THE MASS MATRIX IN DYNAMIC STRUCTURAL ANALYSIS

by

Thomas J. Enneking

Thesis submitted to the Graduate Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Civil Engineering

APPROVED:

S. M. Holzer, Chairman

R. H. Plaut

A. E. Somers, Jr.

December, 1978

Blacksburg, Virginia

ACKNOWLEDGEMENTS

My appreciation is extended to Dr. Siegfried M. Holzer for his suggestions and support throughout this work and for his help in editing the manuscript. I also wish to thank Dr. R. H. Plaut, Dr. A. E. Somers and Dr. L. Meirovitch for their assistance and encouragement, not only in the writing of this report, but throughout my graduate study. Special thanks are for my wife, , who provided me with encouragement and inspiration during the preparation of this report.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Traditionally engineers have lumped the mass of a structure at a discrete number of points based solely on experience. Then with the introduction of consistent mass approximations for continuous systems, it appeared that a proper discretization procedure had evolved. Now, however, there seems to be a trend to return to the lumped or diagonal mass approach, as researchers find that the use of consistent mass matrices does not always lead to improved accuracy in frequency prediction and always involves additional computations (12, 19, 20, 23, 28, 35, 36, 41, 49, 51, 82, 92, 93, 101).

The purpose of this investigation is to present a literature study pertaining to mass matrices and their role in structural analysis and to conduct a comparison study on different types of mass matrices on the basis of frequency prediction. In order to perform the comparison study, a FORTRAN code was developed using beam-column elements to assemble the system mass matrix and calculate the eigenvalues and eigenvectors. This code was then added to the code developed in CE4002-Matrix Structural Analysis and CE5980-Computer Aided Structural Design.

The formulation of the mass matrix has not been as thoroughly investigated as the stiffness matrix. This is probably because it is required for only a limited class of problems. However, as structures become lighter and more sophisticated, vibration analysis, and consequently the mass matrix, becomes critical to a complete structural analysis.

In the same way that the stiffness matrix relates the nodal displacements to the strain energy of an element, the mass matrix relates the no-

dal velocities to the kinetic energy. Let ρ = the mass per unit volume of the element and N = the interpolation matrix containing functions used to discretize the velocity field of the continuum. The mass matrix, m, can be written as

$$m = \iiint_V \rho \, N^T \, N \, dV$$

where V = the volume of the element. The selection of these interpolation functions is arbitrary, but one obvious possibility is to use the same functions that were used to discretize the displacement field. If this is done, the kinetic energy of the elements will be consistent with the strain energy, and the resulting mass matrix is called a consistent mass matrix. Using a mass matrix of this type can at times lead to increased accuracy in frequency prediction; however, there is always an accompanying increase in computational effort (3, 5, 14, 16, 54, 88).

A lumped diagonal mass matrix is, as the name implies, a diagonal matrix containing the structural mass of a system lumped at a discrete number of points. This lumping is normally achieved by ascribing the mass of a certain portion of the structure to each of the discrete points approximating the continuum. There are other methods of forming diagonal mass matrices (24, 26, 35, 36, 49, 82, 86), which will be introduced in Chapter II and examined in greater detail in Chapter III. A diagonal system, formed by any method, leads to a relatively simple and efficient solution process. The computed frequencies and mode shapes, however, may differ from the exact, especially in the higher modes (6, 14, 41, 51, 54, 62, 67, 88).

CHAPTER II

LITERATURE STUDY ON MASS MATRICES AND RELATED STRUCTURAL CHARACTERISTICS

This chapter contains a literature study on mass matrices and related structural characteristics. Element models, solution procedures, and mass matrix types were investigated to determine possible correlations between them and accurate frequency predictions. Also included in the study were methods for reducing the size, and consequently the solution time, of complex eigenproblems. Therefore, the role of the mass matrix in the current state of the art of dynamic structural analysis can be determined.

The information derived from this study is presented in Table 1. The symbols used in Table 1 are defined, in some detail, in Tables 2 - 6. Following these tables are brief summaries of key point.

The study contains information from various books and conference proceedings; however, the major sources were articles found in journals dealing with numerical methods, computers, and structures. Most of these were published since 1963.

Table 1

Literature Study of Mass Matrices and Related Structural Characteristics

Reference Number	Mass Matrix Type			Finite Element Model			Continuum Model				Solution Process			Matrix Reduction Method
References are grouped according to mass type	Consistent	Lumped	Other	Beams	Plates	Shells	Beams	Plates	Frames	Other	Modal Analysis	Direct Integration	Other	
Consistent														
3	X			2			PCB						RL	RNU
4	X			1			CAB						ND	ASH
4	X			1			SSB						ND	ASH
4	X			1			FFB						ND	ASH
5	X			2			CAB						ND	RNU
6	X			1			SSB						ND	RNU
6	X			1			FFB						ND	RNU
7	X				8	14				NP			ND	RNU
12	X			1	8		CAB	CB					ND	RNU
14	X			3						NP			ND	RNU
16	X				9			CCP					ND	RNU
19	X				9	14		CCP		CC	HQRI			RNU
20	X					14				SC			ND	RNU
20	X				18					CC			ND	RNU
22	X			6			CAB						ND	RNU
27	X					NES				NP			ND	RNU
28	X					17				CC			ND	RNU

Table 1

Literature Study of Mass Matrices and Related Structural Characteristics

Reference Number	Mass Matrix Type			Finite Element Model			Continuum Model				Solution Process			Matrix Reduction Method
References are grouped according to mass type	Consistent	Lumped	Other	Beams	Plates	Shells	Beams	Plates	Frames	Other	Modal Analysis	Direct Integration	Other	
Consistent														
29	X					18				SSM			ND	RNU
31	X				9					NP			ND	RNU
32	X					NES				NP			ND	GUY
34	X				9			CAP					ND	HO
34	X					15		COL					ND	HO
38	X			1			CAB		PF		BAT	NB		GUY
38	X			3			SSB				BAT	NB		GUY
50	X				9			CCP					ND	RNU
51	X			1			SSB				HQRI	NB		RNU
51	X			1			FFB				JAC	WO		RNU
54	X			1					PF				ND	RNU
55	X			5			SSC						ND	RNU
55	X			5			CAC						ND	RNU
59	X					16		SSP					ND	RNU
64	X			1			CB						ND	RNU
65	X					11				SC			ND	RNU
70	X				9			CCP					ND	RNU

Table 1

Literature Study of Mass Matrices and Related Structural Characteristics

Reference Number	Mass Matrix Type			Finite Element Model			Continuum Model				Solution Process			Matrix Reduction Method
References are grouped according to mass type	Consistent	Lumped	Other	Beams	Plates	Shells	Beams	Plates	Frames	Other	Modal Analysis	Direct Integration	Other	
Consistent														
70	X				9			SSP					ND	RNU
71	X				9			SSP					ND	RNU
73	X				10			RD					ND	RNU
75	X				9			CAP			HQRI			RNU
75	X				9			SSP			HQRI			RNU
76	X			2			CAB						ND	RNU
79	X				9			CCP					ND	RNU
79	X				9			SSP					ND	RNU
80	X			1			CAB						ND	RNU
80	X			1			SSB						ND	RNU
82	X				9			SSP					ND	RNU
83	X				9			SSP				NB		RNU
84	X					17				CC			ND	RNU
88	X					17				CC			ND	GUY
89	X				8			CAP					ND	RNU
90	X					17				SCP	HQRI		RQ	RNU
93	X			1			CAB						ND	RNU

Table 1

Literature Study of Mass Matrices and Related Structural Characteristics

Reference Number							Continuum Model				Solution Process			Matrix Reduction Method
References are grouped according to mass type	Consistent	Lumped	Other	Beams	Plates	Shells	Beams	Plates	Frames	Other	Modal Analysis	Direct Integration	Other	
Consistent														
94	X			1			SSB						RQ	RNU
95	X			1					SF				ND	RNU
96	X			5			SST						ND	RNU
99	X			1					SF				ND	RNU
Lumped														
1		X		4		16	CAB			CC			ND	RNU
3		X		2			PCB						RL	RNU
6		X		1			SSB						ND	RNU
6		X		1			FFB						ND	RNU
12		X		1	8		CAB	CR				DU		RNU
13		X		3					SF			DU		RNU
17		X		1					PF				ND	RNU
18		X		2			CB		PF				ND	RNU
19		X			9	14		CCP		CC		DU		RNU
20		X				14				SC		NB		RNU
28		X				17				CC			ND	RNU
33		X		1		15	CAB			CC		DU		RNU

Table 1

Literature Study of Mass Matrices and Related Structural Characteristics

Reference Number							Continuum Model				Solution Process			Matrix Reduction Method
References are grouped according to mass type	Consistent	Lumped	Other	Beams	Plates	Shells	Beams	Plates	Frames	Other	Modal Analysis	Direct Integration	Other	
Lumped														
46		X		1			CAB						ND	RNU
52		X		1			FFB						ND	RNU
53		X		1					SF				ND	RNU
54		X		1					SF				ND	RNU
58		X		1			CAB		SF			NB		RNU
60		X		1		13	SSB			SC		NB		RNU
60		X		1		13	SSB			SC		WO		RNU
61		X				13				SC			ND	RNU
62		X		1			CAB						ND	RNU
67		X		1						SSG			ND	RNU
74		X		1					SF				ND	RNU
82		X			9			SSP					ND	RNU
85		X		1					SF				ND	RNU
92		X				12				HS		WO		RNU
93		X		1			CAB						ND	RNU
94		X		1			SSB						RQ	RNU

Table 1

Literature Study of Mass Matrices and Related Structural Characteristics

Reference Number	Mass Matrix Type			Finite Element Model			Continuum Model				Solution Process			Matrix Reduction Method
References are grouped according to mass type	Consistent	Lumped	Other	Beams	Plates	Shells	Beams	Plates	Frames	Other	Modal Analysis	Direct Integration	Other	
Other														
21			DM	5					SWB				ND	RNU
26			NI			7				FM			ND	RNU
35			HRZ		9			TP					ND	RNU
36			HRZ		10			CR				ECD		RNU
36			HRZ		10			CCP				ECD		RNU
36			HRZ		10			SSP				ECD		RNU
42			HCH		9					NP			ND	RNU
49			KB	1			SSB					DU		RNU
51			KB	1			SSB				HQRI	NB		RNU
51			KB	1			FFB				JAC	WO		RNU
56			ML	1			SSB						ND	RNU
56			ML	1			FFB						ND	RNU
82			HRZ		9			SSP					ND	RNU
86			SCH	1						NP			ND	RNU
97			VKC		9			SSP					ND	RNU

The symbol; X; denotes that the general topic is discussed in the article. All other symbols, denoting specific topics discussed in the article, are defined in Tables 2 - 6.

Table 2

Mass Matrix Type

Definition of Symbols

DM - Distributed mass approach examined by Coull and Mukherjee (21)

HCH - Lumping procedure identical to KB except α is selected to allow a larger time step without upsetting convergence (42)

HRZ - Lumping procedure which scales diagonal terms of the consistent mass matrix to preserve the total mass of the system developed and used by Hinton, Rock and Zienkiewicz (35)

KB - Lumping procedure developed by Key and Beisenger (49) based on the consistent mass matrix and a gradient inertia scaling factor, α, selected on the basis of the maximum frequency

ML - Complementary energy representation of mass is used in conjunction with the potential energy representation to yield a modified non-diagonal "consistent" mass matrix (56)

NI - Numerical integration technique to form a diagonal lumped mass matrix studied by Fried and Malkus (26)

SCH - Consistent diagonal mass matrix determined using orthogonal base functions and mixed variational formulation developed by Schreyer (86)

VKC - Truncated mass matrix with only one degree of freedom per node and no coupling determined from the consistent mass matrix (97)

Table 3

Finite Element Model

Definition of Symbols

1 - Two-Dimensional Beam

2 - Two-Dimensional Beam including shear deformation and rotary inertia effects

3 - Three-Dimensional Beam

4 - Sandwich Beam

5 - Thin-Walled Beam

6 - Curved Beam

7 - Triangular Membrane

8 - Triangular Plate

9 - Plate Bending

10 - Axisymmetric Plate

11 - Triangular Shell

12 - Flat Quadrilateral Shell

13 - Isoparametric Shell

14 - Axisymmetric Shell

15 - Thin Shell

16 - Sandwich Shell

17 - Curved Shell

18 - Finite Dynamic Element developed by Gupta (29)

NES - No Element Specified

Table 4

Continuum Model

Definition of Symbols

CAB - Cantilever Beam

CAC - Cantilever Channel Beam

CAP - Cantilever Plate

CB - Continuous Beam

CC - Circular Cylinder

CCB - Clamped-Clamped Beam

CCP - Clamped-Clamped Plate

COL - Column

CR - Clamped Ring

FFB - Free-Free Beam

FM - Fixed Membrane

HS - Hyperboloidal Shell

NP - No Example Problems

PCB - Pre-Twisted Cantilever Beam on a Rotating Disk

PF - Plane Frame

RD - Rotating Disk

SC - Spherical Cap

SCP - Simply-Supported Cylindrical Panel

SF - Space Frame

SSB - Simply-Supported Beam

SSC - Simply-Supported Channel Beam

SSG - Simply-Supported Grid

Table 4

Continuum Model

Definition of Symbols

SSM - Simply-Supported Square Membrane

SSP - Simply-Supported Plate

SST - Simply-Supported Thin-Walled Beam

SWB - Y-Shaped Shear Wall Building

TP - Thick Square Plate

Table 5

Solution Process

Definition of Symbols

BAT - Solution Methods Discussed by Bathe and Wilson (10)

DU - Direct Integration Methods were used; However, a Specific Type was not Discussed

ECD - Explicit Central Difference Time Integration

HQRI - Householder Reduction, QR Eigenvalue Solution and Inverse Iteration to Find Eigenvectors

JAC - Jacobi Solution Method

NB - Newmark-β Method

ND - Solution Process Not Discussed

RL - Ritz Method using Lagrange Functions (3)

RQ - Rayleigh Quotient Minimization Method

WO - Wilson-Θ Method

Table 6

Matrix Reduction Method

Definition of Symbols

ASH - Reduction method proposed by Appa, Smith and Hughes (4) where degrees of freedom producing the highest norms of vectors generated by $K^{-1}M$ are retained

FRI - Reduction method based on the Finite Element, Rayleigh-Ritz, and Power Methods discussed by Fried (27)

GUY - Reduction method discussed by Guyan (32) involving combinations of K and M elements in the elimination of degrees of freedom

HO - Reduction method where degrees of freedom are eliminated based on the lowest frequencies determined neglecting coupling effects (34)

RNU - Reduction method not used

Summary

There appears to be no clear cut preference in the literature for any one form of mass matrix. Although ten different types of mass matrices were mentioned, only four were discussed in more than one article: the consistent mass matrix, lumped diagonal mass matrix, α-factor diagonal mass matrix introduced by Key and Beisinger (49), and the scaled diagonal mass matrix formulated by Hinton, Rock, and Zienkiewicz (35).

The most commonly studied mass matrices were the lumped diagonal mass matrix and the consistent mass matrix. The most common element model used in their examination was the beam. For frames, the lumped diagonal mass matrix was used twice as often as the consistent mass matrix, while for structures composed of plate elements, the consistent mass matrix was more often used. When a direct integration solution scheme was employed, some form of diagonal mass matrix was usually used. Accurate frequencies were determined for certain problems by all methods.

CHAPTER III

DEVELOPMENT OF THE MASS MATRICES FOR THE COMPARISON STUDY

The four types of mass matrices discussed by more than one article in the literature study are included in the comparison study: the consistent mass matrix, lumped diagonal mass matrix, α-factor diagonal mass matrix (49), and the scaled diagonal mass matrix (35). Following an examination of each matrix's development is a discussion of its possible advantages and disadvantages in natural frequency estimation.

Consistent Mass Matrix

The consistent mass matrix can be derived using the kinetic energy, T. In Fig. 1, P is the reference state relative to which the motion of the element is defined, and P^* the location of P at time t. w is the displacement vector of P and V the volume of the element. If ρ is the mass per unit volume (the mass density) of the element, the kinetic energy of the differential volume can be written as

$$dT = \frac{1}{2} \rho \, dV \, \dot{w} \cdot \dot{w} = \frac{1}{2} \rho \, dV \, (\dot{w}_1{}^2 + \dot{w}_2{}^2 + \dot{w}_3{}^2) \tag{1}$$

Eq. 1 can be rewritten in matrix notation as

$$dT = \frac{1}{2} \rho \, \dot{w}^T \, \dot{w} \, dV \tag{2}$$

Both sides of Eq. 2 are integrated to determine the kinetic energy of the continuum model:

$$T = \frac{1}{2} \iiint_V \rho \, \dot{w}^T \, \dot{w} \, dV \tag{3}$$

To discretize the continuum, let

$$\dot{w} = N \, \dot{u} \tag{4}$$

where $\dot{w}$ is the velocity vector, a function of x, y, z, and t. N is the

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \text{Displacement Vector of P}$$

$$w_i = w_i(x,y,z;t)$$

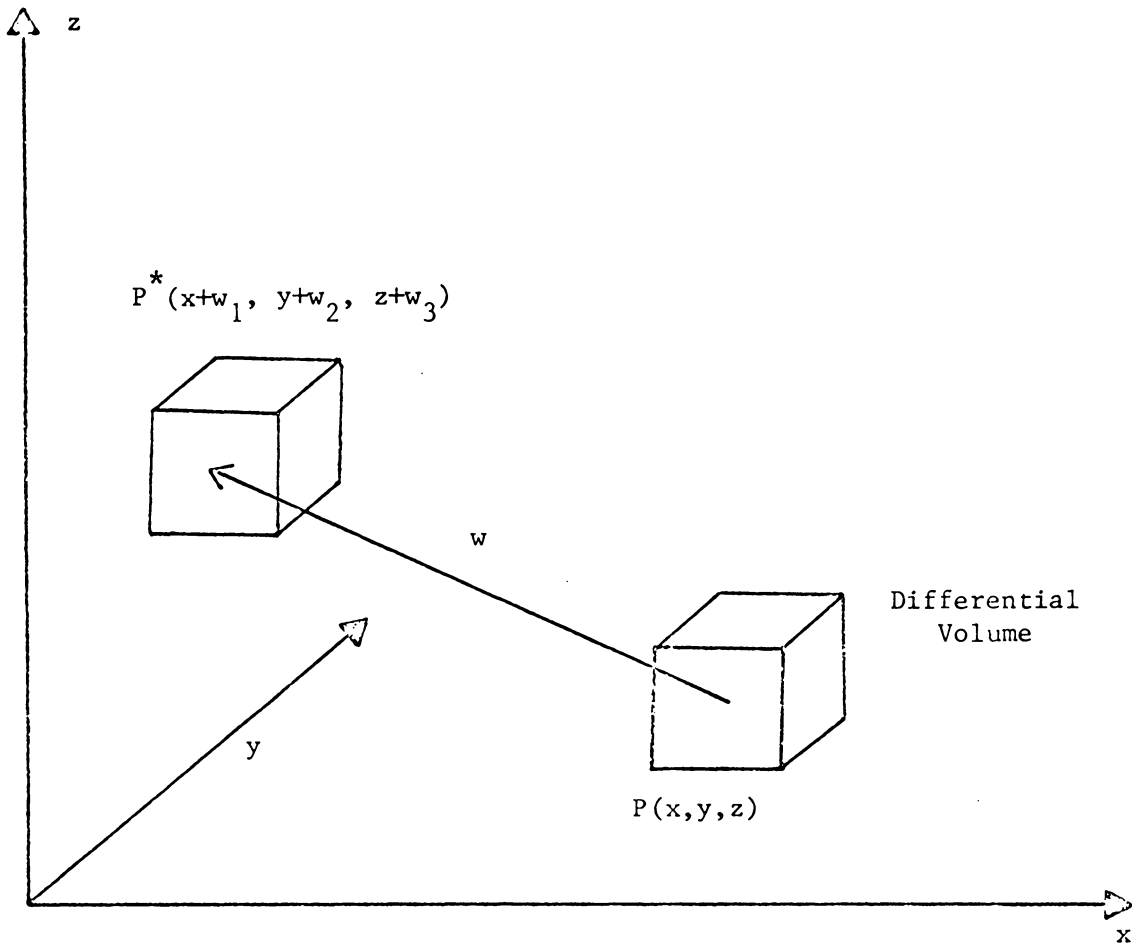


FIGURE 1 - KINETIC ENERGY FORMULATION OF A DIFFERENTIAL VOLUME

interpolation matrix, composed of interpolation functions, which are functions of x, y, and z. $\dot{u}$ is a function of t. Then, by definition,

$$\dot{w}^T = \dot{u}^T N^T \tag{5}$$

Applying Eqs. 4 and 5 to Eq. 3 gives

$$T = \frac{1}{2} \dot{u}^T \left[\iiint_V \rho N^T N \, dV \right] \dot{u}$$

or

$$T = \frac{1}{2} \dot{u}^T m \dot{u}$$

where

$$m = \iiint_V \rho N^T N \, dV \tag{6}$$

m is a consistent mass matrix if the interpolation functions used to discretize the velocity and displacement fields are identical.

The beam-column element, shown in Fig. 2, is used in modeling the test problems for the comparison study. Its consistent mass matrix derived using Lagrange and Hermite interpolation functions (39) is

$$m = \frac{\rho AL}{420} \begin{bmatrix} 140 & 147 & 21L & 70 & 63 & -14L \\ 147 & 156 & 22L & 63 & 54 & -13L \\ 21L & 22L & 4L^2 & 14L & 13L & -3L^2 \\ 70 & 63 & 14L & 140 & 147 & -21L \\ 63 & 54 & 13L & 147 & 156 & -22L \\ -14L & -13L & -3L^2 & -21L & -22L & 4L^2 \end{bmatrix} \tag{7}$$

where A is the area and L the length of the element.

When studying the natural frequencies and mode shapes of structures with finite element analysis, using a consistent mass matrix often produces more accurate results (3, 5, 6, 14, 16, 54, 88). A consistent mass matrix has the advantage that it provides a mathematical approximation of

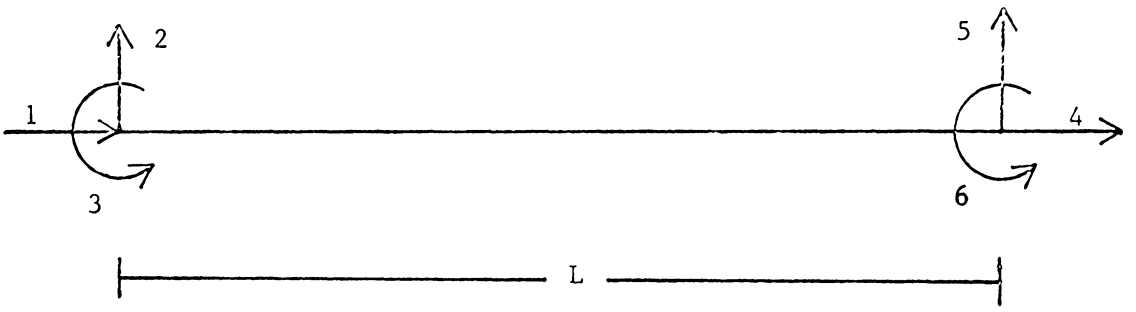


FIGURE 2 - BEAM-COLUMN ELEMENT USED IN THE COMPARISON STUDY

the exact inertia force associated with each degree of freedom rather than some arbitrary lumped value, and accounts for coupling between degrees of freedom. Another advantage is that when using a consistent mass matrix with compatible elements, the computed natural frequencies of the structure will always be upper bounds to the exact frequencies (6, 16, 19, 20, 23).

According to the literature, the primary disadvantage in using the consistent mass matrix is the increased complexity of the matrix computations, that is, the inversion or triangularization of a full or banded matrix rather than a diagonal one.

Lumped Diagonal Mass Matrix

The interpolation functions, N, used in the evaluation of the consistent mass matrix, do not have to be exactly the same as those which discretize the displacement field. Other mass matrices can be derived by using different interpolation functions.

The lumped diagonal mass matrix for the beam element shown in Fig. 3(a) can be calculated from Eq. 6, if the interpolation functions are determined by assuming that half of the element acts like a rigid body unaffected by the remaining half (19, 77). The interpolation functions determined in this manner are

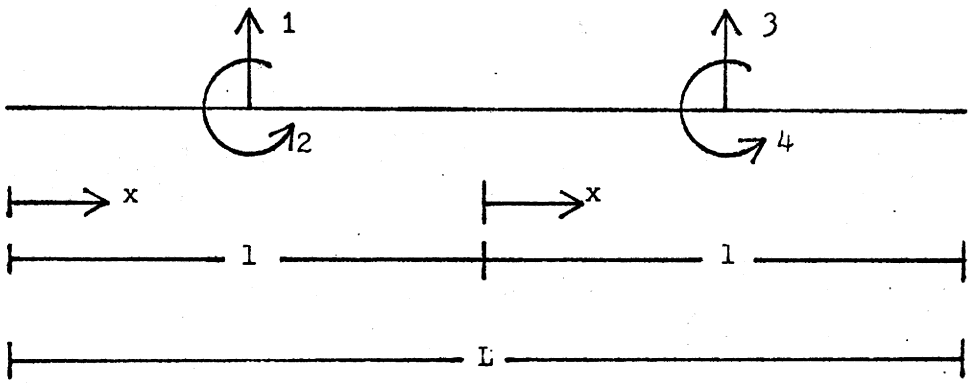
$$N_1 = 1$$

$$N_2 = x$$

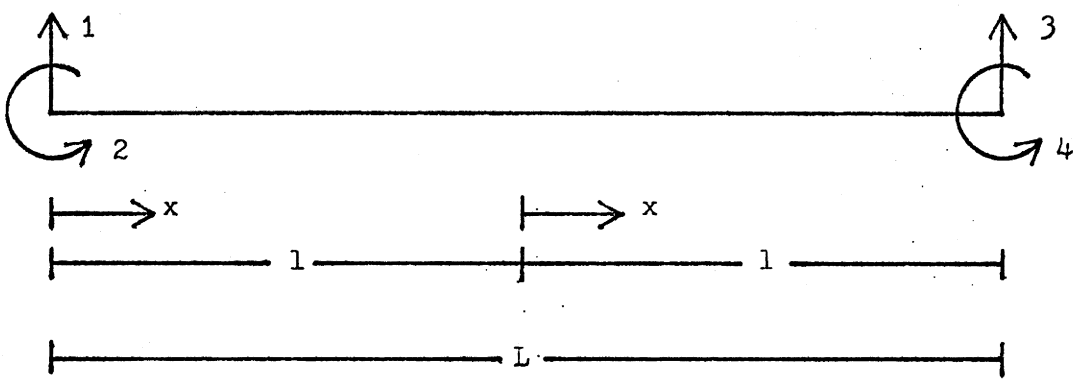
$$N_3 = 1$$

$$N_4 = x$$

and the resulting mass matrix is



(a)



(b)

FIGURE 3 - BEAM ELEMENTS

$$m = \rho A \begin{bmatrix} \frac{L}{2} & & & \\ & \frac{L^3}{96} & & \\ & & \frac{L}{2} & \\ & & & \frac{L^3}{96} \end{bmatrix}$$

Observe that the rotational terms are equal to the mass moments of inertia of each half about its center. However, if this method is used to calculate the interpolation functions for the beam element shown in Fig. 3(b), the type used in Test Problem 1, the resulting mass matrix is not diagonal. The interpolation functions remain

$$N_1 = 1$$

$$N_2 = x$$

$$N_3 = 1$$

$$N_4 = x$$

but the mass matrix becomes

$$m = \rho A \begin{bmatrix} \frac{L}{2} & \frac{L^2}{8} & 0 & 0 \\ \frac{L^2}{8} & \frac{L^3}{24} & 0 & 0 \\ 0 & 0 & \frac{L}{2} & \frac{L^2}{8} \\ 0 & 0 & \frac{L^2}{8} & \frac{L^3}{24} \end{bmatrix}$$

For this element, the rotational terms are equal to the mass moment of inertia of each half about its end. Although the diagonal terms are correct, some coupling between degrees of freedom is retained, that is, some off-diagonal terms are non-zero. Therefore, it appears that to guarantee the correct determination by this method of the lumped diagonal mass ma-

trix for an element, each half of the element should be symmetric about its generalized displacements.

In a more popular method of forming a lumped diagonal mass matrix, the mass of contiguous regions surrounding a node are considered concentrated at that node. For example, a beam is divided in half and ascribed a rotational mass equal to the mass moment of inertia of the adjacent half segment about the node, and a translational mass equal to the mass of the half segment. The lumped diagonal mass matrix for the beam-column element, used in the test problems, is determined in this manner and shown in Eq. 8.

$$m = \rho A \begin{bmatrix} \frac{L}{2} & & & & & \\ & \frac{L}{2} & & & & \\ & & \frac{1}{3}\left(\frac{L}{2}\right)^3 & & & \\ & & & \frac{L}{2} & & \\ & & & & \frac{L}{2} & \\ & & & & & \frac{1}{3}\left(\frac{L}{2}\right)^3 \end{bmatrix} \tag{8}$$

Diagonal mass matrices, formed by any method, require less storage space and are easily inverted. The fundamental frequencies determined are usually accurate, and may actually at times be better than the frequencies determined using the consistent mass system (1, 9, 19, 20, 23, 28, 35). If a diagonal mass system is used, the calculated frequencies may be above or below the actual frequencies. Also, since the mass is concentrated at a point rather than distributed throughout the system, the lumping of the mass overestimates the flexibility of the structure, while the structural model inherently overestimates the stiffness. These could be reasons why in some cases diagonal mass results approximate the

actual frequencies more closely than the consistent mass results. Therefore, the lack of bounding frequencies may not be a great disadvantage to a diagonal mass approach. Also, unless the finite element is conforming, the bounding frequency property of the consistent mass approach does not hold.

The mode shapes determined by a diagonal mass method are less reliable and frequencies are usually less accurate than when using the consistent mass system, although only slightly so in the lower modes. In general, the errors induced by lumping increase as the complexity of the element increases (23, 41, 51, 62, 67, 88, 101).

α-Factor Diagonal Mass Matrix

This method for forming a diagonal mass matrix, described in a paper by Key and Beisinger (49), is an approach which generates a diagonal mass matrix from the non-diagonal consistent mass matrix. Consider the consistent mass matrix terms, from Eq. 7, corresponding to the translational displacements of the beam-column element. The non-zero elements of the α-factor diagonal mass matrix are formed by adding the diagonal terms of the consistent mass matrix to the appropriate off-diagonal terms:

$$m_{11} = \frac{\rho AL}{420} (140 + 70) = \frac{\rho AL}{2}$$

$$m_{22} = \frac{\rho AL}{420} (156 + 54) = \frac{\rho AL}{2}$$

$$m_{44} = \frac{\rho AL}{420} (70 + 140) = \frac{\rho AL}{2} .$$

$$m_{55} = \frac{\rho AL}{420} (54 + 156) = \frac{\rho AL}{2}$$

The same technique is then applied to the rotational inertia terms

$$m_{33} = \frac{\rho AL}{420} (4L^2 - 3L^2) = \frac{\rho AL}{420}$$

$$m_{66} = \frac{\rho AL}{420} (-3L^2 + 4L^2) = \frac{\rho AL}{420}$$

The diagonal mass matrix formed according to the α-factor method is

$$m = \frac{\rho AL}{420} \begin{bmatrix} 210 & & & & & \\ & 210 & & & & \\ & & \alpha L^2 & & & \\ & & & 210 & & \\ & & & & 210 & \\ & & & & & \alpha L^2 \end{bmatrix}$$

where α is the gradient inertia scaling factor. α is selected so that the maximum eigenvalue for the diagonal mass system is equal to the maximum eigenvalue of the consistent mass system. The α value specified by Krieg and Key (51) for a beam-column element is 17.5. Interestingly enough, using this value for α produces a mass matrix equivalent to the lumped diagonal mass matrix defined in Eq. 8. In the comparison study of Chapter IV, the maximum frequency determined using the consistent mass matrix does not coincide with the maximum frequency determined using the α-factor diagonal mass matrix, with α = 17.5. Additional study is needed to determine if the given α value is correct.

The advantages and disadvantages of using this type of mass matrix for natural frequency prediction of structures are those discussed previously for a diagonal mass matrix.

Scaled Diagonal Mass Matrix

For this mass matrix, described in papers by Hinton, Rock and

Zienkiewicz (35) and by Rock and Hinton (82), the diagonal terms of the consistent mass matrix are computed and then scaled so as to preserve the total overall mass of the element.

In mathematical notation, the non-zero elements of the scaled diagonal mass matrix are

$$m_{rr} = \frac{\left(\iiint_V \rho \, N_r \, N_r \, dV \right) \left(\iiint_V \rho \, dV \right)}{\left(\iiint_V \sum_{j=1}^{5} \rho \, N_j \, N_j \, dV \right)} \qquad \begin{array}{l} r = 1, 2, \ldots, 6 \\ j \neq 3 \end{array} \tag{9}$$

Eqs. 7 and 9 are used to compute the non-zero elements of the scaled diagonal mass matrix for the beam-column element

$$m_{11} = m_{44} = \frac{\left(\frac{140\rho AL}{420} \right) \rho AL}{\frac{\rho AL}{420} \left(140 + 156 + 140 + 156 \right)} = \frac{140\rho AL}{592}$$

$$m_{22} = m_{55} = \frac{\left(\frac{156\rho AL}{420} \right) \rho AL}{\frac{\rho AL}{420} \left(140 + 156 + 140 + 156 \right)} = \frac{156\rho AL}{592}$$

$$m_{33} = m_{66} = \frac{\left(\frac{4L^2\rho AL}{420} \right) \rho AL}{\frac{\rho AL}{420} \left(140 + 156 + 140 + 156 \right)} = \frac{4L^2\rho AL}{592}$$

So the diagonal mass matrix formed according to this scaling method is

$$m = \frac{\rho AL}{420} \begin{bmatrix} 140 & & & & & \\ & 156 & & & & \\ & & 4L^2 & & & \\ & & & 140 & & \\ & & & & 156 & \\ & & & & & 4L^2 \end{bmatrix}$$

Since the overall mass of the structure is retained, this appears to be

a more rational method of forming a diagonal mass matrix from the consistent mass matrix than the method proposed by Key and Beisinger.

Again, the advantages and disadvantages pertaining to the use of this type of mass matrix in frequency estimation are those discussed previously for a diagonal mass matrix.

CHAPTER IV

COMPARISON STUDY

A comparison study was conducted in order to determine the accuracy of frequency estimation for different types of mass matrices. Originally the study was to be composed of the four types of mass matrices referred to more than once in the literature study; however, it was reduced to three types after the α-factor diagonal mass matrix was shown to be identical to the lumped diagonal mass matrix for the beam-column element.

Three test problems were used in the study: a simply-supported beam (6), a three member frame (102), and a three story single bay frame (17). Natural frequencies were determined using the computer code listed in Appendix B. Comparisons were made with results presented in the source of the test problem, and to exact results when available.

Description of Analysis Process

The accuracy of a solution and the validity of the results are dependent on the solution process used to obtain them. There is no one best solution method for all types of eigenproblems. The method used in this comparison study was selected after discussions with Dr. Meirovitch and a limited amount of research into the characteristics of various other types of solution routines.

The generalized form of the eigenproblem is

$$K \Phi = \omega^2 M \Phi \tag{10}$$

where K and M are the stiffness and mass matrices respectively. Φ is an eigenvector or mode shape, and ω is a natural frequency of the system. Then for this solution procedure the generalized eigenproblem must be

transformed into a standard eigenproblem of the form

$$\tilde{K}\ \tilde{\Phi} = \omega^2\ \tilde{\Phi}$$

where $\tilde{K}$ is the matrix resulting from the transformation and $\tilde{\Phi}$ is an eigenvector for the $\tilde{K}$ system. The transformation is performed using a symmetric Choleski decomposition. M is transformed into a matrix product of a triangular matrix and its transpose

$$M = \tilde{L}\ \tilde{L}^T \tag{11}$$

Then, letting

$$\tilde{\Phi} = \tilde{L}^T\ \Phi \tag{12}$$

and applying Eqs. 11 and 12 to Eq. 10 gives

$$\tilde{K}\ \tilde{\Phi} = \omega^2\ \tilde{\Phi}$$

where

$$\tilde{K} = \tilde{L}^{-1}\ K\ (\tilde{L}^{-1})^T$$

This decomposition is only applicable if M is positive definite. A consistent mass matrix is always positive definite; however, when using a diagonal mass matrix all elements must be greater than zero. (For a more detailed discussion of this decomposition, see Bathe and Wilson (10) pp. 258, 381-382.)

After the decomposition is performed, the resulting $\tilde{K}$ matrix is reduced to tridiagonal form using a Householder reduction transformation and the eigenvalues are obtained by QR iteration. A tridiagonal matrix is one in which all elements except those on the main diagonal, and the two diagonals adjacent to the main diagonal, are zero. In the QR iterative solution method, $\tilde{K}$ is decomposed into the form

$$\tilde{K} = Q_1\ R_1$$

where Q_1 is an orthogonal and R_1 an upper triangular matrix. Then, let-

ting

$$R_1 Q_1 = Q_1^T \tilde{K}_2 Q_1$$

begins the iterative cycle. The eigenvalues are determined through repeated calculations of RQ, while the eigenvectors are determined by inverse iteration. (For a more detailed discussion of this solution technique (HQRI) see Bathe and Wilson (10) p. 461.) Then, using the same transformations as before, the eigenvectors for the original generalized eigenproblem are determined from those of the tridiagonalized system.

This solution procedure works well for symmetric, positive definite, banded matrices. However, if M is ill-conditioned with respect to inversion, the transformation process will also be ill-conditioned and could result in the inaccurate calculation of eigenvalues and eigenvectors (10). Another disadvantage is that when a banded mass matrix is used in the generalized eigenproblem, a full matrix is obtained from the transformation to the standard eigenproblem. Therefore, bandedness cannot be used to simplify the solution of the standard eigenproblem. Also, since this technique solves for all eigenvalues andeigenvectors, another solution process may be more efficient if all are not required.

Along with the calculation of frequencies and mode shapes, the program listed in Appendix B has the capacity to compute fully stressed designs, responses to member loadings and internal member responses for various loading conditions. If a consistent mass matrix is to be used, it is generated internally and assembled for the system in the same manner as the stiffness matrix. For a diagonal mass matrix, all system mass values must be input.

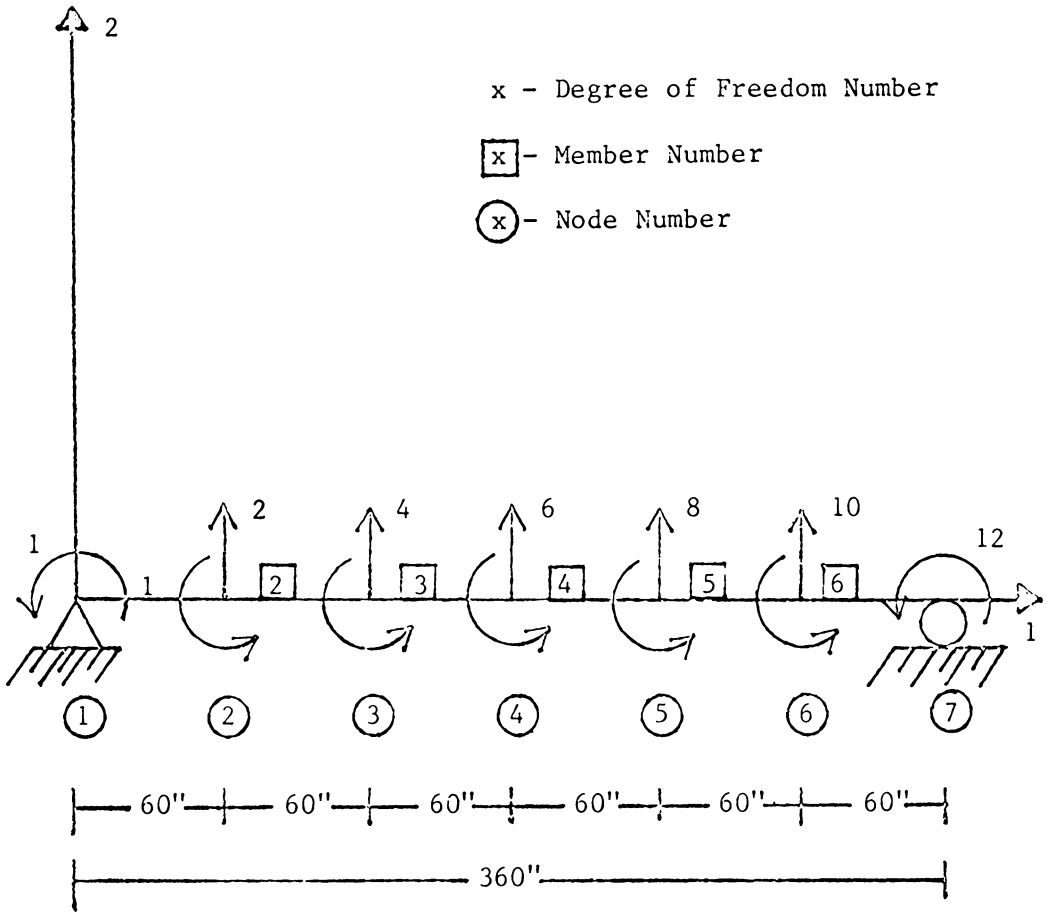
Test Problem 1 - Simply-Supported Beam

This problem, taken from Archer (6), is a simply-supported beam divided into six elements as shown in Fig. 4. The twelve degrees of freedom considered include a translational and rotational degree of freedom at each interior node and a rotational degree of freedom at each end. The mass matrices are developed as described in Chapter III with the exception of the scaled diagonal mass matrix. The use of beam elements rather than beam-column elements causes only a deletion of the first and fourth rows and columns in the consistent mass matrix and the lumped diagonal mass matrix; however, the scaled diagonal mass matrix becomes

$$m = \frac{\rho AL}{312} \begin{bmatrix} 156 & & & \\ & 4L^2 & & \\ & & 156 & \\ & & & 4L^2 \end{bmatrix}$$

The member properties, shown in Fig. 4, are consistent throughout the structure and were chosen so that the output frequencies would correspond directly to those determined by Archer (6). All frequencies were then non-dimensionalized and compared to the exact frequencies for a simply-supported beam (44, 81, 104), as shown in Table 7. The percentage differences between the computed and exact frequencies are shown in Table 8.

An examination of Tables 7 and 8 shows that the computed consistent mass matrix frequencies are verified by the results of the consistent mass study by Archer (6). As expected, the consistent mass matrix frequencies are always higher than the exact frequencies; they are accurate to more than 3% in the first five modes with a maximum error of 38% in the eleventh mode.



Cross Sectional Area - 10.0 in^2

Modulus of Elasticity - 30000.0 ksi

Moment of Inertia - 40.0 in^4

Mass Density - 0.0060014 $\frac{kips\text{-}sec^2}{in^4}$

FIGURE 4 - TEST PROBLEM 1 - SIMPLY-SUPPORTED BEAM

TABLE 7 - NATURAL FREQUENCIES, $\omega_n\left(\frac{\rho A L^4}{EI}\right)^{\frac{1}{2}}$, OF A SIMPLY-SUPPORTED BEAM

Mode Number	Exact	Archer Consistent Mass Study	Consistent Mass Matrix	Lumped Diagonal Mass Matrix	Scaled Diagonal Mass Matrix
1	9.8696	9.8703	9.8801	9.7647	9.8307
2	39.478	39.511	39.519	37.759	38.896
3	88.826	89.177	89.184	80.573	85.697
4	157.91	159.78	159.78	133.50	146.78
5	246.74	253.29	253.29	191.41	212.66
6	355.31	394.37	394.37	249.42	449.64
7	483.61	533.30	533.30	303.23	492.01
8	631.65	733.28	733.28	349.49	573.04
9	799.44	991.28	991.28	386.05	654.30
10	986.96	1312.1	1312.1	411.88	720.75
11	1194.2	1645.2	1645.2	427.03	763.88
12	1421.2	1807.2	1807.2	432.01	778.79

TABLE 8 - PERCENTAGE DIFFERENCE BETWEEN EXACT AND COMPUTED FREQUENCIES

Mode Number	Consistent Mass Matrix	Lumped Diagonal Mass Matrix	Scaled Diagonal Mass Matrix
1	.106	-1.06	-.394
2	.104	-4.35	-1.47
3	.403	-9.29	-3.52
4	1.18	-15.5	-7.05
5	2.65	-22.4	-13.8
6	11.0	-29.8	26.5
7	10.3	-37.3	1.74
8	16.1	-44.7	-9.28
9	24.0	-51.7	-18.2
10	32.9	-58.3	-27.0
11	37.8	-64.3	-36.0
12	27.2	-69.6	-45.2

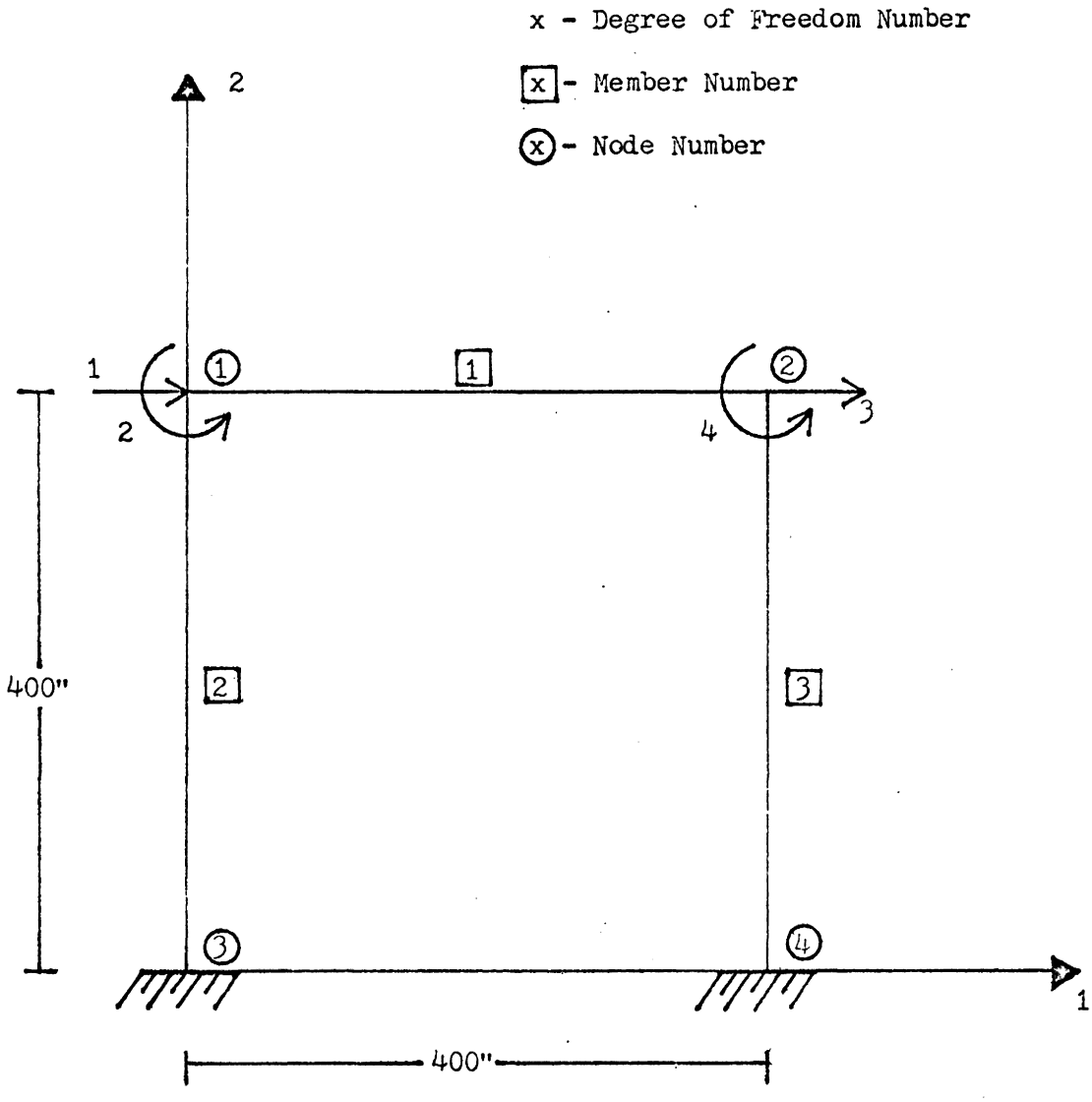
The lumped diagonal mass matrix results are consistently lower than the exact frequencies with an error of more than 20% by the fifth mode. The maximum error was approximately 70% in the twelfth mode.

The scaled diagonal mass matrix frequencies were lower than the exact frequencies in all modes except the sixth and seventh. These computed frequencies were consistently more accurate than the lumped diagonal mass matrix frequencies, and in the higher modes, seven through eleven, were more accurate than the consistent mass matrix frequencies. The error was less than 10% through the first four modes and in modes seven and eight. The maximum error was approximately 45% in the twelfth mode.

Test Problem 2 - Three Member Frame

This problem, taken from Warburton (102), is the three member frame shown in Fig. 5. Joint rotation and lateral translation were allowed at each joint for a total of four degrees of freedom. The mass matrices were developed for the beam-column elements as described in Chapter III. The member properties, included in Fig. 5, are again consistent throughout the structure and chosen so that the output frequencies correspond directly with the Warburton results. The fundamental frequency was determined in the source using two methods, a frame analysis procedure and a single degree of freedom lumped mass model. These frequencies and the computed results are shown in Table 9. The approximate mode shapes were also determined for this problem and are shown in Fig. 6.

The frame analysis procedure is described in detail by Warburton (102). It is essentially a trial and error procedure in which the frame is divided into a number of beams whose displacements and rotations are



Cross Sectional Area - 24.0 in^2

Modulus of Elasticity - 30000.0 ksi

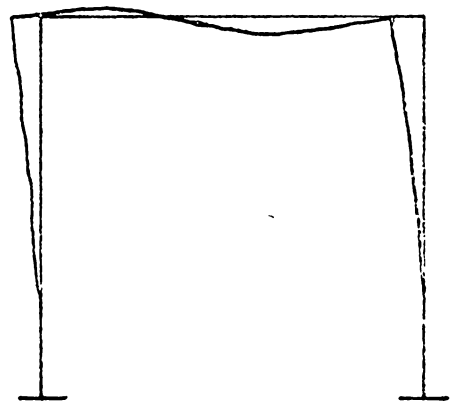
Moment of Inertia - 1600.0 in^4

Mass Density - 0.000078125 $\frac{kips\text{-}sec^2}{in^4}$

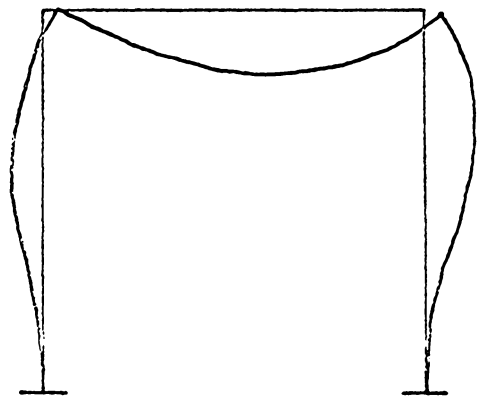
FIGURE 5 - TEST PROBLEM 2 - THREE MEMBER FRAME

TABLE 9 - NATURAL FREQUENCIES, $\omega_n\left(\frac{\rho A L^4}{EI}\right)^{\frac{1}{2}}$, OF A THREE MEMBER FRAME

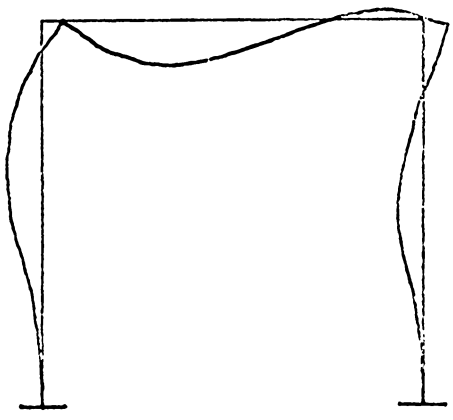
Mode Number	Warburton		Consistent Mass Matrix	Lumped Diagonal Mass Matrix	Scaled Diagonal Mass Matrix
	Frame Analysis	Lumped Single Degree of Freedom System			
1	3.17	3.46	3.118	2.853	4.079
2	--	--	18.01	8.479	21.06
3	--	--	35.35	11.13	27.34
4	--	--	139.9	69.37	98.10



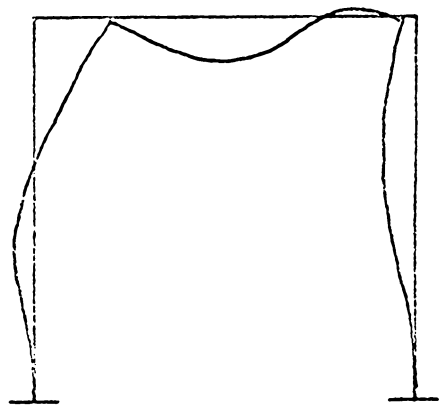
First Mode



Second Mode



Third Mode



Fourth Mode

FIGURE 6 - MODE SHAPES FOR TEST PROBLEM 2

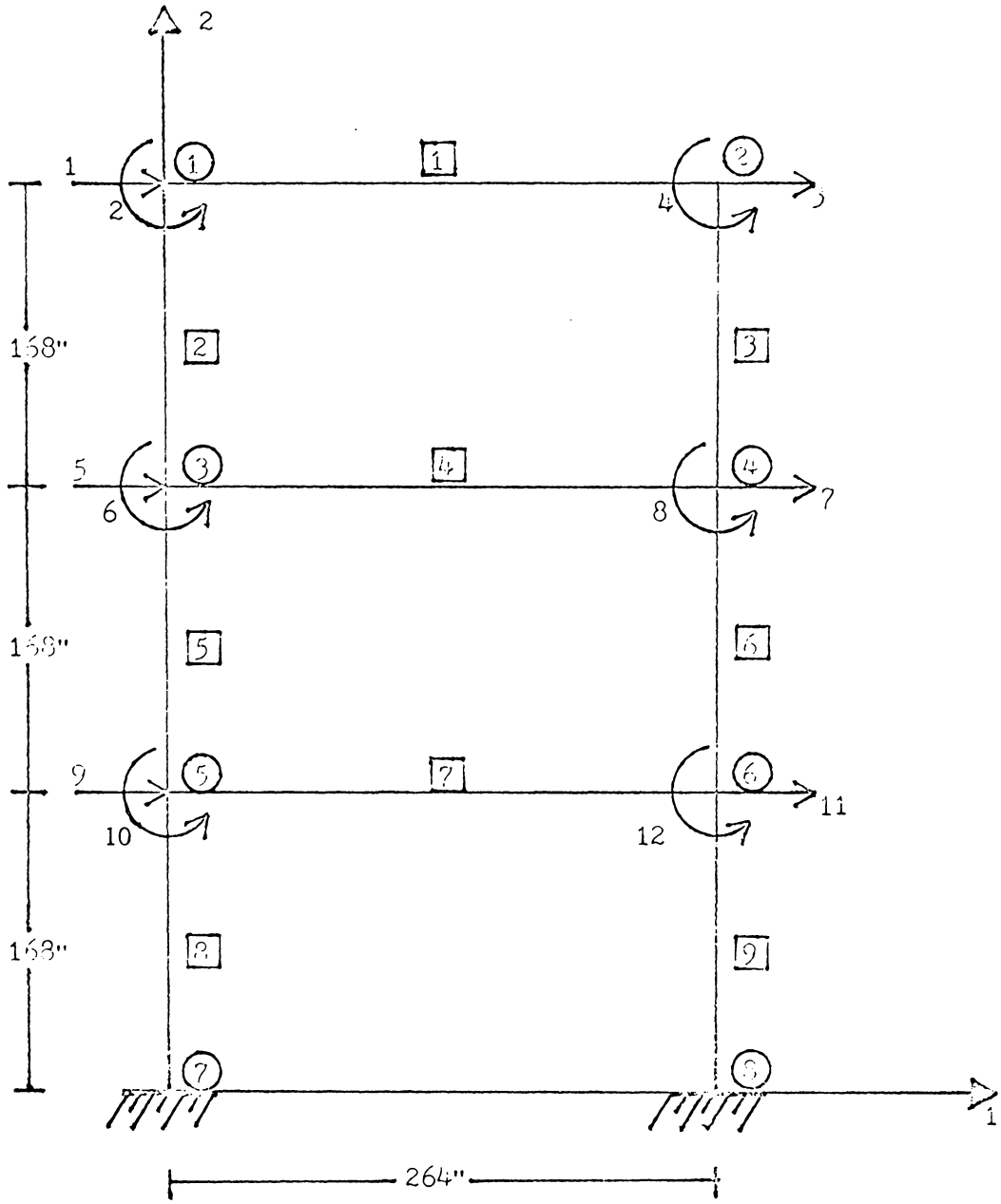
expressed in terms of end forces and moments. Continuity is ensured by equating the appropriate displacements.

The fundamental frequency determined using the consistent mass matrix is within 1% of the fundamental frequency determined using the frame analysis procedure. However, frequencies determined using either of the diagonal mass matrix schemes vary from these by 10% or more. The lumped diagonal mass matrix frequencies are the lowest in all modes. The consistent mass matrix frequencies are the largest in the higher modes, while the scaled diagonal mass matrix frequencies are the largest in the lower modes. The fact that the scaled diagonal mass matrix frequencies could be greater than the consistent mass matrix frequencies was not seen in Test Problem 1.

Test Problem 3 - Three Story Single Bay Frame

This problem, taken from Cheng (17), is the three story single bay frame shown in Fig. 6. Rotation and lateral translation were again allowed at each joint for a total of twelve degrees of freedom. The mass matrices are as described in Chapter III; the given member properties (17) varied throughout the structure and are shown in Table 10. Cheng determined the frequencies for the first three modes using a three degree of freedom lumped mass model. These frequencies and the computed results are shown in Table 11.

Since there were no exact frequencies available for this problem, general comparisons to the lumped mass results of Cheng are all that can be made. As expected, the lumped diagonal mass matrix frequencies are lowest in all modes and closest to the frequencies determined by Cheng.



x - Degree of Freedom Number

[x] - Member Number

(x) - Node Number

FIGURE 7 - TEST PROBLEM 3 - THREE STORY SINGLE BAY FRAME

TABLE 10 - MEMBER PROPERTIES FOR TEST PROBLEM 3 - THREE STORY SINGLE BAY FRAME

Member Number	Cross-Sectional Area (in^2)	Moment of Inertia (in^4)	Modulus of Elasticity (ksi)	Mass Density ($\frac{kips\text{-}sec^2}{in^4}$)
1	10.0	340.0	30000.0	$7.3386(10)^{-7}$
2	9.13	239.0	30000.0	$7.3386(10)^{-7}$
3	9.13	239.0	30000.0	$7.3386(10)^{-7}$
4	10.0	340.0	30000.0	$7.3386(10)^{-7}$
5	13.2	351.0	30000.0	$7.3386(10)^{-7}$
6	13.2	351.0	30000.0	$7.3386(10)^{-7}$
7	10.0	340.0	30000.0	$7.3386(10)^{-7}$
8	25.0	723.0	30000.0	$7.3386(10)^{-7}$
9	25.0	723.0	30000.0	$7.3386(10)^{-7}$

TABLE 11 - NATURAL FREQUENCIES, $\omega_n\left(\frac{\rho A L^4}{EI}\right)^{\frac{1}{2}}$, OF A THREE STORY SINGLE BAY FRAME

Mode Number	Cheng Lumped Three Degree of Freedom System	Consistent Mass Matrix	Lumped Diagonal Mass Matrix	Scaled Diagonal Mass Matrix
1	32.270	34.365	33.486	47.833
2	96.927	103.87	93.263	137.41
3	168.94	185.48	159.44	236.95
4	--	399.58	173.64	431.10
5	--	471.01	232.76	571.80
6	--	602.84	243.81	577.45
7	--	657.66	286.37	681.47
8	--	908.73	286.53	711.27
9	--	1018.3	316.03	777.04
10	--	1226.7	841.75	1178.5
11	--	1478.8	994.34	1400.4
12	--	2506.1	1223.2	1742.9

Again, as in Test Problem 2, the consistent mass matrix frequencies are the largest in the higher modes, while the scaled diagonal mass matrix frequencies are the largest in the lower modes.

Time and Cost Comparison

For each test problem, comparisons were made between the time required to reach a solution when using a banded consistent mass matrix and that required when using a diagonal one. These execution times and the corresponding costs are shown in Table 12.

Although the execution times in all cases were slightly less when using a diagonal mass matrix scheme, the corresponding costs were unaffected. Therefore, any economic incentive for using a diagonal mass matrix was not evident in this study.

TABLE 12 - EXECUTION TIMES AND CORRESPONDING COSTS FOR CONSISTENT AND DIAGONAL MASS SYSTEMS

		Test Problem 1	Test Problem 2	Test Problem 3
Execution Times (CPU)	Consistent Mass System	4.80	3.35	4.99
	Diagonal Mass System	4.68	3.32	4.88
Cost (Cents)	Consistent Mass System	10	6	10
	Diagonal Mass System	10	6	10

CHAPTER V

SUMMARY AND CONCLUSIONS

A major conclusion resulting from this study is that there is no clear cut preference in the literature for any one type of mass matrix; however, based on the results of the comparison study, the frequencies determined using the consistent mass matrix were more accurate, and computed with no significant increase in solution time or cost, than those determined using a diagonal mass system. Additional research is needed to determine if this contradiction of the literature can be substantiated for larger problems.

In Test Problem 1, where direct comparisons with the exact frequencies could be made, the consistent mass matrix frequencies were more accurate through the first six modes than those determined using either diagonal mass matrix. Therefore, since these lower modes are critical for most civil engineering structures, the consistent mass matrix is superior in frequency prediction to the diagonal systems. In the higher modes, the scaled diagonal mass matrix frequencies more closely approximated the exact frequencies; however, in most cases the error was 10% or more. Therefore, the value of frequency estimation for the higher modes is questionable. Although the exact frequencies for the structures in Test Problems 2 and 3 were not available, these problems were useful for frequency and time comparisons between mass matrix types. The exact results could be determined with further study to provide an additional basis for accuracy comparisons.

The results of the time and cost comparisons showed that there were no additional costs incurred when using a consistent mass matrix. There-

fore, there is little reason to use a diagonal mass matrix.

It is hoped that the fairly extensive literature study, presented in Chapter II, will aid the person interested in discovering more about the role of the mass matrix in dynamic structural analysis; although the mass matrix is infrequently the major topic of an article, it is often considered as important within the article. The literature study is perhaps the most valuable and useful part of this thesis.

Included in the literature study were several methods for reducing the size of eigenproblems by a reduction in the number of degrees of freedom. Perhaps, in a later work, an automated degree of freedom reduction method could be incorporated into the program listed in Appendix B, and a study could be done to determine the effects of reduction on frequency estimation.

REFERENCES

1. Abel, J. F. and Popov, E. P., "Static and Dynamic Finite Element Analysis of Sandwich Structures," Proceedings of the Second Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1968.

2. Akesson, B. A., "PFVIBAT - A Computer Program for Plane Frame Vibration Analysis by an Exact Method," International Journal for Numerical Methods in Engineering, Vol. 10, No. 6, 1976, pp. 1221-1231.

3. Ansari, K. A., "Nonlinear Vibrations of a Rotating Pretwisted Blade," Computers and Structures, Vol. 5, No. 2, 1975, pp. 101-118.

4. Appa, K., Smith, G. C. C. and Hughes, J. T., "Rational Reduction of Large-Scale Eigenvalue Problems," Journal of the American Institute of Aeronautics and Astronautics, Vol. 10, No. 7, 1972, pp. 964-965.

5. Archer, J. S., "Consistent Matrix Formulations for Structural Analysis Using Finite Element Techniques," Journal of the American Institute of Aeronautics and Astronautics, Vol. 3, No. 10, 1965, pp. 1910-1918.

6. Archer, J. S., "Consistent Mass Matrices for Distributed Mass Systems," Journal of the Structural Divison, ASCE, Vol. 89, 1963, pp. 161-178.

7. Argyris, J. H. and Scharpe, D. W., "Matrix Displacement Analysis of Shells and Plates Including Transverse and Shear Strain Effects," Computer Methods in Applied Mechanics and Engineering, Vol. 1, No. 1, 1972, pp. 81-139.

8. Armstrong, I. D., "The Natural Frequencies of Multi-Storey Frames," The Structural Engineer, Vol. 47, No. 8, 1969, pp. 299-308.

9. Bathe, K. J. and Wilson, E. L., "Large Eigenvalue Problems in Dynamic Analysis," Journal of the Engineering Mechanics Divison, ASCE, Vol. 98, No. 6, 1972, pp. 1471-1485.

10. Bathe, K. J. and Wilson, E. L., Numerical Methods in Finite Element Analysis, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1976.

11. Bathe, K. J. and Wilson, E. L., "Solution Methods for Eigenvalue Problems in Structural Mechanics," International Journal for Numerical Methods in Engineering, Vol. 6, No. 2, 1973, pp. 213-226.

12. Belytschko, T. and Hseih, B. J., "Non-Linear Transient Finite Element Analysis with Convected Coordinates," International Journal for Numerical Methods in Engineering, Vol. 7, 1973, pp. 255-271.

13. Belytschko, T., Schwer, L. and Klein, M. J., "Large Displacement Transient Analysis of Space Frames," International Journal for Numerical Methods in Engineering, Vol. 11, No. 1, 1977, pp. 65-84.

14. Benton, M. D., Hobbs, G. K. and Dickerson, J. R., "Dynamic Analysis of Complex Structures," Shock and Vibration Bulletin, No. 37, Part 2, 1968, pp. 173-192.

15. Berger, B. S. and Shore, S., "Dynamic Response of Lattice-Type Structures," Journal of the Engineering Mechanics Division, ASCE, Vol. 89, No. 2, 1963, pp. 47-70.

16. Bogner, F. K., Fox, R. L. and Schmit, Jr., L. A., "The Generation of Inter-Element Compatible Stiffness and Mass Matrices by the use of Interpolation Formulas," Proceedings of the First Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1965.

17. Cheng, F. Y., "Dynamic Characteristics of Multistory Buildings," Journal of the Structural Division, ASCE, Vol. 94, No. 8, 1968, pp. 2023-2025.

18. Cheng, F. Y., "Vibrations of Timoshenko Beams and Frameworks," Journal of the Structural Division, ASCE, Vol. 96, No. 3, 1970, pp. 551-571.

19. Clough, R. W., "Analysis of Structural Vibrations and Dynamic Response," Recent Advances in Matrix Methods of Structural Analysis and Design, University of Alabama Press, Huntsville, Alabama, 1971.

20. Clough, R. W. and Wilson, E. L., "Dynamic Finite Element Analysis of Arbitrary Thin Shells," Computers and Structures, Vol. 1, No. 1, 1971, pp. 33-55.

21. Coull, A. and Mukherjee, P. R., "Natural Vibrations of Shear Wall Buildings on Flexible Supports," Earthquake Engineering and Structural Dynamics, Vol. 6, No. 3, 1978, pp. 295-315.

22. Davis, R., Henshell, R. D. and Warburton, G. B., "Curved Beam Finite Elements for Coupled Bending and Torsional Vibration," Earthquake Engineering and Structural Dynamics, Vol. 1, No. 2, 1972, pp. 165-175.

23. Desai, C. S. and Abel, J. F., Introduction to the Finite Element Method, Van Nostrand Reinhold Co., New York, 1972.

24. Farrington, C. C., Gregory, R. T. and Taub, A. H., "On the Numerical Solution of Sturm-Liouville Differential Equations," Mathematical Tables and Other Aids to Computation, Vol. 11, No. 59, 1957, pp. 131-150.

25. Fertis, D. G., Dynamics and Vibrations of Structures, John Wiley and Sons, Inc., New York, 1973.

26. Fried, I. and Malkus, D. S., "Finite Element Mass Matrix Lumping by Numerical Integration with No Convergence Rate Loss," International Journal of Solids and Structures, Vol. 11, No. 4, 1975, pp. 461-466.

27. Fried, I., "Condensation of Finite Element Eigenproblems," Journal of the American Institute of Aeronautics and Astronautics, Vol. 10, No. 11, 1972, pp. 1529-1530.

28. Greene, B. E., Jones, R. E., McLay, R. W. and Strome, D. R., "Dynamic Analysis of Shells Using Doubly Curved Finite Elements," Proceedings of the Second Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1968.

29. Gupta, K. K., "On a Finite Dynamic Element Method for Free Vibration Analysis of Structures," Computer Methods in Applied Mechanics and Engineering, Vol. 9, No. 1, 1976, pp. 105-120.

30. Gupta, K. K., "Vibration of Frames and Other Structures with Banded Stiffness Matrix," International Journal for Numerical Methods in Engineering, Vol. 2, No. 2, 1970, pp. 221-228.

31. Guyan, R. J., "Distributed Mass Matrix for Plate Element Bending," Journal of the American Institute of Aeronautics and Astronautics, Vol. 3, No. 3, 1965, pp. 567-568.

32. Guyan, R. J., "Reduction of Stiffness and Mass Matrices," Journal of the American Institute of Aeronautics and Astronautics, Vol. 3, No. 2, 1965.

33. Hartzman, M. and Hutchinson, J. R., "Non-Linear Dynamics of Solids by the Finite Element Method," Computers and Structures, Vol. 2, 1972, pp. 47-77.

34. Henshell, R. D. and Ong, J. H., "Automatic Masters for Eigenvalue Economization," Earthquake Engineering and Structural Dynamics, Vol. 3, No. 4, 1975, pp. 375-383.

35. Hinton, E., Rock, T. A. and Zienkiewicz, O. C., "Note on Mass Lumping and Related Processes in the Finite Element Method," Earthquake Engineering and Structural Dynamics, Vol. 4, No. 3, 1976, pp. 245-249.

36. Hinton, E., "Dynamic Transient Analysis of Axisymmetric Circular Plates by the Finite Element Method," Journal of Sound and Vibration, Vol. 46, No. 4, 1976, pp. 465-472.

37. Holzer, S. M., "Lecture Notes on CE5640 - Computer Aided Structural Design," VPI & SU, Blacksburg, Virginia, Summer Quarter, 1978.

38. Holzer, S. M., "Lecture Notes on CE6020 - Dynamics of Structures," VPI & SU, Blacksburg, Virginia, Spring Quarter, 1978.

39. Holzer, S. M., "Lecture Notes on CE6030 - Finite Element Analysis of Structures," VPI & SU, Blacksburg, Virginia, Spring Quarter, 1978.

40. Holzer, S. M., "Lecture Notes on CE4002 - Matrix Structural Analysis," VPI & SU, Blacksburg, Virginia, Winter Quarter, 1978.

41. Huebner, K. H., The Finite Element Method for Engineers, John Wiley and Sons, Inc., New York, 1975.

42. Hughes, T. J. R., Cohen, M. and Haroun, M., "Reduced and Selective Integration Techniques in the Finite Element Analysis of Plates," Nuclear Engineering and Design, Vol. 46, No. 1, 1978, pp. 203-222.

43. Hughes, T. J. R., "Reduction Scheme for Some Structural Eigenvalue Problems by a Variational Theorem," International Journal for Numerical Methods in Engineering, Vol. 10, No. 4, 1976, pp. 845-852.

44. Hurty, W. C. and Rubenstein, M. F., Dynamics of Structures, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1964.

45. Irons, B. M. and Barlow, J., "Comments on Matrices for the Direct Stiffness Method," Journal of the American Institute of Aeronautics and Astronautics, Vol. 2, No. 2, 1964, p. 403.

46. Iyengar, H. S., Amin, N. and Carpenter, L., "Computerized Design of World's Tallest Building," Computers and Structures, Vol. 2, No. 5, 1972, pp. 771-783.

47. Johnsen, T. L., "On the Computation of Natural Modes of an Unsupported Vibrating Structure by Simultaneous Iteration," Computer Methods in Applied Mechanics and Engineering, Vol. 2, No. 3, 1973, pp. 305-322.

48. Kamat, M. P., "Effect of Shear Deformations and Rotary Inertia on Optimum Beam Frequencies," International Journal for Numerical Methods in Engineering, Vol. 9, No. 1, 1975, pp. 51-62.

49. Key, S. W. and Beisinger, Z. E., "The Transient Dynamic Analysis of Thin Shells by the Finite Element Method," Proceedings of the Third Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1970.

50. Klein, S. and Sylvester, R. J., "The Linear Elastic Dynamic Analysis of Shells of Revolution by the Matrix Displacement Method," Proceedings of the First Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1965.

51. Krieg, R. D. and Key, S. W., "Transient Shell Response by Numerical Time Integration," International Journal for Numerical Methods in Engineering, Vol. 7, 1973, pp. 273-286.

52. Leckie, F. A. and Lindberg, G. M., "The Effect of Lumped Parameters on Beam Frequencies," Aeronautical Quarterly, Vol. 14, 1963, pp. 224-240.

53. Leimbach, K. R. and McDonald, D., "Free Vibration of Two-Dimensional Frameworks," Journal of the Structural Division, ASCE, Vol. 96, No. 2, 1970, pp. 267-289.

54. Mee, A. L., Jordaan, I. J. and Ward, M. A., "Dynamic Response of a Staggered Wall-Beam Structure," Earthquake Engineering and Structural Dynamics, Vol. 3, No. 4, 1975, pp. 353-364.

55. Mei, C., "Coupled Vibrations of Thin-Walled Beams of Open Section Using the Finite Element Method," International Journal of Mechanical Sciences, Vol. 12, 1970, pp. 883-891.

56. Melosh, R. J. and Lang, T. E., "Modified Potential Energy Mass Representations for Frequency Prediction," Proceedings of the First Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1965.

57. Melosh, R. J., "Manipulation Errors in Finite Element Analysis," Recent Advances in Matrix Methods of Structural Analysis and Design, University of Alabama Press, Huntsville, Alabama, 1971.

58. Mondkar, D. P. and Powell, G. H., "Finite Element Analysis of Non-Linear Static and Dynamic Response," International Journal for Numerical Methods in Engineering, Vol. 11, No. 3, 1977, pp. 499-520.

59. Monforton, G. R. and Schmit, Jr., L. A., "Finite Element Analysis of Sandwich Plates and Cylindrical Shells with Laminated Faces," Proceedings of the Second Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1968.

60. Nagarajan, S. and Popov, E. P., "Elastic-Plastic Dynamic Analysis of Axisymmetric Solids," Computers and Structures, Vol. 4, No. 6, 1974, pp. 1117-1134.

61. Nagarajan, S. and Popov, E. P., "Non-Linear Dynamic Analysis of Axisymmetric Shells," International Journal for Numerical Methods in Engineering, Vol. 9, No. 3, 1975, pp. 535-550.

62. Neubert, V. H., "Lumping of Mass in Calculating Vibration Response," Journal of the Engineering Mechanics Division, ASCE, Vol. 90, No. 1, 1964, pp. 47-67.

63. Newmark, N. M. and Rosenblueth, E., Fundamentals of Earthquake Engineering, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1971.

64. Olson, M. D., "A Consistent Finite Element Method for Random Response Problems," Computers and Structures, Vol. 2, No. 2, 1972, pp. 161-180.

65. Olsen, M. D. and Lindberg, G. M., "Dynamic Analysis of Shallow Shells with a Doubly Curved Triangular Finite Element," Journal of Sound and Vibration, Vol. 19, No. 3, 1971, pp. 299-318.

66. Olson, M. D. and Lindberg, G. M., "Vibration Analysis of Cantilevered Curved Plates Using a New Cylindrical Shell Finite Element," Proceedings of the Second Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1968.

67. Omid'Varan, C., "Free Vibration of Grid-Stiffened Plates," Journal of Sound and Vibration, Vol. 19, No. 4, 1971, pp. 463-472.

68. Ovunc, B. A., "Dynamics of Frameworks by Continuous Mass Methods," Computers and Structures, Vol. 4, No. 5, 1974, pp. 1061-1089.

69. Parameswaran, M. A. and Sukumaran, K., "A Lumped Mass Vibration Model of a Slender Latticed Cantilever," Computers and Structures, Vol. 6, No. 2, 1976, pp. 107-109.

70. Pardoen, G. C., "Static, Vibration and Buckling Analysis of Axisymmetric Circular Plates Using Finite Elements," Computers and Structures, Vol. 3, No. 2, 1973, pp. 355-375.

71. Pardoen, G. C., "Vibration and Buckling Analysis of Axisymmetric Polar Orthotropic Circular Plates," Computers and Structures, Vol. 4, No. 5, 1974, pp. 951-960.

72. Paz, M. and Dung, L., "Power Series Expansion of the General Stiffness Matrix for Beam Elements," International Journal for Numerical Methods in Engineering, Vol. 9, No. 2, 1975, pp. 449-459.

73. Pedersen, P. and Meghed, M. M., "Axisymmetric Element Analysis Using Analytical Computing," Computers and Structures, Vol. 5, No. 4, 1975, pp. 241-247.

74. Penzien, J., Kaul, M. K. and Berge, B., "Stochastic Response of Off-Shore Towers to Random Sea Waves and Strong Motion Earthquakes," Computers and Structures, Vol. 2, No. 5, 1972, pp. 733-756.

75. Popplewell, N. and McDonald, D., "Conforming Rectangular and Triangular Plate-Bending Elements," Journal of Sound and Vibration, Vol. 19, No. 3, 1971, pp. 333-347.

76. Przemieniecki, J. S., "Quadratic Matrix Equations for Determining Vibration Modes and Frequencies of Continuous Elastic Systems," Proceedings of the First Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, 1965.

77. Przemieniecki, J. S., Theory of Matrix Structural Analysis, McGraw-Hill, Inc., New York, 1968.

78. Ramsden, N. N. and Stoker, J. R., "Mass-Condensation: A Semi-Automatic Method for Reducing the Size of Vibration Problems," International Journal for Numerical Methods in Engineering, Vol. 1, 1969, pp. 333-349.

79. Rao, G. V., Raju, I. S. and Raju, K. K., "A Finite Element Formulation for Large Amplitude Flexural Vibrations of Thin Rectangular Plates," Computers and Structures, Vol. 6, No. 3, 1976, pp. 163-167.

80. Rao, G. V., Raju, K. K. and Raju, I. S., "Finite Element Formulation for the Large Amplitude Free Vibrations of Beams and Orthotropic Circular Plates," Computers and Structures, Vol. 6, No. 3, 1976, pp. 169-172.

81. Rayleigh, J. W. S., The Theory of Sound, Vol. 1, Dover Publications, New York, 1945.

82. Rock, T. A. and Hinton, E., "A Finite Element Method for the Free Vibration of Plates Allowing for Transverse Shear Deformation," Computers and Structures, Vol. 6, No. 1, 1976, pp. 37-44.

83. Rock, T. A. and Hinton, E., "Free Vibration and Transient Response of Thick and Thin Plates Using the Finite Element Method," Earthquake Engineering and Structural Dynamics, Vol. 3, No. 1, 1974, pp. 51-63.

84. Ross, C. T. F., "Finite Elements for the Vibration of Cones and Cylinders," International Journal for Numerical Methods in Engineering, Vol. 9, No. 4, 1975, pp. 833-845.

85. Schenker, L., "The Dynamic Response of Tall Structures to Lateral Loads," Journal of the Engineering Mechanics Division, ASCE, Vol. 82, No. 2, 1956, pp. 944-1 - 944-13.

86. Schreyer, H. L., "Consistent Diagonal Mass Matrices and Finite Element Equations for One-Dimensional Problems," International Journal for Numerical Methods in Engineering, Vol. 12, No. 7, 1978, pp. 1171-1184.

87. Schwanz, H. R., Rutishauser, H. and Stiefel, E., Numerical Analysis of Symmetric Matrices, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1973.

88. Sen, S. K. and Gould, P. L., "Free Vibration of Shells of Revolution Using the FEM," Journal of the Engineering Mechanics Division, ASCE, Vol. 100, No. 2, 1974, pp. 283-303.

89. Slyper, H. A., "Development of Explicit Stiffness and Mass Matrices for a Triangular Plate Element," International Journal of Solids and Structures, Vol. 5, No. 3, 1969, pp. 241-249.

90. Stanton, E. L. and McGovern, D. J., "The Application of Gradient Minimization Methods and Higher Order Discrete Elements to Shell Buckling and Vibration Eigenproblems," Computers and Structures, Vol. 1, No. 3, 1971, pp. 413-434.

91. "SUBROUTINES EIGRS, EHOBK, EHOUSS, EQRT2S, UERTST," International Mathematical and Statistical Libraries, Inc., Houston, Texas.

92. Suryoutomo, H., Gould, P. L. and Prodyut, K. B., "Direct Dynamic Analysis of Shells of Revolution Using High-Precision Finite Elements," Computers and Structures, Vol. 7, No. 3, 1977, pp. 425-433.

93. Tolani, S. K. and Rocke, R. D., "A Strain Energy Comparison of Discrete Modeling for Vibrating Continuous Systems," Journal of Engineering for Industry, Vol. 94, No. 1, 1972, pp. 23-30.

94. Tong, P., Pian, T. H. H. and Bucciarelli, L. L., "Mode Shapes and Frequencies by Finite Element Method Using Consistent and Lumped Masses," Computers and Structures, Vol. 1, No. 4, 1971, pp. 623-638.

95. Toridis, T. G. and Khozeimeh, K., "Computer Analysis of Rigid Frames," Computers and Structures, Vol. 1, No. 2, 1971, pp. 193-221.

96. Venancio-Filho, F. and Iguti, F., "Vibrations of Grids by the Finite Element Method," Computers and Structures, Vol. 3, No. 6, 1973, pp. 1331-1344.

97. Vysloukh, V. A., Kandidov, V. P. and Chesnokov, S. S., "Reduction of the Degrees of Freedom in Solving Dynamic Problems by the Finite Element Method," International Journal for Numerical Methods in Engineering, Vol. 7, No. 2, 1973, pp. 185-194.

98. Wang, C. K., Computer Methods in Advanced Structural Analysis, Intext Press, Inc., New York, 1973.

99. Wang, S. L. and Gray, H. P., "Dynamic Response of Elastoplastic Beam-Girder Systems," Computers and Structures, Vol. 2, No. 2, 1972, pp. 223-251.

100. Wang, T. M. and Kinsman, T. A., "Vibrations of Frame Structures According to the Timoshenko Theory," Journal of Sound and Vibration, Vol. 14, No. 2, 1971, pp. 215-227.

101. Warburton, G. B., "Recent Advances in Structural Dynamics," Symposium on Dynamic Analysis of Structures, Birnichill Institute, National Engineering Laboratory, East Kilbride, Glasgow, 1975.

102. Warburton, G. B., The Dynamical Behavior of Structures, Pergamon Press, London, 1964.

103. Weaver, Jr., W. and Yoshida, D. M., "The Eigenvalue Problem for Banded Matrices," Computers and Structures, Vol. 1, No. 4, 1971, pp. 651-664.

104. Young, D. and Felgar, Jr., R. P., Tables of Characteristic Functions Representing Normal Modes of Vibration of a Beam, The University of Texas Publication No. 4913, Austin, Texas, 1949.

APPENDIX A

USERS GUIDE FOR COMPUTER CODE

The computer code listed in Appendix B is an analysis and/or design program with the capability of determining the natural frequencies and mode shapes of beams and of plane frames composed of beam-column elements. Structures are limited to ten members, ten joints, twenty degrees of freedom and four loading conditions. If more complex structures are to be considered the COMMON statements and storage allocations must be adjusted accordingly. The units used internally in the program are inches, kips, seconds and radians. These must also be used in the input data. A description of the data deck is as follows:

FIRST CARD

NDS

NDS - number of data sets, problems, contained in the data deck

DATA SET CARDS

Structure Information Card

NM,NJ,NDOF,MB,NLC

NM - total number of members in the structure

NJ - total number of joints in the structure

NDOF - number of degrees of freedom considered in the analysis

MB - bandwidth of the stiffness matrix and consistent mass matrix if it is used, equal to the maximum difference between the MCODE values of each element excluding zeros plus one

NLC - number of loading conditions in data set

Function Calling Card

INFOC,IFSDC,ITTCON,INFOP,ILDAP,IGENCL,MASTYP

INFOC - SUBROUTINE INTFOR calling flag: if greater than zero, internal member forces are calculated

IFSDC - SUBROUTINE FSD calling flag: if greater than zero, a fully stressed design is determined

ITTCON - iteration control value: maximum number of iterations allowed to reach a fully stressed design; if IFSDC equals zero ITTCON should be set equal to zero

INFOP - if greater than zero, internal member forces are printed; must be set equal to zero if INFOC and IFSDC are zero

ILDAP - if greater than zero, the tilda force matrix is printed

IGENCL - SUBROUTINE EIGEN calling flag: if equal to zero, frequencies and mode shapes are not determined; if equal to one, frequencies and mode shapes are determined but no other analysis and/or design is performed; if equal to two, frequencies and mode shapes are determined along with other analysis and/or design

MASTYP - calling flag for mass matrix selection: if equal to zero, a consistent mass matrix is generated and assembled internally; if equal to one, system diagonal mass matrix values must be read in while non-diagonal values are set equal to zero internally

Member Loading Total Card - not required if IGENCL equals one

NMA(I) I = 1, NLC

NMA(I) - total number of member loadings in each loading case

Member Property Cards

INC(I,1),INC(I,2),A(I),XI(I),E(I),DENS(I) I = 1, NM

INC(I,1), INC(I,2) - beginning, ending joint of element I

A(I), XI(I), E(I), DENS(I) - area, moment of inertia, modulus of elasticity, mass density for each element I

Use one card for each member

Joint Coordinates Cards

X(I,1),X(I,2) I = 1,NJ

X(I,1), X(I,2) - X or 1, Y or 2 coordinate of joint I

Use one card for each joint

Member Action Index Card - not required if IGENCL equals one

MA(N,I) I = 1,NLC and N = 1,NM

MA(N,I) - number of member loads on member N in load case I

Start a new card for each loading case

Member Action Information Cards - not required if IGENCL equals one or if all NMA(I)'s equal zero

MNUM(I,J),LDTYP(I,J),WON(I,J),WTW(I,J),WTH(I,J),WFO(I,J)

J = 1,NLC and I = 1,NMA(J)

MNUM(I,J) - member number; members must be read in sequentially lowest to highest

LDTYP(I,J) - types of loading, values explained below

WON(I,J), WTW(I,J), WTH(I,J), WFO(I,J) - loading parameters, values explained below, all zeros must be read in

Input all member loadings for load case one and then continue for other loading cases.

Explanation of LDTYP and Loading Parameters

LDTYP = 1 Uniform Distributed Load

WON, WTW - starting, ending position of load from the a-end of the member as a fraction of L

WTH - load value

WFO - set equal to zero

LDTYP = 2 Concentrated Transverse Load

WON - load value

WTW - position of load in inches from a-end of the member

WTH, WFO - set equal to zero

LDTYP = 3 Concentrated Axial Load

WON - load value

WTW - position of load in inches from a-end of the member

WTH, WFO - set equal to zero

LDTYP = 4 Uniform Temperature Increase

WON - amount of temperature increase

WTW - coefficient of thermal expansion

WTH, WFO - set equal to zero

LDTYP = 5 Linearly Varying Distributed Load

WON, WTW - starting, ending position of load from the a-end of the member as a fraction of L

WTH, WFO - starting, ending value of load

LDTYP = 6 Fabrication Error

WON - inches too long, (-) if too short

WTW, WTH, WFO - set equal to zero

LDTYP = 7 Linearly Varying Temperature Increase - Used in Conjunction with LDTYP = 4

WON - linear variation in temperature, 0 to WON, or if used with

LDTYP = 4 it is the difference between the beginning and ending temperature values

WTW - coefficient of thermal expansion

WTH - depth of section

WFO - set equal to zero

Nodal Loading Cards - not required if IGENCL equals one

PSTOR(L,K) K = 1,NLC and L = 1,NDOF

PSTOR(L,K) - load applied to degree of freedom L during loading case K

A new card must be started for each loading case and all zeros must be read in.

Yield Strength Card - not required if IGENCL equals one or if IFSDC equals zero

FY

FY - yield strength of member material

MCODE Matrix Cards

MCODE(L,M) L = 1,NM and M = 1,6

MCODE(L,M) - relates system degrees of freedom to element degrees of freedom; if no system degree of freedom corresponds to degree of freedom M of element L a zero is input, otherwise the corresponding system degree of freedom number is input; for each element, each successive number excluding zeros must be greater than all previous ones

Diagonal Mass Values Card - not required if IGENCL equals zero or if MASTYP equals zero

SM(I,1) I = 1,NDOF

SM(I,1) - system diagonal mass matrix values

APPENDIX B

COMPUTER CODE LISTING

```
C     MAIN PROGRAM                                                      MAIN  10
C     ------------------------------------------------------------------MAIN  20
C     THIS IS A MULTIPURPOSE PLANE FRAME PROGRAM UTILIZING BEAM-COLUMN  MAIN  30
C     ELEMENTS.  MEMBER AND JOINT FORCES, NATURAL FREQUENCIES AND MODE  MAIN  40
C     SHAPES AND FULLY STRESSED DESIGNS CAN BE DETERMINED FOR STRUCTURESMAIN  50
C     WITH A MAXIMUM OF 10 MEMBERS AND/OR JOINTS AND 20 DEGREES OF FREE-MAIN  60
C     DOM OR LESS.  LARGER STRUCTURES MAY BE ANALYZED BUT THE COMMON    MAIN  70
C     STATEMENTS AND STORAGE ALLOCATIONS MUST THEN BE MODIFIED.         MAIN  80
C                                                                       MAIN  90
C                                                                       MAIN 100
C     MAJOR VARIABLES ARE DEFINED AS FOLLOWS:                           MAIN 110
C                                                                       MAIN 120
C     NM,NJ-NUMBER OF MEMBERS, NUMBER OF JOINTS                         MAIN 130
C     NDOF-NUMBER OF DEGREES OF FREEDOM OF THE STRUCTURE                MAIN 140
C     MB-HALF BAND WIDTH OF THE STIFFNESS AND MASS MATRICES             MAIN 150
C     NDS-NUMBER OF DATA SETS MAKING UP DATA DECK                       MAIN 160
C     NLC-NUMBER OF LOADING CONDITIONS                                  MAIN 170
C     INFOC-CALLING FLAG FOR CALCULATION OF INTERNAL MEMBER FORCES      MAIN 180
C     IFSDC-CALLING FLAG FOR CALCULATION OF FULLY STRESSED DESIGN       MAIN 190
C     ITTCON-MAXIMUM ALLOWABLE NUMBER OF ITERATIONS, DESIGN CHANGES, TO MAIN 200
C            REACH A FULLY STRESSED DESIGN                              MAIN 210
C     INFOP-FLAG FOR PRINTING INTERNAL MEMBER FORCES                    MAIN 220
C     ILDAP-FLAG FOR PRINTING MEMBER FORCES DUE TO MEMBER LOADS,        MAIN 230
C           F-TILDAS                                                    MAIN 235
C     IGENCL-FLAG FOR CALCULATION OF EIGENVALUES AND EIGENVECTORS       MAIN 240
C     X-JOINT COORDINATE MATRIX, TWO COORDINATES PER JOINT              MAIN 250
C     INC-MEMBER INCIDENCE MATRIX, CONNECTIVITY BETWEEN MEMBERS         MAIN 260
C     A-CROSS-SECTIONAL AREA OF EACH MEMBER                             MAIN 270
C     XI-MOMENT OF INERTIA OF EACH MEMBER                               MAIN 280
C     E-YOUNG'S MODULUS OF ELASTICITY OF EACH MEMBER                    MAIN 290
C     XL-LENGTH OF EACH MEMBER                                          MAIN 300
C     C-COSINE OF THE ANGLE BETWEEN THE MEMBER AND THE CORRESPONDING    MAIN 310
```

```
C       GLOBAL AXIS                                                     MAIN 320
C     S-SINE OF THE ANGLE BETWEEN THE MEMBER AND THE CORRESPONDING      MAIN 330
C       GLOBAL AXIS                                                     MAIN 340
C     MASTYP-CALLING FLAG FOR MASS MATRIX TYPE DESIRED                  MAIN 350
C     ALPH, BETA-STIFFNESS MATRIX PARAMETERS                            MAIN 360
C     DENS-MASS DENSITY OF ELEMENT                                      MAIN 370
C     WEIGHT-WEIGHT OF MEMBER                                           MAIN 380
C     FY-YIELD STRENGTH OF MEMBER MATERIAL                              MAIN 390
C     TWEIHT-TOTAL WEIGHT OF THE STRUCTURE                              MAIN 400
C     MCODE-MATRIX RELATING ELEMENT TO SYSTEM DEGREES OF FREEDOM        MAIN 410
C     AK-UPPER TRIANGULAR BANDED PORTION OF THE STIFFNESS MATRIX        MAIN 420
C     SM-UPPER TRIANGULAR BANDED PORTION OF THE MASS MATRIX             MAIN 430
C     NMA-NUMBER OF MEMBER ACTIONS IN LOADING CASE                      MAIN 440
C     MA-MEMBER ACTION INDEX CONTAINING THE NUMBER OF MEMBER ACTIONS    MAIN 450
C        APPLIED TO EACH MEMBER                                         MAIN 460
C     MNUM-MEMBER NUMBER OF MEMBER TO WHICH A MEMBER LOAD IS APPLIED    MAIN 470
C     LDTYP-TYPE OF MEMBER LOAD APPLIED                                 MAIN 480
C     WON, WTW, WTH, WFO-MEMBER LOAD PARAMETERS DEPENDENT ON LOAD TYPE  MAIN 490
C     PSTOR-APPLIED CONCENTRATED LOAD AT EACH DEGREE OF FREEDOM         MAIN 500
C     FLOC-MEMBER FORCES IN LOCAL COORDINATES                           MAIN 510
C     PP-JOINT FORCES IN GLOBAL COORDINATES                             MAIN 520
C     DISP-DISPLACEMENTS FOR EACH DEGREE OF FREEDOM                     MAIN 530
C     XXX-LOCATION OF PANEL POINTS IN EACH MEMBER WHERE INTERNAL MEMBER MAIN 540
C         RESPONSES ARE CALCULATED                                      MAIN 550
C     V-SHEAR FORCE AT EACH PANEL POINT FOR EACH MEMBER                 MAIN 560
C     BM-INTERNAL BENDING MOMENT AT EACH PANEL POINT FOR EACH MEMBER    MAIN 570
C     BMM-MAXIMUM INTERNAL BENDING MOMENT FOR EACH MEMBER               MAIN 580
C     FBALL, FTALL, FCALL-ALLOWABLE BENDING, TENSILE, COMPRESSIVE STRESSMAIN 590
C     SECMOD-SECTION MODULUS FOR EACH MEMBER                            MAIN 600
C     ELM-EFFECTIVE LENGTH OF EACH MEMBER                               MAIN 610
C     FBACT, FAXACT-ACTUAL BENDING, AXIAL STRESS                        MAIN 620
C     SF-SCALING FACTOR FOR EACH MEMBER USED TO MODIFY DESIGN IN DETER- MAIN 630
```

```
C          MINING A FULLY STRESSED DESIGN                                MAIN 640
C      M-SKYLINE VALUES OF MASS MATRIX                                   MAIN 650
C      DD, AZ-EIGENVALUES, EIGENVECTORS OF STRUCTURE                     MAIN 660
C      Z-EIGENVECTORS OF MODIFIED STRUCTURAL SYSTEM                      MAIN 670
C      D, G, TLINV, TILDAK, SSKTIL-MODIFIED MASS-STIFFNESS MATRICES USED MAIN 680
C                                  EIGENVALUE, EIGENVECTOR CALCULATIONS  MAIN 690
C                                                                        MAIN 700
C                                                                        MAIN 710
C      LATEST REVISION - OCTOBER 1978                                    MAIN 720
C                                                                        MAIN 730
C                                                                        MAIN 740
C      ------------------------------------------------------------------MAIN 750
       COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAMAIN 760
      1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),MAIN 770
      2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,MAIN 780
      34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),MAIN 790
      4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),MAIN 800
      5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLMAIN 810
      6C,INFOP,ILDAP,IGENCL,MASTYP                                       MAIN 820
       DIMENSION T(1000)                                                 MAIN 830
C                                                                        MAIN 840
C      READ NUMBER OF DATA SETS                                          MAIN 850
C                                                                        MAIN 860
       READ (5,*) NDS                                                    MAIN 870
       DO 2 JJJ=1,NDS                                                    MAIN 880
C                                                                        MAIN 890
C      READ CALLS FOR INTERNAL FORCES, FULLY STRESSED DESIGNS, ALLOWABLE MAIN 900
C      ITERATIONS AND CALLS FOR PRINTING THE F-TILDA'S AND INTERNAL      MAIN 910
C      MEMBER FORCES                                                     MAIN 915
C                                                                        MAIN 920
       READ (5,*) INFOC,IFSDC,ITTCON,INFOP,ILDAP,IGENCL,MASTYP           MAIN 930
C                                                                        MAIN 940
```

```
C     READ NUMBER OF MEMBERS, NUMBER OF JOINTS, NUMBER OF DEGREES OF    MAIN 950
C     FREEDOM, BANDWIDTH AND NUMBER OF LOADING CONDITIONS               MAIN 960
C                                                                       MAIN 970
      READ (5,*) NM,NJ,NDOF,MB,NLC                                      MAIN 980
      IF (IGENCL.EQ.1) GO TO 1                                          MAIN 990
C                                                                       MAIN1000
C     READ TOTAL NUMBER OF MEMBER ACTIONS IN EACH LOADING CASE          MAIN1010
C                                                                       MAIN1020
      READ (5,*) (NMA(I),I=1,NLC)                                       MAIN1030
C                                                                       MAIN1040
C     SET UP DYNAMIC DIMENSIONING                                       MAIN1050
C                                                                       MAIN1060
1     N1=1                                                              MAIN1070
      N2=1+NDOF*MB                                                      MAIN1080
      N3=1+N2+NDOF                                                      MAIN1090
      N4=1+N3+NM*6*NLC                                                  MAIN1100
      CALL EXEC (T(N1),T(N2),T(N3),T(N4))                               MAIN1110
      WRITE (6,3)                                                       MAIN1120
2     CONTINUE                                                          MAIN1130
      STOP                                                              MAIN1140
C                                                                       MAIN1150
3     FORMAT (1H-)                                                      MAIN1160
      END                                                               MAIN1170
```

```
      SUBROUTINE EXEC (AK,P,FTIL,SM)                                    EXEC  10
C     ------------------------------------------------------------------EXEC  20
C                                                                       EXEC  30
C                                                                       EXEC  40
C     THIS SUBROUTINE IS THE EXECUTIVE SUBROUTINE WHICH CALLS ALL SUB-  EXEC  50
C     ROUTINES SPECIFIED BY USER OPTIONS.  ALSO DISPLACEMENTS AT EACH   EXEC  60
C     DEGREE OF FREEDOM AND MEMBER AND JOINT FORCES FOR EACH LOADING    EXEC  70
C     CONDITION ARE OUTPUT.  TILDA FORCES AND INTERNAL MEMBER RESPONSES EXEC  80
C     MAY ALSO BE OUTPUT.                                               EXEC  90
C                                                                       EXEC 100
C                                                                       EXEC 110
C     ------------------------------------------------------------------EXEC 120
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAEXEC 130
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),EXEC 140
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,EXEC 150
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),EXEC 160
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),EXEC 170
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLEXEC 180
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       EXEC 190
      DIMENSION AK(NDOF,MB), P(NDOF), FTIL(NM,6,NLC), SM(NDOF,MB)       EXEC 200
C                                                                       EXEC 210
C     CALL ALL OTHER SUBROUTINES DEPENDING ON USER FLAGS                EXEC 220
C                                                                       EXEC 230
      CALL INPUT (AK)                                                   EXEC 240
      CALL MPROP                                                        EXEC 250
      IF (IGENCL.EQ.0) GO TO 1                                          EXEC 260
      CALL STIFF (AK)                                                   EXEC 270
      CALL AMASS (AK,P,FTIL,SM)                                         EXEC 280
      CALL EIGEN (AK,P,FTIL,SM)                                         EXEC 290
      IF (IGENCL.EQ.1) GO TO 15                                         EXEC 300
1     CALL MACT (AK,P,FTIL)                                             EXEC 310
      IF (IGENCL.EQ.2) GO TO 2                                          EXEC 320
```

```
      CALL STIFF (AK)                                                   EXEC 330
2     CALL SOLVE (AK,P)                                                 EXEC 340
      CALL FORCE (AK,P,FTIL)                                            EXEC 350
      IF (INFOC.EQ.0) GO TO 3                                           EXEC 360
      CALL INTFOR                                                       EXEC 370
3     IF (IFSDC.EQ.0) GO TO 5                                           EXEC 380
      IF (INFOC.GT.0) GO TO 4                                           EXEC 390
      CALL INTFOR                                                       EXEC 400
4     CALL FSD (AK,P,FTIL)                                              EXEC 410
5     DO 14 II=1,NLC                                                    EXEC 420
      WRITE (6,16) II                                                   EXEC 430
      IF (ILDAP.EQ.0) GO TO 7                                           EXEC 440
      IF (NMA(II).EQ.0) GO TO 7                                         EXEC 450
      WRITE (6,17)                                                      EXEC 460
C                                                                       EXEC 470
C     WRITE THE MEMBER TILDA FORCES FOR EACH LOADING CASE IF REQUESTED  EXEC 480
C                                                                       EXEC 490
      DO 6 I=1,NM                                                       EXEC 500
6     WRITE (6,18) I,(FTIL(I,J,II),J=1,6)                               EXEC 510
7     WRITE (6,19)                                                      EXEC 520
C                                                                       EXEC 530
C     WRITE DISPLACEMENTS AT EACH DEGREE OF FREEDOM FOR EACH LOADING    EXEC 540
C     CASE                                                              EXEC 550
C                                                                       EXEC 560
      DO 8 I=1,NDOF                                                     EXEC 570
      WRITE (6,20) I,DISP(I,II)                                         EXEC 580
8     CONTINUE                                                          EXEC 590
C                                                                       EXEC 600
C     WRITE MEMBER FORCES FOR EACH LOADING CASE                         EXEC 610
C                                                                       EXEC 620
      WRITE (6,21)                                                      EXEC 630
      DO 9 I=1,NM                                                       EXEC 640
```

```
      WRITE (6,22) I,FLOC(I,1,II),FLOC(I,4,II),FLOC(I,2,II),FLOC(I,5,II)EXEC 650
     1,FLOC(I,3,II),FLOC(I,6,II)                                        EXEC 660
9     CONTINUE                                                          EXEC 670
C                                                                       EXEC 680
C     WRITE JOINT FORCES FOR EACH LOADING CASE                          EXEC 690
C                                                                       EXEC 700
      WRITE (6,23)                                                      EXEC 710
      DO 10 I=1,NJ                                                      EXEC 720
      WRITE (6,24) I,PP(3*I-2,II),PP(3*I-1,II),PP(3*I,II)               EXEC 730
10    CONTINUE                                                          EXEC 740
      IF (INFOC.EQ.0) GO TO 11                                          EXEC 750
      GO TO 12                                                          EXEC 760
11    IF (IFSOC.EQ.0) GO TO 14                                          EXEC 770
12    IF (INFOP.EQ.0) GO TO 14                                          EXEC 780
      WRITE (6,25)                                                      EXEC 790
C                                                                       EXEC 800
C     WRITE THE INTERNAL MEMBER RESPONSES, SHEAR AND BENDING MOMENTS,   EXEC 810
C     FOR EACH LOADING CASE IF REQUESTED                                EXEC 820
C                                                                       EXEC 830
      DO 13 M=1,NM                                                      EXEC 840
      WRITE (6,26) M,(XXX(M,I,II),I=1,7)                                EXEC 850
      WRITE (6,27) (V(M,I,II),I=1,7)                                    EXEC 860
      WRITE (6,28) (BM(M,I,II),I=1,7)                                   EXEC 870
13    CONTINUE                                                          EXEC 880
14    CONTINUE                                                          EXEC 890
15    RETURN                                                            EXEC 900
C                                                                       EXEC 910
16    FORMAT (1H0,///,10X,17HLOADING CONDITION,I2)                      EXEC 920
17    FORMAT (//40X,31HTILDE FORCE MATRIX (TRANSPOSED)/)                EXEC 930
18    FORMAT (8X,6HMEMBER,I3,6(5X,F12.5)/)                              EXEC 940
19    FORMAT (1H-,3HDOF,5X,4HDISP)                                      EXEC 950
20    FORMAT (1H0,1X,I2,4X,F10.6)                                       EXEC 960
```

```
21    FORMAT (1H-,6HMEMBER,5X,12HA-END FORCES,5X,12HB-END FORCES)        EXEC 970
22    FORMAT (1H0,2X,I2,7X,F10.4,7X,F10.4,/,11X,F10.4,7X,F10.4,/,11X,F10EXEC 980
     1.4,7X,F10.4)                                                       EXEC 990
23    FORMAT (1H-,5HJOINT,5X,7H1-FORCE,5X,7H2-FORCE,5X,6HMOMENT)          EXEC1000
24    FORMAT (1H0,1X,I2,5X,F9.3,3X,F9.3,3X,F9.3)                          EXEC1010
25    FORMAT (1H0,///,40X,29H**INTERNAL MEMBER RESPONSES**,//)            EXEC1020
26    FORMAT (/,1X,7HMEMBER ,I2,7(4X,2HX=,F8.3),/)                        EXEC1030
27    FORMAT (5X,5HSHEAR,7(2X,E12.4))                                     EXEC1040
28    FORMAT (4X,6HMOMENT,7(2X,E12.4))                                    EXEC1050
      END                                                                 EXEC1060
```

```
      SUBROUTINE INPUT (AK)                                             INPT  10
C     ------------------------------------------------------------------INPT  20
C                                                                       INPT  30
C                                                                       INPT  40
C     THIS SUBROUTINE READS, STORES AND OUTPUTS THE INFORMATION         INPT  50
C     SUPPLIED BY THE USER                                              INPT  60
C                                                                       INPT  70
C                                                                       INPT  80
C     ------------------------------------------------------------------INPT  90
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAINPT 100
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),INPT 110
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,INPT 120
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),INPT 130
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),INPT 140
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLINPT 150
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       INPT 160
      DIMENSION AK(NDOF,MB)                                             INPT 170
C                                                                       INPT 180
C     WRITE THE NUMBER OF MEMBERS AND JOINTS                            INPT 190
C                                                                       INPT 200
      WRITE (6,19) NM,NJ                                                INPT 210
      WRITE (6,20)                                                      INPT 220
      WRITE (6,21)                                                      INPT 230
C                                                                       INPT 240
C     READ JOINT COORDINATES, MEMBER PROPERTIES AND CONNECTIVITY        INPT 250
C                                                                       INPT 260
      READ (5,*) (INC(N,1),INC(N,2),A(N),XI(N),E(N),DENS(N),N=1,NM)     INPT 270
      READ (5,*) (X(J,1),X(J,2),J=1,NJ)                                 INPT 280
      IF (IGENCL.EQ.1) GO TO 5                                          INPT 290
C                                                                       INPT 300
C     READ THE MEMBER ACTION INDEX                                      INPT 310
C                                                                       INPT 320
```

```
      DO 1 I=1,NLC                                                      INPT 330
      READ (5,*) (MA(N,I),N=1,NM)                                       INPT 340
1     CONTINUE                                                          INPT 350
      DO 2 J=1,NLC                                                      INPT 360
      NMALC=NMA(J)                                                      INPT 370
      IF (NMALC.EQ.0) GO TO 2                                           INPT 380
C                                                                       INPT 390
C     READ MEMBER ACTION LOADING AND APPLIED NODAL LOADING              INPT 400
C                                                                       INPT 410
C                                                                       INPT 420
C     FOR LOAD CASE 1 - UNIFORM DISTRIBUTED LOAD                        INPT 430
C     W1, W2 ARE STARTING, ENDING FRACTION OF L FROM A-END              INPT 440
C     W3 IS LOADING VALUE                                               INPT 450
C                                                                       INPT 460
C     FOR LOAD CASE 2 - CONCENTRATED LOAD                               INPT 470
C     W1 IS LOADING VALUE                                               INPT 480
C     W2 IS DISTANCE FROM A-END IN INCHES                               INPT 490
C                                                                       INPT 500
C     FOR LOAD CASE 3 - CONCENTRATED AXIAL LOAD                         INPT 510
C     W1 IS LOADING VALUE                                               INPT 520
C     W2 IS DISTANCE FROM A-END IN INCHES                               INPT 530
C                                                                       INPT 540
C     FOR LOAD CASE 4 - UNIFORM TEMPERATURE INCREASE                    INPT 550
C     W1 IS UNIFORM INCREASE IN TEMPERATURE                             INPT 560
C     W2 IS COEFFICIENT OF THERMAL EXPANSION                            INPT 570
C                                                                       INPT 580
C     FOR LOAD CASE 5 - LINEARLY VARYING DISTRIBUTED LOAD               INPT 590
C     W1, W2 ARE STARTING, ENDING FRACTIONS OF L FROM THE A-END         INPT 600
C     W3, W4 ARE STARTING, ENDING VALUES OF LOAD                        INPT 610
C                                                                       INPT 620
C     FOR LOAD CASE 6 - FABRICATION ERROR                               INPT 630
C     W1 IS INCHES TOO LONG                                             INPT 640
```

```
C                                                                       INPT 650
C     FOR LOAD CASE 7 - LINEARLY VARYING TEMPERATURE INCREASE USED IN   INPT 660
C                       CONJUNCTION WITH LOAD CASE 4                    INPT 670
C     W1 IS THE LINEAR VARIATION IN TEMPERATURE, 0 TO W1                INPT 680
C     W2 IS THE COEFFICIENT OF THERMAL EXPANSION                        INPT 690
C     W3 IS THE DEPTH OF THE SECTION                                    INPT 700
C                                                                       INPT 710
      READ (5,*) (MNUM(I,J),LDTYP(I,J),WON(I,J),WTW(I,J),WTH(I,J),WFO(I,INPT 720
     1J),I=1,NMALC)                                                     INPT 730
2     CONTINUE                                                          INPT 740
C                                                                       INPT 750
C     READ APPLIED LOADS AT EACH DEGREE OF FREEDOM                      INPT 760
C                                                                       INPT 770
      DO 4 K=1,NLC                                                      INPT 780
      READ (5,*) (PSTOR(L,K),L=1,NDOF)                                  INPT 790
      DO 3 J=1,NDOF                                                     INPT 800
3     PDUP(J,K)=PSTOR(J,K)                                              INPT 810
4     CONTINUE                                                          INPT 820
      IF (IFSDC.EQ.0) GO TO 5                                           INPT 830
C                                                                       INPT 840
C     READ THE YIELD STRENGTH OF MEMBER MATERIAL                        INPT 850
C                                                                       INPT 860
      READ (5,*) FY                                                     INPT 870
C                                                                       INPT 880
C     WRITE THE NODAL COORDINATES AND MEMBER PROPERTIES                 INPT 890
C                                                                       INPT 900
5     DO 6 M=1,NJ                                                       INPT 910
      WRITE (6,22) M,X(M,1),X(M,2)                                      INPT 920
6     CONTINUE                                                          INPT 930
      WRITE (6,23)                                                      INPT 940
      WRITE (6,24) (N,INC(N,1),INC(N,2),A(N),XI(N),E(N),DENS(N),N=1,NM) INPT 950
      WRITE (6,25)                                                      INPT 960
```

```
C                                                                       INPT 970
C     READ AND WRITE THE MCODE MATRIX RELATING ELEMENT TO SYSTEM DEGREESINPT 980
C     OF FREEDOM                                                        INPT 990
C                                                                       INPT1000
      DO 7 LL=1,NM                                                      INPT1010
      READ (5,*) (MCODE(LL,MM),MM=1,6)                                  INPT1020
      WRITE (6,26) (MCODE(LL,MM),MM=1,6)                                INPT1030
7     CONTINUE                                                          INPT1040
      IF (IGENCL.EQ.1) GO TO 18                                         INPT1050
      DO 17 L=1,NLC                                                     INPT1060
      WRITE (6,27) L                                                    INPT1070
C                                                                       INPT1080
C     WRITE LOADING INFORMATION FOR EACH LOADING CONDITION              INPT1090
C                                                                       INPT1100
      WRITE (6,28)                                                      INPT1110
      DO 8 I=1,NDOF                                                     INPT1120
      WRITE (6,29) I,PDUP(I,L)                                          INPT1130
8     CONTINUE                                                          INPT1140
      NMALC=NMA(L)                                                      INPT1150
      IF (NMALC.EQ.0) GO TO 17                                          INPT1160
      DO 16 J=1,NMALC                                                   INPT1170
      MNO=MNUM(J,L)                                                     INPT1180
      LODTYP=LDTYP(J,L)                                                 INPT1190
      W1=WON(J,L)                                                       INPT1200
      W2=WTW(J,L)                                                       INPT1210
      W3=WTH(J,L)                                                       INPT1220
      W4=WFO(J,L)                                                       INPT1230
      GO TO (9,10,11,12,13,14,15), LODTYP                               INPT1240
9     WRITE (6,30) MNO,W3,W1,W2                                         INPT1250
      GO TO 16                                                          INPT1260
10    WRITE (6,31) MNO,W1,W2                                            INPT1270
      GO TO 16                                                          INPT1280
```

```
11    WRITE (6,32) MNO,W1,W2                                            INPT1290
      GO TO 16                                                          INPT1300
12    WRITE (6,33) MNO,W1,W2                                            INPT1310
      GO TO 16                                                          INPT1320
13    WRITE (6,34) MNO,W3,W1,W4,W2                                      INPT1330
      GO TO 16                                                          INPT1340
14    WRITE (6,35) MNO,W1                                               INPT1350
      GO TO 16                                                          INPT1360
15    WRITE (6,36) MNO,W1,W2                                            INPT1370
      WRITE (6,37) W3                                                   INPT1380
16    CONTINUE                                                          INPT1390
17    CONTINUE                                                          INPT1400
18    RETURN                                                            INPT1410
C                                                                       INPT1420
19    FORMAT (1H1,18X,I2,29X,I2)                                        INPT1430
20    FORMAT (1H+,17HNUMBER OF MEMBERS,15X,16HNUMBER OF JOINTS,//,1X,35HINPT1440
     1INPUT UNITS:  INCHES, KIPS, RADIANS,///)                          INPT1450
21    FORMAT (1H-,5HJOINT,3X,6HCOOR-1,3X,6HCOOR-2)                      INPT1460
22    FORMAT (1H0,1X,I2,2F10.2,//)                                      INPT1470
23    FORMAT (1H-,6HMEMBER,3X,7HFROM JT,3X,5HTO JT,3X,8HC/S AREA,6X,1HI,INPT1480
     19X,1HE,9X,7HDENSITY)                                              INPT1490
24    FORMAT (1H0,2X,I2,7X,I2,7X,I2,5X,F6.2,4X,F7.2,3X,F8.1,3X,F10.7,///INPT1500
     1)                                                                 INPT1510
25    FORMAT (1H-,12HMCODE MATRIX)                                      INPT1520
26    FORMAT (1H0,6I5)                                                  INPT1530
27    FORMAT (1H0,///,10X,17HLOADING CONDITION,I2)                      INPT1540
28    FORMAT (1H-,3HDOF,5X,7HLOADING)                                   INPT1550
29    FORMAT (1H0,1X,I2,4X,F8.2)                                        INPT1560
30    FORMAT (/20X,6HMEMBER,I2,34H IS SUBJECTED TO A UNIFORM LOAD OF,F8.INPT1570
     15,9HKIPS/INCH/20X,8HSTARTING,F5.2,12HL AND ENDING,F5.2,30HL FROM TINPT1580
     2HE A-END OF THE MEMBER)                                           INPT1590
31    FORMAT (/20X,6HMEMBER,I2,37H IS SUBJECTED TO A TRANSVERSE LOAD OF,INPT1600
```

```
     1F8.3,5H KIPS,F9.5,36H INCHES FROM THE A-END OF THE MEMBER)        INPT1610
32    FORMAT (/20X,6HMEMBER,I2,32HIS SUBJECTED TO AN AXIAL LOAD OF,F8.5,INPT1620
     14HKIPS,F9.5,35HINCHES FROM THE A-END OF THE MEMBER)               INPT1630
33    FORMAT (/20X,6HMEMBER,I2,52HIS SUBJECTED TO A UNIFORM INCREASE IN INPT1640
     1TEMPERATURE OF,F7.4,10HDEGREES F./45X,40HTHE COEFFICIENT OF THERMAINPT1650
     2L EX-PANSION IS,F10.8)                                            INPT1660
34    FORMAT (/20X,6HMEMBER,I2,32HIS SUBJECTED TO A LINEAR LOAD OF,F8.5,INPT1670
     19HKIPS/INCH/2X,8HSTARTING,F5.2,52HL FROM THE A-END OF THE MEMBER AINPT1680
     2ND  VARYING LINERALY,/20X,2HTO,F8.5,23HKIPS/INCH AT A DISTANCE,F5.INPT1690
     32,30HLFROM THE A-END OF THE MEMBER.)                              INPT1700
35    FORMAT (/20X,6HMEMBER,I2,2HIS,F8.5,42HINCHES TOO LONG DUE TO FABRIINPT1710
     1CA-TION ERRORS)                                                   INPT1720
36    FORMAT (/20X,6HMEMBER,I2,52HIS SUBJECTED TO A LINEAR VARIATION IN INPT1730
     1TEMPERATURE OF,F7.4,10HDEGREES F./45X,39HTHE COEFFICIENT OF THERMAINPT1740
     2L EXPANSION IS,F10.8)                                             INPT1750
37    FORMAT (45X,19HDEPTH OF SECTION IS,F9.6,6HINCHES)                  INPT1760
      END                                                               INPT1770
```

```
      SUBROUTINE MPROP                                                  MPRP  10
C     ------------------------------------------------------------------MPRP  20
C                                                                       MPRP  30
C                                                                       MPRP  40
C     THIS SUBROUTINE CALCULATES THE MEMBER PROPERTIES - LENGTH,        MPRP  50
C     ORIENTATION COSINE AND SINE AND THE ALPHA AND BETA PARAMETERS FOR MPRP  60
C     EACH MEMBER                                                       MPRP  65
C                                                                       MPRP  70
C                                                                       MPRP  80
C     ------------------------------------------------------------------MPRP  90
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAMPRP 100
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),MPRP 110
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,MPRP 120
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),MPRP 130
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),MPRP 140
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLMPRP 150
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       MPRP 160
C                                                                       MPRP 170
C     CALCULATE AND STORE MEMBER PROPERTIES                             MPRP 180
C                                                                       MPRP 190
      DO 1 I=1,NM                                                       MPRP 200
      J=INC(I,1)                                                        MPRP 210
      K=INC(I,2)                                                        MPRP 220
      DX1=X(K,1)-X(J,1)                                                 MPRP 230
      DX2=X(K,2)-X(J,2)                                                 MPRP 240
      XLEN=SQRT(DX1**2+DX2**2)                                          MPRP 250
      XL(I)=XLEN                                                        MPRP 260
      C(I)=DX1/XLEN                                                     MPRP 270
      S(I)=DX2/XLEN                                                     MPRP 280
      ALPH(I)=E(I)*XI(I)/XLEN**3                                        MPRP 290
      BETA(I)=A(I)*XLEN**2/XI(I)                                        MPRP 300
1     CONTINUE                                                          MPRP 310
```

```
RETURN                                                    MPRP 320
END                                                       MPRP 330
```

```
      SUBROUTINE MACT (AK,P,FTIL)                                       MACT  10
C     ------------------------------------------------------------------MACT  20
C                                                                       MACT  30
C                                                                       MACT  40
C     THIS SUBROUTINE CALCULATES AND STORES THE MEMBER FORCES RESULTING MACT  50
C     FROM MEMBER LOADS.  THESE TILDA FORCES ARE THEN TRANSFORMED TO    MACT  60
C     GLOBAL COORDINATES AND, IF THEY ARE CONCENTRATED AT SYSTEM DEGREESMACT  70
C     OF FREEDOM, SUBTRACTED FROM ANY APPLIED LOADS CONCENTRATED THERE. MACT  80
C                                                                       MACT  90
C                                                                       MACT 100
C     ------------------------------------------------------------------MACT 110
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAMACT 120
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),MACT 130
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,MACT 140
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),MACT 150
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),MACT 160
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLMACT 170
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       MACT 180
      DIMENSION AK(NDOF,MB), P(NDOF), FTIL(NM,6,NLC), GTIL(10,6)        MACT 190
C                                                                       MACT 200
C     INITIALIZE TILDA FORCE SYSTEM TO ZERO                             MACT 210
C                                                                       MACT 220
      DO 1 I=1,NM                                                       MACT 230
      DO 1 J=1,6                                                        MACT 240
      GTIL(I,J)=0.0                                                     MACT 250
      DO 1 K=1,NLC                                                      MACT 260
      FTIL(I,J,K)=0.0                                                   MACT 270
1     CONTINUE                                                          MACT 280
      DO 11 N=1,NLC                                                     MACT 290
      IF (NMA(N).EQ.0) GO TO 11                                         MACT 300
      K=1                                                               MACT 310
      DO 10 I=1,NM                                                      MACT 320
```

```
      L=MA(I,N)                                                         MACT 330
      IF (L.EQ.0) GO TO 10                                              MACT 340
      M=K+L-1                                                           MACT 350
      KK=K                                                              MACT 360
      DO 9 J=KK,M                                                       MACT 370
      K=K+1                                                             MACT 380
      MNO=MNUM(J,N)                                                     MACT 390
      LODTYP=LDTYP(J,N)                                                 MACT 400
      W1=WON(J,N)                                                       MACT 410
      W2=WTW(J,N)                                                       MACT 420
      W3=WTH(J,N)                                                       MACT 430
      W4=WFO(J,N)                                                       MACT 440
C                                                                       MACT 450
C     DETERMINE TYPE OF MEMBER LOADING                                  MACT 460
C                                                                       MACT 470
      GO TO (2,3,4,5,6,7,8), LODTYP                                     MACT 480
C                                                                       MACT 490
C     CALCULATE AND STORE F-TILDAS                                      MACT 500
C                                                                       MACT 510
C                                                                       MACT 520
C     LOAD TYPE 1 UNIFORM LOAD                                          MACT 530
C                                                                       MACT 540
2     A1=W2**2/2.0                                                      MACT 550
      A2=W2**3/3.0                                                      MACT 560
      A3=W2**4/4.0                                                      MACT 570
      A4=W1**2/2.0                                                      MACT 580
      A5=W1**3/3.0                                                      MACT 590
      A6=W1**4/4.0                                                      MACT 600
      FTIL(MNO,2,N)=FTIL(MNO,2,N)-(W2-A2*3.0+A3*2.0-W1+A5*3.0-A6*2.0)*W3MACT 610
     1*XL(MNO)                                                          MACT 620
      FTIL(MNO,3,N)=FTIL(MNO,3,N)-(A1-A2*2.0+A3-A4+A5*2.0-A6)*W3*XL(MNO)MACT 630
     1**2                                                               MACT 640
```

```
      FTIL(MNO,5,N)=FTIL(MNO,5,N)-(A2*3.0-A3*2.0-A5*3.0+A6*2.0)*W3*XL(MNMACT 650
     10)                                                              MACT 660
      FTIL(MNO,6,N)=FTIL(MNO,6,N)+(A2-A3-A5+A6)*W3*XL(MNO)**2          MACT 670
      GO TO 9                                                         MACT 680
C                                                                     MACT 690
C     LOAD TYPE 2 SINGLE CONCENTRATED LOAD                            MACT 700
C                                                                     MACT 710
3     B=XL(MNO)-W2                                                    MACT 720
      ALI=XL(MNO)                                                     MACT 730
      FTIL(MNO,2,N)=FTIL(MNO,2,N)-W1*B**2/(ALI**3)*(3.0*W2+B)         MACT 740
      FTIL(MNO,3,N)=FTIL(MNO,3,N)-W1*W2*B**2/(ALI**2)                 MACT 750
      FTIL(MNO,5,N)=FTIL(MNO,5,N)-W1*W2**2*(W2+3.0*B)/(ALI**3)        MACT 760
      FTIL(MNO,6,N)=FTIL(MNO,6,N)+W1*B*W2**2/(ALI**2)                 MACT 770
      GO TO 9                                                         MACT 780
C                                                                     MACT 790
C     LOAD TYPE 3 AXIAL LOAD                                          MACT 800
C                                                                     MACT 810
4     FTIL(MNO,1,N)=FTIL(MNO,1,N)-W1*(XL(MNO)-W2)/XL(MNO)             MACT 820
      FTIL(MNO,4,N)=FTIL(MNO,4,N)-W1*W2/XL(MNO)                       MACT 830
      GO TO 9                                                         MACT 840
C                                                                     MACT 850
C     LOAD TYPE 4 UNIFORM INCREASE IN TEMPERATURE                     MACT 860
C                                                                     MACT 870
5     FTIL(MNO,1,N)=FTIL(MNO,1,N)+A(MNO)*E(MNO)*W1*W2                 MACT 880
      FTIL(MNO,4,N)=FTIL(MNO,4,N)-A(MNO)*E(MNO)*W1*W2                 MACT 890
      GO TO 9                                                         MACT 900
C                                                                     MACT 910
C     LOAD TYPE 5 LINEAR LOAD                                         MACT 920
C                                                                     MACT 930
6     XM=(W4-W3)/(W2-W1)                                              MACT 940
      Q1=-XM*W1+W3                                                    MACT 950
      A1=W2**2/2.0                                                    MACT 960
```

```
      A2=W2**3/3.0                                                      MACT 970
      A3=W2**4/4.0                                                      MACT 980
      A4=W1**2/2.0                                                      MACT 990
      A5=W1**3/3.0                                                      MACT1000
      A6=W1**4/4.0                                                      MACT1010
      A7=W2**5/5.0                                                      MACT1020
      A8=W1**5/5.0                                                      MACT1030
      B1=W2-A2*3.0+A3*2.0-W1+A5*3.0-A6*2.0                              MACT1040
      B2=A1-A3*3.0+A7*2.0-A4+A6*3.0-A8*2.0                              MACT1050
      FTIL(MNO,2,N)=FTIL(MNO,2,N)-(Q1*B1+XM*B2)*XL(MNO)                 MACT1060
      B2=A2-A3*2.0+A7-A5+A6*2.0-A8                                      MACT1070
      B1=A1-A2*2.0+A3-A4+A5*2.0-A6                                      MACT1080
      FTIL(MNO,3,N)=FTIL(MNO,3,N)-(Q1*B1+XM*B2)*XL(MNO)**2              MACT1090
      B1=A2*3.0-A3*2.0-A5*3.0+A6*2.0                                    MACT1100
      B2=A3*3.0-A7*2.0-A6*3.0+A8*2.0                                    MACT1110
      FTIL(MNO,5,N)=FTIL(MNO,5,N)-(Q1*B1+XM*B2)*XL(MNO)                 MACT1120
      FTIL(MNO,6,N)=FTIL(MNO,6,N)-(Q1*B1+XM*B2)*XL(MNO)**2              MACT1130
      GO TO 9                                                           MACT1140
C                                                                       MACT1150
C     LOAD TYPE 6  FABRICATION ERROR-- + MEANS MEMBER TOO LONG          MACT1160
C                                                                       MACT1170
7     FTIL(MNO,1,N)=FTIL(MNO,1,N)+A(MNO)*W1*E(MNO)/XL(MNO)              MACT1180
      FTIL(MNO,4,N)=FTIL(MNO,4,N)-A(MNO)*W1*E(MNO)/XL(MNO)              MACT1190
      GO TO 9                                                           MACT1200
C                                                                       MACT1210
C     LOAD TYPE 7 LINEAR VARIATION IN TEMPERATURE--USED WITH TYPE 4     MACT1220
C                                                                       MACT1230
8     FTIL(MNO,3,N)=FTIL(MNO,3,N)+XI(MNO)*E(MNO)*W1*W2/W3               MACT1240
      FTIL(MNO,6,N)=FTIL(MNO,6,N)-XI(MNO)*E(MNO)*W1*W2/W3               MACT1250
9     CONTINUE                                                          MACT1260
10    CONTINUE                                                          MACT1270
11    CONTINUE                                                          MACT1280
```

```
      DO 14 L=1,NLC                                                     MACT1290
      IF (NMA(L).EQ.0) GO TO 14                                         MACT1300
C                                                                       MACT1310
C     TRANSFORM TILDA FORCES TO GLOBAL COORDINATES                      MACT1320
C                                                                       MACT1330
      DO 13 I=1,NM                                                      MACT1340
      IF (MA(I,L).EQ.0) GO TO 13                                        MACT1350
      CI=C(I)                                                           MACT1360
      SI=S(I)                                                           MACT1370
      GTIL(I,1)=CI*FTIL(I,1,L)-SI*FTIL(I,2,L)                           MACT1380
      GTIL(I,2)=SI*FTIL(I,1,L)+CI*FTIL(I,2,L)                           MACT1390
      GTIL(I,3)=FTIL(I,3,L)                                             MACT1400
      GTIL(I,4)=FTIL(I,4,L)*CI-SI*FTIL(I,5,L)                           MACT1410
      GTIL(I,5)=FTIL(I,4,L)*SI+CI*FTIL(I,5,L)                           MACT1420
      GTIL(I,6)=FTIL(I,6,L)                                             MACT1430
C                                                                       MACT1440
C     IF F-TILDA IS CONCENTRATED AT A SYSTEM DEGREE OF FREEDOM SUBTRACT MACT1450
C     IT FROM THE APPLIED LOAD THERE.                                   MACT1460
C                                                                       MACT1470
      DO 12 JJ=1,6                                                      MACT1480
      IF (MCODE(I,JJ).EQ.0) GO TO 12                                    MACT1490
      KK=MCODE(I,JJ)                                                    MACT1500
      PSTOR(KK,L)=PSTOR(KK,L)-GTIL(I,JJ)                                MACT1510
12    CONTINUE                                                          MACT1520
13    CONTINUE                                                          MACT1530
14    CONTINUE                                                          MACT1540
      RETURN                                                            MACT1550
      END                                                               MACT1560
```

```
      SUBROUTINE STIFF (AK)                                             STIF  10
C     ------------------------------------------------------------------STIF  20
C                                                                       STIF  30
C                                                                       STIF  40
C     THIS SUBROUTINE CALCULATES THE STIFFNESS MATRIX OF THE SYSTEM.    STIF  50
C     THE MEMBER STIFFNESS MATRICES ARE CALCULATED AND THEN ASSEMBLED   STIF  60
C     INTO A SYSTEM STIFFNESS MATRIX OF THE SIZE (NDOF,MB).             STIF  70
C                                                                       STIF  80
C                                                                       STIF  90
C     ------------------------------------------------------------------STIF 100
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETASTIF 110
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),STIF 120
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,STIF 130
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),STIF 140
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),STIF 150
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLSTIF 160
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       STIF 170
      DIMENSION AK(NDOF,MB), AA(7), INDEX(6,6)                          STIF 180
      DATA INDEX/1,2,3,-1,-2,3,2,4,5,-2,-4,5,3,5,6,-3,-5,7,-1,-2,-3,1,2,STIF 190
     1-3,-2,-4,-5,2,4,-5,3,5,7,-3,-5,6/                                 STIF 200
C                                                                       STIF 210
C     INITIALIZE STIFFNESS MATRIX VALUES TO ZERO                        STIF 220
C                                                                       STIF 230
      DO 1 J=1,NDOF                                                     STIF 240
      DO 1 K=1,MB                                                       STIF 250
1     AK(J,K)=0.0                                                       STIF 260
      DO 5 I=1,NM                                                       STIF 270
      CI=C(I)                                                           STIF 280
      SI=S(I)                                                           STIF 290
      AI=ALPH(I)                                                        STIF 300
      BI=BETA(I)                                                        STIF 310
      XLI=XL(I)                                                         STIF 320
```

```
C                                                                       STIF 330
C     VALUES WHICH MAKE UP THE INDIVIDUAL MEMBER STIFFNESS MATRICES ARE STIF 340
C     CALCULATED                                                        STIF 350
C                                                                       STIF 360
      AA(1)=AI*(BI*CI**2+12.0*SI**2)                                    STIF 370
      AA(2)=AI*CI*SI*(BI-12.0)                                          STIF 380
      AA(3)=-AI*6.0*XLI*SI                                              STIF 390
      AA(4)=AI*(BI*SI**2+12.0*CI**2)                                    STIF 400
      AA(5)=AI*6.0*XLI*CI                                               STIF 410
      AA(6)=AI*4.0*XLI**2                                               STIF 420
      AA(7)=AA(6)/2.0                                                   STIF 430
C                                                                       STIF 440
C     ASSEMBLE THE STIFFNESS MATRIX                                     STIF 450
C                                                                       STIF 460
      DO 4 JM=1,6                                                       STIF 470
      J=MCODE(I,JM)                                                     STIF 480
      IF (J.EQ.0) GO TO 4                                               STIF 490
      DO 3 KM=JM,6                                                      STIF 500
      K=MCODE(I,KM)                                                     STIF 510
      IF (K.EQ.0) GO TO 3                                               STIF 520
      KB=K-J+1                                                          STIF 530
      L=INDEX(JM,KM)                                                    STIF 540
      IF (L.LT.0) GO TO 2                                               STIF 550
      AK(J,KB)=AK(J,KB)+AA(L)                                           STIF 560
      GO TO 3                                                           STIF 570
2     L=-L                                                              STIF 580
      AK(J,KB)=AK(J,KB)-AA(L)                                           STIF 590
3     CONTINUE                                                          STIF 600
4     CONTINUE                                                          STIF 610
5     CONTINUE                                                          STIF 620
      RETURN                                                            STIF 630
      END                                                               STIF 640
```

```
      SUBROUTINE SOLVE (AK,P)                                           SOLV  10
C     ------------------------------------------------------------------SOLV  20
C                                                                       SOLV  30
C                                                                       SOLV  40
C     THIS SUBROUTINE DECOMPOSES AND TRIANGULARIZES AK AND THEN SOLVES  SOLV  50
C     FOR THE DISPLACEMENTS USING A GAUSSIAN ELIMINATION TECHNIQUE      SOLV  60
C                                                                       SOLV  70
C                                                                       SOLV  80
C     ------------------------------------------------------------------SOLV  90
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETASOLV 100
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),SOLV 110
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,SOLV 120
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),SOLV 130
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),SOLV 140
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLSOLV 150
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       SOLV 160
      DIMENSION AK(NDOF,MB), P(NDOF)                                    SOLV 170
C                                                                       SOLV 180
C     AK IS DECOMPOSED AND TRIANGULARIZED                               SOLV 190
C                                                                       SOLV 200
      NE=NDOF                                                           SOLV 210
      NRS=NE-1                                                          SOLV 220
      DO 2 N=1,NRS                                                      SOLV 230
      M=N-1                                                             SOLV 240
      MR=NE-M                                                           SOLV 250
      IF (MB.LT.MR) MR=MB                                               SOLV 260
      PIVOT=AK(N,1)                                                     SOLV 270
      DO 2 L=2,MR                                                       SOLV 280
      CP=AK(N,L)/PIVOT                                                  SOLV 290
      I=M+L                                                             SOLV 300
      J=0                                                               SOLV 310
      DO 1 K=L,MR                                                       SOLV 320
```

```
      J=J+1                                                             SOLV 330
1     AK(I,J)=AK(I,J)-CP*AK(N,K)                                        SOLV 340
2     AK(N,L)=CP                                                        SOLV 350
C                                                                       SOLV 360
C     BACK SUBSTITUTION IS USED TO SOLVE FOR DISPLACEMENTS              SOLV 370
C                                                                       SOLV 380
      DO 7 II=1,NLC                                                     SOLV 390
      DO 3 JJ=1,NDOF                                                    SOLV 400
3     P(JJ)=PSTOR(JJ,II)                                                SOLV 410
      DO 4 N=1,NRS                                                      SOLV 420
      M=N-1                                                             SOLV 430
      MR=NE-M                                                           SOLV 440
      IF (MB.LT.MR) MR=MB                                               SOLV 450
      CP=P(N)                                                           SOLV 460
      P(N)=CP/AK(N,1)                                                   SOLV 470
      DO 4 L=2,MR                                                       SOLV 480
      I=M+L                                                             SOLV 490
4     P(I)=P(I)-AK(N,L)*CP                                              SOLV 500
      P(NE)=P(NE)/AK(NE,1)                                              SOLV 510
      DO 5 I=1,NRS                                                      SOLV 520
      N=NE-I                                                            SOLV 530
      M=N-1                                                             SOLV 540
      MR=NE-M                                                           SOLV 550
      IF (MB.LT.MR) MR=MB                                               SOLV 560
      DO 5 K=2,MR                                                       SOLV 570
      L=M+K                                                             SOLV 580
5     P(N)=P(N)-AK(N,K)*P(L)                                            SOLV 590
      DO 6 KK=1,NDOF                                                    SOLV 600
6     DISP(KK,II)=P(KK)                                                 SOLV 610
7     CONTINUE                                                          SOLV 620
      RETURN                                                            SOLV 630
      END                                                               SOLV 640
```

```
      SUBROUTINE FORCE (AK,P,FTIL)                                      FORC  10
C     ------------------------------------------------------------------FORC  20
C                                                                       FORC  30
C                                                                       FORC  40
C     THIS SUBROUTINE CALCULATES BOTH MEMBER AND JOINT FORCES.          FORC  50
C                                                                       FORC  60
C                                                                       FORC  70
C     ------------------------------------------------------------------FORC  80
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAFORC  90
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),FORC 100
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,FORC 110
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),FORC 120
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),FORC 130
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLFORC 140
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       FORC 150
      DIMENSION AK(NDOF,MB), P(NDOF), FTIL(NM,6,NLC), DTOT(10,6)        FORC 160
C                                                                       FORC 170
C     INITIALIZE JOINT FORCES TO ZERO                                   FORC 180
C                                                                       FORC 190
      DO 1 N=1,NLC                                                      FORC 200
      NJF=3*NJ                                                          FORC 210
      DO 1 J=1,NJF                                                      FORC 220
1     PP(J,N)=0.0                                                       FORC 230
      DO 6 II=1,NLC                                                     FORC 240
      DO 5 I=1,NM                                                       FORC 250
C                                                                       FORC 260
C     DETERMINE JOINT DISPLACEMENTS                                     FORC 270
C                                                                       FORC 280
      DO 3 L=1,6                                                        FORC 290
      IF (MCODE(I,L).EQ.0) GO TO 2                                      FORC 300
      M=MCODE(I,L)                                                      FORC 310
      DTOT(I,L)=DISP(M,II)                                              FORC 320
```

```
      GO TO 3                                                           FORC 330
2     DTOT(I,L)=0.0                                                     FORC 340
3     CONTINUE                                                          FORC 350
      J=INC(I,1)                                                        FORC 360
      K=INC(I,2)                                                        FORC 370
      J1=3*J-2                                                          FORC 380
      J2=J1+1                                                           FORC 390
      J3=J2+1                                                           FORC 400
      K1=3*K-2                                                          FORC 410
      K2=K1+1                                                           FORC 420
      K3=K2+1                                                           FORC 430
      CI=C(I)                                                           FORC 440
      SI=S(I)                                                           FORC 450
      AI=ALPH(I)                                                        FORC 460
      BI=BETA(I)                                                        FORC 470
      XLEN=XL(I)                                                        FORC 480
C                                                                       FORC 490
C     DISPLACEMENTS ARE TRANSFORMED FROM LOCAL TO GLOBAL COORDINATES    FORC 500
C                                                                       FORC 510
      U1=CI*DTOT(I,1)+SI*DTOT(I,2)                                      FORC 520
      U2=-SI*DTOT(I,1)+CI*DTOT(I,2)                                     FORC 530
      U3=DTOT(I,3)                                                      FORC 540
      U4=CI*DTOT(I,4)+SI*DTOT(I,5)                                      FORC 550
      U5=-SI*DTOT(I,4)+CI*DTOT(I,5)                                     FORC 560
      U6=DTOT(I,6)                                                      FORC 570
C                                                                       FORC 580
C     CALCULATE ELEMENT FORCES                                          FORC 590
C                                                                       FORC 600
      F1=AI*BI*(U1-U4)                                                  FORC 610
      F2=AI*(12.0*U2+6.0*XLEN*U3-12.0*U5+6.0*XLEN*U6)                   FORC 620
      F3=AI*(6.0*XLEN*U2+4.0*XLEN**2*U3-6.0*XLEN*U5+2.0*XLEN**2*U6)     FORC 630
      F4=-F1                                                            FORC 640
```

```
      F5=-F2                                                            FORC 650
      F6=-F3+XLEN*F2                                                    FORC 660
      IF (NMA(II).EQ.0) GO TO 4                                         FORC 670
      F1=F1+FTIL(I,1,II)                                                FORC 680
      F2=F2+FTIL(I,2,II)                                                FORC 690
      F3=F3+FTIL(I,3,II)                                                FORC 700
      F4=F4+FTIL(I,4,II)                                                FORC 710
      F5=F5+FTIL(I,5,II)                                                FORC 720
      F6=F6+FTIL(I,6,II)                                                FORC 730
C                                                                       FORC 740
C     MEMBER FORCES ARE TRANSFORMED FROM LOCAL TO GLOBAL COORDINATES    FORC 750
C                                                                       FORC 760
4     GF1=CI*F1-SI*F2                                                   FORC 770
      GF2=SI*F1+CI*F2                                                   FORC 780
      GF3=F3                                                            FORC 790
      GF4=CI*F4-SI*F5                                                   FORC 800
      GF5=SI*F4+CI*F5                                                   FORC 810
      GF6=F6                                                            FORC 820
C                                                                       FORC 830
C     MEMBER FORCES ARE STORED IN LOCAL COORDINATES                     FORC 840
C                                                                       FORC 850
      FLOC(I,1,II)=F1                                                   FORC 860
      FLOC(I,2,II)=F2                                                   FORC 870
      FLOC(I,3,II)=F3                                                   FORC 880
      FLOC(I,4,II)=F4                                                   FORC 890
      FLOC(I,5,II)=F5                                                   FORC 900
      FLOC(I,6,II)=F6                                                   FORC 910
C                                                                       FORC 920
C     JOINT FORCES ARE CALCULATED                                       FORC 930
C                                                                       FORC 940
      PP(J1,II)=PP(J1,II)+GF1                                           FORC 950
      PP(J2,II)=PP(J2,II)+GF2                                           FORC 960
```

```
      PP(J3,II)=PP(J3,II)+GF3                                   FORC 970
      PP(K1,II)=PP(K1,II)+GF4                                   FORC 980
      PP(K2,II)=PP(K2,II)+GF5                                   FORC 990
      PP(K3,II)=PP(K3,II)+GF6                                   FORC1000
5     CONTINUE                                                  FORC1010
6     CONTINUE                                                  FORC1020
      RETURN                                                    FORC1030
      END                                                       FORC1040
```

```
      SUBROUTINE INTFOR                                                 INFO  10
C     ------------------------------------------------------------------INFO  20
C                                                                       INFO  30
C                                                                       INFO  40
C     THIS SUBROUTINE CALCULATES SHEAR AND MOMENT VALUES AT PANEL POINTSINFO  50
C     ALONG A MEMBER BY APPLICATION OF NEWMARK'S METHOD.                INFO  60
C                                                                       INFO  70
C                                                                       INFO  80
C     ------------------------------------------------------------------INFO  90
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAINFO 100
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),INFO 110
     2FLOC(10,6,4),BM(10,7,4),PSTCR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,INFO 120
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),INFO 130
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),INFO 140
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLINFO 150
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       INFO 160
      DIMENSION Q(9), R(9)                                              INFO 170
      DO 17 KK=1,NLC                                                    INFO 180
      ILOAD=0                                                           INFO 190
C                                                                       INFO 200
C     FOR EACH MEMBER, CONVERT APPLIED MEMBER LOADING TO A SET OF STATICINFO 210
C     EQUIVALENT LOADS AT PANEL POINTS.                                 INFO 220
C                                                                       INFO 230
      DO 16 M=1,NM                                                      INFO 240
C                                                                       INFO 250
C     INITIALIZE EQUIVALENT PANEL POINT LOADS TO ZERO, ESTABLISH HOW    INFO 260
C     MANY MEMBER ACTIONS THE ELEMENTS FEELS.  IF NONE, GO STRAIGHT TO  INFO 270
C     V-M CALCULATIONS                                                  INFO 275
C                                                                       INFO 280
      DO 1 I=1,7                                                        INFO 290
1     R(I)=0.0                                                          INFO 300
      H=XL(M)/6.                                                        INFO 310
```

```
      NLOAD=MA(M,KK)                                                    INFO 320
      IF (NLOAD.EQ.0) GO TO 14                                          INFO 330
C                                                                       INFO 340
C     ASSEMBLE EQUIVALENT PANEL POINT LOADS RESULTING FROM ALL ACTIONS  INFO 350
C     ON THE MEMBER                                                     INFO 360
C                                                                       INFO 370
      DO 13 JJ=1,NLOAD                                                  INFO 380
      ILOAD=ILOAD+1                                                     INFO 390
      ID=LDTYP(ILOAD,KK)                                                INFO 400
      GO TO (4,2,13,13,4,13,13), ID                                     INFO 410
C                                                                       INFO 420
C     CONCENTRATED TRANSVERSE LOADS                                     INFO 430
C                                                                       INFO 440
2     XX=0.0                                                            INFO 450
      II=1                                                              INFO 460
      XLOC=WTW(ILOAD,KK)                                                INFO 470
      XLOAD=WON(ILOAD,KK)                                               INFO 480
C                                                                       INFO 490
C     LOCATE PANEL WHERE LOAD ACTS                                      INFO 500
C                                                                       INFO 510
3     XX=XX+H                                                           INFO 520
      II=II+1                                                           INFO 530
      IF (XLOC.GT.XX) GO TO 3                                           INFO 540
C                                                                       INFO 550
C     REDUCE CONCENTRATED LOAD TO TWO PANEL POINT LOADS                 INFO 560
C                                                                       INFO 570
      CC=XLOC+H-XX                                                      INFO 580
      R(II-1)=R(II-1)+XLOAD*(1.-CC/H)                                   INFO 590
      R(II)=R(II)+XLOAD*CC/H                                            INFO 600
      GO TO 13                                                          INFO 610
C                                                                       INFO 620
C     DISTINGUISH BETWEEN UNIFORM AND LINEAR LOAD                       INFO 630
```

```
C                                                                       INFO 640
4     IF (ID.EQ.1) GO TO 5                                              INFO 650
      XBEGIN=WON(ILOAD,KK)*XL(M)                                        INFO 660
      XEND=WTW(ILOAD,KK)*XL(M)                                          INFO 670
      QBEGIN=WTH(ILOAD,KK)                                              INFO 680
      QEND=WFO(ILOAD,KK)                                                INFO 690
      GO TO 6                                                           INFO 700
5     XBEGIN=WON(ILOAD,KK)*XL(M)                                        INFO 710
      XEND=WTW(ILOAD,KK)*XL(M)                                          INFO 720
      QBEGIN=WTH(ILOAD,KK)                                              INFO 730
      QEND=QBEGIN                                                       INFO 740
C                                                                       INFO 750
C     DISTRIBUTED TRANSVERSE LOAD (UNIFORM OR LINEAR)                   INFO 760
C                                                                       INFO 770
6     SLOPE=(QEND-QBEGIN)/(XEND-XBEGIN)                                 INFO 780
C                                                                       INFO 790
C     LOCATE FIRST PANEL POINT INSIDE LOADED AREA (FROM A-END)          INFO 800
C                                                                       INFO 810
      XX=0.0                                                            INFO 820
      II=1                                                              INFO 830
7     XX=XX+H                                                           INFO 840
      II=II+1                                                           INFO 850
      IF (XBEGIN.GT.XX) GO TO 7                                         INFO 860
      I1=II                                                             INFO 870
      AA=XX-XBEGIN                                                      INFO 880
C                                                                       INFO 890
C     IF LOAD STARTS AND ENDS IN ONE PANEL, HANDLE AS A SEPARATE CASE   INFO 900
C                                                                       INFO 910
      IF (XEND.LT.XX) GO TO 12                                          INFO 920
      Q(I1)=QBEGIN+AA*SLOPE                                             INFO 930
C                                                                       INFO 940
C     LOCATE FIRST PANEL POINT INSIDE LOADED AREA (FROM B-END)          INFO 950
```

```
C                                                                       INFO 960
8     XX=XX+H                                                           INFO 970
      II=II+1                                                           INFO 980
      IF (XEND.GT.XX) GO TO 8                                           INFO 990
      I2=II-1                                                           INFO1000
      BB=XEND+H-XX                                                      INFO1010
C                                                                       INFO1020
C     IF LOAD COVERS ONLY ONE PANEL POINT, HANDLE SPECIALLY             INFO1030
C                                                                       INFO1040
      IF (I1.EQ.I2) GO TO 10                                            INFO1050
C                                                                       INFO1060
C     EVALUATE DISTRIBUTED LOAD AT INTERMEDIATE PANEL POINTS            INFO1070
C                                                                       INFO1080
      II=I1                                                             INFO1090
9     II=II+1                                                           INFO1100
      Q(II)=Q(II-1)+H*SLOPE                                             INFO1110
      IF (II.LT.I2) GO TO 9                                             INFO1120
C                                                                       INFO1130
C     IF THERE ARE PANELS BETWEEN THE FIRST AND LAST PANELS, ADD THEIR  INFO1140
C     EFFECTS TO THE INSIDE PANEL POINT LOAD OF THE OUTER TWO LOADED    INFO1150
C     PANELS                                                            INFO1155
C                                                                       INFO1160
      R(I1)=R(I1)+H/6.*(2.*Q(I1)+Q(I1+1))                               INFO1170
      R(I2)=R(I2)+H/6.*(2.*Q(I2)+Q(I2-1))                               INFO1180
C                                                                       INFO1190
C     COMPUTE EFFECTIVE LOADS AT PANEL POINTS DUE TO TWO OUTSIDE LOADED INFO1200
C                                                                       INFO1210
10    R2=(AA/(2.*H))*((H-2.*AA/3.)*QBEGIN+(H-AA/3.)*Q(I1))              INFO1220
      R1=(QBEGIN+Q(I1))*AA/2.-R2                                        INFO1230
      R3=(BB/(2.*H))*((H-2.*BB/3.)*QEND+(H-AA/3.)*Q(I2))                INFO1240
      R4=(QEND+Q(I2))*BB/2.-R3                                          INFO1250
      R(I1)=R(I1)+R2                                                    INFO1260
```

```
      R(I1-1)=R(I1-1)+R1                                                INF01270
      R(I2)=R(I2)+R3                                                    INF01280
      R(I2+1)=R(I2+1)+R4                                                INF01290
C                                                                       INF01300
C     IF INTERMEDIATE PANEL POINTS BETWEEN "I1" AND "I2" ARE PRESENT,   INF01310
C     CALCULATE THEIR EQUIVALENT LOAD BY RECURSIVE FORMULA              INF01320
C                                                                       INF01330
      IF ((I2-I1).LT.2) GO TO 13                                        INF01340
      I1P1=I1+1                                                         INF01350
      I2M1=I2-1                                                         INF01360
      DO 11 I=I1P1,I2M1                                                 INF01370
11    R(I)=R(I)+H/6.*(Q(I-1)+Q(I+1)+4.*Q(I))                            INF01380
      GO TO 13                                                          INF01390
C                                                                       INF01400
C     HANDLE SPECIAL CASE WHERE LOAD STARTS AND ENDS IN ONE PANEL       INF01410
C                                                                       INF01420
12    AA=XBEGIN+H-XX                                                    INF01430
      BB=XEND-XBEGIN                                                    INF01440
      RI1=BB/(2.*H)*((AA+BB/3.)*QBEGIN+(AA+2.*BB/3.)*QEND)              INF01450
      R(I1)=R(I1)+RI1                                                   INF01460
      R(I1-1)=R(I1-1)+BB/2.*(QBEGIN+QEND)-RI1                           INF01470
13    CONTINUE                                                          INF01480
C                                                                       INF01490
C     COMPUTE SHEAR AND BENDING MOMENT AT PANEL POINTS                  INF01500
C                                                                       INF01510
14    BM(M,1,KK)=(-1.0)*FLOC(M,3,KK)                                    INF01520
      V(M,1,KK)=FLOC(M,2,KK)+R(1)                                       INF01530
      XXX(M,1,KK)=0.0                                                   INF01540
C                                                                       INF01550
C     DETERMINE AND STORE MAXIMUM BENDING MOMENT FOR EACH MEMBER        INF01560
C                                                                       INF01570
      BMM(M,KK)=ABS(BM(M,1,KK))                                         INF01580
```

```
      DO 15 I=2,7                                                       INF01590
      V(M,I,KK)=V(M,I-1,KK)+R(I)                                        INF01600
      BM(M,I,KK)=BM(M,I-1,KK)+V(M,I-1,KK)*H                             INF01610
      CHECK=ABS(BM(M,I,KK))                                             INF01620
      IF (CHECK.GT.BMM(M,KK)) BMM(M,KK)=CHECK                           INF01630
15    XXX(M,I,KK)=XXX(M,I-1,KK)+H                                       INF01640
16    CONTINUE                                                          INF01650
17    CONTINUE                                                          INF01660
      RETURN                                                            INF01670
      END                                                               INF01680
```

```
      SUBROUTINE FSD (AK,P,FTIL)                                        FSDR  10
C     ------------------------------------------------------------------FSDR  20
C                                                                       FSDR  30
C                                                                       FSDR  40
C     THIS SUBROUTINE DETERMINES A FULLY STRESSED DESIGN BASED ON       FSDR  50
C     RESTRICTIONS SET BY THE AISC CODE.  THE DESIGN REACHED MUST BE    FSDR  60
C     WITHIN TWO PERCENT OF THE CODE LIMITATIONS.  AFTER THE DESIGN, OR FSDR  70
C     THE MAXIMUM NUMBER OF ITERATIONS, ITTCON, IS REACHED, MOMENTS OF  FSDR  80
C     INERTIA, AREAS AND SCALING FACTORS ARE OUTPUT FOR EACH MEMBER,    FSDR  90
C     ALONG WITH THE TOTAL WEIGHT OF THE STRUCTURE.                     FSDR 100
C                                                                       FSDR 110
C                                                                       FSDR 120
C     ------------------------------------------------------------------FSDR 130
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAFSDR 140
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),FSDR 150
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,FSDR 160
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),FSDR 170
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),FSDR 180
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLFSDR 190
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       FSDR 200
      DIMENSION AK(NDOF,MB), P(NDOF), FTIL(NM,6,NLC), SF(10)            FSDR 210
C                                                                       FSDR 220
C     DETERMINE ALLOWABLE STRESSES IN TENSION AND BENDING               FSDR 230
C                                                                       FSDR 240
      FBALL=0.66*FY                                                     FSDR 250
      FTALL=0.6*FY                                                      FSDR 260
      ITCONT=0                                                          FSDR 270
C                                                                       FSDR 280
C     CHECK ITERATION COUNT                                             FSDR 290
C                                                                       FSDR 300
1     IF (ITCONT.EQ.ITTCON) GO TO 20                                    FSDR 310
      ITCONT=ITCONT+1                                                   FSDR 320
```

```
C                                                                       FSDR 330
C     INITIALIZE SCALING FACTORS TO ZERO                                FSDR 340
C                                                                       FSDR 350
      DO 2 M=1,NM                                                       FSDR 360
2     SF(M)=0.0                                                         FSDR 370
      DMAX=0.0                                                          FSDR 380
      DO 14 L=1,NLC                                                     FSDR 390
C                                                                       FSDR 400
C     DETERMINE KL/R RATIO FOR EACH MEMBER                              FSDR 410
C                                                                       FSDR 420
      DO 13 I=1,NM                                                      FSDR 430
      IA=MCODE(I,3)                                                     FSDR 440
      IB=MCODE(I,6)                                                     FSDR 450
      IF (IA.EQ.0) GO TO 3                                              FSDR 460
      IF (IB.EQ.0) GO TO 4                                              FSDR 470
      EK=1.0                                                            FSDR 480
      GO TO 6                                                           FSDR 490
3     IF (IB.EQ.0) GO TO 5                                              FSDR 500
4     EK=2.0                                                            FSDR 510
      GO TO 6                                                           FSDR 520
5     EK=1.2                                                            FSDR 530
6     R=SQRT(XI(I)/A(I))                                                FSDR 540
      ELM=EK*XL(I)/R                                                    FSDR 550
C                                                                       FSDR 560
C     DETERMINE ACTUAL STRESSES IN EACH MEMBER                          FSDR 570
C                                                                       FSDR 580
      FAXACT=FLOC(I,1,L)/A(I)                                           FSDR 590
      SECMOD=0.58*XI(I)**0.75                                           FSDR 600
      FBACT=BMM(I,L)/SECMOD                                             FSDR 610
      IF (FAXACT.GT.0.0) GO TO 7                                        FSDR 620
      SFAX=-FAXACT/FTALL                                                FSDR 630
      SFBEN=FBACT/FBALL                                                 FSDR 640
```

```
      SFLC=SFAX+SFBEN                                                   FSDR 650
      GO TO 12                                                          FSDR 660
C                                                                       FSDR 670
C     DETERMINE ALLOWABLE STRESS IN COMPRESSION                         FSDR 680
C                                                                       FSDR 690
7     PI=3.1415926                                                      FSDR 700
      CC=SQRT(2.0*PI**2*E(I)/FY)                                        FSDR 710
      IF (ELM.GT.CC) GO TO 8                                            FSDR 720
      FNUMER=FY*(1.0-ELM**2/2.0/CC**2)                                  FSDR 730
      FDENOM=5.0/3.0+3.0*ELM/8.0/CC-ELM**3/8.0/CC**3                    FSDR 740
      FCALL=FNUMER/FDENOM                                               FSDR 750
      GO TO 9                                                           FSDR 760
8     FCALL=12.0*PI**2*E(I)/(23.0*ELM**2)                               FSDR 770
9     SFAX=FAXACT/FCALL                                                 FSDR 780
      IF (SFAX.LE.0.15) GO TO 11                                        FSDR 790
      CM=0.85                                                           FSDR 800
      FPRIME=12.0*PI**2*E(I)/(23.0*ELM**2)                              FSDR 810
      SFBEN=CM*FBACT/FBALL/(1.0-FAXACT/FPRIME)                          FSDR 820
      SF1=SFAX+SFBEN                                                    FSDR 830
      SF2=FAXACT/(0.6*FY)+FBACT/FBALL                                   FSDR 840
      IF (SF2.GT.SF1) GO TO 10                                          FSDR 850
      SFLC=SF1                                                          FSDR 860
      GO TO 12                                                          FSDR 870
10    SFLC=SF2                                                          FSDR 880
      GO TO 12                                                          FSDR 890
11    SFLC=FAXACT/FCALL+FBACT/FBALL                                     FSDR 900
C                                                                       FSDR 910
C     DETERMINE AND STORE THE LARGEST SCALING FACTOR FOR EACH MEMBER    FSDR 920
C                                                                       FSDR 930
12    IF (SFLC.GT.SF(I)) SF(I)=SFLC                                     FSDR 940
13    CONTINUE                                                          FSDR 950
14    CONTINUE                                                          FSDR 960
```

```
      DO 15 I=1,NM                                                      FSDR 970
      TEST=ABS(SF(I)-0.99)                                              FSDR 980
      IF (TEST.GT.DMAX) DMAX=TEST                                       FSDR 990
15    CONTINUE                                                          FSDR1000
C                                                                       FSDR1010
C     IF SUFFICIENT PRECISION IS REACHED WRITE FSD REACHED AND MEMBER   FSDR1020
C     PROPERTIES                                                        FSDR1030
C                                                                       FSDR1040
      IF (DMAX.LE.0.01) GO TO 21                                        FSDR1050
      TWEIHT=0.0                                                        FSDR1060
C                                                                       FSDR1070
C     IF SUFFICIENT PRECISION IS NOT REACHED, DETERMINE IF THE OVER-    FSDR1080
C     RELAXATION FACTOR SHOULD BE APPLIED                               FSDR1090
C                                                                       FSDR1100
      IF (DMAX.LE.0.05) GO TO 17                                        FSDR1110
C                                                                       FSDR1120
C     APPLY OVER-RELAXATION FACTOR IF APPLICABLE                        FSDR1130
C                                                                       FSDR1140
      DO 16 K=1,NM                                                      FSDR1150
16    SF(K)=SF(K)**1.2                                                  FSDR1160
C                                                                       FSDR1170
C     SCALE MEMBER PROPERTIES AND DETERMINE NEW TOTAL STRUCTURAL WEIGHT FSDR1180
C                                                                       FSDR1190
17    DO 18 J=1,NM                                                      FSDR1200
      XI(J)=SF(J)*XI(J)/0.99                                            FSDR1210
      A(J)=SQRT(XI(J))*0.58                                             FSDR1220
      WEIGHT(J)=XL(J)*A(J)*32.17*DENS(J)*12.0                           FSDR1230
      TWEIHT=TWEIHT+WEIGHT(J)                                           FSDR1240
18    CONTINUE                                                          FSDR1250
      DO 19 JJ=1,NLC                                                    FSDR1260
      DO 19 K=1,NDOF                                                    FSDR1270
19    PSTOR(K,JJ)=PDUP(K,JJ)                                            FSDR1280
```

```
C                                                                       FSDR1290
C     REPEAT ANALYSIS PROCEDURE                                         FSDR1300
C                                                                       FSDR1310
      CALL MPROP                                                        FSDR1320
      CALL MACT (AK,P,FTIL)                                             FSDR1330
      CALL STIFF (AK)                                                   FSDR1340
      CALL SOLVE (AK,P)                                                 FSDR1350
      CALL FORCE (AK,P,FTIL)                                            FSDR1360
      CALL INTFOR                                                       FSDR1370
      GO TO 1                                                           FSDR1380
C                                                                       FSDR1390
C     IF SUFFICIENT PRECISION IS NOT REACHED WRITE FSD NOT REACHED AND  FSDR1400
C     MEMBER PROPERTIES AT FINAL ITERATION                              FSDR1410
C                                                                       FSDR1420
20    WRITE (6,24) ITTCON                                               FSDR1430
      IF (ITCONT.EQ.ITTCON) GO TO 22                                    FSDR1440
21    WRITE (6,25) ITCONT                                               FSDR1450
22    WRITE (6,26)                                                      FSDR1460
      DO 23 M=1,NM                                                      FSDR1470
      WRITE (6,27) M,XI(M),A(M),SF(M)                                   FSDR1480
23    CONTINUE                                                          FSDR1490
      WRITE (6,28) TWEIHT                                               FSDR1500
      RETURN                                                            FSDR1510
C                                                                       FSDR1520
24    FORMAT (1H0,//19H FSD NOT REACHED IN,I3,12H ITERATIONS //72H THE MFSDR1530
     1EMBERPROPERTIES AND SAFETY FACTORS FOR EACH MEMBER ARE AS FOLLOWS:FSDR1540
     2///)                                                              FSDR1550
25    FORMAT (1H0,//,15H FSD REACHED IN,I3,13H ITERATIONS. //80H THE MEMFSDR1560
     1BER  PROPERTIES AND SAFETY FACTOR RATIOS FOR EACH MEMBER ARE AS FOFSDR1570
     2LLOWS:///)                                                        FSDR1580
26    FORMAT (1H0,6HMEMBER,5X,1HI,11X,1HA,11X,2HSF)                     FSDR1590
27    FORMAT (1H0,2X,I2,3X,F10.4,2X,F10.4,2X,F10.4)                     FSDR1600
```

```
28    FORMAT (1H0,5X,15HTOTAL WEIGHT IS,F8.4,5H KIPS)                FSDR1610
      END                                                            FSDR1620
```

```
      SUBROUTINE AMASS (AK,P,FTIL,SM)                                   AMAS  10
C     ------------------------------------------------------------------AMAS  20
C                                                                       AMAS  30
C                                                                       AMAS  40
C     THIS SUBROUTINE DETERMINES THE MASS MATRIX OF THE SYSTEM.  IF A   AMAS  50
C     CONSISTENT MASS MATRIX IS USED, THE MEMBER MASS MATRICES ARE      AMAS  60
C     CALCULATED AND THEN ASSEMBLED INTO A SYSTEM MASS MATRIX OF THE    AMAS  70
C     SIZE (NDOF,MB).  IF A DIAGONAL MASS MATRIX IS USED, SYSTEM VALUES AMAS  80
C     ARE INPUT.                                                        AMAS  90
C                                                                       AMAS 100
C                                                                       AMAS 110
C     ------------------------------------------------------------------AMAS 120
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAAMAS 130
     1(10),SSKTIL(210),WGN(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),AMAS 140
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,AMAS 150
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),AMAS 160
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),AMAS 170
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLAMAS 180
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       AMAS 190
      DIMENSION AK(NDOF,MB), P(NDOF), FTIL(NM,6,NLC), SM(NDOF,MB), NEEDEAMAS 200
     1D(6,6), SS(12)                                                    AMAS 210
      DATA NEEDED/1,2,3,4,5,6,2,7,8,5,9,10,3,8,11,-6,-10,12,4,5,-6,1,2,-AMAS 220
     13,5,9,-10,2,7,-8,6,10,12,-3,-8,11/                                AMAS 230
C                                                                       AMAS 240
C     INITIALIZE MASS MATRIX VALUES TO ZERO                             AMAS 250
C                                                                       AMAS 260
      DO 1 I=1,NDOF                                                     AMAS 270
      DO 1 J=1,MB                                                       AMAS 280
1     SM(I,J)=0.0                                                       AMAS 290
C                                                                       AMAS 300
C     CHECK IF A CONSISTENT OR DIAGONAL MASS MATRIX IS TO BE USED       AMAS 310
C                                                                       AMAS 320
```

```
      IF (MASTYP.EQ.1) GO TO 6                                          AMAS 330
C                                                                       AMAS 340
C     IF USING A CONSISTENT MASS MATRIX, VALUES WHICH MAKE UP THE INDI- AMAS 350
C     VIDUAL MEMBER MASS MATRICES ARE CALCULATED.                       AMAS 360
C                                                                       AMAS 370
      DO 5 I=1,NM                                                       AMAS 380
      XLI=XL(I)                                                         AMAS 390
      CI=C(I)                                                           AMAS 400
      SI=S(I)                                                           AMAS 410
      XMU=A(I)*XLI*DENS(I)                                              AMAS 420
      SS(1)=XMU*(140.0*CI**2-294.0*SI*CI+156.0*SI**2)/420.0             AMAS 430
      SS(2)=XMU*(147.0*CI**2-16.0*SI*CI-147.0*SI**2)/420.0              AMAS 440
      SS(3)=XMU*XLI*(21.0*CI-22.0*SI)/420.0                             AMAS 450
      SS(4)=XMU*(70.0*CI**2-126.0*SI*CI+54.0*SI**2)/420.0               AMAS 460
      SS(5)=XMU*(63.0*CI**2+16.0*SI*CI-63.0*SI**2)/420.0                AMAS 470
      SS(6)=XMU*XLI*(13.0*SI-14.0*CI)/420.0                             AMAS 480
      SS(7)=XMU*(156.0*CI**2+294.0*SI*CI+140.0*SI**2)/420.0             AMAS 490
      SS(8)=XMU*XLI*(21.0*SI+22.0*CI)/420.0                             AMAS 500
      SS(9)=XMU*(54.0*CI**2+126.0*SI*CI+70.0*SI**2)/420.0               AMAS 510
      SS(10)=-XMU*XLI*(14.0*SI+13.0*CI)/420.0                           AMAS 520
      SS(11)=XMU*4.0*XLI**2/420.0                                       AMAS 530
      SS(12)=-XMU*3.0*XLI**2/420.0                                      AMAS 540
C                                                                       AMAS 550
C     ASSEMBLE SYSTEM MASS MATRIX                                       AMAS 560
C                                                                       AMAS 570
      DO 4 JM=1,6                                                       AMAS 580
      J=MCODE(I,JM)                                                     AMAS 590
      IF (J.EQ.0) GO TO 4                                               AMAS 600
      DO 3 KM=JM,6                                                      AMAS 610
      K=MCODE(I,KM)                                                     AMAS 620
      IF (K.EQ.0) GO TO 3                                               AMAS 630
      KB=K-J+1                                                          AMAS 640
```

```
      L=NEEDED(JM,KM)                                                   AMAS 650
      IF (L.LT.0) GO TO 2                                               AMAS 660
      SM(J,KB)=SM(J,KB)+SS(L)                                           AMAS 670
      GO TO 3                                                           AMAS 680
2     L=-L                                                              AMAS 690
      SM(J,KB)=SM(J,KB)-SS(L)                                           AMAS 700
3     CONTINUE                                                          AMAS 710
4     CONTINUE                                                          AMAS 720
5     CONTINUE                                                          AMAS 730
      GO TO 7                                                           AMAS 740
C                                                                       AMAS 750
C     READ SYSTEM MASS MATRIX VALUES IF A DIAGONAL MASS MATRIX IS TO BE AMAS 760
C     USED                                                              AMAS 770
C                                                                       AMAS 780
6     READ (5,*) (SM(I,1),I=1,NDOF)                                     AMAS 790
7     RETURN                                                            AMAS 800
      END                                                               AMAS 810
```

```
      SUBROUTINE EIGEN (AK,P,FTIL,SM)                                   EIGN  10
C     ------------------------------------------------------------------EIGN  20
C                                                                       EIGN  30
C                                                                       EIGN  40
C     THIS SUBROUTINE DETERMINES THE EIGENVALUES AND EIGENVECTORS OF THEEIGN  50
C     GENERALIZED EIGENVALUE PROBLEM INVOLVING THE K AND M MATRICES.    EIGN  60
C     A CHOLESKI DECOMPOSITION IS USED TO TRANSFORM THE MASS MATRIX INTOEIGN  70
C     TWO TRIANGULAR MATRICES, THAT IS, M=L*L-TRANSPOSE.  THEN THE TRI- EIGN  80
C     ANGULAR MATRIX IS INVERTED AND A NEW K-TILDA MATRIX IS FORMED     EIGN  90
C     WHERE K-TILDA=L-INVERSE*AK*L-TRANSPOSE-INVERSE.  THE K-TILDA      EIGN 100
C     MATRIX IS THEN STORED IN THE SYMMETRIC MODE SUITABLE FOR USING THEEIGN 110
C     IMSL SUBROUTINE EIGRS.  THIS SUBROUTINE TRIDIAGONALIZES THE K-    EIGN 120
C     TILDA MATRIX BY A HOUSEHOLDER TRANSFORMATION AND SOLVES FOR THE   EIGN 130
C     EIGENVALUES USING THE QL METHOD.  THE EIGENVECTORS, FOR THE TRI-  EIGN 140
C     DIAGONAL SYSTEM ARE THEN DETERMINED BY INVERSE ITERATION.  THESE  EIGN 150
C     ARE THEN TRANSFORMED INTO THE EIGENVECTORS FOR THE ORIGINAL       EIGN 160
C     SYSTEM.  OTHER REQUIRED SUBROUTINES ARE CHODEC, SYMST AND THE IMSLEIGN 170
C     SUBROUTINE EIGRS WHICH REQUIRES THE OTHER IMSL SUBROUTINES EHOBKS,EIGN 180
C     EHOUSS, EQRT2S, AND UERTST.                                       EIGN 190
C                                                                       EIGN 200
C                                                                       EIGN 210
C     ------------------------------------------------------------------EIGN 220
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETAEIGN 230
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),EIGN 240
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,EIGN 250
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),EIGN 260
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),EIGN 270
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLEIGN 280
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       EIGN 290
      DIMENSION AK(NDOF,MB), P(NDOF), FTIL(NM,6,NLC), SM(NDOF,MB), M(20)EIGN 300
     1, DD(20), Z(400,1), WK(20)                                        EIGN 310
C                                                                       EIGN 320
```

```
C     INITIALIZE THE EIGENVECTORS TO ZERO                                EIGN 330
C                                                                        EIGN 340
      DO 1 I=1,NDOF                                                      EIGN 350
      DO 1 J=1,NDOF                                                      EIGN 360
1     AZ(I,J)=0.0                                                        EIGN 370
      CALL CHODEC (AK,P,FTIL,SM)                                         EIGN 380
      CALL SYMST                                                         EIGN 390
      IJOB=1                                                             EIGN 400
      CALL EIGRS (SSKTIL,NDOF,IJOB,DD,Z,NDOF,WK,IER)                     EIGN 410
      IF (MASTYP.LT.1) GO TO 2                                           EIGN 420
      WRITE (6,8)                                                        EIGN 430
      GO TO 3                                                            EIGN 440
2     WRITE (6,9)                                                        EIGN 450
3     WRITE (6,10)                                                       EIGN 460
C                                                                        EIGN 470
C     WRITE THE EIGENVALUES                                              EIGN 480
C                                                                        EIGN 490
      DO 4 I=1,NDOF                                                      EIGN 500
      WRITE (6,11) DD(I)                                                 EIGN 510
4     CONTINUE                                                           EIGN 520
      IF (IJOB.EQ.0) GO TO 7                                             EIGN 530
C                                                                        EIGN 540
C     DETERMINE THE EIGENVECTORS FOR THE ORIGINAL SYSTEM FROM THE        EIGN 550
C     EIGENVECTORS OF THE K-TILDA SYSTEM                                 EIGN 560
C                                                                        EIGN 570
      DO 5 I=1,NDOF                                                      EIGN 580
      JCOR=0                                                             EIGN 590
      DO 5 K=1,NDOF                                                      EIGN 600
      JCOR=JCOR+1                                                        EIGN 610
      DO 5 J=I,NDOF                                                      EIGN 620
      JJ=J+(JCOR-1)*NDOF                                                 EIGN 630
5     AZ(I,K)=TLINV(I,J)*Z(JJ,1)+AZ(I,K)                                 EIGN 640
```

```
      WRITE (6,12)                                                      EIGN 650
C                                                                       EIGN 660
C     WRITE THE EIGENVECTORS                                            EIGN 670
C                                                                       EIGN 680
      DO 6 I=1,NDOF                                                     EIGN 690
      WRITE (6,13) (AZ(I,J),J=1,NDOF)                                   EIGN 700
6     CONTINUE                                                          EIGN 710
7     RETURN                                                            EIGN 720
C                                                                       EIGN 730
8     FORMAT (1H0,5X,31HA DIAGONAL MASS MATRIX WAS USED)                EIGN 740
9     FORMAT (1H0,5X,33HA CONSISTENT MASS MATRIX WAS USED)              EIGN 750
10    FORMAT (1H0,///,12H EIGENVALUES)                                  EIGN 760
11    FORMAT (1H0,F20.7)                                                EIGN 770
12    FORMAT (1H0,///,13H EIGENVECTORS)                                 EIGN 780
13    FORMAT (1H0,10(F9.4,3X))                                          EIGN 790
      END                                                               EIGN 800
```

```
      SUBROUTINE CHODEC (AK,P,FTIL,SM)                                  CHOD  10
C     ------------------------------------------------------------------CHOD  20
C                                                                       CHOD  30
C                                                                       CHOD  40
C     THIS SUBROUTINE DETERMINES THE SKYLINE VALUES OF THE MASS MATRIX. CHOD  50
C     A CHOLESKI DECOMPOSITION IS THEN PERFORMED.  THE RESULTING        CHOD  60
C     TRIANGULAR MATRIX L IS INVERTED AND THE K-TILDA MATRIX IS FORMED. CHOD  70
C                                                                       CHOD  80
C                                                                       CHOD  90
C     ------------------------------------------------------------------CHOD 100
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETACHOD 110
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),CHOD 120
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,CHOD 130
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),CHOD 140
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),CHOD 150
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLCHOD 160
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       CHOD 170
      DIMENSION AK(NDOF,MB), P(NDOF), FTIL(NM,6,NLC), SM(NDOF,MB), M(20)CHOD 180
     1, PARTK(20)                                                       CHOD 190
C                                                                       CHOD 200
C     INITIALIZE K-TILDA AND L VALUES TO ZERO                           CHOD 210
C                                                                       CHOD 220
      DO 1 I=1,NDOF                                                     CHOD 230
      DO 1 J=1,NDOF                                                     CHOD 240
      D(I,J)=0.0                                                        CHOD 250
      TILDAK(I,J)=0.0                                                   CHOD 260
      TLINV(I,J)=0.0                                                    CHOD 270
1     G(I,J)=0.0                                                        CHOD 280
C                                                                       CHOD 290
C     INITIALIZE SKYLINE VALUES TO THE MAXIMUM POSSIBLE, NDOF           CHOD 300
C                                                                       CHOD 310
      DO 2 L=1,NDOF                                                     CHOD 320
```

```
2     M(L)=NDOF                                                CHOD 330
C                                                              CHOD 340
C     DETERMINE SKYLINE VALUES                                 CHOD 350
C                                                              CHOD 360
      DO 5 I=1,NM                                              CHOD 370
      LEAST=NDOF                                               CHOD 380
      DO 3 J=1,6                                               CHOD 390
      K=MCODE(I,J)                                             CHOD 400
      IF (K.EQ.0) GO TO 3                                      CHOD 410
      IF (K.LT.LEAST) LEAST=K                                  CHOD 420
3     CONTINUE                                                 CHOD 430
      DO 4 JJ=1,6                                              CHOD 440
      KK=MCODE(I,JJ)                                           CHOD 450
      IF (KK.EQ.0) GO TO 4                                     CHOD 460
      MM=M(KK)                                                 CHOD 470
      IF (LEAST.LT.MM) MM=LEAST                                CHOD 480
      M(KK)=MM                                                 CHOD 490
4     CONTINUE                                                 CHOD 500
5     CONTINUE                                                 CHOD 510
C                                                              CHOD 520
C     PERFORM A CHOLESKI DECOMPOSITION                         CHOD 530
C                                                              CHOD 540
      D(1,1)=SM(1,1)                                           CHOD 550
      IF (NDOF.EQ.1) GO TO 11                                  CHOD 560
      DO 10 J=2,NDOF                                           CHOD 570
      K=M(J)                                                   CHOD 580
      MCC=K-1                                                  CHOD 590
      JN=J-MCC                                                 CHOD 600
      G(K,J)=SM(K,JN)                                          CHOD 610
      L=J-1                                                    CHOD 620
      IF (J.EQ.2) GO TO 8                                      CHOD 630
      I=K+1                                                    CHOD 640
```

```
      DO 7 MI=I,L                                                       CHOD 650
      SUBG=0.0                                                          CHOD 660
      IBR=M(MI)                                                         CHOD 670
      IF (K.GT.IBR) IBR=K                                               CHOD 680
      IER=MI-1                                                          CHOD 690
      DO 6 MR=IBR,IER                                                   CHOD 700
6     SUBG=SUBG+D(MR,MI)*G(MR,J)                                        CHOD 710
      MCC=MI-1                                                          CHOD 720
      JN=J-MCC                                                          CHOD 730
7     G(MI,J)=SM(MI,JN)-SUBG                                            CHOD 740
8     SUBD=0.0                                                          CHOD 750
      DO 9 LR=K,L                                                       CHOD 760
      D(LR,J)=G(LR,J)/D(LR,LR)                                          CHOD 770
9     SUBD=SUBD+D(LR,J)*G(LR,J)                                         CHOD 780
10    D(J,J)=SM(J,1)-SUBD                                               CHOD 790
11    DO 13 J=1,NDOF                                                    CHOD 800
      D(J,J)=SQRT(D(J,J))                                               CHOD 810
      IF (J.EQ.NDOF) GO TO 13                                           CHOD 820
      KB=J+1                                                            CHOD 830
      DO 12 K=KB,NDOF                                                   CHOD 840
12    D(J,K)=D(J,K)*D(J,J)                                              CHOD 850
C                                                                       CHOD 860
C     D IS NOW THE TRIANGULAR MATRIX L                                  CHOD 870
C                                                                       CHOD 880
13    CONTINUE                                                          CHOD 890
C                                                                       CHOD 900
C     INVERT THE TRIANGULAR MATRIX L (D)                                CHOD 910
C                                                                       CHOD 920
      DO 18 I=1,NDOF                                                    CHOD 930
      IF (I-1) 17,17,14                                                 CHOD 940
14    ILSS=I-1                                                          CHOD 950
      DO 16 K=1,ILSS                                                    CHOD 960
```

```
      SUM=0.0                                                           CHOD 970
      DO 15 J=K,ILSS                                                    CHOD 980
      SUM=SUM+D(J,I)*TLINV(K,J)                                         CHOD 990
15    CONTINUE                                                          CHOD1000
      TLINV(K,I)=-SUM/D(I,I)                                            CHOD1010
16    CONTINUE                                                          CHOD1020
17    TLINV(I,I)=1.0/D(I,I)                                             CHOD1030
18    CONTINUE                                                          CHOD1040
C                                                                       CHOD1050
C     DETERMINE THE K-TILDA MATRIX                                      CHOD1060
C                                                                       CHOD1070
      DO 22 I=1,NDOF                                                    CHOD1080
      DO 20 K=1,NDOF                                                    CHOD1090
      PARTK(K)=0.0                                                      CHOD1100
      DO 20 J=1,I                                                       CHOD1110
      MCC=J-1                                                           CHOD1120
      KN=K-MCC                                                          CHOD1130
      IF (J.GT.K) GO TO 19                                              CHOD1140
      IF (KN.GT.MB) GO TO 20                                            CHOD1150
      PARTK(K)=TLINV(J,I)*AK(J,KN)+PARTK(K)                             CHOD1160
      GO TO 20                                                          CHOD1170
19    MC=K-1                                                            CHOD1180
      JN=J-MC                                                           CHOD1190
      IF (JN.GT.MB) GO TO 20                                            CHOD1200
      PARTK(K)=TLINV(J,I)*AK(K,JN)+PARTK(K)                             CHOD1210
20    CONTINUE                                                          CHOD1220
      DO 21 KK=1,NDOF                                                   CHOD1230
      DO 21 JJ=1,KK                                                     CHOD1240
21    TILDAK(I,KK)=PARTK(JJ)*TLINV(JJ,KK)+TILDAK(I,KK)                  CHOD1250
22    CONTINUE                                                          CHOD1260
      RETURN                                                            CHOD1270
      END                                                               CHOD1280
```

```
      SUBROUTINE SYMST                                                  SYMS  10
C     ------------------------------------------------------------------SYMS  20
C                                                                       SYMS  30
C                                                                       SYMS  40
C     THIS SUBROUTINE TRANSFORMS THE K-TILDA MATRIX TO THE SYMMETRIC    SYMS  50
C     STORAGE MODE USED BY THE IMSL SUBROUTINE EIGRS.                   SYMS  60
C                                                                       SYMS  70
C                                                                       SYMS  80
C     ------------------------------------------------------------------SYMS  90
      COMMON X(10,2),A(10),XI(10),E(10),XL(10),C(10),S(10),ALPH(10),BETASYMS 100
     1(10),SSKTIL(210),WON(10,4),WTW(10,4),WTH(10,4),WFO(10,4),PP(30,4),SYMS 110
     2FLOC(10,6,4),BM(10,7,4),PSTOR(20,4),PDUP(20,4),BMM(10,4),XXX(10,7,SYMS 120
     34),V(10,7,4),DISP(20,4),WEIGHT(10),D(20,20),G(20,20),TLINV(20,20),SYMS 130
     4TILDAK(20,20),AZ(20,20),DENS(10),MCODE(10,6),INC(10,2),MNUM(10,4),SYMS 140
     5MA(10,4),LDTYP(10,4),NMA(4),FY,NM,NJ,NDOF,MB,INFOC,IFSDC,ITTCON,NLSYMS 150
     6C,INFOP,ILDAP,IGENCL,MASTYP                                       SYMS 160
      K=1                                                               SYMS 170
      DO 2 I=1,NDOF                                                     SYMS 180
      DO 1 J=1,I                                                        SYMS 190
      SSKTIL(K)=TILDAK(J,I)                                             SYMS 200
      K=K+1                                                             SYMS 210
1     CONTINUE                                                          SYMS 220
2     CONTINUE                                                          SYMS 230
      RETURN                                                            SYMS 240
      END                                                               SYMS 250
```

```
      SUBROUTINE EIGRS (A,N,IJOB,D,Z,IZ,WK,IER)                         EIRS  10
C     ------------------------------------------------------------------EIRS  20
C                                                                       EIRS  30
C                                                                       EIRS  40
C      THIS SUBROUTINE AND ALL REMAINING SUBROUTINES ARE USED IN THE    EIRS  50
C     CALCULATIONS OF EIGENVALUES AND EIGENVECTORS.  THEY ARE A PART OF EIRS  60
C     THE INTERNATIONAL MATHEMATICAL AND STATISTICAL LIBRARIES, INC.    EIRS  70
C     (IMSL) IN HOUSTON, TEXAS.  THEY ARE REPRODUCED HERE WITH THE      EIRS  80
C     PERMISSION OF IMSL AND MAY NOT BE EXTRACTED AS A BASIS FOR ANY    EIRS  90
C     SOFTWARE DEVELOPMENT.                                             EIRS 100
C                                                                       EIRS 110
C                                                                       EIRS 120
C     -EIGRS---------S/D-----LIBRARY 1----------------------------------EIRS 130
C     FUNCTION             - TO CALCULATE EIGENVALUES AND (OPTIONALLY)  EIRS 140
C                            EIGENVECTORS OF A REAL SYMMETRIC MATRIX.   EIRS 150
C     USAGE                - CALL EIGRS (A,N,IJOB,D,Z,IZ,WK,IER)      EIRS 160
C     PARAMETERS   A       - THE INPUT REAL SYMMETRIC MATRIX OF ORDER N, EIRS 170
C                            STORED IN SYMMETRIC STORAGE MODE,          EIRS 180
C                            WHOSE EIGENVALUES AND EIGENVECTORS         EIRS 190
C                            ARE TO BE COMPUTED. INPUT A IS             EIRS 200
C                            DESTROYED IF IJOB IS EQUAL TO 0 OR 1.      EIRS 210
C                N       - THE ORDER OF THE MATRIX A.(INPUT)            EIRS 220
C                IJOB    - INPUT OPTION PARAMETER, WHEN                 EIRS 230
C                            IJOB = 0, COMPUTE EIGENVALUES ONLY         EIRS 240
C                            IJOB = 1, COMPUTE EIGENVALUES AND EIGEN-   EIRS 250
C                              VECTORS.                                 EIRS 260
C                            IJOB = 2, COMPUTE EIGENVALUES, EIGENVECTORS EIRS 270
C                              AND PERFORMANCE INDEX.                   EIRS 280
C                            IJOB = 3, COMPUTE PERFORMANCE INDEX ONLY.  EIRS 290
C                            IF THE PERFORMANCE INDEX IS COMPUTED, IT IS EIRS 300
C                            RETURNED IN WK(1). THE ROUTINES HAVE       EIRS 310
C                            PERFORMED (WELL, SATISFACTORILY, POORLY) IF EIRS 320
```

```
C                 WK(1) IS (LESS THAN 1, BETWEEN 1 AND 100,    EIRS 330
C                 GREATER THAN 100).                           EIRS 340
C         D     - THE OUTPUT VECTOR OF LENGTH N,               EIRS 350
C                 CONTAINING THE EIGENVALUES OF A.             EIRS 360
C         Z     - THE OUTPUT N BY N MATRIX CONTAINING          EIRS 370
C                 THE EIGENVECTORS OF A.                       EIRS 380
C                 THE EIGENVECTOR IN COLUMN J OF Z CORRES-     EIRS 390
C                 PONDS TO THE EIGENVALUE D(J).                EIRS 400
C                 IF IJOB = 0, Z IS NOT USED.                  EIRS 410
C         IZ    - THE ROW DIMENSION OF THE MATRIX Z IN THE     EIRS 420
C                 CALLING PROGRAM. IZ MUST BE GREATER THAN     EIRS 430
C                 OR EQUAL TO N IF IJOB IS NOT EQUAL TO ZERO.  EIRS 440
C         WK    - WORK AREA, THE LENGTH OF WK DEPENDS          EIRS 450
C                 ON THE VALUE OF IJOB, WHEN                   EIRS 460
C                 IJOB = 0, THE LENGTH OF WK IS AT LEAST N.    EIRS 470
C                 IJOB = 1, THE LENGTH OF WK IS AT LEAST N.    EIRS 480
C                 IJOB = 2, THE LENGTH OF WK IS AT LEAST       EIRS 490
C                   N(N+1)/2+N.                                EIRS 500
C                 IJOB = 3, THE LENGTH OF WK IS AT LEAST 1.    EIRS 510
C         IER   - ERROR PARAMETER                              EIRS 520
C                 TERMINAL ERROR                               EIRS 530
C                 IER = 128+J, INDICATES THAT EQRT2S FAILED    EIRS 540
C                   TO CONVERGE ON EIGENVALUE J. EIGENVALUES   EIRS 550
C                   AND EIGENVECTORS 1,...,J-1 HAVE BEEN       EIRS 560
C                   COMPUTED CORRECTLY, BUT THE EIGENVALUES    EIRS 570
C                   ARE UNORDERED. THE PERFORMANCE INDEX       EIRS 580
C                   IS SET TO 1000.0                           EIRS 590
C                 WARNING ERROR (WITH FIX)                     EIRS 600
C                 IER = 66, INDICATES IJOB IS LESS THAN 0 OR   EIRS 610
C                   IJOB IS GREATER THAN 3. IJOB SET TO 1.     EIRS 620
C                 IER = 67, INDICATES IJOB IS NOT EQUAL TO     EIRS 630
C                   ZERO, AND IZ IS LESS THAN THE ORDER OF     EIRS 640
```

```
C                                MATRIX A. IJOB IS SET TO ZERO.                EIRS 650
C     PRECISION            - SINGLE/DOUBLE                                     EIRS 660
C     REQD. IMSL ROUTINES - EHOBKS,EHOCSS,EQRT2S,UERTST                        EIRS 670
C     LANGUAGE             - FORTRAN                                           EIRS 680
C     ------------------------------------------------------------------------EIRS 690
C     LATEST REVISION      - MARCH 9, 1977                                     EIRS 700
C                                                                              EIRS 710
C                                                                              EIRS 720
      DIMENSION A(1), D(1), WK(1), Z(IZ,1)                                     EIRS 730
C     DOUBLE PRECISION   A,D,WK,Z,ANORM,ASUM,PI,SUMZ,SUMR,AN,S                 EIRS 740
C     DOUBLE PRECISION   TEN,RDELP,ZERO,ONE,THOUS                              EIRS 750
C     DATA               RDELP/Z3410000000000000/                              EIRS 760
C     2    DATA                ZERO,ONE,TEN,THOUS/0.0D0,1.0D0,10.0D0,1000EIRS 770
      DATA RDELP/Z3C100000/                                                    EIRS 780
      DATA ZERO,ONE/0.0,1.0/,TEN/10.0/,THOUS/1000.0/                           EIRS 790
C                                     INITIALIZE ERROR PARAMETERS              EIRS 800
      IER=0                                                                    EIRS 810
      JER=0                                                                    EIRS 820
      IF (IJOB.GE.0.AND.IJOB.LE.3) GO TO 1                                     EIRS 830
C                                     WARNING ERROR - IJOB IS NOT IN THE       EIRS 840
C                                       RANGE                                  EIRS 850
      IER=66                                                                   EIRS 860
      IJOB=1                                                                   EIRS 870
      GO TO 2                                                                  EIRS 880
1     IF (IJOB.EQ.0) GO TO 4                                                   EIRS 890
2     IF (IZ.GE.N) GO TO 3                                                     EIRS 900
C                                     WARNING ERROR - IZ IS LESS THAN N        EIRS 910
C                                       EIGENVECTORS CAN NOT BE COMPUTED,      EIRS 920
C                                       IJOB SET TO ZERO                       EIRS 930
      IER=67                                                                   EIRS 940
      IJOB=0                                                                   EIRS 950
3     IF (IJOB.EQ.3) GO TO 12                                                  EIRS 960
```

```
4     NA=(N*(N+1))/2                                                    EIRS 970
      IF (IJOB.NE.2) GO TO 6                                            EIRS 980
      DO 5 I=1,NA                                                       EIRS 990
      WK(I)=A(I)                                                        EIRS1000
5     CONTINUE                                                          EIRS1010
C                                   SAVE INPUT A IF IJOB = 2            EIRS1020
6     ND=1                                                              EIRS1030
      IF (IJOB.EQ.2) ND=NA+1                                            EIRS1040
C                                   REDUCE A TO SYMMETRIC TRIDIAGONAL   EIRS1050
C                                     FORM                              EIRS1060
      CALL EHOUSS (A,N,D,WK(ND),WK(ND))                                 EIRS1070
      IIZ=0                                                             EIRS1080
      IF (IJOB.GT.0) IIZ=IZ                                             EIRS1090
      IF (IIZ.EQ.0) GO TO 9                                             EIRS1100
C                                   SET Z TO THE IDENTITY MATRIX        EIRS1110
      DO 8 I=1,N                                                        EIRS1120
      DO 7 J=1,N                                                        EIRS1130
      Z(I,J)=ZERO                                                       EIRS1140
7     CONTINUE                                                          EIRS1150
      Z(I,I)=ONE                                                        EIRS1160
8     CONTINUE                                                          EIRS1170
C                                   COMPUTE EIGENVALUES AND EIGENVECTORS EIRS1180
9     CALL EQRT2S (D,WK(ND),N,Z,IIZ,JER)                                EIRS1190
      IF (IJOB.EQ.0) GO TO 18                                           EIRS1200
      IF (JER.GT.128) GO TO 10                                          EIRS1210
C                                   BACK TRANSFORM EIGENVECTORS         EIRS1220
      CALL EHOBKS (A,N,1,N,Z,IZ)                                        EIRS1230
10    IF (IJOB.LE.1) GO TO 18                                           EIRS1240
C                                   MOVE INPUT MATRIX BACK TO A         EIRS1250
      DO 11 I=1,NA                                                      EIRS1260
      A(I)=WK(I)                                                        EIRS1270
11    CONTINUE                                                          EIRS1280
```

```
      WK(1)=THOUS                                                       EIRS1290
      IF (JER.NE.0) GO TO 18                                            EIRS1300
C                                   COMPUTE 1 - NORM OF A               EIRS1310
12    ANORM=ZERO                                                        EIRS1320
      IBEG=1                                                            EIRS1330
      DO 14 I=1,N                                                       EIRS1340
      ASUM=ZERO                                                         EIRS1350
      IL=IBEG                                                           EIRS1360
      KK=1                                                              EIRS1370
      DO 13 L=1,N                                                       EIRS1380
C     1           ASUM =ASUM+DABS(A(IL))                                EIRS1390
      ASUM=ASUM+ABS(A(IL))                                              EIRS1400
      IF (L.GE.I) KK=L                                                  EIRS1410
      IL=IL+KK                                                          EIRS1420
13    CONTINUE                                                          EIRS1430
C     1        ANORM = DMAX1(ANORM,ASUM)                                EIRS1440
      ANORM=AMAX1(ANORM,ASUM)                                           EIRS1450
      IBEG=IBEG+I                                                       EIRS1460
14    CONTINUE                                                          EIRS1470
      IF (ANORM.EQ.ZERO) ANORM=ONE                                      EIRS1480
C                                   COMPUTE PERFORMANCE INDEX           EIRS1490
      PI=ZERO                                                           EIRS1500
      DO 17 I=1,N                                                       EIRS1510
      IBEG=1                                                            EIRS1520
      S=ZERO                                                            EIRS1530
      SUMZ=ZERO                                                         EIRS1540
      DO 16 L=1,N                                                       EIRS1550
      LK=IBEG                                                           EIRS1560
      KK=1                                                              EIRS1570
C     1           SUMZ = SUMZ+DABS(Z(L,I))                              EIRS1580
      SUMZ=SUMZ+ABS(Z(L,I))                                             EIRS1590
      SUMR=-D(I)*Z(L,I)                                                 EIRS1600
```

```
      DO 15 K=1,N                                                       EIRS1610
      SUMR=SUMR+A(LK)*Z(K,I)                                            EIRS1620
      IF (K.GE.L) KK=K                                                  EIRS1630
      LK=LK+KK                                                          EIRS1640
15    CONTINUE                                                          EIRS1650
C     1           S = S+DABS(SUMR)                                      EIRS1660
      S=S+ABS(SUMR)                                                     EIRS1670
      IBEG=IBEG+L                                                       EIRS1680
16    CONTINUE                                                          EIRS1690
      IF (SUMZ.EQ.ZERO) GO TO 17                                        EIRS1700
C     1        PI = DMAX1(PI,S/SUMZ)                                    EIRS1710
      PI=AMAX1(PI,S/SUMZ)                                               EIRS1720
17    CONTINUE                                                          EIRS1730
      AN=N                                                              EIRS1740
      PI=PI/(ANORM*TEN*AN*RDELP)                                        EIRS1750
      WK(1)=PI                                                          EIRS1760
18    CONTINUE                                                          EIRS1770
      IF (IER.NE.0) CALL UERTST (IER,6HEIGRS )                          EIRS1780
      IF (JER.EQ.0) GO TO 19                                            EIRS1790
      IER=JER                                                           EIRS1800
      CALL UERTST (IER,6HEIGRS )                                        EIRS1810
19    RETURN                                                            EIRS1820
      END                                                               EIRS1830
```

```
      SUBROUTINE EHOBKS (A,N,M1,M2,Z,IZ)                                EHOB  10
C     -EHOBKS--------S/D-----LIBRARY 1-----------------------------------EHOB  20
C                                                                       EHOB  30
C     FUNCTION              - PERFORM A BACK TRANSFORMATION TO FORM THE EHOB  40
C                             EIGENVECTORS OF THE ORIGINAL SYMMETRIC    EHOB  50
C                             MATRIX FROM THE EIGENVECTORS OF THE       EHOB  60
C                             TRIDIAGONAL MATRIX.                       EHOB  70
C     USAGE                 - CALL EHOBKS (A,N,M1,M2,Z,IZ)              EHOB  80
C     PARAMETERS   A        - THE ARRAY CONTAINS THE DETAILS OF THE HOUSE EHOB  90
C                             HOLDER REDUCTION OF THE ORIGINAL MATRIX A ASEHOB 100
C                             GENERATED BY IMSL ROUTINE 'EHOUSS'.       EHOB 110
C                N      - ORDER OF THE REAL SYMMETRIC MATRIX.           EHOB 120
C                M1     - M1 AND M2 ARE TWO INPUT SCALARS SUCH THAT     EHOB 130
C                         EIGENVECTORS M1 TO M2 OF THE TRIDIAGONAL      EHOB 140
C                         MATRIX A HAVE BEEN FOUND AND NORMALIZED       EHOB 150
C                         ACCORDING TO THE EUCLIDEAN NORM.              EHOB 160
C                M2     - SEE ABOVE - M1                                EHOB 170
C                Z      - A TWO DIMENSIONAL ARRAY OF SIZE N X (M2-M1+1) EHOB 180
C                         WHICH CONTAINS EIGENVECTORS M1 TO M2 OF       EHOB 190
C                         TRIDIAGONAL MATRIX T, NORMALIZED ACCORDING    EHOB 200
C                         TO EUCLIDEAN NORM. INPUT Z CAN BE PRODUCED    EHOB 210
C                         BY IMSL ROUTINE 'EQRT2S', THE RESULTANT       EHOB 220
C                         MATRIX OVERWRITES THE INPUT Z.                EHOB 230
C                IZ     - ROW DIMENSION OF Z IN CALLING PROGRAM.        EHOB 240
C     PRECISION             - SINGLE/DOUBLE                             EHOB 250
C     LANGUAGE              - FORTRAN                                   EHOB 260
C     ------------------------------------------------------------------EHOB 270
C     LATEST REVISION       - JULY 21, 1972                             EHOB 280
C                                                                       EHOB 290
C                                                                       EHOB 300
      DIMENSION A(1), Z(IZ,1)                                           EHOB 310
C     DOUBLE PRECISION   A,Z,H,S                                        EHOB 320
```

```
      IF (N.EQ.1) GO TO 5                                               EHOB 330
      DO 4 I=2,N                                                        EHOB 340
      L=I-1                                                             EHOB 350
      IA=I*L/2                                                          EHOB 360
      H=A(IA+I)                                                         EHOB 370
      IF (H.EQ.0.) GO TO 4                                              EHOB 380
C                                 DERIVES EIGENVECTORS M1 TO M2 OF      EHOB 390
C                                 THE ORIGINAL MATRIX FROM EIGENVECTORSEHOB 400
C                                 M1 TO M2 OF THE SYMMETRIC             EHOB 410
C                                 TRIDIAGONAL MATRIX                    EHOB 420
      DO 3 J=M1,M2                                                      EHOB 430
      S=0.0                                                             EHOB 440
      DO 1 K=1,L                                                        EHOB 450
      S=S+A(IA+K)*Z(K,J)                                                EHOB 460
1     CONTINUE                                                          EHOB 470
      S=S/H                                                             EHOB 480
      DO 2 K=1,L                                                        EHOB 490
      Z(K,J)=Z(K,J)-S*A(IA+K)                                           EHOB 500
2     CONTINUE                                                          EHOB 510
3     CONTINUE                                                          EHOB 520
4     CONTINUE                                                          EHOB 530
5     RETURN                                                            EHOB 540
      END                                                               EHOB 550
```

```
      SUBROUTINE EHOUSS (A,N,D,E,E2)                                    EHOU  10
C     -EHOUSS--------S/D-----LIBRARY 1-----------------------------------EHOU  20
C                                                                       EHOU  30
C     FUNCTION            - REDUCE A SYMMETRIC MATRIX A TO SYMMETRIC    EHOU  40
C                           TRIDIAGONAL FORM USING HOUSEHOLDER'S        EHOU  50
C                           REDUCTION.                                  EHOU  60
C     USAGE               - CALL EHOUSS(A,N,D,E,E2)                     EHOU  70
C     PARAMETERS   A      - THE GIVEN N X N, REAL SYMMETRIC MATRIX A,   EHOU  80
C                           WHERE A IS STORED IN SYMMETRIC STORAGE MODE.EHOU  90
C                           THE INPUT A IS REPLACED BY                  EHOU 100
C                           THE DETAILS OF THE HOUSEHOLDER              EHOU 110
C                           REDUCTION OF A.                             EHOU 120
C                N      - ORDER OF A AND THE LENGTH OF D,E, AND E2      EHOU 130
C                D      - THE OUTPUT ARRAY OF LENGTH N, GIVING THE      EHOU 140
C                           DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX.EHOU 150
C                E      - THE OUTPUT ARRAY OF LENGTH N, GIVING THE SUB- EHOU 160
C                           DIAGONAL IN THE LAST (N-1) ELEMENTS, E(1) ISEHOU 170
C                           SET TO ZERO.                                EHOU 180
C                E2     - OUTPUT ARRAY OF LENGTH N.  E2(I) = E(I)**2.  EHOU 190
C     PRECISION           - SINGLE/DOUBLE                               EHOU 200
C     LANGUAGE            - FORTRAN                                     EHOU 210
C     ------------------------------------------------------------------EHOU 220
C     LATEST REVISION     - APRIL 11,1975                               EHOU 230
C                                                                       EHOU 240
C                                                                       EHOU 250
      DIMENSION A(1), D(N), E(N), E2(N)                                 EHOU 260
C     1    DOUBLE PRECISION   A,D,E,E2,ZERO,H,SCALE,ONE,SCALE1,F,G,HH  EHOU 270
      REAL A,D,E,E2,ZERO,H,SCALE,ONE,SCALE1,F,G,HH                      EHOU 280
C     1    DATA                ZERO/0.0D0/,ONE/1.0D0/                  EHOU 290
      DATA ZERO/0.0/,ONE/1.0/                                           EHOU 300
      NP1=N+1                                                           EHOU 310
      NN=(N*NP1)/2-1                                                    EHOU 320
```

```
      NBEG=NN+1-N                                                       EHOU 330
      DO 14 II=1,N                                                      EHOU 340
      I=NP1-II                                                          EHOU 350
      L=I-1                                                             EHOU 360
      H=ZERO                                                            EHOU 370
      SCALE=ZERO                                                        EHOU 380
      IF (L.LT.1) GO TO 2                                               EHOU 390
C                                   SCALE ROW (ALGOL TOL THEN NOT NEEDED)EHOU 400
      NK=NN                                                             EHOU 410
      DO 1 K=1,L                                                        EHOU 420
C     1           SCALE = SCALE+DABS(A(NK))                             EHOU 430
      SCALE=SCALE+ABS(A(NK))                                            EHOU 440
      NK=NK-1                                                           EHOU 450
1     CONTINUE                                                          EHOU 460
      IF (SCALE.NE.ZERO) GO TO 3                                        EHOU 470
2     E(I)=ZERO                                                         EHOU 480
      E2(I)=ZERO                                                        EHOU 490
      GO TO 13                                                          EHOU 500
3     NK=NN                                                             EHOU 510
      SCALE1=ONE/SCALE                                                  EHOU 520
      DO 4 K=1,L                                                        EHOU 530
      A(NK)=A(NK)*SCALE1                                                EHOU 540
      H=H+A(NK)*A(NK)                                                   EHOU 550
      NK=NK-1                                                           EHOU 560
4     CONTINUE                                                          EHOU 570
      E2(I)=SCALE*SCALE*H                                               EHOU 580
      F=A(NN)                                                           EHOU 590
C     1        G = -DSIGN(DSQRT(H),F)                                   EHOU 600
      G=-SIGN(SQRT(H),F)                                                EHOU 610
      E(I)=SCALE*G                                                      EHOU 620
      H=H-F*G                                                           EHOU 630
      A(NN)=F-G                                                         EHOU 640
```

```
      IF (L.EQ.1) GO TO 11                                              EHOU 650
      F=ZERO                                                            EHOU 660
      JK1=1                                                             EHOU 670
      DO 8 J=1,L                                                        EHOU 680
      G=ZERO                                                            EHOU 690
      IK=NBEG+1                                                         EHOU 700
      JK=JK1                                                            EHOU 710
C                                   FORM ELEMENT OF A*U                 EHOU 720
      DO 5 K=1,J                                                        EHOU 730
      G=G+A(JK)*A(IK)                                                   EHOU 740
      JK=JK+1                                                           EHOU 750
      IK=IK+1                                                           EHOU 760
5     CONTINUE                                                          EHOU 770
      JP1=J+1                                                           EHOU 780
      IF (L.LT.JP1) GO TO 7                                             EHOU 790
      JK=JK+J-1                                                         EHOU 800
      DO 6 K=JP1,L                                                      EHOU 810
      G=G+A(JK)*A(IK)                                                   EHOU 820
      JK=JK+K                                                           EHOU 830
      IK=IK+1                                                           EHOU 840
6     CONTINUE                                                          EHOU 850
C                                   FORM ELEMENT OF P                   EHOU 860
7     E(J)=G/H                                                          EHOU 870
      F=F+E(J)*A(NBEG+J)                                                EHOU 880
      JK1=JK1+J                                                         EHOU 890
8     CONTINUE                                                          EHOU 900
      HH=F/(H+H)                                                        EHOU 910
C                                   FORM REDUCED A                      EHOU 920
      JK=1                                                              EHOU 930
      DO 10 J=1,L                                                       EHOU 940
      F=A(NBEG+J)                                                       EHOU 950
      G=E(J)-HH*F                                                       EHOU 960
```

```
      E(J)=G                                                            EHOU 970
      DO 9 K=1,J                                                        EHOU 980
      A(JK)=A(JK)-F*E(K)-G*A(NBEG+K)                                    EHOU 990
      JK=JK+1                                                           EHOU1000
9     CONTINUE                                                          EHOU1010
10    CONTINUE                                                          EHOU1020
11    DO 12 K=1,L                                                       EHOU1030
      A(NBEG+K)=SCALE*A(NBEG+K)                                         EHOU1040
12    CONTINUE                                                          EHOU1050
13    D(I)=A(NBEG+I)                                                    EHOU1060
      A(NBEG+I)=H*SCALE*SCALE                                           EHOU1070
      NBEG=NBEG-I+1                                                     EHOU1080
      NN=NN-I                                                           EHOU1090
14    CONTINUE                                                          EHOU1100
      RETURN                                                            EHOU1110
      END                                                               EHOU1120
```

```
      SUBROUTINE EQRT2S (D,E,N,Z,IZ,IER)                                EQRT  10
C     -EQRT2S--------S/D-----LIBRARY 1-----------------------------------EQRT  20
C                                                                       EQRT  30
C     FUNCTION              - FIND THE EIGENVALUES AND (OPTIONALLY)     EQRT  40
C                             EIGENVECTORS OF A TRIDIAGONAL MATRIX, T,  EQRT  50
C                             USING THE QL METHOD.                      EQRT  60
C     USAGE                 - CALL EQRT2S (D,E,N,Z,IZ,IER)              EQRT  70
C     PARAMETERS   D        - ON INPUT, THE VECTOR D OF LENGTH N CONTAINS EQRT  80
C                             THE DIAGONAL ELEMENTS OF THE TRIDIAGONAL  EQRT  90
C                             MATRIX T.                                 EQRT 100
C                             ON OUTPUT, D CONTAINS THE EIGENVALUES OF  EQRT 110
C                             T IN ASCENDING ORDER.                     EQRT 120
C                  E        - ON INPUT, THE VECTOR E OF LENGTH N CONTAINS EQRT 130
C                             THE SUB-DIAGONAL ELEMENTS OF T IN POSITION EQRT 140
C                             2,...,N. ON OUTPUT, E IS DESTROYED.       EQRT 150
C                  N        - ORDER OF TRIDIAGONAL MATRIX T.(INPUT)     EQRT 160
C                  Z        - ON INPUT, Z CONTAINS THE IDENTITY MATRIX OF EQRT 170
C                             ORDER N.                                  EQRT 180
C                             ON OUTPUT, Z CONTAINS THE EIGENVECTORS    EQRT 190
C                             OF T. THE EIGENVECTOR IN COLUMN J OF Z    EQRT 200
C                             CORRESPONDS TO THE EIGENVALUE D(J).       EQRT 210
C                  IZ       - ROW DIMENSION OF Z IN THE CALLING PROGRAM. EQRT 220
C                             IF IZ IS LESS THAN N, THE EIGENVECTORS ARE EQRT 230
C                             NOT COMPUTED. IN THIS CASE Z IS NOT USED. EQRT 240
C                  IER      - ERROR PARAMETER                           EQRT 250
C                             TERMINAL ERROR                            EQRT 260
C                             IER = 128+J, INDICATES THAT EQRT2S FAILED EQRT 270
C                               TO CONVERGE ON EIGENVALUE J. EIGENVALUES EQRT 280
C                               AND EIGENVECTORS 1,...,J-1 HAVE BEEN    EQRT 290
C                               COMPUTED CORRECTLY, BUT THE EIGENVALUES EQRT 300
C                               ARE UNORDERED.                          EQRT 310
C     PRECISION             - SINGLE/DOUBLE                             EQRT 320
```

```
C     REQD. IMSL ROUTINES - UERTST                                      EQRT 330
C     LANGUAGE            - FORTRAN                                     EQRT 340
C     ------------------------------------------------------------------EQRT 350
C     LATEST REVISION     - MARCH 11, 1977                              EQRT 360
C                                                                       EQRT 370
C                                                                       EQRT 380
      DIMENSION D(1), E(1), Z(IZ,1)                                     EQRT 390
C     DOUBLE PRECISION   D,E,Z,B,C,F,G,H,P,R,S,RDELP,ONE,ZERO           EQRT 400
C     1    DATA                RDELP/Z3410000000000000/                 EQRT 410
      DATA RDELP/Z3C100000/                                             EQRT 420
C     1    DATA                ZERO,ONE/0.0D0,1.0D0/                    EQRT 430
      DATA ZERO,ONE/0.0,1.0/                                            EQRT 440
C                                     MOVE THE LAST N-1 ELEMENTS        EQRT 450
C                                     OF E INTO THE FIRST N-1 LOCATIONS EQRT 460
      IER=0                                                             EQRT 470
      IF (N.EQ.1) GO TO 18                                              EQRT 480
      DO 1 I=2,N                                                        EQRT 490
      E(I-1)=E(I)                                                       EQRT 500
1     CONTINUE                                                          EQRT 510
      E(N)=ZERO                                                         EQRT 520
      B=ZERO                                                            EQRT 530
      F=ZERO                                                            EQRT 540
      DO 12 L=1,N                                                       EQRT 550
      J=0                                                               EQRT 560
C     1        H = RDELP*(DABS(D(L))+DABS(E(L)))                        EQRT 570
      H=RDELP*(ABS(D(L))+ABS(E(L)))                                     EQRT 580
      IF (B.LT.H) B=H                                                   EQRT 590
C                                     LOOK FOR SMALL SUB-DIAGONAL ELEMENT EQRT 600
      DO 2 M=L,N                                                        EQRT 610
      K=M                                                               EQRT 620
C     1          IF (DABS(E(K)) .LE. B) GO TO 15                        EQRT 630
      IF (ABS(E(K)).LE.B) GO TO 3                                       EQRT 640
```

```
2     CONTINUE                                                          EQRT 650
3     M=K                                                               EQRT 660
      IF (M.EQ.L) GO TO 11                                              EQRT 670
4     IF (J.EQ.30) GO TO 17                                             EQRT 680
      J=J+1                                                             EQRT 690
      L1=L+1                                                            EQRT 700
      G=D(L)                                                            EQRT 710
      P=(D(L1)-G)/(E(L)+E(L))                                           EQRT 720
C        R = DSQRT(P*P+ONE)                                             EQRT 730
C     2        D(L) = E(L)/(P+DSIGN(R,P))                              EQRT 740
      R=SQRT(P*P+ONE)                                                   EQRT 750
      D(L)=E(L)/(P+SIGN(R,P))                                           EQRT 760
      H=G-D(L)                                                          EQRT 770
      DO 5 I=L1,N                                                       EQRT 780
      D(I)=D(I)-H                                                       EQRT 790
5     CONTINUE                                                          EQRT 800
      F=F+H                                                             EQRT 810
C                                   QL TRANSFORMATION                   EQRT 820
      P=D(M)                                                            EQRT 830
      C=ONE                                                             EQRT 840
      S=ZERO                                                            EQRT 850
      MM1=M-1                                                           EQRT 860
      MM1PL=MM1+L                                                       EQRT 870
      IF (L.GT.MM1) GO TO 10                                            EQRT 880
      DO 9 II=L,MM1                                                     EQRT 890
      I=MM1PL-II                                                        EQRT 900
      G=C*E(I)                                                          EQRT 910
      H=C*P                                                             EQRT 920
C     1           IF (DABS(P).LT.DABS(E(I))) GO TO 30                   EQRT 930
      IF (ABS(P).LT.ABS(E(I))) GO TO 6                                  EQRT 940
      C=E(I)/P                                                          EQRT 950
C     1           R = DSQRT(C*C+ONE)                                    EQRT 960
```

```
      R=SQRT(C*C+ONE)                                                   EQRT 970
      E(I+1)=S*P*R                                                      EQRT 980
      S=C/R                                                             EQRT 990
      C=ONE/R                                                           EQRT1000
      GO TO 7                                                           EQRT1010
6     C=P/E(I)                                                          EQRT1020
C     1            R = DSQRT(C*C+ONE)                                   EQRT1030
      R=SQRT(C*C+ONE)                                                   EQRT1040
      E(I+1)=S*E(I)*R                                                   EQRT1050
      S=ONE/R                                                           EQRT1060
      C=C*S                                                             EQRT1070
7     P=C*D(I)-S*G                                                      EQRT1080
      D(I+1)=H+S*(C*G+S*D(I))                                           EQRT1090
      IF (IZ.LT.N) GO TO 9                                              EQRT1100
C                                   FORM VECTOR                         EQRT1110
      DO 8 K=1,N                                                        EQRT1120
      H=Z(K,I+1)                                                        EQRT1130
      Z(K,I+1)=S*Z(K,I)+C*H                                             EQRT1140
      Z(K,I)=C*Z(K,I)-S*H                                               EQRT1150
8     CONTINUE                                                          EQRT1160
9     CONTINUE                                                          EQRT1170
10    E(L)=S*P                                                          EQRT1180
      D(L)=C*P                                                          EQRT1190
C     1        IF (DABS(E(L)) .GT.B) GO TO 20                           EQRT1200
      IF (ABS(E(L)).GT.B) GO TO 4                                       EQRT1210
11    D(L)=D(L)+F                                                       EQRT1220
12    CONTINUE                                                          EQRT1230
C                                   ORDER EIGENVALUES AND EIGENVECTORS  EQRT1240
      DO 16 I=1,N                                                       EQRT1250
      K=I                                                               EQRT1260
      P=D(I)                                                            EQRT1270
      IP1=I+1                                                           EQRT1280
```

```
      IF (IP1.GT.N) GO TO 14                                            EQRT1290
      DO 13 J=IP1,N                                                     EQRT1300
      IF (D(J).GE.P) GO TO 13                                           EQRT1310
      K=J                                                               EQRT1320
      P=D(J)                                                            EQRT1330
13    CONTINUE                                                          EQRT1340
14    IF (K.EQ.I) GO TO 16                                              EQRT1350
      D(K)=D(I)                                                         EQRT1360
      D(I)=P                                                            EQRT1370
      IF (IZ.LT.N) GO TO 16                                             EQRT1380
      DO 15 J=1,N                                                       EQRT1390
      P=Z(J,I)                                                          EQRT1400
      Z(J,I)=Z(J,K)                                                     EQRT1410
      Z(J,K)=P                                                          EQRT1420
15    CONTINUE                                                          EQRT1430
16    CONTINUE                                                          EQRT1440
      GO TO 18                                                          EQRT1450
17    IER=128+L                                                         EQRT1460
      CALL UERTST (IER,6HEQRT2S)                                        EQRT1470
18    RETURN                                                            EQRT1480
      END                                                               EQRT1490
```

```
      SUBROUTINE UERTST (IER,NAME)                                      UERT  10
C     -UERTST----------------LIBRARY 1-----------------------------------UERT  20
C                                                                       UERT  30
C     FUNCTION              - ERROR MESSAGE GENERATION                  UERT  40
C     USAGE                 - CALL UERTST(IER,NAME)                     UERT  50
C     PARAMETERS   IER      - ERROR PARAMETER. TYPE + N  WHERE          UERT  60
C                             TYPE= 128 IMPLIES TERMINAL ERROR          UERT  70
C                                    64 IMPLIES WARNING WITH FIX        UERT  80
C                                    32 IMPLIES WARNING                 UERT  90
C                             N   = ERROR CODE RELEVANT TO CALLING ROUTINEUERT 100
C                 NAME    - INPUT VECTOR CONTAINING THE NAME OF THE     UERT 110
C                             CALLING ROUTINE AS A SIX CHARACTER LITERAL UERT 120
C                             STRING.                                   UERT 130
C     LANGUAGE              - FORTRAN                                   UERT 140
C     ------------------------------------------------------------------UERT 150
C     LATEST REVISION       - JANUARY 18, 1974                          UERT 160
C                                                                       UERT 170
C                                                                       UERT 180
      DIMENSION ITYP(5,4), IBIT(4)                                      UERT 190
      INTEGER*2 NAME(3)                                                 UERT 195
      INTEGER WARN,WARF,TERM,PRINTR                                     UERT 200
      EQUIVALENCE (IBIT(1),WARN), (IBIT(2),WARF), (IBIT(3),TERM)        UERT 210
      DATA ITYP/4HWARN,4HING ,4H    ,4H    ,4H    ,4HWARN,4HING(,4HWITH,UERT 220
     14H FIX,4H)   ,4HTERM,4HINAL,4H    ,4H    ,4H    ,4HNON-,4HDEFI,4HNUERT 230
     2ED ,4H    ,4H    /,IBIT/32,64,128,0/                              UERT 240
      DATA PRINTR/6/                                                    UERT 250
      IER2=IER                                                          UERT 260
      IF (IER2.GE.WARN) GO TO 1                                         UERT 270
C                                   NON-DEFINED                         UERT 280
      IER1=4                                                            UERT 290
      GO TO 4                                                           UERT 300
1     IF (IER2.LT.TERM) GO TO 2                                         UERT 310
```

```
C                                       TERMINAL                                UERT 320
      IER1=3                                                                    UERT 330
      GO TO 4                                                                   UERT 340
2     IF (IER2.LT.WARF) GO TO 3                                                 UERT 350
C                                       WARNING(WITH FIX)                       UERT 360
      IER1=2                                                                    UERT 370
      GO TO 4                                                                   UERT 380
C                                       WARNING                                 UERT 390
3     IER1=1                                                                    UERT 400
C                                       EXTRACT 'N'                             UERT 410
4     IER2=IER2-IBIT(IER1)                                                      UERT 420
C                                       PRINT ERROR MESSAGE                     UERT 430
      WRITE (PRINTR,5) (ITYP(I,IER1),I=1,5),NAME,IER2,IER                       UERT 440
      RETURN                                                                    UERT 450
C                                                                               UERT 460
5     FORMAT (26H *** I M S L(UERTST) ***  ,5A4,4X,3A2,4X,I2,8H (IER = ,UERT 470
     1I3,1H))                                                                   UERT 480
      END                                                                       UERT 490
```

The vita has been removed from the scanned document

THE MASS MATRIX IN DYNAMIC STRUCTURAL ANALYSIS

by

Thomas J. Enneking

(ABSTRACT)

This thesis is concerned with the use and development of mass matrices. A literature study is performed to determine the role of the mass matrix in the current state of the art of dynamic structural analysis. For simplicity and efficiency, the information obtained from the literature study is presented in a tabular format.

A comparison study of three different types of mass matrices on the basis of frequency prediction is conducted. In order to perform the comparison study, a computer code was developed using beam-column elements to assemble the system mass matrix and calculate the eigenvalues and eigenvectors. This code was then added to the code developed in CE4002 - Matrix Structural Analysis and CE5980 - Computer Aided Structural Design. Test problems are presented and comparisons made with exact solutions and solutions from the literature.