $\left[A^*\right]$ transpose.

(4) RRED (printed in subroutine PSD1)

RRED(1) is the number of numerator reals
RRED(2) is the number of numerator complex pairs
RRED(3) is the number of numerator zeroes
RRED(4) is the number of denominator reals
RRED(5) is the number of denominator complex pairs
RRED(6) is the number of denominator zeroes
RRED(7) is the Bode gain (G_B),

Based on the values of the first six quantities, come numerator time constants (τ), damping (ζ), and resonant frequencies (ω), followed by denominator time constants, damping and resonant frequencies.

Subroutine PUNCHO punches a matrix of octal numbers onto cards without round-off error. Octal representation of the matrix elements is used because it gives an exact replica of the binary number used by a digital computer. A decimal representation will not give an exact replica. The matrix on punched cards is to be used only as an emergency backup for the matrix written on a storage tape.

This matrix on cards is compatible with the input form for Subroutine READO. A group of up to three consecutive elements from a row of the matrix are punched on each card. If all of the elements of a group are zero, punching of this card is suppressed.

The first card punched contains the matrix name (in card columns 1-6), the matrix row size (in card columns 7-10), and the matrix column size (card columns 11-15). This is followed by the matrix data. On any card of the matrix data the first integer number (card columns 1-5) is the row number of the matrix elements on that card. The second integer number (card columns 6-10) is the column number of the matrix element in the first data field (card columns 14-25). The next group contains octal numbers (up to three numbers in card columns 14-25, 29-40, 44-55) that are the values of the matrix elements. This group of matrix elements is given in consecutive column order. The last card punched contains ten zeroes in card columns 1-10 to indicate the end of the matrix.

Subroutine RBTG1 calculates a rigid body transformation matrix, [RBT] , such that

$$\{q\} = [RBT]\{q\}_{REF}$$

where $\{q\}$ is displacements at selected points on a structure, and $\{q\}_{REF}$ is displacements at a reference point. The cartesian coordinates (x, y, z) of the selected points are given by the input matrix [XYZ] . Each row of [XYZ] has the x, y, z coordinates of one point. The x, y, z coordinates of the reference point are given by [XYZREF] . This reference point need not be one of the points given in [XYZ] . A matrix of integers, [JDOF] is also input to this subroutine to define the degree of freedom number of the displacement of each point. This degree of freedom number will define the row number of the degree of freedom in [RBT] . Each of these degrees of freedom are assumed to be in the same direction as its corresponding reference degree of freedom. A negative value in [JDOF] indicates a relative (versus absolute) degree of freedom (e.g., slosh, modal). The negative value will cause that row of the resulting [RBT] matrix to be zero. A row of integers, [JVEC] , is input to define the degrees of freedom associated with the reference point. Negative signs on [JVEC] enables change from assumed right hand system to one you wish to specify. Use of these matrices will be illustrated in the example problem.

DESCRIPTION OF TECHNIQUE

The translation $(\delta_x, \delta_y, \delta_z)$ at the i^{th} point is given in terms of the translation $(\delta_x, \delta_y, \delta_z)$ and rotation $(\theta_x, \theta_y, \theta_z)$ of the reference point by the vector equation

$$\vec{\delta}_i = \vec{\delta}_R + \vec{\theta}_R \times \vec{V}_i$$

where $\vec{V}_i$ is the vector from the reference point to the i^{th} point. All vectors are in terms of x, y, z components.

The rotation at the i^{th} point is equal to the rotation at the reference point, as shown by the vector equation

$$\vec{\theta}_i = \vec{\theta}_R.$$

A typical rigid body transformation for the i^{th} point is size 6x6 and looks like

$$\begin{Bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_i = \left[\begin{array}{ccc|ccc} 1. & & & 0. & V_z & -V_y \\ & 1. & & -V_z & 0. & V_x \\ & & 1. & V_y & -V_x & 0. \\ \hline & & & 1. & & \\ & & & & 1. & \\ & & & & & 1. \end{array}\right]_i \begin{Bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_R$$

where

V_x = the x component of the vector distance from the reference point to point i,

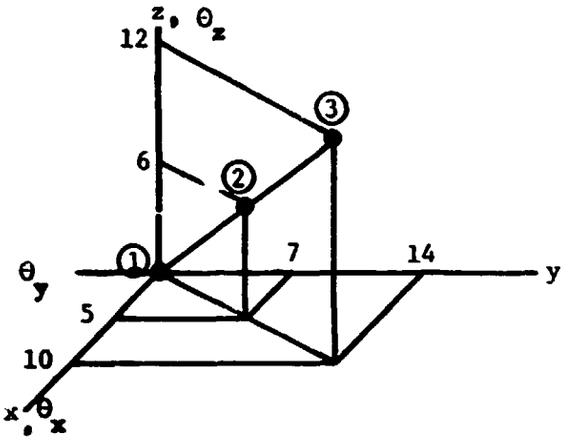
V_y = the y component of the vector distance from the reference point to point i,

V_z = the z component of the vector distance from the reference point to point i.

The vector distance is calculated by using the i^{th} row of the matrix [XYZ] and the vector [XYZREF] , e.g., $V_{x,i}$ = XYZ (i, 1)-XYZREF(1). Using the Forma Subroutine REVADD, the 6x6 formed above is then "revadded" into the output matrix [RBT] . The IVEC required for REVADD is formed from the i^{th} row of the integer matrix [JDOF] , and sign modified by the corresponding signs of the values in JVEC. The JVEC required for REVADD uses the integer vector [JVEC] .

EXAMPLE

A rigid body transformation will be calculated for the beam as shown in the sketch below that will express the rigid body motions of the 15 dof in terms of node point 1 (the reference point).



Node 1 = 6 dof
Node 2 = 3 dof
Node 3 = 6 dof
15 dof

Input data:

[XYZ] = A matrix of size NNODES (number of nodes) x 3 used to describe the coordinate locations of the node points.

	x	y	z
Node 1	0.	0.	0.
Node 2	5.	7.	6.
Node 3	10.	14.	12.

[XYZREF] = A vector of size 3 used to describe the coordinate location of the reference point. The reference point is assumed to be Node 1.

	x	y	z
Reference	0.	0.	0.

[JDOF] = An integer matrix of size NNODES x 6 used to describe the degree-of-freedom table. In this example there are no θ degrees of freedom at Node 2.

	δ_x	δ_y	δ_z	θ_x	θ_y	θ_z
Node 1	1	2	3	4	5	6
Node 2	7	8	9	0	0	0
Node 3	10	11	12	13	14	15

[JVEC] = An integer vector of size 6 used to define the degrees-of-freedom of the reference point.

	δ_x	δ_y	δ_z	θ_x	θ_y	θ_z
Reference	1	2	3	4	-5	6

Using the above input data, the rigid body transformation matrix, [RBT], is calculated such that

$$\{q\}_{15x1} = [RBT]_{15x6} \{q_{REF}\}_{6x1}.$$

That is,

$$\begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}_{\text{Node 1}} \\ \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \end{bmatrix}_{\text{Node 2}} \\ \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}_{\text{Node 3}} = \left[\begin{array}{cccccc} 1. & & & & & \\ & 1. & & & & \\ & & 1. & & & \\ & & & 1. & & \\ & & & & 1. & \\ & & & & & 1. \\ \hline 1. & & & 0. & -6. & -7. \\ & 1. & & -6. & 0. & 5. \\ & & 1. & 7. & +5. & 0. \\ \hline 1. & & & 0. & -12. & -14. \\ & 1. & & -12. & 0. & 10. \\ & & 1. & 14. & +10. & 0. \\ & & & 1. & & \\ & & & & 1. & \\ & & & & & 1. \end{array}\right] \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \hline \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}_{\text{REF}}$$

Subroutine RBTG2 calculates a rigid body transformation matrix, [RBT] , such that

$$\{q\} = [RBT]\ \{q\}_{REF}$$

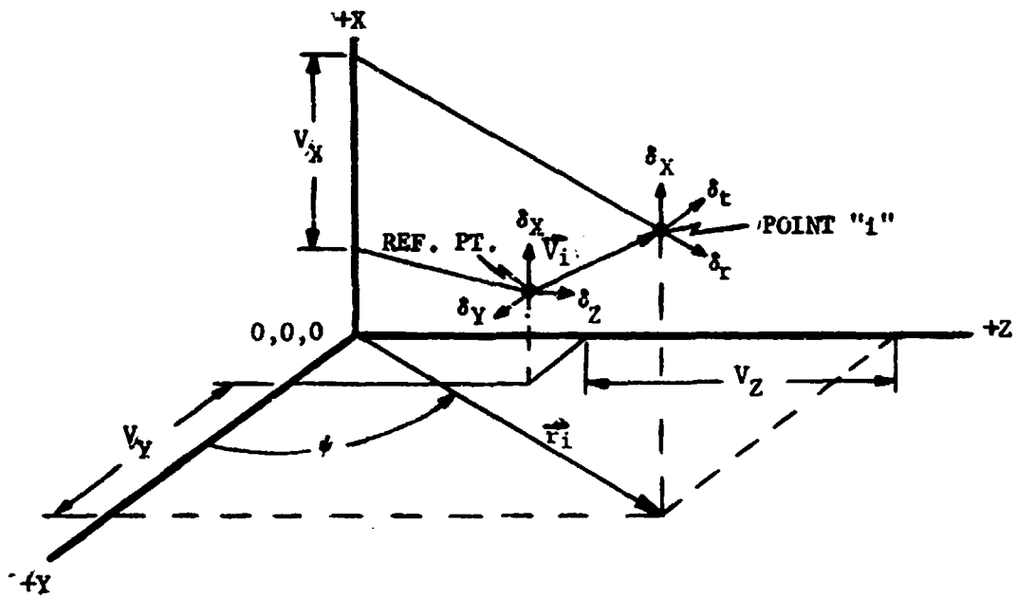
where

$\{q\}$ = displacements (δ_x, δ_r, δ_t, θ_x, θ_r, θ_t) at selected points on a structure

and

$\{q\}_{REF}$ = displacements (δ_x, δ_y, δ_z, θ_x, θ_y, θ_z) at a reference point.

The following sketch is useful for identifying the variables involved.



The cylindrical coordinates (x, r, ψ) of the selected points are given by the input matrix [XRT] . The i^{th} row of [XRT] has the x, r, ψ coordinate of the i^{th} point. The x, y, z coordinates of the reference point are given by the input matrix [XYZREF] .

The reference point need not be one of the points defined in [XRT] . A matrix of integers, [JDOF] , is input to the routine to define the degree-of-freedom number of the displacements of each point. This degree-of-freedom number will define the row number of the degree-of-freedom in [RBT] . A negative value in [JDOF] indicates a relative (versus absolute) degree of freedom (e.g., slosh, modal). The negative value will cause that row of the resulting [RBT] matrix to be zero. A row of integers, [JVEC] , is input to define the degree-of-freedom associated with the reference point (i.e., the columns of [RBT]). Use of these matrices will be illustrated in the example problem.

DESCRIPTION OF TECHNIQUE

A cartesian rigid body transformation is formed first. The translations (δ_x, δ_y, δ_z) at the i^{th} point are given in terms of the translations and rotations (δ_x, δ_y, δ_z, θ_x, θ_y, θ_z) of the reference point by the vector equation

$$\vec{\delta}_i = \vec{\delta}_R + \vec{\theta}_R \times \vec{V}_i$$

where V_i is the vector from the reference point to the i· point. The rotations at the i^{th} point are equal to the tions at the reference point as shown by the vector equatic ı

$$\vec{\theta}_i = \vec{\theta}_R.$$

A typical cartesian rigid body transformation for the i^{th} point is size 6x6 and looks like

$$\begin{Bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \hline \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_i = \left[\begin{array}{ccc:ccc} 1. & & & 0. & V_z & -V_y \\ & 1. & & -V_z & 0. & V_x \\ & & 1. & V_y & -V_x & 0. \\ \hdashline & & & 1. & & \\ & & & & 1. & \\ & & & & & 1. \end{array}\right]_i \begin{Bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \hline \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_{REF} \qquad (1)$$

where

V_x = the x component of the vector distance from the reference point to point "i",

= XRT(i,1) - XYZREF(1)

V_y = the y component of the vector distance from the reference point to point "i",

= XRT(i,2)*cos(XRT(i,3)) - XYZREF(2)

V_z = the z component of the vector distance from the reference point to point "i",

= XRT(i,2)*sin(XRT(i,3)) - XYZREF(3)

Equation (1) may be written in matrix form as

$$\{q_{cart}\}_i = [T2]_i \{q_{cart}\}_{REF}. \qquad (2)$$

It is now desired to transform $\{q_{cart}\}_i$ from cartesian coordinates to cylindrical coordinates. This may be done by using a simple rotation transformation such that

$$\begin{Bmatrix} \delta_x \\ \delta_r \\ \delta_t \\ \hdashline \theta_x \\ \theta_r \\ \theta_t \end{Bmatrix}_i = \left[\begin{array}{ccc:ccc} 1 & & & & & \\ & C & S & & & \\ & -S & C & & & \\ \hdashline & & & 1 & & \\ & & & & C & S \\ & & & & -S & C \end{array}\right] \begin{Bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \hdashline \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_i \qquad (3)$$

where

$c = \cos\psi = \cos\ (XRT(i,3))$

$s = \sin\psi = \sin\ (XRT(i,3))$.

Equation (3) may be expressed as

$$\{q_{cyl}\}_i = [T1]\ \{q_{cart}\}_i \qquad (4)$$

Substituting equation (2) into equation (4) yields

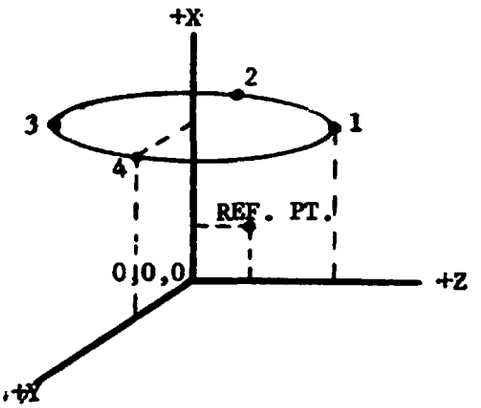
$$\begin{aligned} \{q_{cyl}\}_i &= [T1]\ [T2]\ \{q_{cart}\}_{REF} \\ &= [RBT]_i\ \{q_{cart}\}_{REF} \end{aligned} \qquad (5)$$

Using the FORMA Subroutine REVADD, the resultant 6x6 formed in equation (5) is "revadded" into the output matrix [RBT] . The IVEC required for REVADD is formed from the i^{th} row of the integer matrix [JDOF] , and the JVEC required uses the integer vector [JVEC] .

EXAMPLE

A rigid body transformation will be calculated for the

ring, as shown in the sketch below, that will express the rigid body motion of the 18 DOF's in terms of the reference point.



Node 1 = 6 DOF
Node 2 = 3 DOF
Node 3 = 6 DOF
Node 4 = 3 DOF
18 DOF

Ring Geometry:
X = 100.
Diameter = 180.
Node Points 90. Degrees Apart

Input date:

[XRT] : A matrix of size NNODES (number of nodes)x3, used to describe coordinate locations of the node points.

	X	r	ψ
Node 1	100.	90.	90.
Node 2	100.	90.	180.
Node 3	100.	90.	270.
Node 4	100.	90.	360.

[XYZREF] : A scalar vector of size 3 used to describe the coordinate location of the reference point.

	x	y	z
REFPT	30.	0.	30.

[JDOF] : An integer matrix of size NNODESx6, used to describe the degree-of-freedom table. In this

example, there are no rotational degrees-of-freedom for node point 2 and no translational degrees-of-freedom for node point 4.

	DOF					
	δ_x	δ_r	δ_t	θ_x	θ_r	θ_t
Node 1	1	2	3	4	5	6
Node 2	7	8	9	0	0	0
Node 3	10	11	12	13	14	15
Node 4	0	0	0	16	17	18

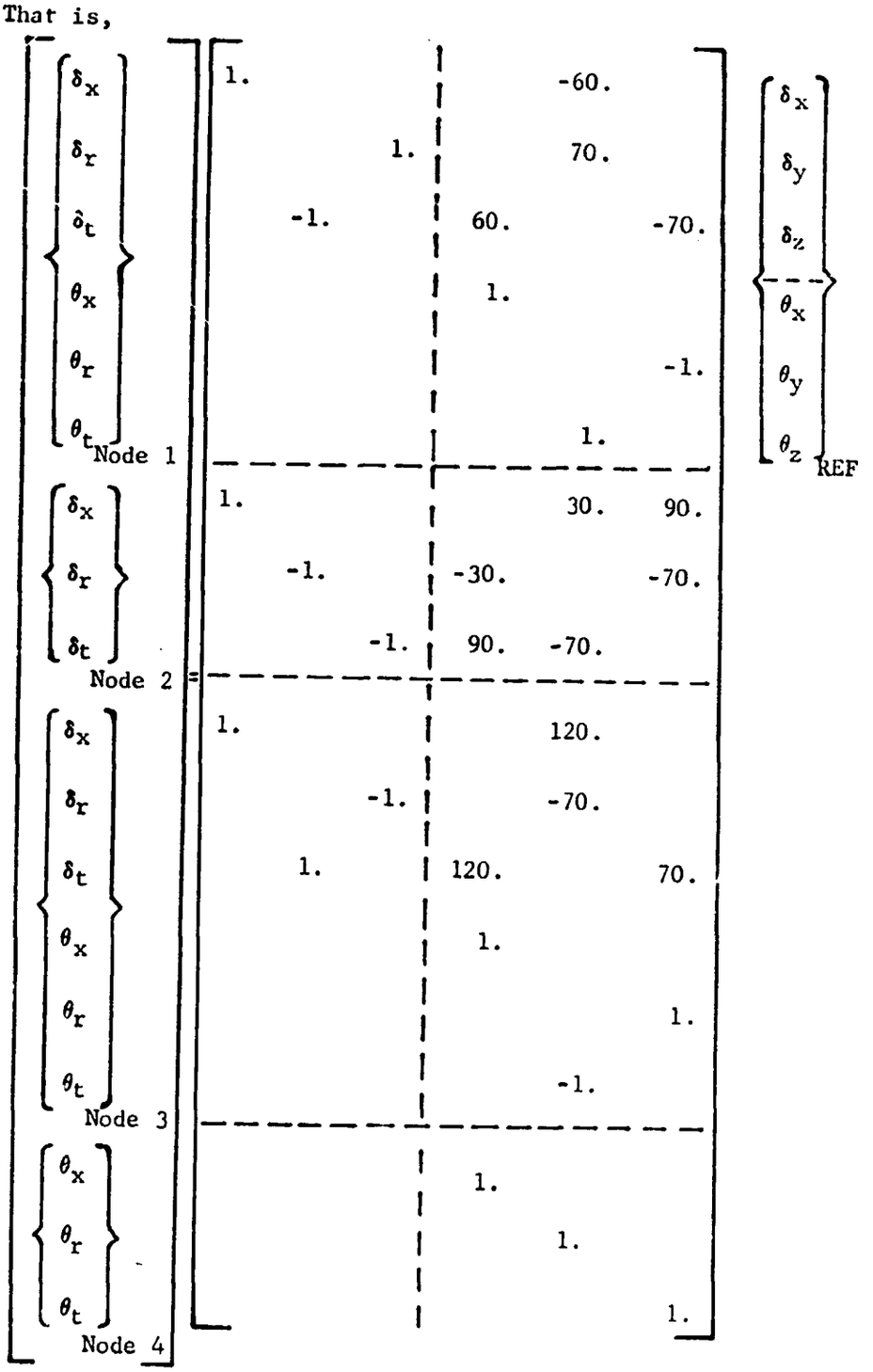
[JVEC] : An integer vector of size 6, used to define the degrees-of-freedom of the reference point.

	DOF					
	δ_x	δ_y	δ_z	θ_x	θ_y	θ_z
Reference Point	1	2	3	4	-5	6

Using the above input data, a rigid body transformation matrix [RBT] , is calculated such that

$$\{q\}_{13x1} = [RBT]_{18x6} \{q\}_{REF_{6x1}}$$

That is,


δx
δr
δt
θx
θr
θt
Node 1
δx
δr
δt
Node 2
δx
δr
δt
θx
θr
θt
Node 3
θx
θr
θt
Node 4
1. -60.
1. 70.
-1. 60. -70.
1.
-1.
1.
1. 30. 90.
-1. -30. -70.
-1. 90. -70.
1. 120.
-1. -70.
1. 120. 70.
1.
1.
-1.
1.
1.
1.
δx
δy
δz
θx
θy
θz
REF

Subroutine READ reads a matrix of real numbers (a Fortran term for numbers with a decimal point) from either cards or tape into the computer. The matrix is then printed so that these input data are recorded with the answers of a run. A print suppression option is available for a matrix read from tape. On option, the matrix read from either cards or tape may be written on a tape (by Subroutine WTAPE).

The first data card read by Subroutine READ contains the information to indicate whether cards or tape will be used. The information entered on this card (and subsequent cards for card input) is given below.

Card Data Input Form

Required entries are denoted by an * symbol below. Any other entry is optional

	Card Columns	Format Type (1)		Entry
First Card	1-6	A		*Matrix Name. Will appear in printout.
	7-10	I		*Matrix Row Size.
	11-15	I		*Matrix Column Size.
	16-69	A		Any remarks to further identify the input matrix.
Write-Tape Options	72			$. Only if the Write-Tape is to be initialized by Subroutine INTAPE. The Write-Tape identification will be from card columns 73-78.
	72		or	Anything other than $ is the Write-Tape is not to be initialized.
	73-78	A		The Write-Tape identification. (e.g., T1234). Use with $ in card column 72.
	73-78		or	REWIND. The Write-Tape will be rewound before being used.
	73-76		or	LIST. The Write-Tape will be listed by Subroutine LTAPE after the matrix has been written on the Write-Tape.
	73-78		or	Anything else will be ignored.
	79-80	I		The Write-Tape Number. (e.g., 21).
			or	Blank if the matrix is not to be written on tape.

	Card Columns	Format Type (1)	Entry
Middle Cards	1-5	I	*Row Number of matrix elements on card.
	6-10	I	*Column Number of matrix element in first data field.
	11-27	E	*First data field with matrix elements. (2)
	28-44	E	*Second data field with matrix elements. (2)
	45-61	E	*Third data field with matrix elements. (2)
	62-78	E	*Fourth data field with matrix elements. (2)
Last Card	1-10	I	*Ten zeroes.

Note (1) Format Type A allows any keypunch symbol. Format Type I allows only integer numbers right justified in the field. Format Type E allows only real numbers (a Fortran term for numbers with a decimal point) anywhere in the field.

Note (2) Only nonzero elements need be entered.

As an example of card input to Subroutine READ consider the following matrix:

$$[A1*C]_{3x6} = \begin{bmatrix} 1. & 0. & 3. & 0. & 6. & 5. \\ 0. & 2. & 4. & 0. & 0. & 0. \\ 0. & 7. & 0. & 0. & 0. & 0. \end{bmatrix}.$$

This matrix is also to be written on tape number 21 that is to be initialized and identified as T4334. Figure 1 demonstrates how this information could be written on a coding form to facilitate keypunching to cards.

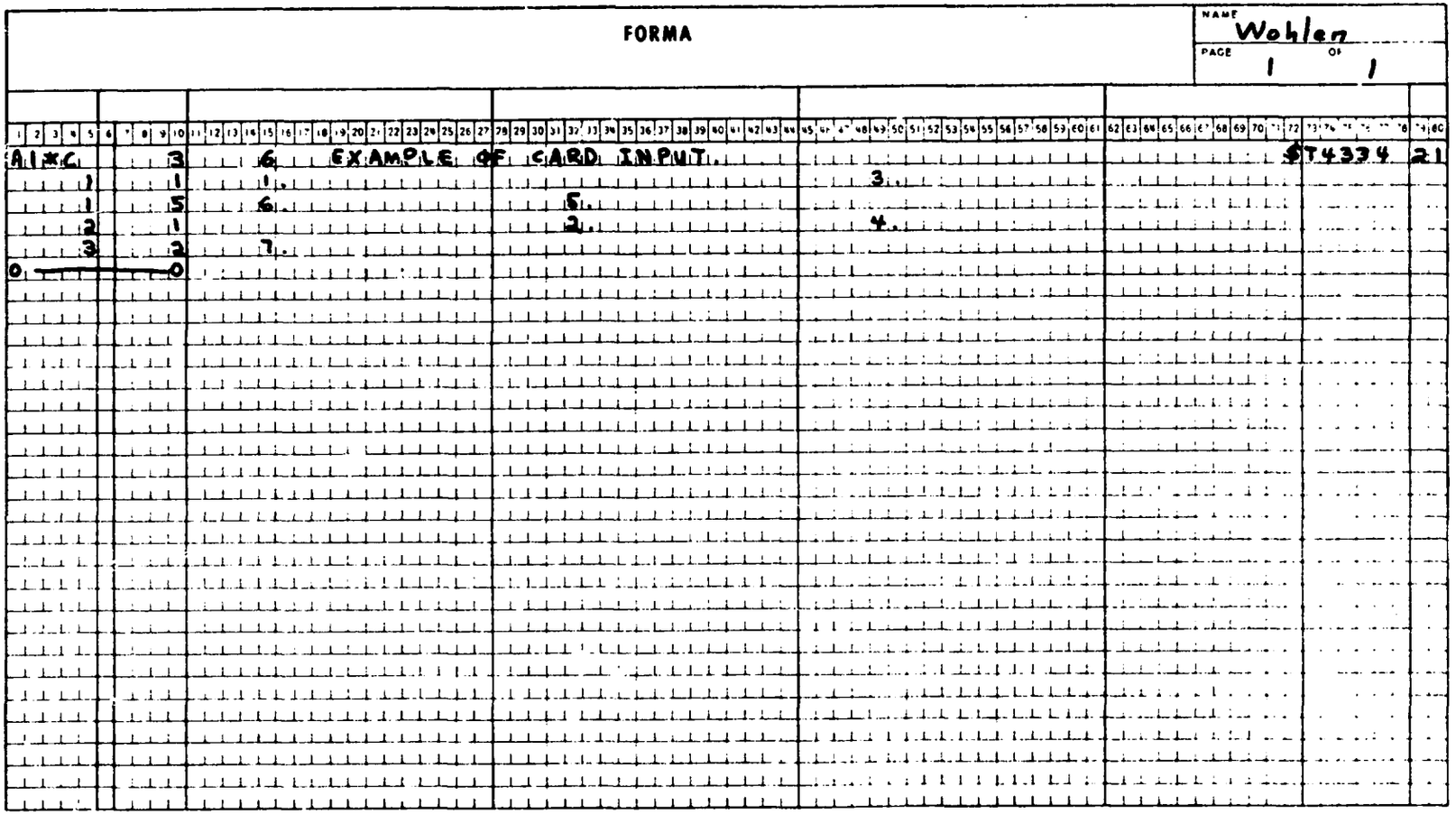


Figure 1 Example of Card Input for Subroutine READ

Tape Data Input Form

Required entries are denoted with an * symbol below. Any other entry is optional. Only one card is used for each matrix read.

	Card Columns	Format Type (1)		Entry
One Card	1-6	A		*Name of matrix to be read from the Read-Tape.
	10			Zero. The Read-Tape will move forward from its present position and search to the end of the tape. If the matrix is not found upon the first end-of-tape encounter, the tape will automatically rewind and make one more pass. If it is not found on the second end-of-tape encounter, an error message will be printed and the program will stop.
	7-10	I	or	Minus the location number of matrix on the Read-Tape. Tape will be positioned at the beginning of the location specified and then continue as described above for a zero in column 10.
	11-15	I		*The Read-Tape Number. (e.g., 11). If positive, the matrix read will be printed in the output. If negative, the matrix read will not be printed in the output.
	16-21	A		*Run number of matrix to be read from the Read-Tape.
	22-27			REWIND. The Read-Tape will be rewound before being used.
	22-25		or	LIST. The Read-Tape will be listed by Subroutine LTAPE.
	22-27		or	Anything else will be considered as part of the remarks described below.

	Card Columns	Format Type (1)	Entry
	28-69	A	Any remarks to further identify the input matrix.
Write-Tape Options	72		$. Only if the Write-Tape is to be initialized by Subroutine INTAPE. The Write-Tape identification will be from card columns 73-78.
	72	or	Anything other than $ if the Write-Tape is not to be initialized.
	73-78	A	The Write-Tape identification. (e.g., T1234). Use with $ in card column 72.
	73-78	or	REWIND. The Write-Tape will be rewound before being used.
	73-76	or	LIST. The Write-Tape will be listed by Subroutine LTAPE after the matrix has been written on the Write-Tape.
	73-78	or	Anything else will be ignored.
	79-80	I	The Write-Tape Number. (e.g., 21).
		or	Blank if the matrix is not to be written on tape.

Note (1) Format Type A allows any keypunch symbol.
Format Type I allows only integer numbers right justified in the field.

As examples of tape input to Subroutine Read consider:

Example 1. A matrix named AB2 with run number of RUN-46 is to be read from tape number 11 into the computer and printed. This matrix is also to be written on tape number 22 that is to be initialized and identified as T4321.

Example 2. A matrix named XYZ4 with run number of TKD is on tape number 13 twice. The first time is at location 29 and the second time is at location 54. It is desired to read the second matrix.

Figure 2 demonstrates how these two examples would be written on a coding form to facilitate keypunching to cards.

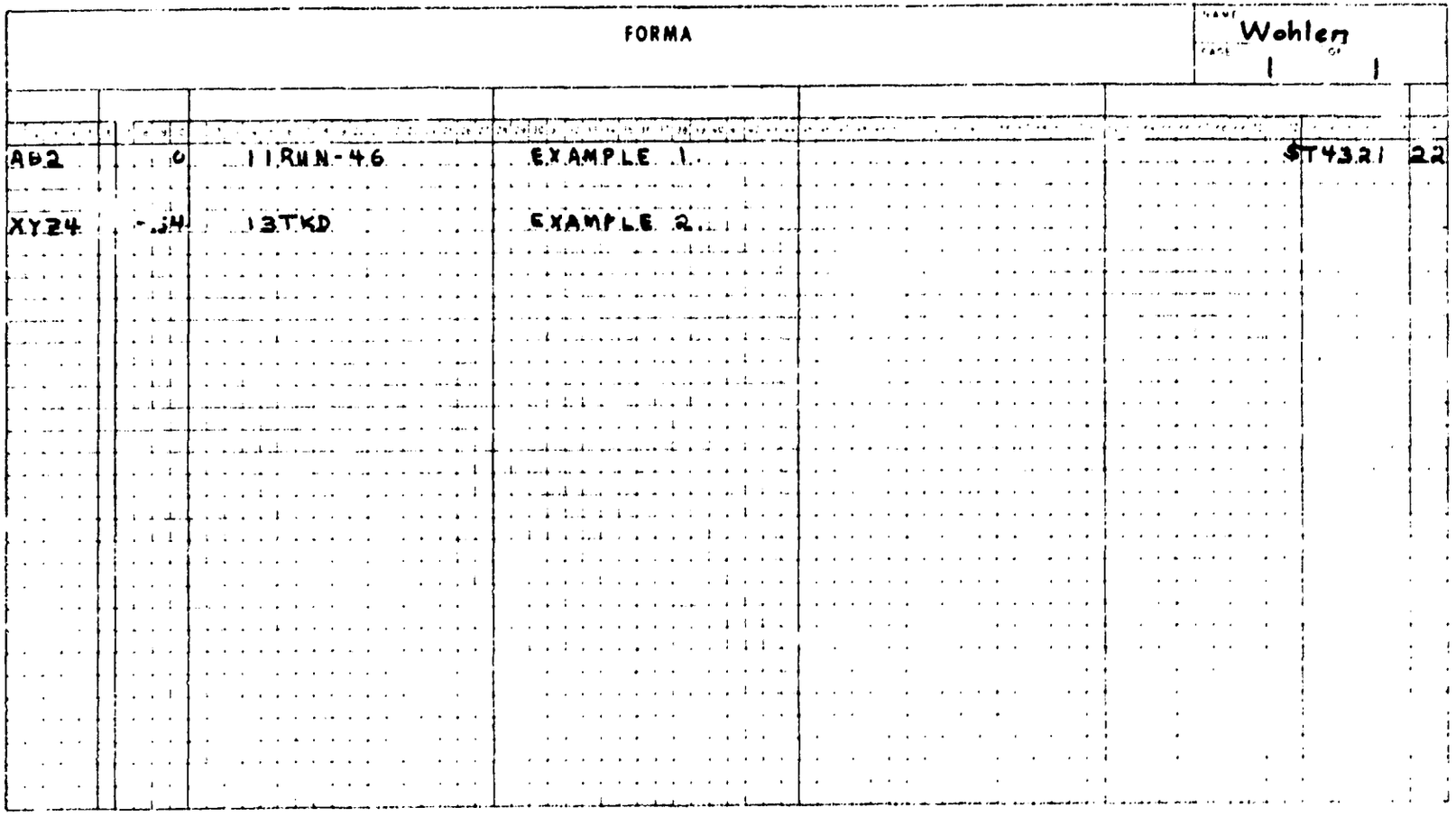


Figure 2 Examples of Tape Input for Subroutine READ

Subroutine READIM reads a matrix of integer numbers from either cards or tape into the computer. The matrix is then printed so that these input data are recorded with the answers of a run. A print suppression option is available for a matrix read from tape. On option, the matrix read from either cards or tape may be written on a tape (by Subroutine WTAPE).

The first data card read by Subroutine READIM contains the information to indicate whether cards or tape will be used. The information entered on this card (and subsequent cards for card input) is given below.

Card Data Input Form

Required entries are denoted by an * symbol below. Any other entry is optional.

	Card Columns	Format Type (1)		Entry
First Card	1-6	A		*Matrix Name. Will appear in printout.
	7-10	I		*Matrix Row Size.
	11-15	I		*Matrix Column Size.
	16-69	A		Any remarks to further identify the input matrix.
Write-Tape Options	72			$. Only if the Write-Tape is to be initialized by Subroutine INTAPE. The Write-Tape identification will be from card columns 73-78.
	72		or	Anything other than $ if the Write-Tape is not to be initialized.
	73-78	A		The Write-Tape identification. (e.g., T1234). Use with $ card column 72.
	73-78		or	REWIND. The Write-Tape will be rewound before being used.
	73-76		or	LIST. The Write-Tape will be listed by Subroutine LTAPE after the matrix has been written on the Write-Tape.
	73-78		or	Anything else will be ignored.
	79-80	I		The Write-Tape Number. (e.g., 21).
			or	Blank if the matrix is not to be written on tape.

	Card Columns	Format Type (1)	Entry
Middle Cards	1-5	I	*Row Number of matrix elements on card.
	6-10	I	*Column Number of matrix element in first data field.
	11-15	I	*First data field with matrix elements. (2)
	16-20	I	*Second data field with matrix elements. (2)
	etc		
	76-80	I	*Fourteenth data field with matrix elements. (2)
Last Card	1-10	I	*Ten zeroes.

Note (1) Format Type A allows any keypunch symbol. Format Type I allows only integer numbers right justified in the field.

Note (2) Only nonzero elements need be entered.

As an example of card input to Subroutine READIM consider the following matrix:

$$[A1*C]_{3x6} = \begin{bmatrix} 1 & 0 & 3 & 0 & 6 & 5 \\ 0 & 2 & 4 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This matrix is also to be written on tape number 21 that is to be initialized and identified as T4334. Figure 1 demonstrates how this information could be written on a coding form to facilitate keypunching to cards.

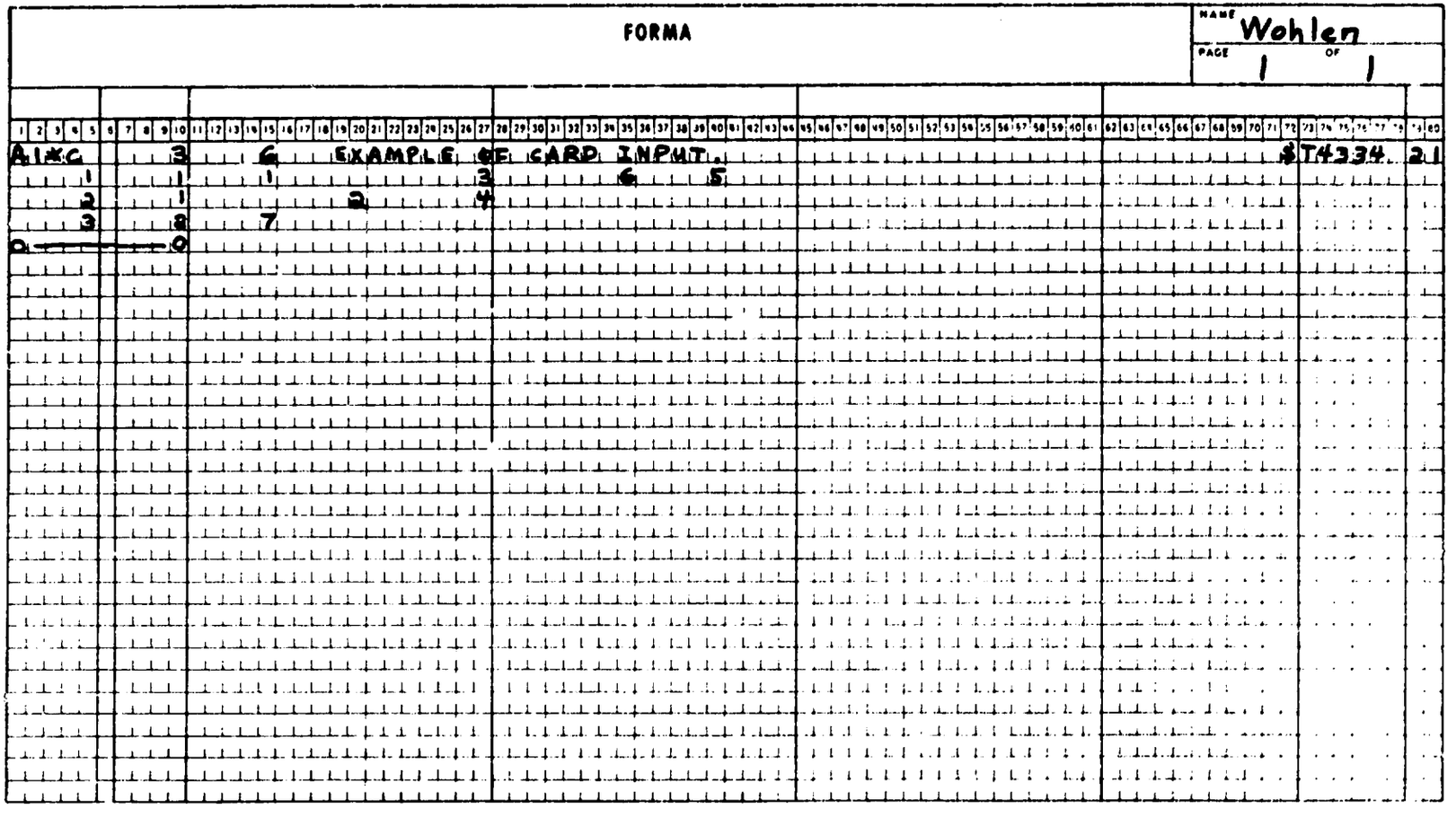


Figure 1 Example of Card Input for Subroutine READIM

Tape Data Input Form

Required entries are denoted with an * symbol below. Any other entry is optional. Only one card is used for each matrix read.

	Card Columns	Format Type (1)		Entry
One Card	1-6	A		*Name of matrix to be read from the Read-Tape.
	10			Zero. The Read-Tape will move forward from its present position and search to the end of the tape. If the matrix is not found upon the first end-of-tape encounter, the tape will automatically rewind and make one more pass. If it is not found on the second end-of-tape encounter, an error message will be printed and the program will stop.
	7-10	I	or	Minus the location number of matrix on the Read-Tape. Tape will be positioned at the beginning of the location specified and then continue as described above for a zero in column 10.
	11-15	I		*The Read-Tape Number. (e.g., 11). If positive, the matrix read will be printed in the output. If negative, the matrix read will not be printed in the output.
	16-21	A		*Run number of matrix to be read from the Read-Tape.
	22-27			REWIND. The Read-Tape will be rewound before being used.
	22-25		or	LIST. The Read-Tape will be listed by Subroutine LTAPE.
	22-27		or	Anything else will be considered as part of the remarks described below.

	Card Columns	Format Type (1)	Entry
	28-69	A	Any remarks to further identify the input matrix.
Write-Tape Options	72		$. Only if the Write-Tape is to be initialized by Subroutine INTAPE. The Write-Tape identification will be from card columns 73-78.
	72	or	Anything other than $ if the Write-Tape is not to be initialized.
	73-78	A	The Write-Tape identification. (e.g., T1234). Use with $ in card column 72.
	73-78	or	REWIND. The Write-Tape will be rewound before being used.
	73-76	or	LIST. The Write-Tape will be listed by Subroutine LTAPE after the matrix has been written on the Write-Tape.
	73-78	or	Anything else will be ignored.
	79-80	I	The Write-Tape Number. (e.g., 21).
		or	Blank if the matrix is not to be written on tape.

Note (1) Format Type A allows any keypunch symbol. Format Type I allows only integer numbers right justified in the field.

As examples of tape input to Subroutine READIM consider:

Example 1. A matrix named AB2 with run number of RUN-46 is to be read from tape number 11 into the computer and printed. This matrix is also to be written on tape number 22 that is to be initialized and identified as T4321.

Example 2. A matrix named XYZ4 with run number of TKD is on tape number 13 twice. The first time is at location 29 and the second time is at location 54. It is desired to read the second matrix.

Figure 2 demonstrates how these two examples would be written on a coding form to facilitate keypunching to cards.

FCRMA

NAME Wohlen

PAGE 1 OF 1

Cols 1–5	Cols 6–10	Cols 11–27	Cols 28–44	Cols 45–61	Cols 62–72	Cols 73–78	Cols 79–80
AB2	0	11RUN-46	EXAMPLE 1.			$T4321	22
XY24	-54	13TKD	EXAMPLE 2.				

Figure 2 Examples of Tape Input for Subroutine READIM

Subroutine READO reads a matrix of octal numbers from cards into the computer. The matrix is then printed side by side in both octal and decimal so that these input data are recorded with the answers of a run.

The primary purpose of Subroutine READO is to read a matrix from punched cards without round off error. The cards are punched by Subroutine PUNCHO. Octal representation of the matrix elements is used because it gives an exact replica of the binary number used by a digital computer. A decimal representation will not give an exact replica. The matrix on punched cards is to be used only as an emergency backup for the matrix written on a storage tape.

Because of the emergency backup nature of input data to this Subroutine READO, only cards are read. No tape reading or writing options are available.

The information entered on the data cards is given below. Required entries are denoted by an * symbol. Any other entry is optional.

	Card Columns	Format Type (1)	Entry
First Card	1-6	A	*Matrix Name. (Will appear in printout.)
	7-10	I	*Matrix Row Size.
	11-15	I	*Matrix Column Size.
	16-69	A	Any remarks to further identify the input matrix.
Middle Cards	1-5	I	*Row Number of matrix elements on card.
	6-10	I	*Column Number of matrix element in first data field.
	14-25	O	*First data field with matrix elements. (2)
	29-40	O	*Second data field with matrix elements. (2)
	44-55	O	*Third data field with matrix elements. (2)
Last Card	1-10	I	*Ten zeroes.

Note (1) Format Type A allows any keypunch symbol. Format Type I allows only integer numbers right justified in the field. Format Type O allows only octal numbers.

Note (2) Only nonzero elements need be entered.

No examples of input are given because data would not be key-punched for input to Subroutine READO but rather obtained from Subroutine PUNCHO.

Subroutine REVADD rearranges (revises) the rows and columns of a matrix [A], multiplies [A] by a scalar alpha, and adds the result to a previously defined matrix [Z]. The revision of [A] is specified by two vectors. The first vector $\{IVEC\}$ gives the new row location of each row of [A] in [Z]. The second vector $\{JVEC\}$ gives the new column location of each column of [A] in [Z]. The [Z] matrix must be defined before the use of this subroutine. For instance, if [Z] is to be originally defined as all zeros, Subroutine ZERO could be used. The REVADD operation can be thought of in subscript notation as

$$z_{k\ell}\ (\text{out}) = z_{k\ell}\ (\text{in}) + \alpha\, a_{ij} \qquad \begin{pmatrix} i = 1, \text{NRA} \\ j = 1, \text{NCA} \end{pmatrix}$$

where

$k = IVEC(i)$,

$\ell = JVEC(j)$.

NRA is the number of rows of [A], and NCA is the number of columns of [A].

Values in $\{IVEC\}$ and $\{JVEC\}$ may be positive, negative or zero. A negative value changes the sign of the corresponding row or column of [A] in [Z]. A zero value omits the corresponding row or column of [A] from [Z]. The values are integer numbers.

This subroutine may be called repeatedly to form [Z] from the revision/addition of several [A] matrices.

An important use of Subroutine REVADD is to revise and add the stiffness matrix of a structural component to the stiffness matrix of the complete structure to account for the difference in coordinate systems.

EXAMPLES

The first example to illustrate the REVADD operation is as follows:

Matrix [Z] has been previously defined as

$$[Z]_{4x5} = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 2. & 0. & 0. & 0. \\ 0. & 0. & 3. & 4. & 5. \\ 0. & 0. & 0. & 0. & 6. \end{bmatrix}.$$

Let [A] be defined as

$$[A]_{3x2} = \begin{bmatrix} 1. & 2. \\ 3. & 4. \\ 5. & 6. \end{bmatrix}.$$

The first row of [A] is to be added to the third row of [Z], the second row of [A] is to be omitted from [Z], the third row of [A] is to be added to the first row of [Z], with the sign of each element reversed.

Thus $\{IVEC\}_{3x1} = \begin{bmatrix} 3 \\ 0 \\ -1 \end{bmatrix}$.

The first column of [A] is to be added to the second column of [Z] with the sign of each element reversed, the second column of [A] is to be added to the fifth column of [Z].

Thus $\{JVEC\}_{2x1} = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$.

Then, assuming $\alpha = 1.0$ and placing $\{IVEC\}$ and $\{JVEC\}$ adjacent to [A] to aid in visualizing the revision of [A], we have

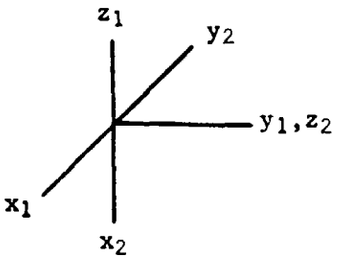
$$[Z]_{4x5} = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 2. & 0. & 0. & 0. \\ 0. & 0. & 3. & 4. & 5. \\ 0. & 0. & 0. & 0. & 6. \end{bmatrix} + 1.0 \begin{bmatrix} 3 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} -2 & 5 \end{bmatrix} \begin{bmatrix} 1. & 2. \\ 3. & 4. \\ 5. & 6. \end{bmatrix}$$

$$= \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 2. & 0. & 0. & 0. \\ 0. & 0. & 3. & 4. & 5. \\ 0. & 0. & 0. & 0. & 6. \end{bmatrix} + 1.0 \begin{bmatrix} 3 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 0. & -1. & 0. & 0. & 2. \\ 0. & -3. & 0. & 0. & 4. \\ 0. & -5. & 0. & 0. & 6. \end{bmatrix}$$

$$= \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 2. & 0. & 0. & 0. \\ 0. & 0. & 3. & 4. & 5. \\ 0. & 0. & 0. & 0. & 6. \end{bmatrix} + 1.0 \begin{bmatrix} 0. & 5. & 0. & 0. & -6. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & -1. & 0. & 0. & 2. \\ 0. & 0. & 0. & 0. & 0. \end{bmatrix}$$

$$= \begin{bmatrix} 1. & 5. & 0. & 0. & -6. \\ 0. & 2. & 0. & 0. & 0. \\ 0. & -1. & 3. & 4. & 7. \\ 0 & 0. & 0. & 0. & 6. \end{bmatrix}.$$

A second example of the use of this subroutine demonstrates the coordinate transformation concept which is a very important application of REVADD. It is desired to transform a stiffness matrix from an original coordinate system (subscript 1) to a final coordinate system (subscript 2) as shown in the sketch below.



For this example, the stiffness matrix in the original coordinate system is assumed to be

$$\begin{bmatrix} 1. & 2. & 4. \\ 2. & 3. & 5. \\ 4. & 5. & 6. \end{bmatrix}.$$

From inspection of the coordinate system axes in the above sketch,

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} -z_1 \\ -x_1 \\ y_1 \end{bmatrix}.$$

To obtain the vector on the right hand side of the equation:

x_1 of the original coordinate system is to be the second row of the final coordinate system with the sign reversed.

y_1 of the original coordinate system is to be the third row of the final coordinate system.

z_1 of the original coordinate system is to be the first row of the final coordinate system with the sign reversed.

The $\{IVEC\}$ to accomplish this change is

$$\{IVEC\}_{3x1} = \begin{bmatrix} -2 \\ 3 \\ -1 \end{bmatrix}.$$

Apply this $\{IVEC\}$ as the $\{IVEC\}$ and $\{JVEC\}$ to the original stiffness matrix to obtain the final stiffness matrix. This application is analogous to the change in coordinates made in the triple matrix product procedure.

$$\begin{bmatrix} -2 \\ 3 \\ -1 \end{bmatrix} \begin{bmatrix} 1. & 2. & 4. \\ 2. & 3. & 5. \\ 4. & 5. & 6. \end{bmatrix} \begin{bmatrix} -2 & 3 & -1 \end{bmatrix} \xrightarrow{\text{REVADD}} \begin{bmatrix} 6. & 4. & -5. \\ 4. & 1. & -2. \\ -5. & -2. & 3. \end{bmatrix}.$$

The above REVADD procedure may be compared with the more conventional triple matrix product procedure below.

The equation for the coordinate transformation would be

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} & -1. & \\ & & 1. \\ -1. & & \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \tag{1}$$

The strain energy expression with the stiffness matrix is given by $U = \frac{1}{2} \{q\}^T [K] \{q\}$ or for coordinate system 1,

$$U = \frac{1}{2} [x_1 \; y_1 \; z_1] \begin{bmatrix} 1. & 2. & 4. \\ 2. & 3. & 5. \\ 4. & 5. & 6. \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}. \tag{2}$$

Substituting Equation (1) into (2) gives

$$U = \frac{1}{2} [x_2 \; y_2 \; z_2] \begin{bmatrix} & & -1. \\ -1. & & \\ & 1. & \end{bmatrix} \begin{bmatrix} 1. & 2. & 4. \\ 2. & 3. & 5. \\ 4. & 5. & 6. \end{bmatrix} \begin{bmatrix} & -1. & \\ & & 1. \\ -1. & & \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix},$$

thus the inner triple matrix product gives the stiffness matrix in coordinate system 2, i.e.,

$$\begin{bmatrix} 6. & 4. & -5. \\ 4. & 1. & -2. \\ -5. & -2. & 3. \end{bmatrix}.$$

which is the same result as that obtained from the REVADD procedure.

The advantages of using the REVADD procedure over the triple matrix product procedure are:

1) Less computer time;
2) Less computer core is used;
3) Usually easier to code the $\{IVEC\}$ (thus $\{JVEC\}$) than to code the transformation matrix.

Subroutine ROWMLT evaluates a special matrix operation by multiplying each row of a matrix [B] by a scalar. That is,

$$[Z]_{NRxNC} = \begin{bmatrix} a_1 \{b_1\}^T_{1xNC} \\ a_2 \{b_2\}^T_{1xNC} \\ \vdots \\ a_{NR} \{b_{NR}\}^T_{1xNC} \end{bmatrix}$$

where $z_{ij} = a_i b_{ij}$ $\begin{pmatrix} i = 1, NR \\ j = 1, NC \end{pmatrix}$

$\{b_i\}^T$ denotes row i of [B]. Each scalar a_i is an element of the input vector {AVEC}. NR is the number of rows in [B] and [Z] and the size of {AVEC}. NC is the number of columns in [B] and [Z]. The number of elements of {AVEC} must be equal to the number of rows of [B].

EXAMPLE

Consider the input of

$$\{AVEC\}_{2\times1} = \begin{bmatrix} 2. \\ -3. \end{bmatrix} \text{ and } [B]_{2\times3} = \begin{bmatrix} 7. & 2. & -3. \\ 4. & 5. & 6. \end{bmatrix}.$$

The output will be

$$[Z]_{2\times3} = \begin{bmatrix} 2. [7. & 2. & -3.] \\ -3. [4. & 5. & 6.] \end{bmatrix}$$

$$= \begin{bmatrix} 14. & 4. & -6. \\ -12. & -15. & -18. \end{bmatrix}.$$

Subroutine RTAPE reads a selected matrix from tape (disk) into the computer core. The matrix to be selected is identified by the desired run number and matrix name. This procedure is accomplished by searching the matrix headings (see Subroutine WTAPE writeup) until a match with the desired run number and matrix name is obtained and then reading the matrix elements from tape (disk) into the computer core. The search starts from the current position (does not rewind) of the tape (disk) and proceeds to the EOT (end of tape defined in Subroutine WTAPE writeup). If the desired matrix was not found upon reaching the EOT, a rewind is performed and one more search to the EOT is made. If the desired matrix is again not found, (1) an error message is printed, (2) a listing of the matrix headings is printed (see Subroutine LTAPE writeup), and (3) transfer is made to Subroutine ZZBOMB where the program is terminated.

Subroutine SIGMA generates a square matrix with elements on and below the diagonal equal to one and above the diagonal equal to zero. That is,

$$z_{ij} = 1. \qquad (i \geq j)$$

$$z_{ij} = 0. \qquad (i < j)$$

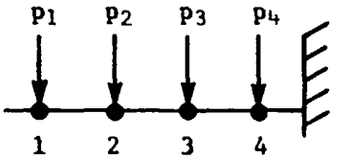
In matrix notation,

$$[Z]_{NxN} = \begin{bmatrix} 1. & & & \\ 1. & 1. & & \\ . & . & & \\ . & . & . & \\ . & . & . & \\ 1. & . & . & 1. \end{bmatrix}$$

where N is the size of [Z].

This subroutine is useful in calculating the sum of all quantities of a given collection. The name of the subroutine comes from the algebraic notation for this process, that is, Σ.

As an example of the use of this subroutine, consider the following beam with loads p_i at discrete points on the beam:



The shear to the right of each point (i) on the beam is

$$v_i = \sum_{j=1}^{i} p_j$$

or in matrix notation,

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 1. & 0. & 0. & 0. \\ 1. & 1. & 0. & 0. \\ 1. & 1. & 1. & 0. \\ 1. & 1. & 1. & 1. \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}.$$

Subroutine SMEQ1 gives the solution of linear simultaneous algebraic equations. These equations are expressed in matrix form as

$$[A]_{NxN} \{Z\}_{Nx1} = \{B\}_{Nx1}$$

where [A] and {B} are known and {Z} is to be calculated. N is the size of the system. The method, with which Gauss's name is associated, consists of a chain of successive eliminations by which the original system is transformed into a system with an upper triangular matrix whose solution is easily calculated. A modification of the standard technique is used in the subroutine. That is, the largest pivotal divisor is searched for and used. The rows are interchanged when necessary to accomplish this. This technique will give the most accurate results since division by small numbers will be avoided.

DESCRIPTION OF TECHNIQUE

Given a system of "N" linear algebraic equations,

$$a_{11} z_1 + a_{12} z_2 + \ldots + a_{1N} z_N = b_1$$

$$a_{21} z_1 + a_{22} z_2 + \ldots + a_{2N} z_N = b_2$$

$$\vdots$$

$$a_{N1} z_1 + a_{N2} z_2 + \ldots + a_{NN} z_N = b_N.$$

The procedure used to obtain {Z} is as follows:

1) The largest coefficient of z_1 is found. The equation in which this occurs is interchanged with the first equation;

2) Divide the new first equation by the new a_{11};

3) Alter Equations 2 through N by subtracting from the i^{th} equation the value a_{i1} times the first equation from step 2). This is done for i = 2, 3, . . . N.

After these three steps, a new but equivalent system of equations results:

$$z_1 + a_{12.1}\, z_2 + a_{13.1}\, z_3 + \cdots + a_{1N.1}\, z_N = b_{1.1}$$

$$a_{22.1}\, z_2 + a_{23.1}\, z_3 + \cdots + a_{2N.1}\, z_N = b_{2.1}$$

$$\vdots$$

$$a_{N2.1}\, z_2 + a_{N3.1}\, z_3 + \cdots + a_{NN.1}\, z_N = b_{N.1}$$

where the subscript (.1) designates the first elimination. The variable z_1 has thus been eliminated from Equations 2 through N.

Steps 1), 2), and 3) above are repeated on Equations 2 through N such that z_2 is also eliminated. This process is continued until an equivalent system of equations is of triangular form as shown below. This process is referred to as the forward solution.

$$\begin{bmatrix} 1 & a_{12.1} & a_{13.1} & & a_{1N.1} \\ & 1 & & & \\ & & a_{23.2} & \cdots & a_{2N.2} \\ & & \ddots & & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} b_{1.1} \\ b_{2.2} \\ \vdots \\ b_{N.N} \end{bmatrix}.$$

The back solution for {Z} from the above system of equations gives no problem since z_N is immediately given as $b_{N.N}$. Next z_{N-1} is easily found from the (N-1) Equation and so on until all z_i are known.

EXAMPLE

The following data are given as input,

$$[A]_{3\times3} = \begin{bmatrix} 2. & 3. & 4. \\ 5. & 6. & 7. \\ 8. & 9. & 9. \end{bmatrix} \text{ and } \{B\}_{3\times1} = \begin{bmatrix} 2. \\ -3. \\ 4. \end{bmatrix};$$

Thus the equations to be solved are

$$\begin{bmatrix} 2. & 3. & 4. \\ 5. & 6. & 7. \\ 8. & 9. & 9. \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 2. \\ -3. \\ 4. \end{bmatrix}.$$

Using the technique described previously, the reader can verify the output to be

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} -19.00 \\ 29.33 \\ -12.00 \end{bmatrix}.$$

REFERENCE

Faddeeva, V. N.: *Computational Methods of Linear Algebra.* Dover Publications Inc., New York, 1959.

Subroutine SRED2 operates on the stiffness matrix [A] to form a reduced stiffness matrix [R] and/or the reducing transformation [T]. The relation between the stiffness matrix [A], displacements {X}, and applied forces {B} may be expressed in matrix form as

$$[A]\ \{X\} = \{B\} \qquad [1]$$

The reduction method assumes Eq [1] to be partitioned as

$$\begin{bmatrix} [A_{11}] & [A_{12}] \\ [A_{21}] & [A_{22}] \end{bmatrix} \begin{Bmatrix} \{X_1\} \\ \{X_2\} \end{Bmatrix} = \begin{Bmatrix} \{B_1\} \\ \{B_2\} \end{Bmatrix} \qquad [2]$$

where $\{X_1\}$ are the displacements to be reduced out and $\{X_2\}$ are the displacements to be retained. The applied forces acting on the coordinates to be reduced are assumed to be zero, such that

$$\{B_1\} = \{0\} \qquad [3]$$

Substituting Eq [3] into Eq [2] and expanding the upper partition, will yield the reduced displacements in terms of the retained displacements as

$$\{X_1\} = -\ [A_{11}]^{-1}\ [A_{12}]\ \{X_2\} \qquad [4]$$

Expanding the lower partitions of Eq [2] and substituting Eq [4] will yield the reduced stiffness matrix as

$$[R]\ \{X_2\} = \{B_2\} \qquad [5]$$

where [R] is the reduced stiffness matrix and is expressed as

$$[R] = [A_{22}] - [A_{21}]\ [A_{11}]^{-1}\ [A_{12}] \qquad [6]$$

The reducing transformation [T] may be expressed using Eq [4] as

$$\begin{Bmatrix} \{X_1\} \\ \{X_2\} \end{Bmatrix} = [T]\ \{X_2\} \qquad [7]$$

where

$$[T] = \begin{bmatrix} -[A_{11}]^{-1} & [A_{12}] \\ [I] & \end{bmatrix} \quad [8]$$

Also

$$[R] = [T]^T [A] [T] \quad [9]$$

DESCRIPTION OF TECHNIQUE

This subroutine uses Gauss reduction partially completed to form matrix [R] and [T] from stiffness matrix [A]. As an example of the method, consider three simultaneous equations of the following form

$$\begin{aligned} a_{11} x_1 + a_{12} x_2 + a_{13} x_3 &= b_1 \\ a_{21} x_1 + a_{22} x_2 + a_{23} x_3 &= b_2 \\ a_{31} x_1 + a_{32} x_2 + a_{33} x_3 &= b_3 \end{aligned} \quad [10]$$

These equations may be written in matrix form as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix} \quad [11]$$

Solve the first Equation for x_1 as

$$x_1 = - \frac{a_{12}}{a_{11}} x_2 - \frac{a_{13}}{a_{11}} x_3 + \frac{b_1}{a_{11}} \quad [12]$$

Substituting Eq [12] into the second and third equations in Eq [11] and divide the first by a_{11} results in

$$\begin{bmatrix} 1 & a^*_{12} & a^*_{13} \\ 0 & a^*_{22} & a^*_{23} \\ 0 & a^*_{32} & a^*_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b^*_1 \\ b^*_2 \\ b^*_3 \end{Bmatrix} \quad [13]$$

where

$$a_{12}^{*} = \frac{a_{12}}{a_{11}} \qquad [14]$$

$$a_{13}^{*} = \frac{a_{13}}{a_{11}} \qquad [15]$$

$$a_{22}^{*} = a_{22} - \frac{a_{12}\, a_{21}}{a_{11}} \qquad [16]$$

$$a_{23}^{*} = a_{23} - \frac{a_{13}\, a_{21}}{a_{11}} \qquad [17]$$

$$a_{32}^{*} = a_{32} - \frac{a_{12}\, a_{31}}{a_{11}} \qquad [18]$$

$$a_{33}^{*} = a_{33} - \frac{a_{13}\, a_{31}}{a_{11}} \qquad [19]$$

$$b_{1}^{*} = \frac{b_{1}}{a_{11}} \qquad [20]$$

$$b_{2}^{*} = b_{2} - \frac{b_{1}\, a_{21}}{a_{11}} \qquad [21]$$

$$b_{3}^{*} = b_{3} - \frac{b_{1}\, a_{31}}{a_{11}} \qquad [22]$$

Solve the second equation for x_2 which will yield

$$x_{2} = - \frac{a_{23}^{*}}{a_{22}^{*}}\, x_{3} + \frac{b_{2}^{*}}{a_{22}^{*}} \qquad [23]$$

Substitute Eq [23] into the third equation in Eq [13] and divide the second equation by a_{22}^{*}. This results in

$$\begin{bmatrix} 1 & a_{12}^{*} & a_{13}^{*} \\ 0 & 1 & a_{23}^{**} \\ 0 & 0 & a_{33}^{**} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1^{*} \\ b_2^{**} \\ b_3^{**} \end{Bmatrix} \qquad [24]$$

where

$$a_{23}^{**} = \frac{a_{23}^{*}}{a_{22}^{*}} \quad [25]$$

$$a_{33}^{**} = a_{33}^{*} - \frac{a_{23}^{*}\, a_{32}^{*}}{a_{22}^{*}} \quad [26]$$

$$b_{2}^{**} = \frac{b_{2}^{*}}{a_{22}^{*}} \quad [27]$$

$$b_{3}^{**} = b_{3}^{*} - \frac{b_{2}^{*}\, a_{32}^{*}}{a_{22}^{*}} \quad [28]$$

The reduced stiffness matrix has been formed and is contained as the a_{33}^{**} element. This can be shown if in Eq [2] we let

$$\{X_1\} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \quad [29]$$

$$\{X_2\} = \{x_3\} \quad [30]$$

$$[A_{11}] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad [31]$$

$$[A_{12}] = \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \quad [32]$$

$$[A_{21}] = [a_{31}\ a_{32}] \quad [33]$$

$$[A_{22}] = a_{33} \quad [34]$$

$$\{B_1\} = \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix} \quad [35]$$

$$\{B_2\} = b_3 \quad [36]$$

Substituting Eq [31] through [34] into Eq [6] results in a reduced stiffness matrix of the form

$$[R] = a_{33} + \frac{a_{21}\,a_{23}\,a_{31} + a_{12}\,a_{13}\,a_{32} - a_{13}\,a_{22}\,a_{31} - a_{11}\,a_{23}\,a_{32}}{a_{11}\,a_{22} - a_{12}\,a_{21}} \qquad [37]$$

Equation [37] is identical to the result obtained by expanding Eq [26]. Thus, Gauss reduction partially completed yields the reduced stiffness matrix.

The reducing transformation may also be obtained using Gauss reduction if additional operations are performed. From Eq [24], solve the second equation for x_2

$$x_2 = - a_{23}^{**}\ x_3 + b_2^{**} \qquad [38]$$

Substitute Eq [38] into the first equation in Eq [24], which will yield

$$\begin{bmatrix} 1 & 0 & a_{13}^{***} \\ 0 & 1 & a_{23}^{**} \\ 0 & 0 & a_{33}^{**} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1^{***} \\ b_2^{**} \\ b_3^{**} \end{Bmatrix} \qquad [39]$$

where

$$a_{13}^{***} = a_{13}^{*} - a_{12}^{*}\ a_{23}^{**} \qquad [40]$$

$$b_1^{***} = b_1^{*} - a_{12}^{*}\ b_2^{**} \qquad [41]$$

Inspection of Eq [39] shows that we have formed a unity matrix partition where the row-columns were reduced. The a_{13}^{***} and the a_{23}^{**} elements contain the necessary information to form the reducing transformation. To show this, substitute Eq [31] and [32] into Eq [8] to yield

$$[T] = \begin{bmatrix} \dfrac{a_{21}\,a_{23} - a_{13}\,a_{22}}{a_{11}\,a_{22} - a_{21}\,a_{12}} \\ \\ \dfrac{a_{12}\,a_{13} - a_{11}\,a_{23}}{a_{11}\,a_{22} - a_{21}\,a_{12}} \\ \\ 1 \end{bmatrix} \qquad [42]$$

The second and third rows of Eq [42] are equal to the negative of elements a_{13}^{***} and a_{23}^{**} in Eq [39]. Thus, Gauss reduction also yields the reducing transformation.

Subroutine START performs the following operations:

1) Reads Input Card 1 for the run number (any keypunch symbol in card columns 1 thru 6) and the user's name (any keypunch symbol in card columns 11 thru 28).

   If the run number is equal to STOP (Card columns 1 thru 4), the run is terminated.

   If the run number is not equal to STOP, the run continues in Subroutine START as follows.

2) Reads Input Card 2 for Title Card 1. Any keypunch symbols may be used in Card columns 1 thru 72.

3) Reads Input Card 3 for Title Card 2. Any keypunch symbols may be used in Card columns 1 thru 72.

4) Initializes page number as zero for use in Subroutine PAGEHD.

5) Interrogates computer for the date.

Run number, date, page number, user's name, Title Card 1, and Title Card 2 are transferred by a COMMON block labeled LSTART for use in other subroutines PAGEHD, PLOT1, PLOT2, PLOT3, and WTAPE.

Subroutine START is used to start each computer run in the FORMA system and will normally be the first subroutine called in a computer program. As an example, pertinent statements from a program using START could be:

```
1 CALL START
      .
      .
      .
  GO TO 1
  END
```

Subroutine STIF1 takes distributed longitudinal stiffness of a rod and replaces it with a "free-free" stiffness matrix. Longitudinal stiffness will be discussed here but the results are also applicable to the torsional stiffness case. The elements of the stiffness matrix are representative stiffness values of the rod at selected points on the rod. These elements are calculated by assuming constant axial force between pairs of the selected points.

The x-stations of the selected points (panel points) are given in {PP}. These x-stations must be in increasing order.

The distributed stiffness, AE(x), is assumed to be piecewise linear and is represented by straight line segments as shown in Figure 1.

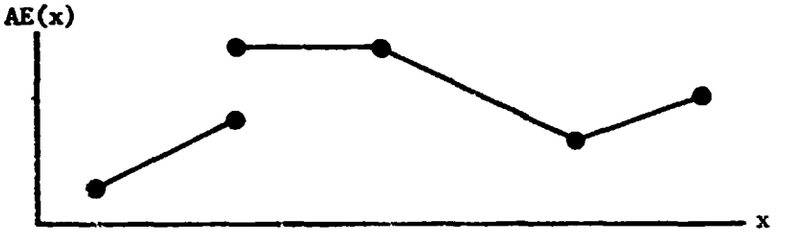


Figure 1 Distributed Stiffness

The x-stations of the first and last points for distributed stiffness must coincide with the first and last panel point x-stations, respectively. All other x-stations of the end points for the line segments giving the distributed stiffness are independent of the panel point x-stations. The line segments may or may not be joined; however, there must not be any x voids or overlaps. The distributed stiffness is defined in [DAE]. Each row of [DAE] represents one nonvertical line segment. The form of each row of [DAE] is $[x_1\ x_2\ AE_1\ AE_2]$ where x_1, AE_1 give the first end point and x_2, AE_2 give the second end point of a line segment. The x_2 of row 1 of [DAE] must be equal to x_1 of row 2, etc.

The calculated representative stiffness values at the selected panel points are placed in a stiffness matrix given by [Z] which is symmetrical and tri-diagonal. That is,

$$[Z]_{n\times n} = \begin{bmatrix} z_{1,1} & z_{1,2} & & & & & \\ z_{2,1} & z_{2,2} & z_{2,3} & & & & \\ & z_{3,2} & z_{3,3} & z_{3,4} & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & z_{n-1,n-2} & z_{n-1,n-1} & z_{n-1,n} \\ & & & & & z_{n,n-1} & z_{n,n} \end{bmatrix}$$

where $z_{i,j} = z_{j,i}$ and n is the number of panel points. The generalized coordinates associated with [Z] are axial deflections at each of the panel points.

As mentioned before, the results of the longitudinal stiffness case considered in this paper are also applicable to the torsional stiffness case. The following table gives the relationship of variables

Longitudinal	Torsional
x	x
δ	θ
AE	GJ
P (axial force)	T (torque)

DESCRIPTION OF TECHNIQUE

The replacement of distributed longitudinal stiffness of a rod by a stiffness matrix is obtained using a strain energy approach as follows. Consider a portion of a rod that is loaded with an axial force at panel point k and restrained at panel point k + 1 as shown in Figure 2. The region between panel points k and k + 1 is referred to as bay k.

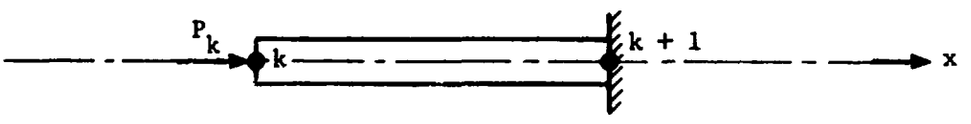


Figure 2 Loading on Bay k

The strain energy in bay k is defined by

$$U_k = \tfrac{1}{2} \int_{x_k}^{x_{k+1}} \frac{P(x)^2}{A(x)\ E(x)}\ dx. \qquad (1)$$

where

P(x) is the axial force,

A(x) is the cross-sectional area of the rod,

E(x) is the modulus of elasticity of the rod material, and

x is the longitudinal axis of the beam.

To integrate Equation (1), the axial force in bay k is assumed *constant* and equal to the axial force at panel point k. That is,

$$P(x) = P_k. \qquad (2)$$

Also, the product A(x) E(x) is assumed to vary linearly as shown in Figure 3.

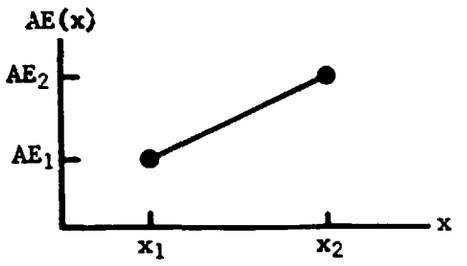


Figure 3 Stiffness Distribution

The equation for a straight line segment as shown in Figure 3 is

$$AE(x) = AE_1 + (x - x_1)\ (AE_2 - AE_1)/(x_2 - x_1). \qquad (3)$$

Substituting Equations (2) and (3) into Equation (1) gives the strain energy of the axial stiffness represented by one line segment i in bay k as

$$U_{i,k} = \tfrac{1}{2}P_k^2 \int_{x_p}^{x_q} \frac{1}{AE_p + \left(x - x_p\right)\left(AE_q - AE_p\right)/\left(x_q - x_p\right)}\, dx. \qquad (4)$$

The subscripts p and q have been introduced to handle the possibility of a line segment extending past the bay limits. Thus, x_p is the greater of x_1 or x_k and x_q is the lesser of x_2 or x_{k+1}. Similarly, AE_p is either AE_1 or AE_k, and AE_q is either AE_2 or AE_{k+1}. The integration is continued for the line segment in adjacent bays, if necessary, until the entire line segment has been used. Performing the integration of Equation (4) yields

$$U_{i,k} = \tfrac{1}{2}P_k^2\, f_k \qquad (5)$$

where

$$f_k = \frac{1}{b}\, \ln \frac{AE_q}{AE_p}, \text{ and} \qquad (5a)$$

$$b = \left(AE_q - AE_p\right)/\left(x_q - x_p\right). \qquad (5b)$$

For constant stiffness, i.e., $AE_p = AE_q$, Equation (5a) is of indefinite form. For this case, integration of Equation (4) yields

$$f_k = \frac{\left(x_q - x_p\right)}{AE_p}. \qquad (5c)$$

Equations (5) are evaluated for every line segment of distributed stiffness. Then, application of Castigliano's Theorem gives the axial deflection of panel point k relative to panel point k + 1 as

$$\Delta\delta_k = \frac{\partial U_k}{\partial P_k} = P_k\, f_k$$

from which

$$P_k = \frac{\Delta\delta_k}{f_k}. \qquad (6)$$

The restraint at panel point k + 1 is removed by application of the transformation

$$\Delta\delta_k = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \delta_k \\ \delta_{k+1} \end{bmatrix}. \tag{7}$$

Substitution of Equations (6) and (7) into Equation (5) (evaluated for all line segments i in bay k) gives the strain energy in bay k as

$$U_k = \tfrac{1}{2} \begin{bmatrix} \delta_k & \delta_{k+1} \end{bmatrix} \begin{bmatrix} z_{k,k} & z_{k,k+1} \\ (\text{sym}) & z_{k+1,k+1} \end{bmatrix} \begin{bmatrix} \delta_k \\ \delta_{k+1} \end{bmatrix} \tag{8}$$

where

$$z_{k,k} = z_{k+1,k+1} = 1/f_k, \text{ and} \tag{8a}$$

$$z_{k,k+1} = -1/f_k. \tag{8b}$$

The kernel matrix in the triple matrix product of Equation (8) is the stiffness matrix that represents the longitudinal stiffness in bay k.

The stiffness matrix for the entire rod is obtained by evaluating Equations (8), (5a), and (5b) for every bay. Each resulting 2x2 bay stiffness matrix is added to previous 2x2 bay stiffness matrices at like panel points to form the stiffness matrix (free-free) for the entire rod.

SPECIAL CASE

For constant stiffness (AE) extending from x_p to x_q in bay k, the stiffness matrix is

$$\frac{AE}{x_q - x_p} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Subroutine STIF2 takes distributed bending stiffness and (on option) distributed shear stiffness of a beam and replaces them with a "free-free" stiffness matrix. The elements of the stiffness matrix are representative stiffness values of the beam at selected points on the beam. These elements are calculated by assuming constant shear and linearly varying bending moment between pairs of the selected points.

The x-stations of the selected points (panel points) are given in {PP}. These x-stations must be in increasing order.

The distributed bending stiffness, EI(x), is assumed to be piecewise linear and is represented by straight line segments as shown in Figure 1. The x-stations of the first and last points for distributed stiffness must coincide with the first and last panel point x-stations, respectively. All other x-stations of the end points for the line segments giving the distributed stiffness are independent of the panel point x-stations. The line segments may or may not be joined; however, there must not be any x voids or overlaps. The distributed bending stiffness is defined in [DEI]. Each row of [DEI] represents one nonvertical line segment. The form of each row of [DEI] is [x_1 x_2 EI_1 EI_2] where x_1, EI_1 give the first end point and x_2, EI_2 give the second end point of a line segment. The x_2 of row 1 of [DEI] must be equal to x_1 of row 2, etc.

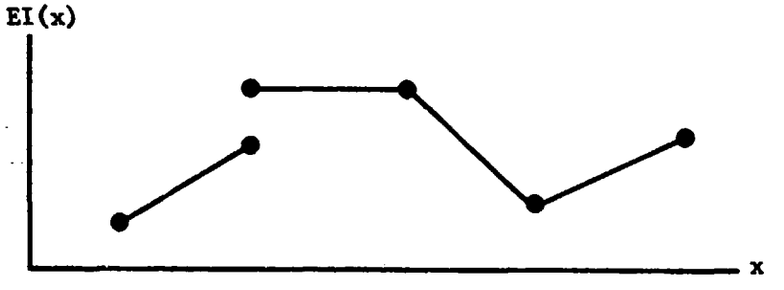


Figure 1 Distributed Bending Stiffness

The distributed shear stiffness, KAG(x), is also assumed to be represented by straight line segments. All statements used above for distributed bending stiffness are applicable to distributed shear stiffness. The distributed shear stiffness is defined in [DKAG]. The form of each row of [DKAG] is [x_1 x_2 KAG_1 KAG_2]. The end point stations x_1 and x_2 for the shear stiffness line segments are independent from the end point stations x_1 and x_2 for the bending stiffness line segments.

The calculated representative stiffness values at the selected panel points are placed in a stiffness matrix given by [Z]. The form of [Z] is

$$[Z]_{2n\times 2n} = \left[\begin{array}{c|c} [Z_{\delta,\delta}] & [Z_{\delta,\theta}] \\ \hline [Z_{\theta,\delta}] & [Z_{\theta,\theta}] \end{array}\right]$$

where n is the number of panel points. Matrix [Z] is symmetric, i.e., $z_{ij} = z_{ji}$. The partition form of [Z] results from the two generalized coordinates (lateral translation δ and rotation θ) at each panel point arranged with all translation coordinates first followed by all rotation coordinates. Each partition of [Z] is square and tri-diagonal, for example,

$$[Z_{\delta,\delta}]_{n\times n} = \begin{bmatrix} z_{1,1} & z_{1,2} & & & & & \\ z_{2,1} & z_{2,2} & z_{2,3} & & & & \\ & z_{3,2} & z_{3,3} & z_{3,4} & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & \cdot & \\ & & & z_{n-1,n-2} & z_{n-1,n-1} & z_{n-1,n} \\ & & & & z_{n,n-1} & z_{n,n} \end{bmatrix}.$$

The sign convention used in this paper to obtain the stiffness matrix [Z] is shown in Figure 2.

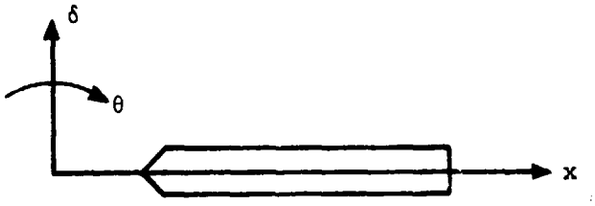


Figure 2 Sign Convention for This Paper

The beam stiffness matrix obtained in this paper is applicable for either the pitch or yaw plane with a change of variables and possible sign changes. For the axis system shown in Figure 3, the following variables would be used.

This Paper (δ, x, θ)	Axis System of Figure 3	
	Pitch (z, x, θ_y)	Yaw (y, x, θ_z)
x	x	x
δ	δ_z	δ_y
θ	θ_y	$-\theta_z$

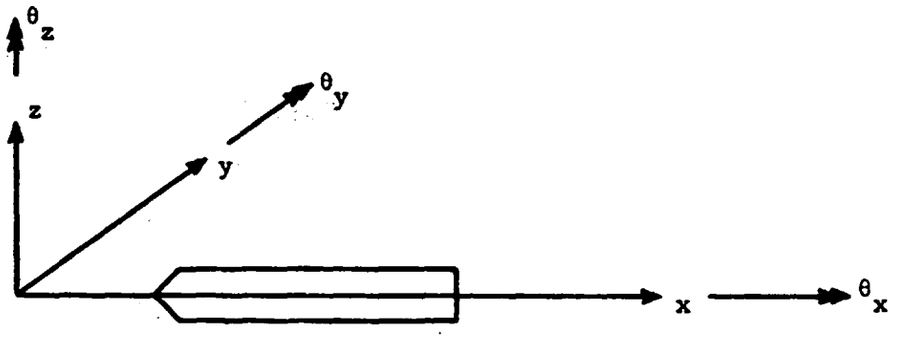


Figure 3 Beam Axis System (Right-Hand)

Note that in order to have a right-hand system such as that shown in Figure 3, the sign of the $\left[Z_{\delta,\theta}\right]$ and $\left[Z_{\theta,\delta}\right]$ partitions of [Z] from this subroutine would have to be changed for the yaw plane.

DESCRIPTION OF TECHNIQUE

The replacement of distributed bending and shear stiffness of a beam by a stiffness matrix is obtained using a strain energy approach as follows. Consider a portion of a beam that is loaded with a shear and moment at panel point k and restrained at panel point k + 1 as shown in Figure 4. The region between panel points k and k + 1 is referred to as bay k.

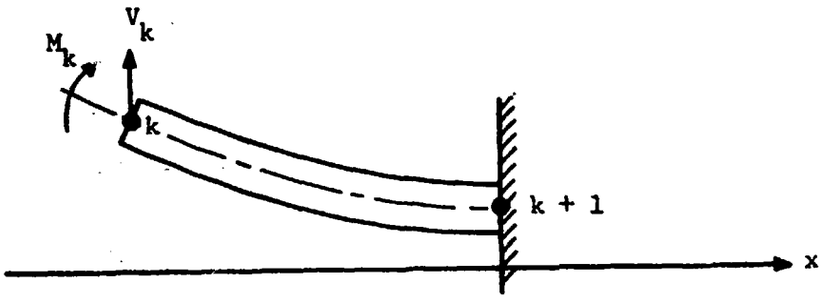


Figure 4 Loading on Bay k

The strain energy in bay k is defined by

$$U_k = \tfrac{1}{2} \int_{x_k}^{x_{k+1}} \left(\frac{M(x)^2}{E(x)\ I(x)} + \frac{V(x)^2}{KA(x)\ G(x)} \right) dx \qquad (1)$$

where

M(x) is the bending moment,

V(x) is the shear,

E(x) is the bending modulus of elasticity of the material,

I(x) is the cross-sectional moment of inertia about the beam's neutral axis,

K is the shape factor (e.g., K = 1 for a solid circular cylinder, K = 0.5 for a thin walled circular cylinder),

A(x) is the cross-sectional area,

G(x) is the shear modulus of elasticity of the material, and

x is the undeformed longitudinal axis of the beam.

To integrate Equation (1), the following assumptions are made. First, the shear in bay k is assumed *constant* and equal to the shear force at panel point k, that is,

$$V(x) = V_k. \qquad (2)$$

Second, the bending moment in bay k is assumed to *vary linearly*, that is,

$$M(x) = M_k + V_k(x - x_k). \tag{3}$$

Third, the products E(x) I(x) and KA(x) G(x) are assumed to vary linearly as shown in Figure 5. The equation for a straight line segment as shown in Figure 5 is

$$EI(x) = EI_1 + (x - x_1)\ (EI_2 - EI_1)/(x_2 - x_1). \tag{4}$$

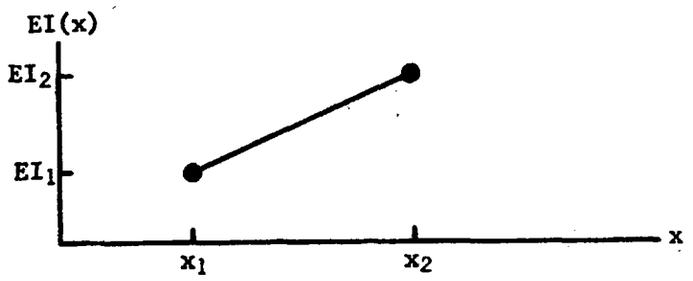


Figure 5 Stiffness Distribution

The strain energy of bending and shear stiffness will be considered separately. The bending stiffness is considered first. Substituting Equations (3) and (4) into Equation (1) gives the strain energy of the bending stiffness represented by one line segment i in bay k as

$$U_{i,k} = \tfrac{1}{2}\begin{bmatrix} V_k & M_k \end{bmatrix}\left(\int_{x_p}^{x_q} \frac{1}{C}\begin{bmatrix} (x - x_k)^2 & (x - x_k) \\ (x - x_k) & 1 \end{bmatrix} dx\right)\begin{bmatrix} V_k \\ M_k \end{bmatrix} \tag{5}$$

where $C = EI_p + (x - x_p)\ (EI_q - EI_p)/(x_q - x_p)$. The subscripts p and q have been introduced to handle the possibility of a line segment extending past the bay limits. Thus x_p is the greater of x_1 or x_k and x_q is the lesser of x_2 or x_{k+1}. Similarly, EI_p is either EI_1 or EI_k and EI_q is either EI_2 or EI_{k+1}. The integration is continued for the line segment in adjacent bays, if necessary, until the entire line segment has been used. Performing the integration of Equation (5) yields the strain energy of bending stiffness as

$$U_{i,k} = \tfrac{1}{2}\begin{bmatrix} V_k & M_k \end{bmatrix} \begin{bmatrix} f_{k,k} & f_{k,k+1} \\ (\text{sym}) & f_{k+1,k+1} \end{bmatrix} \begin{bmatrix} V_k \\ M_k \end{bmatrix} \tag{6}$$

where

$$f_{k,k} = \frac{1}{2b}\left(H_q^2 - H_p^2\right) - \frac{aL_p}{b^2} + \frac{a^2}{b^3} \ln \frac{EI_q}{EI_p} \tag{6a}$$

$$f_{k,k+1} = \frac{L_p}{b} - \frac{a}{b^2} \ln \frac{EI_q}{EI_p} \tag{6b}$$

$$f_{k+1,k+1} = \frac{1}{b} \ln \frac{EI_q}{EI_p} \tag{6c}$$

$$a = \left(EI_p H_q - EI_q H_p\right)\Big/ L_p \tag{6d}$$

$$b = \left(EI_q - EI_p\right)\Big/ L_p \tag{6e}$$

$$H_p = x_p - x_k \tag{6f}$$

$$H_q = x_q - x_k \tag{6g}$$

and

$$L_p = x_q - x_p. \tag{6h}$$

For constant bending stiffness, i.e., $EI_p = EI_q$, Equations (6a) through (6c) are of indefinite form. For this case, integration of Equation (5) yields

$$f_{k,k} = \left(H_q^3 - H_p^3\right)\Big/ 3EI_p \tag{6i}$$

$$f_{k,k+1} = \left(H_q^2 - H_p^2\right)\Big/ 2EI_p \tag{6j}$$

$$f_{k+1,k+1} = L_p/EI \tag{6k}$$

where H_p, H_q, and L_p are given above.

The strain energy of shear stiffness is considered next. The shear stiffness distribution is similar to that shown in Figure 5 and is given by the equation

$$KAG(x) = KAG_1 + (x - x_1)\ (KAG_2 - KAG_1)/(x_2 - x_1). \tag{7}$$

Substituting Equations (2) and (4) into Equation (1) gives the strain energy of the shear stiffness represented by one line segment i in bay k as

$$U_{i,k} = \tfrac{1}{2}\begin{bmatrix} V_k & M_k \end{bmatrix} \begin{bmatrix} \int_{x_p}^{x_q} \frac{dx}{C} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_k \\ M_k \end{bmatrix} \tag{8}$$

where $C = KAG_p + (x - x_p)\ (KAG_q - KAG_p)/(x_q - x_p)$. The subscripts p and q have similar meaning as has been previously discussed for bending stiffness distribution. Performing the integration of Equation (8) yields the same equation as was obtained previously for bending stiffness and is repeated here as

$$U_{i,k} = \tfrac{1}{2}\begin{bmatrix} V_k & M_k \end{bmatrix} \begin{bmatrix} f_{k,k} & f_{k,k+1} \\ (sym) & f_{k+1,k+1} \end{bmatrix} \begin{bmatrix} V_k \\ M_k \end{bmatrix}. \tag{9}$$

Now however,

$$f_{k,k} = \frac{1}{b}\ \ell n\ \frac{KAG_q}{KAG_p} \tag{9a}$$

$$f_{k,k+1} = 0 \tag{9b}$$

$$f_{k+1,k+1} = 0 \tag{9c}$$

and

$$b = (KAG_q - KAG_p)/L_p \tag{9d}$$

As before

$$L_p = x_q - x_p. \tag{9e}$$

For constant shear stiffness, i.e., $KAG_p = KAG_q$, Equation (9a) is of indefinite form. For this case, integration of Equation (5) yields

$$f_{k,k} = L_p/KAG_p. \tag{9f}$$

Equations (6) are evaluated for every line segment of bending stiffness and Equations (9) are evaluated for every line segment of shear stiffness. Then, application of Castigliano's Theorem gives the lateral translation and rotation of panel point k relative to panel point k+1 as

$$\begin{bmatrix} \Delta\delta_k \\ \Delta\theta_k \end{bmatrix} = \begin{bmatrix} \frac{\partial U_k}{\partial V_k} \\ \frac{\partial U_k}{\partial Mk} \end{bmatrix} = \begin{bmatrix} f_{k,k} & f_{k,k+1} \\ (\text{sym}) & f_{k+1,k+1} \end{bmatrix} \begin{bmatrix} V_k \\ M_k \end{bmatrix}. \tag{10}$$

Solving Equation (10) for V_k and M_k in terms of $\Delta\delta_k$ and $\Delta\theta_k$ and substituting into Equation (6) [or (9)] gives the strain energy in bay k as

$$U_k = \tfrac{1}{2}\begin{bmatrix} \Delta\delta_k & \Delta\theta_k \end{bmatrix} \begin{bmatrix} K_{\delta\delta} & K_{\delta\theta} \\ (\text{sym}) & K_{\theta\theta} \end{bmatrix} \begin{bmatrix} \Delta\delta_k \\ \Delta\theta_k \end{bmatrix} \tag{11}$$

where

$$K_{\delta\delta} = \frac{f_{k+1,k+1}}{D} \tag{11a}$$

$$K_{\delta\theta} = -\frac{f_{k+1,k+1}}{D} \tag{11b}$$

$$K_{\theta\theta} = \frac{f_{k,k}}{D} \tag{11c}$$

and

$$D = f_{k,k}\, f_{k+1,k+1} - f^2_{k,k+1}. \tag{11d}$$

The restraint at panel point k + 1 is removed by application of the transformation

$$\begin{bmatrix} \Delta\delta_k \\ \Delta\theta_k \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & -L_k \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \delta_k \\ \delta_{k+1} \\ \theta_k \\ \theta_{k+1} \end{bmatrix} \tag{12}$$

where

$$L_k = x_{k+1} - x_k. \tag{12a}$$

Using this transformation in Equation (11) gives the final strain energy expression as

$$U_k = \tfrac{1}{2} \begin{bmatrix} \delta_k \\ \delta_{k+1} \\ \theta_k \\ \theta_{k+1} \end{bmatrix}^T \begin{bmatrix} z_{k,k} & z_{k,k+1} & z_{k,k+n} & z_{k,k+n+1} \\ & z_{k+1,k+1} & z_{k+1,k+n} & z_{k+1,k+n+1} \\ & & z_{k+n,k+n} & z_{k+n,k+n+1} \\ (\text{sym}) & & & z_{k+n+1,k+n+1} \end{bmatrix} \begin{bmatrix} \delta_k \\ \delta_{k+1} \\ \theta_k \\ \theta_{k+1} \end{bmatrix} \tag{13}$$

where

$$z_{k,k} = K_{\delta,\delta} \tag{13a}$$

$$z_{k,k+1} = -K_{\delta,\delta} \tag{13b}$$

$$z_{k,k+n} = K_{\delta\theta} \tag{13c}$$

$$z_{k,k+n+1} = -L_k K_{\delta\delta} - K_{\delta\theta} \tag{13d}$$

$$z_{k+1,k+1} = K_{\delta\delta} \tag{13e}$$

$$z_{k+1,k+n} = -K_{\delta\theta} \tag{13f}$$

$$z_{k+1,k+n+1} = L_k K_{\delta\delta} + K_{\delta\theta} \tag{13g}$$

$$z_{k+n,k+n} = K_{\theta\theta} \tag{13h}$$

$$z_{k+n,k+n+1} = -L_k K_{\delta\theta} - K_{\theta\theta} \tag{13i}$$

$$z_{k+n+1,k+n+1} = L_k^2 K_{\delta\delta} + 2L_k K_{\delta\theta} + K_{\theta\theta} \tag{13j}$$

and n is the number of panel points. The kernel matrix in the triple matrix product of Equation (13) is the stiffness matrix which represents the bending and shear stiffness in bay k.

The stiffness matrix for the entire beam is obtained by evaluating Equations (11) and (13) for every bay. Each resulting 4x4 bay stiffness matrix is added to previous 4x4 bay stiffness matrices at like panel points to form the stiffness matrix (free-free) for the entire beam.

SPECIAL CASE

For constant bending stiffness (EI) and infinite shear stiffness extending to the bay limits $\left(x_k \text{ and } x_{k+1}\right)$, the stiffness matrix is

$$\frac{EI}{L_k^3}\begin{bmatrix} 12 & -12 & -6L_k & -6L_k \\ & 12 & 6L_k & 6L_k \\ (\text{sym}) & & 4L_k^2 & 2L_k^2 \\ & & & 4L_k^2 \end{bmatrix}$$

where

$$L_k = x_{k+1} - x_k.$$

Subroutine SYMLH symmetrizes a square matrix by placing values from above the diagonal below the diagonal. That is,

$$a_{ji} = a_{ij} \qquad (i < j)$$

EXAMPLE

If [A] is input to Subroutine SYMLH as

$$[A]_{3x3} = \begin{bmatrix} 1. & 2. & 3. \\ 4. & 5. & 6. \\ 7. & 8. & 9. \end{bmatrix},$$

the matrix output from this subroutine will be

$$[A]_{3x3} = \begin{bmatrix} 1. & 2. & 3. \\ 2. & 5. & 6. \\ 3. & 6. & 9. \end{bmatrix}.$$

Subroutine SYMUH symmetrizes a square matrix by placing values from below the diagonal above the diagonal. That is,

$$a_{ij} = a_{ji} \qquad (i \neq j)$$

EXAMPLE

If [A] is input to Subroutine SYMUH as

$$[A]_{3x3} = \begin{bmatrix} 1. & 2. & 3. \\ 4. & 5. & 6. \\ 7. & 8. & 9. \end{bmatrix},$$

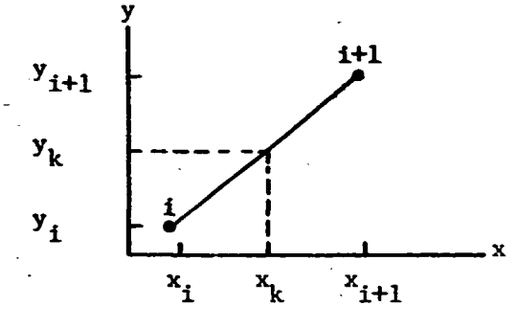
the matrix output from this subroutine will be

$$[A]_{3x3} = \begin{bmatrix} 1. & 4. & 7. \\ 4. & 5. & 8. \\ 7. & 8. & 9. \end{bmatrix}.$$

Subroutine TERP1 performs interpolation assuming a linear function between known points. The x and y coordinates of the known points are given by the elements of {XA} and the corresponding elements in a column of [YA], respectively. Each column of [YA] gives the y coordinates of a different set of points. Interpolated y coordinates are calculated at selected x coordinates which are given by the elements of {XZ}. These interpolated y coordinates are placed in [YZ]. Each column of [YZ] has interpolated values of the respective column of [YA]. Extrapolation assuming a linear function is performed when any element of {XZ} exceeds the limits of {XA}.

DERIVATION OF TECHNIQUE

Given the x,y coordinates of points i and i+1, the coordinate y_k at x_k is to be found by interpolation assuming a linear function.



The equation of a straight line is $y(x) = ax + b$, or in matrix notation, $y(x) = [x \quad 1] \begin{vmatrix} a \\ b \end{vmatrix}$. The coefficients a and b can be determined because the coordinates x_i, y_i and x_{i+1}, y_{i+1} are known. That is,

$$\begin{bmatrix} y_i \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i & 1 \\ x_{i+1} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

from which

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{x_i - x_{i+1}} \begin{bmatrix} 1 & -1 \\ -x_{i+1} & x_i \end{bmatrix} \begin{bmatrix} y_i \\ y_{i+1} \end{bmatrix}$$

giving

$$y(x) = \frac{1}{x_i - x_{i+1}} \begin{bmatrix} x & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -x_{i+1} & x_i \end{bmatrix} \begin{bmatrix} y_i \\ y_{i+1} \end{bmatrix}.$$

Therefore

$$y(x_k) = y_k = y_i + (x_k - x_i)\,(y_{i+1} - y_i)/(x_{i+1} - x_i). \qquad (1)$$

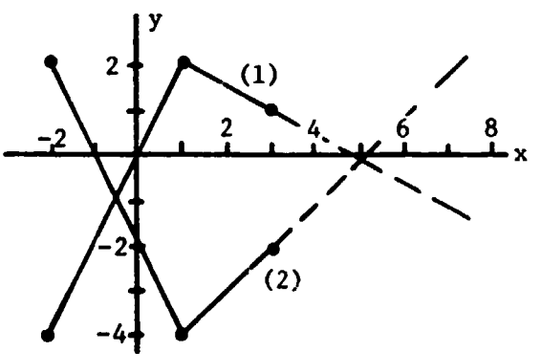
The following table gives the correlation between the nomenclature of Equation (1) and that used in the Fortran coding in the subroutine.

Equation (1)	Fortran Coding*
x_i	XA(I)
x_{i+1}	XA(I+1)
y_i	YA(I) *
y_{i+1}	YA(I+1) *
x_k	XZ(K)
y_k	YZ(K) *

*Subscript j, denoting different sets of points, has been omitted for clarity.

EXAMPLE

Consider the following two sets of points denoted by (1) and (2) in the sketch.



The coordinates of the points are given by

$$\{XA\} = \begin{bmatrix} -2. \\ 1. \\ 3. \end{bmatrix} \quad \text{and} \quad [YA] = \begin{bmatrix} -4. & 2. \\ 2. & -4. \\ 1. & -2. \end{bmatrix}.$$

{XA} gives the x coordinates of the points in both sets (1) and (2). Column 1 of [YA] gives the y coordinates of the points in set (1) and column 2 of [YA] gives the y coordinates of the points in set (2). The y values are wanted at x = -1. and x = 7., that is, at $\{XZ\} = \begin{vmatrix} -1. \\ 7. \end{vmatrix}$. At x = -1., interpolated values of y = -2. and y = 0. are calculated from columns 1 and 2 of [YA], respectively, using rows 1 and 2 of {XA} and [YA]. At x = 7., extrapolated values of y = -1. and y = 2. are calculated from columns 1 and 2 of [YA], respectively, using rows 2 and 3 of {XA} and [YA]. The final result is

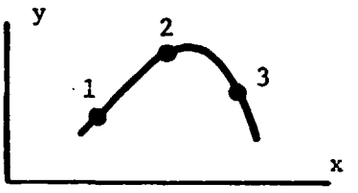
$$[YZ] = \begin{bmatrix} -2. & 0. \\ -1. & 2. \end{bmatrix}.$$

To relate this example problem to a practical problem, consider {XA} to be the collocation points (panel points) of a vehicle and [YA] to be the modal displacements for two modes. Gyros are to be placed at stations x = -1. and 7. (i.e., {XZ}). The modal displacements ([YZ]) are to be found at these stations.

Subroutine TERP2 performs interpolation assuming a diparabolic function between known points. A parabolic function is used where only three points are available. A diparabolic function is obtained from the weighted average of two adjacent parabolas and will be explained later. The x and y coordinates of the known points are given by the elements of {XA} and the corresponding elements in a column of [YA], respectively. Each column of [YA] gives the y coordinates of a different set of points. Interpolated y coordinates are calculated at selected x coordinates which are given by the elements of {XZ}. These interpolated y coordinates are placed in [YZ]. Each column of [YZ] has interpolated values of the respective column of [YA]. Extrapolation assuming a parabolic function is performed when any element of {XZ} exceeds the limits of {XA}.

DERIVATION OF TECHNIQUE

The diparabolic interpolation procedure is obtained as follows. Because this procedure is dependent upon using parabolas, the parabola will be considered first. Given the x,y coordinates of points 1, 2, and 3 below, a parabola is to be fitted to these points.



The equation for a parabola with axis parallel to the y axis is

$$y(x) = Ax^2 + Bx + C$$

or

$$y(H) = \begin{bmatrix} H^2 & H & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

where

$$H = \frac{x - x_1}{x_2 - x_1} \qquad (1a)$$

or

$$H = \frac{x - x_2}{x_3 - x_2} \qquad (1b)$$

is used for ease in later algebraic calculations. The coefficients a, b, and c can be determined because the x (or H) and y coordinates at points 1, 2, and 3 are known. That is,

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} H_1^2 & H_1 & 1 \\ H_2^2 & H_2 & 1 \\ H_3^2 & H_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

from which

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = [\psi] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

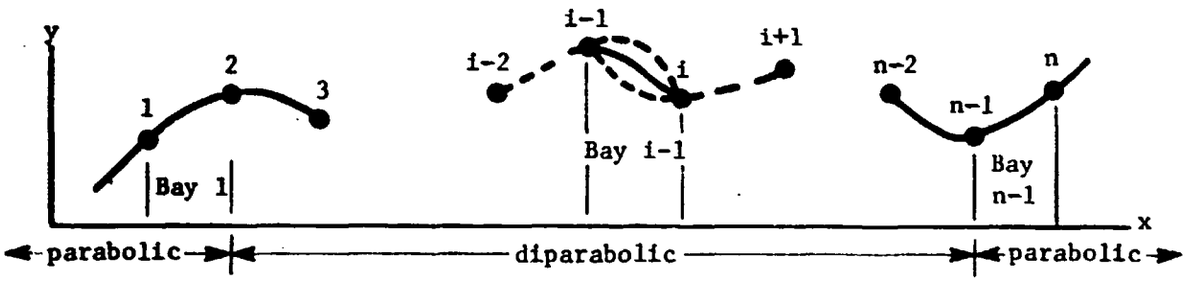
where

$$[\psi] = \frac{\begin{bmatrix} H_2-H_3 & -(H_1-H_3) & H_1-H_2 \\ -(H_2-H_3)(H_2+H_3) & (H_1-H_3)(H_1+H_3) & -(H_1-H_2)(H_1+H_2) \\ H_2H_3(H_2-H_3) & -H_1H_3(H_1-H_3) & H_1H_2(H_1-H_2) \end{bmatrix}}{H_1^2\ (H_2-H_3) - H_2^2\ (H_1-H_3) + H_3^2\ (H_1-H_2)} \qquad (2)$$

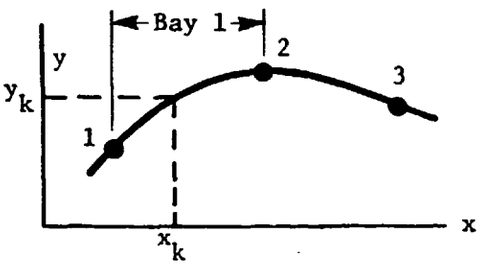
therefore

$$y(H) = [H^2 \quad H \quad 1]\ [\psi] \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}. \qquad (3)$$

For a given set of points, as shown on the following page, a parabolic function is used to the left of point 1 and between points 1 and 2. Also, a parabolic function is used to the right of the last point n and between point n-1 and n. Diparabolic functions are used between all other points.



Bay 1 (and to the left of point 1)



The coordinate y_k at x_k is found as follows:

From Equation (1a),

$$H_k = (x_k - x_1)/(x_2 - x_1) \tag{4a}$$

$$H_1 = (x_1 - x_1)/(x_2 - x_1) = 0$$

$$H_2 = (x_2 - x_1)/(x_2 - x_1) = 1$$

$$H_3 = (x_3 - x_1)/(x_2 - x_1) = D. \tag{4b}$$

Using these expressions with Equations (2) and (3), the final result is obtained as

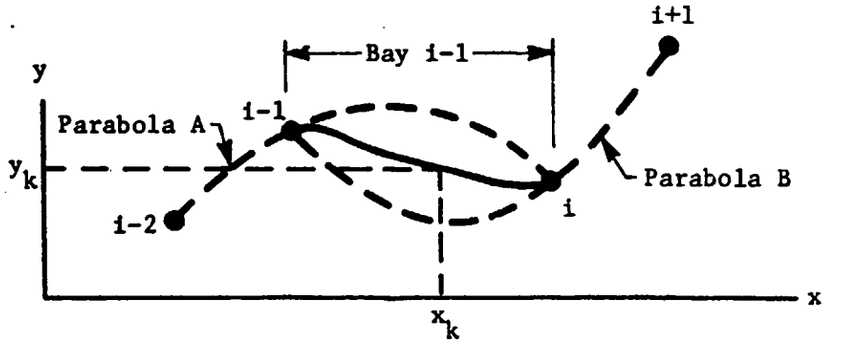
$$y(H_k) = y_k = \begin{vmatrix} H_k^2 & H_k & 1 \end{vmatrix} \begin{bmatrix} 1/D & 1/(1-D) & -1/D(1-D) \\ -(1+D)/D & -D/(1-D) & 1/D(1-D) \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{4c}$$

The following table gives the correlation between the nomenclature of Equations (4a, b, c) and that used in the Fortran coding in the subroutine.

Equations (4a,b,c)	Fortran Coding*	
x_m	XA(m)	m = 1, 2, 3
y_m	YA(m) *	m = 1, 2, 3
H_k	H	
x_k	XZ(K)	
y_k	YZ(K) *	

*Subscript j, denoting different sets of points, has been omitted for clarity.

Interior Bay i-1



The coordinate y_k at x_k is found as follows. A diparabolic function in bay i-1 of the above sketch is obtained as the weighted average of parabolas, A and B. That is,

$$y(H) = (1-H) \cdot y_A + H \cdot y_B \qquad (5)$$

where

$$H = \frac{x - x_{i-1}}{x_i - x_{i-1}} \qquad (6a)$$

as in either Equations (1a) or (1b).

For parabola A:

$$H_{i-2} = (x_{i-2} - x_{i-1})/(x_i - x_{i-1}) \equiv C \qquad (6b)$$

$$H_{i-1} = (x_{i-1} - x_{i-1})/(x_i - x_{i-1}) = 0$$

$$H_i = (x_i - x_{i-1})/(x_i - x_{i-1}) = 1$$

From Equations (2) and (3),

$$y_A(H) = [H^2 \quad H \quad 1] \left[\psi_A\right] \begin{bmatrix} y_{i-2} \\ y_{i-1} \\ y_i \end{bmatrix}$$

where

$$\left[\psi_A\right] = \begin{bmatrix} -1/C(1-C) & 1/C & 1/(1-C) \\ 1/C(1-C) & -(1+C)/C & -C/(1-C) \\ 0 & 1 & 0 \end{bmatrix}.$$

For parabola B:

$$H_{i-1} = (x_{i-1} - x_{i-1})/(x_i - x_{i-1}) = 0$$

$$H_i = (x_i - x_{i-1})/(x_i - x_{i-1}) = 1$$

$$H_{i+1} = (x_{i+1} - x_{i-1})/(x_i - x_{i-1}) \equiv D \qquad (6c)$$

From Equations (2) and (3)

$$y_B(H) = [H^2 \quad H \quad 1] \left|\psi_B\right| \begin{bmatrix} y_{i-1} \\ y_i \\ y_{i+1} \end{bmatrix}$$

where

$$\left|\psi_B\right| = \begin{bmatrix} 1/D & 1/(1-D) & -1/D(1-D) \\ -(1+D)/D & -D/(1-D) & 1/D(1-D) \\ 1 & 0 & 0 \end{bmatrix}.$$

Substituting these expressions for y_A and y_B into Equation (5) and evaluating at point k results in

$$y(H_k) = y_k = \begin{bmatrix} H_k^3 & H_k^2 & H_k & 1 \end{bmatrix} [\psi] \begin{bmatrix} y_{i-2} \\ y_{i-1} \\ y_i \\ y_{i+1} \end{bmatrix} \qquad (6d)$$

where

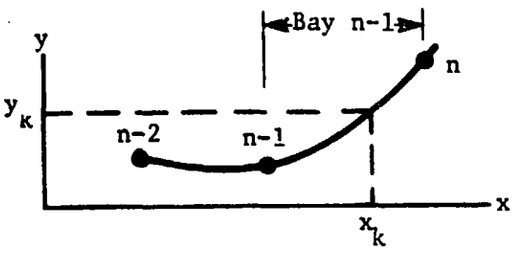
$$[\psi] = \begin{bmatrix} \frac{1}{C(1-C)} & \frac{C-D}{CD} & \frac{D-C}{(1-C)(1-D)} & \frac{-1}{D(1-D)} \\ \frac{-2}{C(1-C)} & \frac{2D-C}{CD} & \frac{1-2D+C}{(1-C)(1-D)} & \frac{1}{D(1-D)} \\ \frac{1}{C(1-C)} & \frac{-(1+C)}{C} & \frac{-C}{1-C} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The following table gives the correlation between the nomenclature of Equations (6a, b, c, d) and that used in the Fortran coding in the subroutine.

Equations (6a,b,c,d)	Fortran Coding*	
x_{i-m}	XA(I-m)	m = 2, 1, 0, -1
y_{i-m}	YA(I-m) *	m = 2, 1, 0, -1
H_k	H	
x_k	XZ(K)	
y_k	YZ(K) *	

*Subscript j, denoting different sets of points, has been omitted for clarity.

Bay n-1 (and to the right of point n)



The coordinate y_k at x_k is found as follows.

From Equation (1b)

$$H_k = \left(x_k - x_{n-1}\right)/\left(x_n - x_{n-1}\right) \qquad (7a)$$

$$H_{n-2} = \left(x_{n-2} - x_{n-1}\right)/\left(x_n - x_{n-1}\right) = C \qquad (7b)$$

$$H_{n-1} = \left(x_{n-1} - x_{n-1}\right)/\left(x_n - x_{n-1}\right) = 0$$

$$H_n = \left(x_n - x_{n-1}\right)/\left(x_n - x_{n-1}\right) = 1$$

Using these expressions with Equations (2) and (3), the final result is obtained as

$$y\left(H_k\right) = y_k = \begin{bmatrix} H_k^2 & H_k & 1 \end{bmatrix} \begin{bmatrix} -1/C(1-C) & 1/C & 1/(1-C) \\ 1/C(1-C) & -(1+C)/C & -C/(1-C) \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}. \qquad (7c)$$

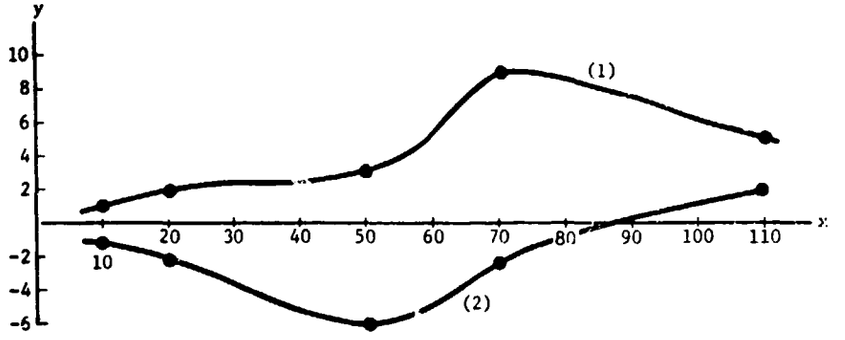
The following table gives the correlation between the nomenclature of Equations (7a, b, c) and that used in the Fortran coding in the subroutine.

Equations (7a,b,c)	Fortran Coding*	
x_{n-m}	XA(NXA-m)	m = 0, 1, 2
y_{n-m}	YA(NXA-m) *	m = 0, 1, 2
H_k	H	
x_k	XZ(k)	
y_k	YZ(K) *	

*Subscript j, denoting different sets of points, has been omitted for clarity.

EXAMPLE

Consider the following two sets of points denoted by (1) and (2).



The coordinates of the points are given by

$$\{XA\} = \begin{bmatrix} 10. \\ 20. \\ 50. \\ 70. \\ 110. \end{bmatrix} \quad \text{and} \quad [YA] = \begin{bmatrix} 1. & -1. \\ 2. & -2. \\ 3. & -6. \\ 9. & -2. \\ 5. & 2. \end{bmatrix}.$$

{XA} gives the x coordinates of the points in both sets (1) and (2). Column 1 of [YA] gives the y coordinates of the points in set (1) and column (2) of [YA] gives the y coordinates of the points in set (2). The y values are wanted at x = 5., x = 100.,

and x = 65., that is, at

$$\{XZ\} = \begin{bmatrix} 5. \\ 100. \\ 65. \end{bmatrix}.$$

At x = 5., extrapolated values of y = 0.375 and y = -0.5625 are calculated from columns 1 and 2 of [YA], respectively, using rows 1, 2, and 3 of {XA} and [YA]. At x = 100., interpolated values of y = 8. and y = 1.5 are calculated from columns 1 and 2 of [YA], respectively, using rows 3, 4, and 5 of {XA} and [YA]. At x = 65., interpolated values of y = 7.775 and y = -3.03125 are calculated from columns 1 and 2 of [YA], respectively, using rows 2, 3, 4, and 5 of {XA} and [YA]. The result is

$$[YZ] = \begin{bmatrix} 0.375 & -0.5625 \\ 8. & 1.5 \\ 7.775 & -3.03125 \end{bmatrix}.$$

To relate this example problem to a practical problem, consider {XA} to be the collocation points (panel points) of a vehicle and [YA] to be the modal displacements for two modes. Gyros are to be placed at stations x = 5., 100., and 65. (i.e., {XZ}). The modal displacements ([YZ]) are to be found at these stations.

REFERENCE

Griffin, J.A.: "A Diparabolic Method of Four-Point Interpolation." *Journal of the Aeronautical Sciences*, Vol. 28, No. 2, Readers' Forum, February 1961.

Subroutine TRAE2 calculates time response additional equations. That is

$$\{Z(t)\} = [A]\{\ddot{X}(t)\} + [B]\{\dot{X}(t)\} + [C]\{X(t)\} + [D]\{F(t)\} + \{E\} \qquad (1)$$

where $\{\ddot{X}(t)\}$, $\{\dot{X}(t)\}$, $\{X(t)\}$, and $\{F(t)\}$ are the time response and force previously calculated in a time response subroutine, for example TRSP1. The TRSP1 tape is read in the subroutine TRAE2 to calculate $\{Z(t)\}$. The coefficient matrices [A], [B], [C], [D], and {E} are supplied or omitted, on option, to this subroutine to calculate $\{Z(t)\}$ which may be shear, bending moment, displacement, etc, depending on the choice of coefficient matrices.

Subroutine TRANS calculates the transpose (interchange of rows and columns) of a matrix. If $[A]_{NRAxNCA}$ is the matrix to be transposed, then the result is

$$[Z]_{NCAxNRA} = [A]^T$$

where

$$z_{ji} = a_{ij} \qquad \begin{pmatrix} i = 1, \text{ NRA} \\ j = 1, \text{ NCA} \end{pmatrix}$$

NRA is the number of rows of [A], and NCA is the number of columns of [A].

Theorem: The transpose of the product of a set of matrices is equal to the product of the transposed matrices taken in reverse order. That is,

$$([A][B][C])^T = [C]^T [B]^T [A]^T.$$

Theorem: The transpose of the sum of a set of matrices is equal to the sum of the transposed matrices. That is,

$$([A] + [B] + [C])^T = [A]^T + [B]^T + [C]^T.$$

EXAMPLE

If the input matrix is

$$[A]_{2x3} = \begin{bmatrix} 1. & 2. & 3. \\ 4. & 5. & 6. \end{bmatrix}$$

then

$$[Z]_{3x2} = [A]^T_{2x3} = \begin{bmatrix} 1. & 4. \\ 2. & 5. \\ 3. & 6. \end{bmatrix}.$$

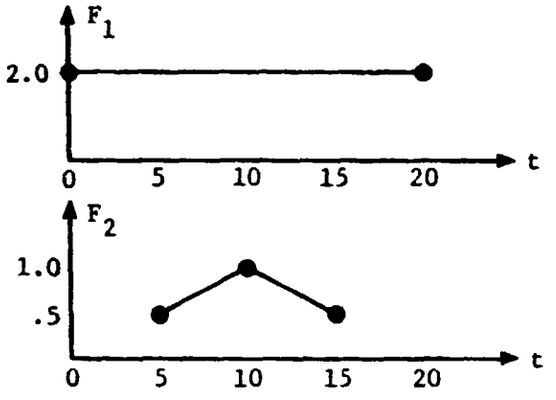
Subroutine TRSP1 solves the second order differential equation

$$[A]\{\ddot{X}(t)\} + [B]\{\dot{X}(t)\} + [C]\{X(t)\} = [D]\{F(t)\} \qquad (1)$$

for

$$\{\ddot{X}(t)\},\ \{\dot{X}(t)\},\ \text{and}\ \{X(t)\}$$

using a fourth order Runge-Kutta numerical integration procedure (Ref 1). Matrices [A], [B], [C], and [D] are input directly to this subroutine. For a structural problem, [A] is the mass matrix, [B] is the damping matrix, [C] is the stiffness matrix, and [D] is the transpose of the vibration mode shapes if the equations are a modal representation of the structure. The force {F(t)} is calculated internally in the subroutine using linear interpolation with [TABT] and [TABF] which are both input to the subroutine. As an illustration of the use of [TABT] and [TABF] consider the following example.



The table (A table is defined in this report as a matrix that may have incomplete column data in some rows.) giving the independent variable coordinate t is

$$[TABT] = \begin{bmatrix} 0. & 20. & \\ 5. & 10. & 15. \end{bmatrix}.$$

The table giving the corresponding coordinates of the dependent variable F is

$$[TABF] = \begin{bmatrix} 2. & 2. & \\ .5 & 1. & .5 \end{bmatrix}.$$

The following values for {F(t)} will be obtained by linear interpolation at t = 7.

$$\{F(t = 7)\} = \begin{bmatrix} 2. \\ .7 \end{bmatrix}.$$

The Runge-Kutta method of numerical integration has several desirable features that may be summarized as follows. The method is generally considered to have good convergence qualities. Forward integration and iteration procedures can sometimes be unstable so that a calculated solution oscillates with rapidly increasing amplitude about the true solution. The Runge-Kutta method does not seem to be so susceptible to this difficulty.

DESCRIPTION OF TECHNIQUE

Matrix [A] must be nonsingular because the highest derivative will be calculated from modification of Equation (1).

$$\{\ddot{X}(t)\} = [A]^{-1}[D]\{F(t)\} - [A]^{-1}[B]\{\dot{X}(t)\} - [A]^{-1}[C]\{X(t)\} \qquad (1a)$$

Runge-Kutta numerical integration is then used to integrate $\{\ddot{X}(t)\}$ to obtain $\{\dot{X}(t)\}$ and then again used to integrate $\{\dot{X}(t)\}$ to obtain $\{X(t)\}$.

REFERENCES

1. S. Gill: "A Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computing Machine," *Proceedings Cambridge Philosophical Society* 47:96-108 (1951).

Subroutine TRSP1A solves the second order differential equation

$$[A]\{\ddot{X}(t)\} + [B]\{\dot{X}(t)\} + [C]\{X(t)\} = [D]\{F(t)\} \qquad (1)$$

for

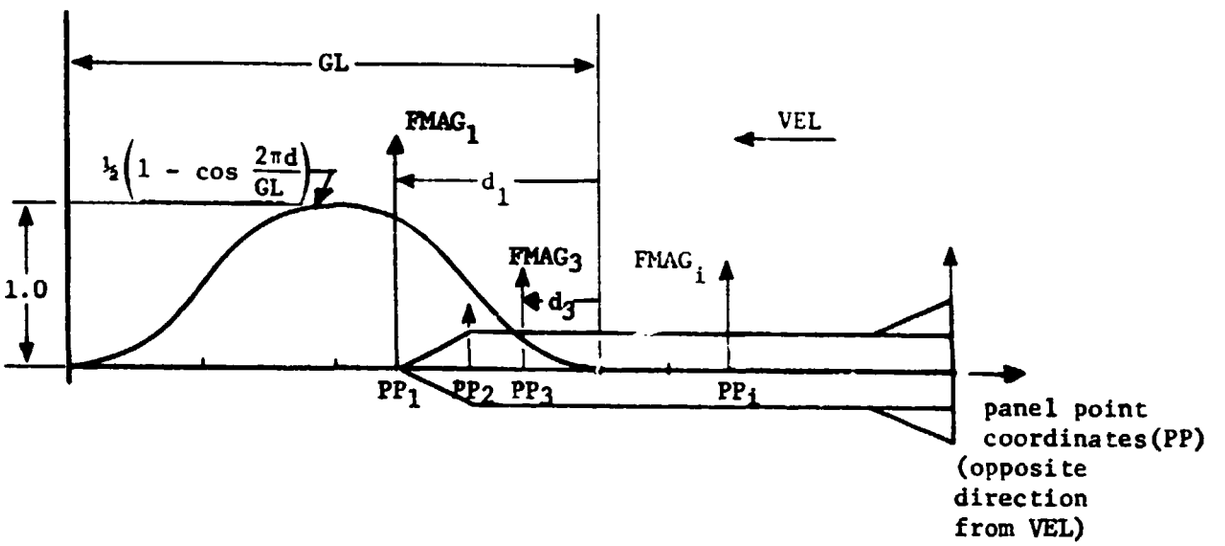
$$\{\ddot{X}(t)\},\ \{\dot{X}(t)\},\ \text{and}\ \{X(t)\}$$

using a fourth order Runge-Kutta numerical integration procedure (Ref 1). Matrices [A], [B], [C], and [D] are input directly to this subroutine. For a structural problem, [A] is the mass matrix, [B] is the damping matrix, [C] is the stiffness matrix, and [D] is the transpose of the vibration mode shapes if the equations are a modal representation of the structure. The force {F(t)} is calculated internally in the subroutine as follows.

{F(t)} is size (NF×1) and is obtained considering a (1 - cos)/2 variation in the coordinate force. Consider the following examples.

EXAMPLE 1:

Missile Penetrating a Stationary Gust at the Rate - VEL -



Note that the ratio GL/VEL is the period of the gust.

If $FMAG_i$ is defined to be the maximum normal force at station (i), then the i^{th} normal force F(t) varies with time as -

$$F_i(t) = FMAG_i \ (\tfrac{1}{2}) \left(1 - \cos \frac{2\pi\, d_i}{GL}\right) \text{ for } 0 \leq d_i \leq GL$$

$$F_i(t) = 0 \qquad \text{for} \begin{cases} d_i \leq 0, \\ GL \leq d_i \end{cases}$$

where, $d_i = d_1 - \left(PP_i - PP_1 \right)$ and

$d_1 = VEL \times (t - STARTT)$.

(assumes PP_1 is at beginning of gust at t = STARTT.)

EXAMPLE 2:

In the event a sudden envelopment forcing function (zero lag time for all stations) is desired, two alternatives exist:

1) Supply FMAG and PP as scalar quantities and $PP_1 = PP$. This sets the lag time to zero, $\left(PP_i - PP_i = 0 \right)$; or

2) Supply FMAG as a vector and PP as a vector with all stations the same and equal to PP_1. This sets the lag time at all stations to zero.

The Runge-Kutta method of numerical integration has several desirable features that may be summarized as follows. The method is generally considered to have good convergence qualities. Forward integration and iteration procedures can sometimes be unstable so that a calculated solution oscillates with rapidly increasing amplitude about the true solution. The Runge-Kutta method does not seem to be so susceptible to this difficulty.

DESCRIPTION OF TECHNIQUE

Matrix [A] must be nonsingular because the highest derivative will be calculated from modification of Equation (1).

$$\{\ddot{X}(t)\} = [A]^{-1}\,[D]\,\{F(t)\} - [A]^{-1}\,[B]\,\{\dot{X}(t)\} - [A]^{-1}\,[C]\,\{X(t)\}. \quad (1a)$$

Runge-Kutta numerical integration is then used to integrate $\{\ddot{X}(t)\}$ to obtain $\{\dot{X}(t)\}$ and then again used to integrate $\{\dot{X}(t)\}$ to obtain $\{X(t)\}$.

REFERENCES

1. S. Gill: "A Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computing Machine." *Proceedings Cambridge Philosophical Society* 47.96-108 (1951).

Subroutine TRSP1B is a modification of Subroutine TRSP1 to use one less matrix space so that larger size matrices can be used. The second order differential equation to be solved is

$$\{\ddot{X}(t)\} + [B]\{\dot{X}(t)\} + [C]\{X(t)\} = [D]\{F(t)\}. \qquad (1)$$

Comparing this equation with Equation (1) of Subroutine TRSP1 shows that

TRSP1B

$[B]$ is $[A]^{-1}[B]$,

$[C]$ is $[A]^{-1}[C]$,

$[D]$ is $[A]^{-1}[D]$.

All other comments in the explanation of Subroutine TRSP1 apply to this subroutine.

Subroutine TRSP1C is a modification of Subroutine TRSP1A to use one less matrix space so that larger size matrices can be used. The second order differential equation to be solved is

$$\{\ddot{X}(t)\} + [B]\{\dot{X}(t)\} + [C]\{X(t)\} = [D]\{F(t)\}. \qquad (1)$$

Comparing this equation with Equation (1) of subroutine TRSP1A shows that

TRSP1C

$[B]$ is $[A]^{-1}[B]$,

$[C]$ is $[A]^{-1}[C]$,

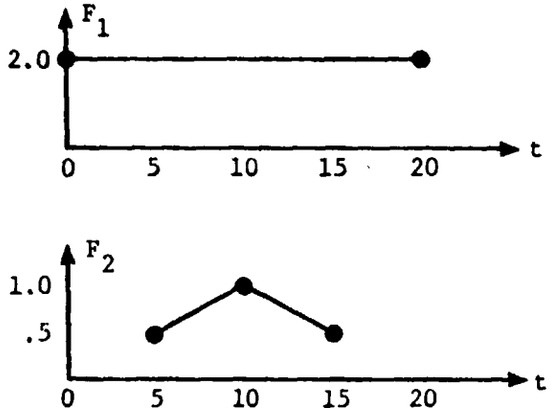
$[D]$ is $[A]^{-1}[D]$.

All other comments in the explanation of subroutine TRSP1A apply to this subroutine.

Subroutine TRSP2 solves the second order differential equation

$$[A]\{\ddot{X}(t)\} + [B]\{\dot{X}(t)\} + [C]\{X(t)\} = [D]\{F(t)\} \qquad (1)$$

for $\{\ddot{X}(t)\}$, $\{\dot{X}(t)\}$, and $\{X(t)\}$ using a third order Newmark-Chan-Beta numerical integration procedure (Ref 1). Matrices [A], [B], [C], and [D] are input directly to this subroutine. For a structural problem, [A] is the mass matrix, [B] is the damping matrix, [C] is the stiffness matrix, and [D] is the transpose of the vibration mode shapes if the equations are a modal representation of the structure. The force {F(t)} is calculated internally in the subroutine using linear interpolation with [TABT] and [TABF] which are both input to the subroutine. As an illustration of the use of [TABT] and [TABF] consider the following example.



The table (A table is defined in this report as a matrix that may have incomplete column data in some rows.) giving the independent variable coordinate t is

$$[TABT] = \begin{bmatrix} 0. & 20. & \\ 5. & 10. & 15. \end{bmatrix}.$$

The table giving the corresponding coordinates of the dependent variable F is

$$[TABF] = \begin{bmatrix} 2. & 2. & \\ .5 & 1. & .5 \end{bmatrix}.$$

The following values for $\{F(t)\}$ will be obtained by linear interpolation at t = 7.

$$\{F(t = 7)\} = \begin{bmatrix} 2. \\ .7 \end{bmatrix}.$$

DESCRIPTION OF TECHNIQUE

Because the difference equation contains displacement terms for three consecutive time intervals, it is required to express a displacement quantity in terms of initial velocity and initial displacement in order to start the procedure. Because [A] must be nonsingular in this method, it was decided to modify Equation (1) into the form

$$\{\ddot{X}(t)\} + [A]^{-1}\,[B]\,\{\dot{X}(t)\} + [A]^{-1}\,[C]\,\{X(t)\} = [A]^{-1}\,[D]\,\{F(t)\}.$$

Thus, only three matrices are needed compared to four matrices in Equation (1). The starting equation is thus given as:

$$\{X\}_1 = [S^{-1}\,P]\,\{X_0\} + [S^{-1}\,Q]\,\{\dot{X}_0\} + \beta h^2\,[S^{-1}\,A^{-1}\,D]\,\{F\}_1 + [S^{-1}\,R\,A^{-1}\,D]\,\{F\}_0$$

where

$$[P] = [I] + \frac{h}{2}\,[A^{-1}\,B] = (\tfrac{1}{2} - \beta)\,h^2\,[A^{-1}\,C]$$

$$- (\tfrac{1}{4} - \beta)\,h^3\,[A^{-1}\,B]\,[A^{-1}\,C]$$

$$[Q] = h\,[I] - (\tfrac{1}{4} - \beta)\,h^3\,[A^{-1}\,B]\,[A^{-1}\,B]$$

$$[R] = (\tfrac{1}{2} - \beta)\,h^2\,[I] + (\tfrac{1}{4} - \beta)\,h^3\,[A^{-1}\,B]$$

$$[S] = [I] + \frac{h}{2}\,[A^{-1}\,B] + \beta h^2\,[A^{-1}\,C].$$

[I] is a diagonal of ones (unity matrix)

h is the integration step size (Δt).

After obtaining $\{X\}_1$, the following difference equation is used to obtain the rest of the values of $\{X\}$.

$$\{X\}_{i+1} = [S^{-1}\ T]\ \{X\}_i - [S^{-1}\ U]\ \{X\}_{i-1}$$
$$+ \beta h^2\ [S^{-1}\ A^{-1}\ D] \left(\{F_{i+1}\} + \left(\frac{1}{\beta} - 2\right)\ \{F\}_i + \{F_{i-1}\} \right)$$

where

$$[T] = 2\ [I] - (1 - 2\beta)\ h^2\ [A^{-1}\ C].$$

$$[U] = [I] - \frac{h}{2}\ [A^{-1}\ B] + \beta h^2\ [A^{-1}\ C].$$

The acceleration and velocity are then obtained using the following difference equations.

$$\{\ddot{X}\}_{i+1} = \left(\{X_{i+1}\} - \{X\}_i - h\ \{\dot{X}\}_i - (\tfrac{1}{2} - \beta)\ h^2\ \{\ddot{X}\}_i \right) \Big/ \beta h^2$$

$$\{\dot{X}\}_{i+1} = \{\dot{X}_i\} + \frac{h}{2}\ \{\ddot{X}\}_i + \frac{h}{2}\ \{\ddot{X}\}_{i+1}$$

Beta (β) is the parameter of generalized acceleration and can be anywhere between 0 and $\frac{1}{4}$.

REFERENCES

1. Chan, S. P., Cox, H. L, and Benfield, W. A.: "Transient Analysis of Forced Vibrations of Complex Structural-Mechanical Systems." *Journal of the Royal Aeronautical Society*, July 1962.

Subroutine TRSP2A solves the second order differential equation

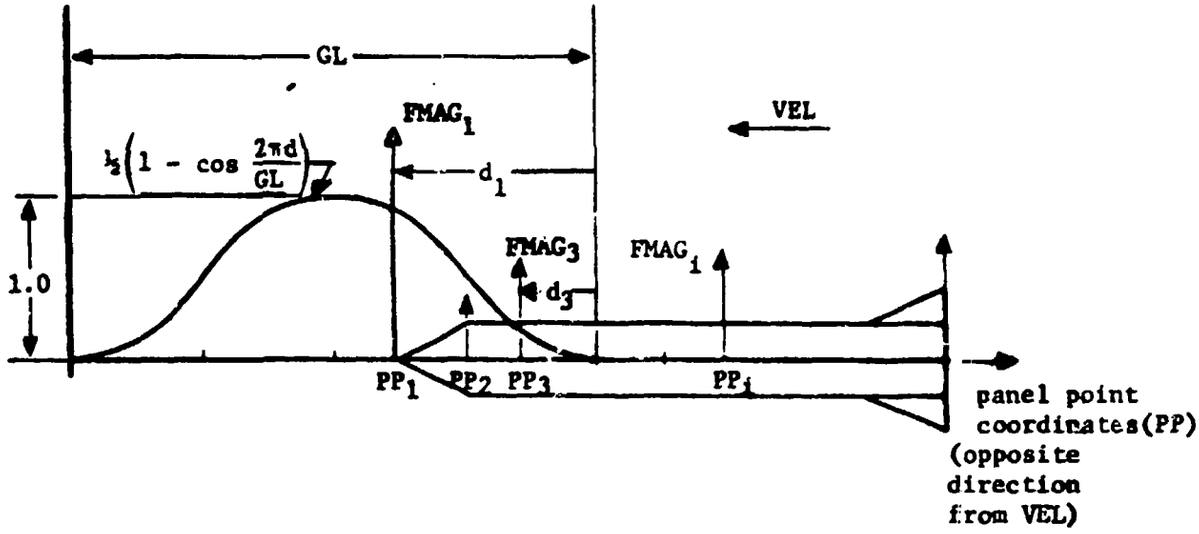
$$[A]\{\ddot{X}(t)\} + [B]\{\dot{X}(t)\} + [C]\{X(t)\} = [D]\{F(t)\} \qquad (1)$$

for $\{\ddot{X}(t)\}$, $\{\dot{X}(t)\}$, and $\{X(t)\}$ using a third order Newmark-Chan-Beta numerical integration procedure (Ref 1). Matrices [A], [B], [C], and [D] are input directly to this subroutine. For a structural problem, [A] is the mass matrix, [B] is the damping matrix, [C] is the stiffness matrix, and [D] is the transpose of the vibration mode shapes if the equations are a modal representation of the structure. The force $\{F(t)\}$ is calculated internally in the subroutine as follows.

$\{F(t)\}$ is size (NF×1) and is obtained considering a (1 - cos)/2 variation in the coordinate force. Consider the following examples.

EXAMPLE 1:

Missile Penetrating a Stationary Gust at the Rate - VEL -



Note that the ratio GL/VEL is the period of the gust.

If $FMAG_i$ is defined to be the maximum normal force at station (i), then the i^{th} normal force F(t) varies with time as -

$$F_i(t) = FMAG_i \ (\tfrac{1}{2}) \left(1 - \cos \frac{2\pi \ d_i}{GL}\right) \text{ for } 0 < d_i < GL$$

$$F_i(t) = 0 \qquad \text{for} \begin{cases} d_i \leq 0, \\ GL \leq d_i \end{cases}$$

where, $d_i = d_1 - \left(PP_i - PP_1 \right)$ and

$d_1 = VEL \times (t - STARTT)$.

Assumes PP_1 is at beginning of gust at t = STARTT.

EXAMPLE 2:

In the event a sudden envelopment forcing function (zero lag time for all stations) is desired, two alternatives exist:

1) Supply FMAG and PP as scalar quantities and $PP_1 = PP$. This sets the lag time to zero, $\left(PP_i - PP_1 = 0 \right)$; or

2) Supply FMAG as a vector and PP as a vector with all stations the same and equal to PP_1. This sets the lag time at all stations to zero.

DESCRIPTION OF TECHNIQUE

Because the difference equation contains displacement terms for three consecutive time intervals, it is required to express a displacement quantity in terms of initial velocity and initial displacement in order to start the procedure. Because [A] must be nonsingular in this method, it was decided to modify Equation (1) into the form

$$\{\ddot{X}(t)\} + [A]^{-1}\,[B]\,\{\dot{X}(t)\} + [A]^{-1}\,[C]\,\{X(t)\} = [A]^{-1}\,[D]\,\{F(t)\}.$$

Thus, only three matrices are needed compared to four matrices in Equation (1). The starting equation is thus given as:

$$\{X\}_1 = [S^{-1}\,P]\,\{X_0\} + [S^{-1}\,Q]\,\{\dot{X}_0\} + \beta h^2\,[S^{-1}\,A^{-1}\,D]\,\{F\}_1 + [S^{-1}\,R\,A^{-1}\,D]\,\{F\}_0$$

where

$$[P] = [I] + \frac{h}{2}\,[A^{-1}\,B] = (\tfrac{1}{2} - \beta)\,h^2\,[A^{-1}\,C] - (\tfrac{1}{4} - \beta)\,h^3\,[A^{-1}\,B]\,[A^{-1}\,C]$$

$$[Q] = h\,[I] - (\tfrac{1}{4} - \beta)\,h^3\,[A^{-1}\,B]\,[A^{-1}\,B]$$

$$[R] = (\tfrac{1}{2} - \beta)\,h^2\,[I] + (\tfrac{1}{4} - \beta)\,h^3\,[A^{-1}\,B]$$

$$[S] = [I] + \frac{h}{2}\,[A^{-1}\,B] + \beta h^2\,[A^{-1}\,C]$$

[I] is a diagonal of ones (unity matrix)

h is the integration step size (Δt)

After obtaining $\{X\}_1$, the following difference equation is used to obtain the rest of the values of {X}.

$$\{X\}_{i+1} = [S^{-1}\,T]\,\{X\}_i - [S^{-1}\,U]\,\{X\}_{i-1} + \beta h^2\,[S^{-1}\,A^{-1}\,D]\left(\{F_{i+1}\} + \left(\frac{1}{\beta} - 2\right)\{F\}_i + \{F_{i-1}\}\right)$$

where

$$[T] = 2\,[I] - (1 - 2\beta)\,h^2\,[A^{-1}\,C].$$

$$[U] = [I] - \frac{h}{2}\,[A^{-1}\,B] + \beta h^2\,[A^{-1}\,C].$$

The acceleration and velocity are then obtained using the following difference equations.

$$\{\ddot{X}\}_{i+1} = \left(\{X_{i+1}\} - \{X\}_i - h\,\{\dot{X}\}_i - (\tfrac{1}{2} - \beta)\,h^2\,\{\ddot{X}\}_i\right) \Big/ \beta h^2$$

$$\{\dot{X}\}_{i+1} = \{\dot{X}_i\} + \frac{h}{2}\,\{\ddot{X}\}_i + \frac{h}{2}\,\{\ddot{X}\}_{i+1}$$

Beta (β) is the parameter of generalized acceleration and can be anywhere between 0 and $\frac{1}{4}$.

REFERENCES

1. Chan, S. P., Cox, H. L, and Benfield, W. A.: "Transient Analysis of Forced Vibrations of Complex Structural-Mechanical Systems." *Journal of the Royal Aeronautical Society*, July 1962.

Subroutine TR3 solves the second order differential equation

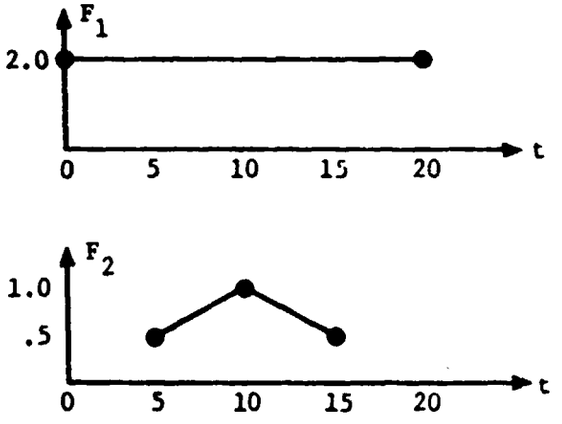
$$[\ A\ ]\ \{\ddot{X}(t)\} + [\ B\ ]\ \{\dot{X}(t)\} + [\ C\ ]\ \{X(t)\} = [D]\ \{F(t)\} \qquad (1)$$

for $\{\ddot{X}(t)\}$, $\{\dot{X}(t)\}$, and $\{X(t)\}$ in closed form. The results are used to solve the matrix equation

$$\{Z(t)\} = [AA]\ \{\ddot{X}(t)\} + [BB]\ \{\dot{X}(t)\} + [CC]\ \{X(t)\} + [DD]\ \{F(t)\} + \{EE\} \qquad (2)$$

that may be used to determine dynamic loads or displacements. The coefficient matrices [AA], etc are supplied or omitted on option.

For a structural problem, [A] is the generalized mass, [B] is the generalized damping, [C] is the generalized stiffness, and [D] is the transpose of the vibration mode shapes. Matrices [A], [B], and [C] are of uncoupled form. The force {F(t)} is calculated internally in the subroutine using linear interpolation with [TABT] and [TABF] which are both input to the subroutine. As an illustration of the use of [TABT] and [TABF] consider the following example.



The table (A table is defined in this report as a matrix which may have incomplete column data in some rows.) giving the independent variable coordinate t is

$$[TABT] = \begin{bmatrix} 0. & 20. & \\ 5. & 10. & 15. \end{bmatrix}.$$

The table giving the corresponding coordinates of the dependent variable F is

$$[TABF] = \begin{bmatrix} 2. & 2. & \\ .5 & 1. & .5 \end{bmatrix}.$$

The following values for {F(t)} will be obtained by linear interpolation at t = 7.

$$\{F(t = 7)\} = \begin{bmatrix} 2. \\ .7 \end{bmatrix}.$$

The matrixes [AA], [BB], [CC], [DD], and {EE} are input, on option, to calculate {Z(t)} which may be shear, bending moment, displacement, etc, depending on the choice of coefficient matrices.

DESCRIPTION OF TECHNIQUE

Equation (1) may be considered as a set of single degree of freedom equations because the mass, damping, and stiffness matrices are of uncoupled form. Thus the equation to consider is

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t) \tag{3}$$

where

f(t) is a row of [D] {F(t)}.

Equation (3) is solved by using Laplace transformations.

$$\text{In general, } \mathcal{L}[g(t)] = G(s) = \int_0^\infty e^{-st}\, g(t)\, dt\,. \tag{4}$$

$$\mathcal{L}[x(t)] = X(s) \tag{4a}$$

$$\mathcal{L}[\dot{x}(t)] = sX(s) - x(0) \tag{4b}$$

$$\mathcal{L}[\ddot{x}(t)] = s^2X(s) - sx(0) - \dot{x}(0) \tag{4c}$$

$$\mathcal{L}[f(t)] = F(s). \tag{4d}$$

Using the above Laplace transforms with Equation (3) gives

$$X(s) = \frac{F(s)}{m\,[(s + a)^2 + b^2]} + \frac{(s + 2a)\,x(0)}{(s + a)^2 + b^2} + \frac{\dot{x}(0)}{(s + a)^2 + b^2} \tag{5}$$

where

$$a = \frac{c}{2m} \tag{5a}$$

$$b = \sqrt{\frac{k}{m} - \frac{c^2}{4m^2}}\ . \tag{5b}$$

The time response x(t) is then found by taking the inverse Laplace transformation of Equation (5). This can be done conveniently by solving each term of the equation separately. Because the force may be nonzero at the start time, and because the force is assumed to vary linearly, the forcing function is broken down into a step force and a ramp force. Thus, the response is obtained as the sum of four separate solutions. That is,

x(t) = x(t) due to initial displacement

+ x(t) due to initial velocity

+ x(t) due to step force

+ x(t) due to ramp force.

In the solution of Equation (5), we will also define $\omega = \sqrt{\frac{k}{m}}$. Terms such as sin (bt + ψ) will use a trigonometic expansion to avoid inaccuracy of combining ψ with large values of bt. For example,

if

$$\psi = \tan^{-1}\frac{b}{a}$$

then

$$\sin(bt + \psi) = \frac{1}{\omega}(a \sin bt + b \cos bt).$$

The response due to initial displacement, initial velocity, a step force, and a ramp force are summarized below. These results are derived in Reference 1.

1. Initial Displacement, x(0)

$$x(t) = \frac{x(0)}{b} e^{-at} (a \sin bt + b \cos bt)$$

$$\dot{x}(t) = \frac{-x(0)}{b} \omega^2 e^{-at} \sin bt$$

$$\ddot{x}(t) = \frac{x(0)}{b} \omega^2 e^{-at} (a \sin bt - b \cos bt)$$

For the special case $c = k = 0$ (i.e., $a = b = 0$), the above equations give no solution. For this case,

$$x(t) = x(0)$$

$$\dot{x}(t) = 0$$

$$\ddot{x}(t) = 0$$

2. Initial Velocity, $\dot{x}(0)$

$$x(t) = \frac{\dot{x}(0)}{b} e^{-at} \sin bt$$

$$\dot{x}(t) = \frac{\dot{x}(0)}{b} e^{-at} (-a \sin bt + b \cos bt)$$

$$\ddot{x}(t) = \frac{\dot{x}(0)}{b} e^{-at} [(a^2 - b^2) \sin bt - 2ab \cos bt]$$

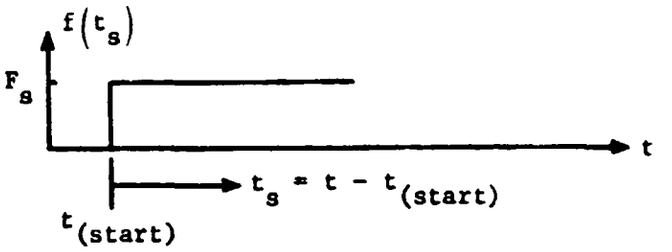
For the special case $c = k = 0$, (i.e., $a = b = 0$), the above equations give no solution. For this case,

$$x(t) = \dot{x}(0)\, t$$

$$\dot{x}(t) = \dot{x}(0)$$

$$\ddot{x}(t) = 0$$

3. Step Force



$$x(t_s) = \frac{F_s}{m\omega^2}\left[1 - \frac{e^{-at_s}}{b}\left(a \sin bt_s + b \cos bt_s\right)\right]$$

$$\dot{x}(t_s) = \frac{F_s}{mb}\, e^{-at_s} \sin bt_s$$

$$\ddot{x}(t_s) = \frac{F_s}{mb}\, e^{-at_s}\left[-a \sin bt_s + b \cos bt_s\right]$$

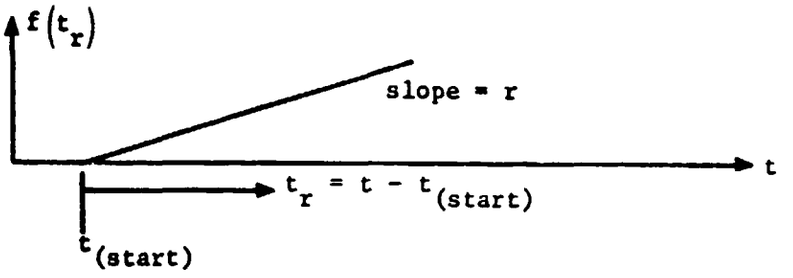
For the special case c = k = 0 (i.e., a = b = 0), the above equations give no solution. For this case,

$$x(t_s) = \tfrac{1}{2}\frac{F_s}{m}\, t_s^2$$

$$\dot{x}(t_s) = \frac{F_s}{m}\, t_s$$

$$\ddot{x}(t_s) = \frac{F_s}{m}$$

4. Ramp Force



$$x(t_r) = \frac{r}{m\omega^2}\left[t_r + \frac{1}{\omega^2}\left(-2a + \frac{e^{-at_r}}{b}\left\{(a^2 - b^2)\sin bt_r + 2ab\cos bt_r\right\}\right)\right]$$

$$\dot{x}(t_r) = \frac{r}{m\omega^2}\left[1 - \frac{e^{-at_r}}{b}\left(a\sin bt_r + b\cos bt_r\right)\right]$$

$$\ddot{x}(t_r) = \frac{r}{m}\frac{e^{-at_r}}{b}\sin bt_r$$

For the special case c = k = 0 (i.e., a = b = 0), the above equations give no solution. For this case,

$$x(t_r) = \frac{r}{6m}t_r^3$$

$$\dot{x}(t_r) = \frac{r}{2m}t_r^2$$

$$\ddot{x}(t_r) = \frac{r}{m}t_r$$

REFERENCES

1. Wohlen, R. L., and Benfield, W. A.: *Closed Form Solution of the Second Order Differential Equation*, Martin Marietta Corporation, Denver Division, Dynamics Section, Memo 108, January 1967.

Subroutine UMAM1 calculates a matrix that relates inertia plus applied loads to applied loads for an inertially restrained (free-free) system.

A physical interpretation of this transformation matrix may be obtained from the following beam with a single degree of freedom (translation) at each panel point.

The inertia forces are defined by

$$\{F_I\} = - [A] \{\ddot{z}\} \tag{1}$$

where [A] is the mass matrix of the beam and $\{\ddot{z}\}$ is the panel point acceleration.

The panel point accelerations are given by

$$\{\ddot{z}\} = \left[\{1\} \{x_R - x\}\right] \begin{bmatrix} \ddot{z}_R \\ \ddot{\theta} \end{bmatrix} \tag{2}$$

where x_R denotes a reference station that may have any value and may or may not coincide with a panel point station.

Let us define

$$[RBM] = \left[\{1\} \{x_R - x\}\right], \tag{3}$$

that is, the rigid body modes for the beam. The rigid body translation mode is given by {1} and the rigid body rotation mode is given by $\{x_R - x\}$. Substituting Equations (2) and (3) into (1) gives

$$\{F_I\} = - [A] [RBM] \begin{bmatrix} \ddot{z}_R \\ \ddot{\theta} \end{bmatrix}. \tag{4}$$

For loads equilibrium, the sum of the forces and the moments about a reference station must be zero. That is

$$\begin{bmatrix} \{1\}^T \\ \{x_R - x\}^T \end{bmatrix} \left[\{F_A\} + \{F_I\} \right] = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5}$$

where $\{F_A\}$ is the applied external force on the beam.

From Equation (5), noting that

$$\begin{bmatrix} \{1\}^T \\ \{x_R - x\}^T \end{bmatrix} = [RBM]^T,$$

$$- [RBM]^T \{F_I\} = [RBM]^T \{F_A\}. \tag{6}$$

Substituting Equation (4) into (6)

$$[RBM]^T [A] [RBM] \begin{bmatrix} \ddot{z}_R \\ \ddot{\theta} \end{bmatrix} = [RBM]^T \{F_A\}$$

from which

$$\begin{bmatrix} \ddot{z}_R \\ \ddot{\theta} \end{bmatrix} = \left[[RBM]^T [A] [RBM] \right]^{-1} [RBM]^T \{F_A\}. \tag{7}$$

Substituting Equation (7) into (4) gives the inertia loads in terms of the applied loads as

$$\{F_I\} = - [A] [MESS] \{F_A\} \tag{8}$$

where we have defined

$$[MESS] = [RBM] \left[[RBM]^T [A] [RBM] \right]^{-1} [RBM]^T. \tag{9}$$

Now the applied plus inertia loads can be expressed as

$$\{F_A\} + \{F_I\} = \{F_A\} - [A]\,[MESS]\,\{F_A\}$$

$$= [I] - [A]\,[MESS]\,\{F_A\}. \qquad (10)$$

$\left[[I] - [A]\,[MESS]\right]$ is the matrix relating the applied plus inertia loads to the applied loads output from the subroutine. It can be shown that $\left[[I] - [A]\,[MESS]\right]$ is independent of the reference station x_R.

Any number of rigid body modes may be used; however, this subroutine is limited to six.

Several important features of $\left[[I] - [A]\,[MESS]\right]$ exist. The first deals with the triple matrix product

$$\left[[I] - [A]\,[MESS]\right]^T \left[E_R\right] \left[[I] - [A]\,[MESS]\right]$$

where $\left[E_F\right]$ is the structural influence coefficient matrix of a structure restrained in a statically determinant fashion. The result of the operation is a free-free structural influence coefficient matrix where the grounding restraints of the structure have been removed. It is somewhat helpful to think that the grounding restraints of the structure have been replaced by inertial restraints. A second important feature is that the product

$$\left[E_F\right] \left[K_F\right] = \left[[I] - [A]\,[MESS]\right]$$

where $\left[E_F\right]$ is the free-free structural influence coefficient and $\left[K_F\right]$ is the free-free structural stiffness matrix.

An interesting property of $\left[[I] - [A]\,[MESS]\right]$ is that it is an *idempotent* matrix, that is, a matrix whose produce with itself is equal to itself. This can be easily shown as follows.

$$\Big[[I] - [A]\,[MESS]\Big]\ \Big[[I] - [A]\,[MESS]\Big]$$

$$= [I] - [A]\,[MESS] - [A]\,[MESS] + [A]\,[MESS]\,[A]\,[MESS].$$

Now

$$[A]\,[MESS]\,[A]\,[MESS]$$

$$= [A]\,[RBM]\Big[[RBM]^T\,[A]\,[RBM]\Big]^{-1}\,[RBM]^T\,[A]\,[RBM]$$

$$\Big[[RBM]^T\,[A]\,[RBM]\Big]^{-1}\,[RBM]^T$$

$$= [A]\,[RBM]\Big[[RBM]^T\,[A]\,[RBM]\Big]^{-1}\,[RBM]^T$$

$$= [A]\,[MESS].$$

$$\therefore \Big[[I] - [A]\,[MESS]\Big]\Big[[I] - [A]\,[MESS]\Big] = [I] - [A]\,[MESS].$$

This technique was formulated by David Lang at Chance Vought Aircraft in an informal memorandum in 1959 and expanded upon by myself and Carl Bodley in 1960.

Subroutine UNITY generates a square matrix with diagonal element equal to one and all off-diagonal elements equal to zero. That is,

$$z_{ij} = 1. \qquad (i = j)$$

$$z_{ij} = 0. \qquad (i \neq j)$$

In matrix notation,

$$[Z]_{NxN} = \begin{bmatrix} 1. & & 0. \\ & \ddots & \\ 0. & & 1. \end{bmatrix}$$

where N is the size of [Z] (square). A synonym for the unity matrix is the identity matrix, thus the usual designation as [I].

A matrix is unaltered when multiplied by the unity matrix and the process is commutative. In matrix notation,

$$[A][I] = [I][A] = [A].$$

Subroutine "PDATE transfers data written by Subroutines WTAPE and YWTAPE from on. or more FORMA tapes onto another (new or existing) FORMA tape*. It also provides an option to unload and return the read-tapes and drives to the system upon completion of updating from them.

The selection of matrices to be updated from one tape to the other, is accomplished by specifying the matrix location number (see Subroutine WTAPE writeup) limits. These limits can include only one matrix, several (consecutive) matrices, or all the matrices (i.e., the whole tape).

*Because Subroutine UPDATE both reads and writes on the "write" tape, and because most computer tape drives do not have sufficient tolerance control for a mixed mode (i.e., read/write) operation, the "write" tape should initially be a disk file. After the update, the disk file may be copied onto a tape.

Subroutine VCROSS calculates the cross product of two vectors in three-dimensional space. By definition, the cross product of two vectors $\vec{A}$ and $\vec{B}$ is a vector $\vec{Z}$ whose magnitude is the product of the magnitudes of $\vec{A}$ and $\vec{B}$ with the sine of the angle between their positive directions; that is,

$$\vec{A} \times \vec{B} = \vec{Z}$$

$$|\vec{Z}| = |\vec{A}|\ |\vec{B}| \sin(\vec{A}, \vec{B}).$$

The direction of $\vec{Z}$ is perpendicular to the plane determined by $\vec{A}$ and $\vec{B}$ and so sensed that the configuration of $\vec{A}$, $\vec{B}$, $\vec{Z}$ constitutes a right-hand system (providing the reference axes themselves are a right-hand system).

In terms of the components of $\vec{A}$ and $\vec{B}$ we have,

$$z_1 = a_2 b_3 - a_3 b_2$$

$$z_2 = a_3 b_1 - a_1 b_3$$

$$z_3 = a_1 b_2 - a_2 b_1$$

which is the expanded form of the determinant

$$\begin{vmatrix} z_1 & z_2 & z_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}.$$

If $\vec{A} \times \vec{B} = 0$, then either at least one of the vectors $(\vec{A}, \vec{B})$ is zero or $\vec{A}$ and $\vec{B}$ are parallel.

The cross product can be represented in matrix notation as

$$\vec{A} \times \vec{B} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

Note that the first matrix is skew-symmetric.

Theorem: Cross multiplication is distributive. That is,

$$\vec{A} \times (\vec{B} + \vec{C}) = \vec{A} \times \vec{B} + \vec{A} \times \vec{C}.$$

Theorem: Cross multiplication is *not* commutative and is sometimes called anti-commutative since

$$\vec{A} \times \vec{B} = -\vec{B} \times \vec{A}.$$

EXAMPLE

If the input is $\vec{A}$ = [-3., 5., 2.] and $\vec{B}$ = [4., 1., -6.], then the output will be

$$\vec{Z} = \vec{A} \times \vec{B}$$

$$= \begin{bmatrix} 0. & -2. & 5. \\ 2. & 0. & 3. \\ -5. & -3. & 0. \end{bmatrix} \begin{bmatrix} 4. \\ 1. \\ -6. \end{bmatrix}$$

$$= \begin{bmatrix} -32. \\ -10. \\ -23. \end{bmatrix}$$

$$|\vec{A}| = \sqrt{(-3.)^2 + 5.^2 + 2.^2} = 6.1644$$

$$|\vec{B}| = \sqrt{4.^2 + 1.^2 + (-6.)^2} = 7.2801$$

$$|\vec{Z}| = \sqrt{(-32.)^2 + (-10.)^2 + (-23.)^2} = 40.6571$$

$$\sin(\vec{A}, \vec{B}) = \frac{|\vec{Z}|}{|\vec{A}||\vec{B}|} = .90596$$

Subroutine VDOT calculates the dot product of two vectors in three-dimensional space. The dot product is also called the scalar product or inert product. By definition, the dot product of two vectors $\vec{A}$ and $\vec{B}$ is a scalar equal to the product of the magnitude of each vector and the cosine of the angle between their positive directions; that is,

$$\vec{A} \cdot \vec{B} = |\vec{A}| \ |\vec{B}| \cos (\vec{A}, \vec{B}).$$

In terms of the components of $\vec{A}$ and $\vec{B}$, we have

$$\vec{A} \cdot \vec{B} = a_1 b_1 + a_2 b_2 + a_3 b_3.$$

If $\vec{A} \cdot \vec{B} = 0$, then either at least one of the vectors $(\vec{A}, \vec{B})$ is zero or $\vec{A}$ and $\vec{B}$ are perpendicular.

The dot product can be represented in matrix notation as

$$\vec{A} \cdot \vec{B} = \{A\}^T \{B\}$$

$$= \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

Theorem: Dot multiplication is distributive. That is,

$$\vec{A} \cdot (\vec{B} + \vec{C}) = \vec{A} \cdot \vec{B} + \vec{A} \cdot \vec{C}$$

Theorem: Dot multiplication is commutative. That is,

$$\vec{A} \cdot \vec{B} = \vec{B} \cdot \vec{A}$$

EXAMPLE

If the input is $\vec{A}$ = [-3., 5., 2.] and $\vec{B}$ = [4., 1., -6.], then the output will be

$$\vec{A} \cdot \vec{B} = \begin{bmatrix} -3. & 5. & 2. \end{bmatrix} \begin{bmatrix} 4. \\ 1. \\ -6. \end{bmatrix}$$

$$= -19.$$

$$|\vec{A}| = \sqrt{(-3.)^2 + 5.^2 + 2.^2} = 6.1644$$

$$|\vec{B}| = \sqrt{4.^2 + 1.^2 + (-6.)^2} = 7.2801$$

$$\cos(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|} = -.42337$$

Subroutine VM1 integrates pressure or weight distribution along with concentrated forces or weights to obtain shears and bending moments at a set of selected points on a beam. The pressure or weight distribution and the concentrated forces or weights are multiplied by an amplification function (normally the acceleration).

The x-stations of the selected points are given in {XVEC}. These x-stations must be in increasing order.

The distributed pressure, p(x), is assumed to be piecewise linear and is represented by straight line segments as shown in Figure 1.

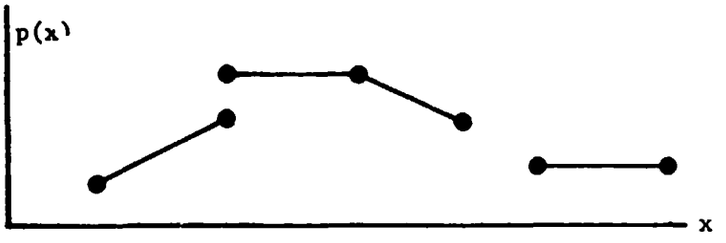


Figure 1 Distributed Pressure

The x-stations of the end points for the line segments giving the distributed pressure are independent of the x-stations in {XVEC}. The line segments representing the distributed pressure may or may not be joined but must not overlap. On any interval where the distributed pressure is not defined, the pressure is assumed to be zero. The distributed pressure is defined in [DIS]. Each row of [DIS] represents one nonvertical line segment. The form of each row of [DIS] is $[x_1\ x_2\ p_1\ p_2]$ where x_1, p_1 give the first end point and x_2, p_2 give the second end point of a line segment. The line segments given by [DIS] must be in increasing order of x. The distributed weight may also be given by [DIS].

The amplification function is also assumed to be represented by straight line segments. All statements given above for distributed pressure are applicable. The distributed amplification function is defined in [AMP].

The concentrated mass items are defined in [CONC]. Each row of [CONC] represents one concentrated mass item and contains:

1) x_a, the station at which the item is attached to the beam;
2) M_c, the mass of the item;
3) x_{cg}, the center of gravity of the item; and
4) I_c^{cg}, the moment of inertia of the item about its own center of gravity.

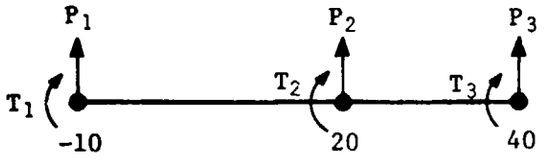
These four elements are given in the form $\left[x_a \; M_c \; x_{cg} \; I_c^{cg}\right]$. The attachment station, x_a, of an item to the beam must be within the panel point limits.

The calculated shears and bending moments at the selected points are placed in {ZV} and {ZM}, respectively.

Subroutine VMTR1 calculates a matrix [Z] that can be used to obtain internal shears and bending moments of a beam in terms of external forces and moments acting on the beam. The internal shears and bending moments, as well as the external forces and moments, are at specific locations on the beam. These locations are specified by the elements of {PP} which is input to the subroutine.

EXAMPLE

Consider the beam below with external forces P_i and moments T_i acting at the specified locations.



The locations of the external load application points is given by

$$\{PP\} = \begin{bmatrix} -10. \\ 20. \\ 40. \end{bmatrix}.$$

The internal shears (V) and bending moments (M) at the locations specified by {PP} can be obtained in terms of the external loads by the following equations:

$$V_j = \sum_{i=1}^{j} P_i . \tag{1}$$

$$M_j = \sum_{i=1}^{j} \left(x_j - x_i\right) P_i + \sum_{i=1}^{j} T_i . \tag{2}$$

These equations can be expressed in matrix form for the example beam as

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \hline M_1 \\ M_2 \\ M_3 \end{bmatrix} = \underbrace{\left[\begin{array}{ccc:ccc} 1. & & & & & \\ 1. & 1. & & & & \\ 1. & 1. & 1. & & & \\ \hdashline 0. & & & 1. & & \\ 30. & 0. & & 1. & 1. & \\ 50. & 20. & 0. & 1. & 1. & 1. \end{array}\right]}_{[Z]} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \hline T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

where [Z] is the matrix output from this subroutine.

Subroutine WRITE writes a matrix of real numbers (a Fortran term for numbers with a decimal point) on paper. A group of up to ten consecutive elements from a row of the matrix are printed on each line. If all of the elements of a group are zero, printing of this line is suppressed.

Each matrix printed begins on a new page. On each page of printout is the page heading given by Subroutine PAGEHD, the name of the matrix, and the row size and column size of the matrix. This is followed by the matrix data. On any line of matrix data the first integer number is the row number of the matrix elements on that line. The second integer number is the column number of the matrix element in the first data field. The next group of real numbers (up to ten) are the values of the matrix elements. This group of matrix elements is given in consecutive column order.

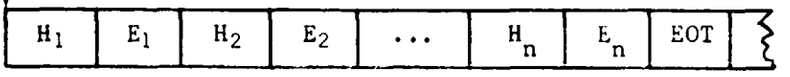
Subroutine WRITIM writes a matrix of integer numbers on paper. A group of up to twenty consecutive elements from a row of the matrix are printed on each line. If all of the elements of a group are zero, printing of this line is suppressed.

Each matrix printed begins on a new page. On each page of printout is the page heading given by Subroutine PAGEHD, the name of the matrix, and the row size and column size of the matrix. This is followed by the matrix data. On any line of matrix data the first integer number is the row number of the matrix elements on that line. The second integer number is the column number of the matrix element in the first data field. The next group of integer numbers (up to twenty) are the values of the matrix elements. This group of matrix elements is given in consecutive column order.

Subroutine WTAPE writes matrix data at the end of existing written matrix data on a FORMA tape (disk is preferred, see below). Each set of matrix data consists of two logical records. The first record contains the matrix heading (tape identification, location number, run number, matrix name, number of rows of matrix, number of columns of matrix, date, and the word "dense"). The second record consists of the matrix elements.

A schematic representation of the tape (disk) is given by the following sketch.

Beginning
of
tape (disk)



where

H_i = Matrix heading of the i^{th} written matrix,

E_i = Matrix elements of the i^{th} written matrix,

EOT = End of Tape. Data written by Subroutine WTAPE or INTAPE that all FORMA tape subroutines recognize as being the end of written data.

Each vertical line is an end of logical record put on by computer system's routines. The tape is written in binary form as opposed to binary coded decimal (BCD) form.

To find the end of written matrix data, a search is made of the matrix headings until the EOT is found. For this reason, a "new" tape (disk) must be initialized with Subroutine INTAPE so that the tape (disk) contains an EOT. A "new" tape (disk) is defined to be a tape (disk) for which it is desired to start writing matrix data at the front of the tape (disk). Thus, a "new" tape (disk) could be one with obsolete FORMA matrix data on it as well as one that has never been written on by the FORMA system. When the EOT is found, a backspace operation is done over the EOT, and then the current matrix heading, current matrix elements, and a new EOT is written.

A disk is preferred to a tape for the following reason. Because of the physical separation of the read and write heads on most tape drives there may be tape tolerance problems thus backspacing over the EOT is usually not successful. Instead of ending up positioned in front of the EOT, the write head is often positioned in front of the previous matrix elements (E_n in the above sketch). The current matrix heading will be written over the previous matrix elements. This causes problems later when trying to read the records written on the tape. To alleviate this problem, it is strongly recommended that all FORMA tape subroutines (INITAPE, LTAPE, RTAPE, WTAPE, and UPDATE) use an intermediate device such as a disk. At the start of a computer run, the existing tape should be copied onto the disk by using computer control cards. Likewise, at the end of the run, the disk should be copied back onto tape by using computer control cards.

Subroutine XLORD contains a sorting procedure used to arrange matrix element locations into ascending order. The associated matrix elements are arranged correspondingly. This subroutine operates only on matrix elements stored in core Subroutine XLORD scans the elements to be sorted and selects one of two possible methods to be used in sorting. The first method (employed primarily when the elements are in a random order) is similar to SORTAG by R. C. Singleton*. The second method (employed when the elements are broken down to two previously ordered groups) is a merging procedure.

---

*R. C. Singleton: *An Efficient Algorithm for Sorting with Minimal Storage.* Research Memorandum, SRI Project 38753'-132, September 1968.

Subroutine ZERO generates a matrix with each element equal to zero. That is,

$$z_{ij} = 0. \qquad \begin{pmatrix} i = 1, NR \\ j = 1, NC \end{pmatrix}$$

In matrix notation,

$$[Z]_{NR \times NC} = \begin{bmatrix} 0. & 0. & \cdots & 0. \\ 0. & 0. & & \vdots \\ \vdots & & & \\ 0. & . & . & 0. \end{bmatrix}$$

where NR is the number of rows of [Z], and NC is the number of columns of [Z].

This subroutine is useful in setting a matrix array to zero before performing subsequent operations such as matrix assembly (ASSEM) or revision/addition of one matrix into another (REVADD).

Subroutine ZEROLH sets the elements below the diagonal of a square matrix equal to zero. That is,

$$a_{ij} = 0. \qquad (i > j)$$

EXAMPLE

If [A] is input to Subroutine ZEROLH as

$$[A]_{3x3} = \begin{bmatrix} 1. & 2. & 3. \\ 4. & 5. & 6. \\ 7. & 8. & 9. \end{bmatrix},$$

the matrix output from this subroutine will be

$$[A]_{3x3} = \begin{bmatrix} 1. & 2. & 3. \\ 0. & 5. & 6. \\ 0. & 0. & 9. \end{bmatrix}.$$

Subroutine ZEROUH sets the elements above the diagonal of a square matrix equal to zero. That is,

$$a_{ij} = 0. \qquad (i > j)$$

EXAMPLE

If [A] is input to Subroutine ZEROUH as

$$[A]_{3x3} = \begin{bmatrix} 1. & 2. & 3. \\ 4. & 5. & 6. \\ 7. & 8. & 9. \end{bmatrix},$$

the matrix output from this subroutine will be

$$[A]_{3x3} = \begin{bmatrix} 1. & 0. & 0. \\ 4. & 5. & 0. \\ 7. & 8. & 9. \end{bmatrix}.$$

Subroutine ZZBOMB controls a computer run after an error message has been encountered in any of the FORMA subroutines. This entails

1) Printing of subroutine name and error number where error occurred;

2) Printing of all of the core memory used in octal form;

3) Program termination.