FREE AND FORCED VIBRATION ANALYSIS OF LINEAR, ELASTIC BUILDING AND ARCH TYPE PLANE FRAMES

Balamir Yasa

September, 1985

Boğaziçi University

FREE AND FORCED VIBRATION ANALYSIS
OF
LINEAR, ELASTIC
BUILDING AND ARCH TYPE
PLANE FRAMES

by

Balamir Yasa

B.S., in C.E., Boğaziçi University, 1982

Submitted to the Institute for Graduate Studies in Science and Engineering in Partial Fulfillment of the requirements for the degree of

Master of Science

in

Civil Engineering

Boğaziçi University

1985

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to those who have assisted me during the development of this study, especially to my thesis supervisor Doç. Dr. H. Faruk Karadoğan for his invaluable suggestions, guidence and encouragement.

I am greatly indebted to the Staff of the Boğaziçi University Civil Engineering Department for their generous help and constructive critisizms.

I am also indebted to all Boğaziçi University Computer Center personnel for making facilities available to me.

I am also grateful to Miss Ayşe Özen for her patience, care and conscientiousness in typing the manuscript.

Istanbul, September 1985.

BALAMİR YASA

ABSTRACT

In this study, behaviour of linear, elastic building and arch type plane frame structures subjected to lateral dynamic loads are investigated. For this purpose, computer programs were developed for the static analysis of plane frame structures, frequency/mode shape computations, time history analysis by the mode superposition and the direct step-by-step integration methods.

A static analysis program is designed, which enables the solution of large scale structures with many degrees-of-freedom by employing an out-of-core solution algorithm. For reasonable half-band width values, there is no limit to the number of equations to be solved, as long as the pheripheral memory may be assumed "infinitely large."

Effective computer utilaziton in terms of time, memory and precision is considered in developing these computer programs. These three factors gain significance, especially, in the case of using low speed, small memory and low precision microcomputers in the analysis and design of large structural systems.

Using the computer programs that were developed during this study, two types of structures were examined : high-rise buildings (multi-storey building frames and towers), and arches.

The following aspects of the problem are studied.

1. The accuracy of the results of the static analysis. For large number of unknowns, error accummulation due to rounding may be a serious problem.

2. Effect of the slenderness and aspect ratios of arch type structures to the symmetry or antimetry of the modeshapes and to the value of the participation factors.

3. Effect of the choice of solution time step Δt, to the accuracy and stability of the integrations performed in direct step-by-step integration method.

4. Effect of the number and type of the mode shape vectors belonging to each of the modes superposed, to the accuracy of the results. The detected maximum displacement values from the modal analysis were compared to those obtained from the direct step-by-step integration of the coupled equations of motion.

Some of the findings from the case studies are summarized below.

1. The algorithm used in the static analysis program is very effective to analyse very high multi-storey structures. The accuracy of the results is significantly affected by the wordsize of the computer and the solution algorithm employed.

2. The choice of the solution time step Δt, affects both the accuracy of the results and the execution time. As Δt decreases, the solution times increase and the weighted percent errors of the results decrease. However, very small Δt values may cause error accumulation due to rounding.

3. For larger Δt values, the numerical integration performed in step by step procedures may diverge Appropriate values for Δt, which will give a converging result may not be chosen depending on a previously stated emphirical formula.

4. Superposing only a few of the highest modes of vibration gives satisfactory results. However, when the lower modes of vibration are superposed, the error in the results may increase. This is due to the computation error made in the lower modes of vibration.

5. Free vibration characteristics of arches with symmetric stiffness and mass distribution depends on the slenderness and aspect ratios. Their participation factors for symmetric mode shapes are equal to zero. Therefore, in modal analysis, superposing only those modes with non-zero participation factors will further reduce the computation time.

Ö Z E T

Bu çalışmada, bina ve kemer tipi düzlem çerçeve sistemlerin dinamik yükler altındaki davranışı incelenmiştir. Bu amaçla; düzlem çerçevelerin statik analizi, serbest titreşim hesapları, modların süperpozisyonu ve zaman artımı yöntemleri ile zaman tarihçesi analizi konularında bilgisayar programları geliştirilmiştir.

Çok serbestlik dereceli büyük yapıların statik yükler altında çözümünü yapabilen bir bilgisayar programı tasarlanmıştır. Bu programda, ana bellek ile birlikte çevre bellekten de faydalanan bir algoritma kullanılmıştır. Çevre bellek kapasitesinde herhangi bir sınırlama yoksa, makul yarı bant genişlikleri için, çözülebilecek bilinmeyen sayısında herhangi bir sınırlama yoktur.

Bu programların geliştirilmesi sırasında; çözüm süresinin kısalığı, bellek sınırlamaları ve sonuçların doğruluğu konuları üzerinde durulmuştur. Bu üç etkenin, büyük yapıların çözümlenmesinde ve tasarımında, düşük hızlı, küçük hafızalı ve hassasiyeti az olan mikrobilgisayarlar kullanıldığında önem kazandığı bilinmektedir.

Bu çalışma sırasında geliştirilen programları kul-

lanarak çok katlı yapılar ve onlara göre davranışı daha az bilinen kemer tipi yapılar incelenmiştir.

Yapılan çözümlemeler sırasında aşağıdaki kavramlar üzerinde durulmuştur.

1. Çok bilinmeyenli sistemlerin statik analizinde sonuçların doğruluğu.

2. Kemer tipi yapılarda, narinlik ve basıklık oranlarının mod şekillerinin simetrik veya antimetrik olmasına ve katılma oranlarının değerine etkisi.

3. Zaman artımı yönteminde seçilen zaman aralığının integrasyonun yakınsaklığına ve doğruluğuna etkisi.

4. Modal analizde, süperpoze edilen modların sayısı ve tiplerinin sonuçların doğruluğuna etkisi.

Elde edilen sonuçlardan çıkarılan bulguların bazıları şunlardır:

1. Statik analiz programında kullanılan algoritma çok bilinmeyenli yapıların hesabında başarılı sonuç vermiştir. Elde edilen sonuçların doğruluğu kullanılan bilgisayarın kelime uzunluğu ve çözüm yöntemi ile yakından ilgilidir.

2. Seçilen zaman aralığı çözüm süresini ve sonuçların doğruluğunu etkiler. Zaman aralığı küçüldükçe, çözüm süresinin uzadığı ve sonuçların doğruluğunun arttığı

gözlenmiştir. Bununla birlikte, çok küçük zaman aralıkları seçildiğinde, yuvarlatmadan dolayı hata birikimi olabileceği unutulmamalıdır.

3. Zaman artımı yönteminde, daha büyük zaman aralığı değerleri için, hesaplanan integral ıraksak olabilir. Yakınsak sonuçlar verecek uygun zaman aralığı seçimi için genel bir kural çıkartılamamıştır.

4. Modal analizde, yanlızca, en küçük frekanslı bir kaç modun katılmasının yeterli doğrulukta sonuçlar verdiği gözlenmiştir. Büyük frekanslı modlar süperpoze edildiğinde ise bir miktar hatanın kaldığı görülmüştür. Bunun sebebi; büyük frekanslı modların hesaplanması sırasında birikmiş olan hataların varlığı olabilir.

5. Simetrik rijidlik ve kütle dağılımı olan kemer tipi yapılarınserbest titreşim özellikleri narinlik ve basıklık oranlarına bağımlıdır. Simetrik modlara ait katılma oranlarının sıfır değeri aldığı görülmüştür. Simetrik modların katkısının modal analizde göz önüne alınmaması hesaplama süresini daha da kısaltacaktır.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	cross-sectional area
[C]	damping matrix
E	modulus of elasticity
{F}	earthquake force vector
$[\bar{F}]$	reduced flexibility matrix
I	moment of inertia
[I]	identity matrix
[K]	structure stiffness matrix
$[\bar{K}]$	reduced stiffness matrix
L	length
[M]	mass matrix
{P}	vector of end forces
S_a	spectral acceleration
S_d	spectral displacement
S_v	spectral velocity
T	period of vibration
[T]	transformation matrix
{U}	vector used to couple the mass coefficients in horizontal direction to the horizontal earthquake motion
{V}	vector
XYZ	global coordinates
{Y}	time independent shape of the system
a(t)	input acceleration function
c	damping
$\ddot{d}_x$	horizontal component of the ground acceleration
{d}	displacement vector
{e}	eigenvector

$f(t)$	applied load function
$\{\bar{f}\}$	vector of fixed end forces
g	gravitational acceleration constant
k	stiffness
$[k]$	element stiffness matrix
m	mass influence coefficient
p	response of the system
r	participation factor
t	time
xyz	local coordinates
$\{y\}$	displacement vector in geometric coordinates
$\{\bar{y}\}$	displacement vector in generalized coordinates
$\{\dot{y}\}$	velocity vector in geometric coordinates
$\{\dot{\bar{y}}\}$	velocity vector in generalized coordinates
$\{\ddot{y}\}$	acceleration vector in geometric coordinates
$\{\ddot{\bar{y}}\}$	acceleration vector in generalized coordinates
α	proportionality factor used to form Rayleigh damping matrix
β	proportionality factor used to form Rayleigh damping matrix
γ	unit weight
Δt	solution time step
ζ	damping ratio
λ	eigenvalue
ω	undamped free vibration frequency
$\{\phi\}$	mass orthonormalized mode shape vector
$[\phi]$	modal matrix
θ	phase angle

I. INTRODUCTION

Methods to predict the response of structural systems subjected to earthquake forces can be classified in three groups. The direct step-by-step integration and the mode superposition methods are consisting the first group, and are commonly called the dynamic procedures.

In direct step-by-step integration method, the response of the structure at a certain step can be computed depending on the accelerations, velocities and the displacements of the preceeding step. This method is generally used in research studies and for predicting the dynamic response of important structures such as high-rise towers, nuclear power plants, off-shore structures, and the like.

The mode superposition method employs a generalized coordinate transformation which serves to change the set of coupled equations of motion into a set of uncoupled equations. These uncoupled equations are integrated, and the response of the whole system is then computed by a linear superposition of these individual single degree-of-freedom responses. Although damping forces can be considered, the method is essentially applicable only to linear elastic structures.

In practice, the procedure so called the response spectra analysis, which is a special case of the mode superposition method, is used very frequently. In this method, only the maximum values of the response for each uncoupled equation of motion is considered, which reduces the execution time considerably. The most probable values for the structural response can be computed by "summing" these individual modal responses. For both the mode superposition method and the response spectra analysis procedure, the contributions of only a few of the smallest frequencies are the most significant, which further reduces the computation time.

The second group of procedures, called the semi-dynamic procedures, are very simple to apply, hence they are used in many design codes. In these procedures, the dynamic problem is reduced to a static problem by defining fictituous static loads depending on, generally, the fundamental mode of vibration, and the results are multiplied by correction factors to satisfy other conditions.

In the third group, static loads are used which do not depend on the dynamic characteristics of the structure. This method is not anymore used alone to predict the response of important structures.

In this study, computer programs were developed for the static and dynamic analysis of linear, elastic building and arch type plane frame structures. For the static analysis program, practically, there is no limit to the size of the problem to be solved. This is achieved by using an out-of-core solution algorithm.

Several sample problems were solved to examine the free and forced vibration characteristics of building and arch type structures. In terms of their dynamic properties, these two types of systems were observed to be behaving in a different manner.

The displacements obtained from step-by-step analysis and modal superposition were compared. The effect of solution time step, Δt, and number and shape of the modes to be superposed on accuracy were investigated.

The choice of fictitious static loads corresponding to each mode and a basis to sum the individual responses due to these static loads are illustrated on one example.

II. METHODS OF ANALYSIS

In this chapter, fundamental aspects of the static and dynamic analysis of plane frame structures are given.

The member stiffness matrix for a general straight, constant cross-section plane frame element and the local and global coordinate systems are defined. Code number technique is illustrated which may be used for the assembly of the equilibrium equations.

The details of the frontal solution technique is illustrated on a simple continuous beam with three unknowns. The advantages and disadvantages of this technique is briefly discussed.

Computation of the frequencies and mode shape vectors of a freely vibrating undamped system, basic orthogonality relationships and ortho-normalization of the mode shape vectors are given in detail.

The recurrance equations which will be used to integrate the equation of motion for a single degree-of-freedom damped system are derived.

The procedure for modal analysis is given, including; the generalized coordinate transformation, integration of the uncoupled equations of motion, and the superposition of the individual modal contributions.

In the derivation of the recurrance equations for the direct step-by-step integration method, the accelerations were assumed to vary linearly within a time interval Δt.

The response spectra analysis procedure is described and three summation formulas are given.

Finally, the relationship between the elastic forces acting on a structure and the inertia forces is derived, and a formula to compute equivalent fictitious static loads is given. These fictitious static loads may be used to predict dynamic response.

II.A MATRIX DISPLACEMENT METHOD

In matrix displacement method, a continuous structure is represented by a set of discrete points in the system. The structural system is discretized into smaller elements, and the properties of the structure is assumed to be concentrated at the points (nodes) within the system. The equilibrium equations belonging to these nodes are solved, and the response is found by computing the response of the elements composing the structure. In the following sections fundamental aspects of the matrix displacement method are given.

II.A.1 PRISMATIC PLANE FRAME ELEMENT

The definition of a stiffness coefficient k_{ij} can be stated as "the force that should be applied in direction i in order to sustain a unit displacement in direction j, while all other prescribed displacements are equal to zero." The group of forces that should be applied in the directions of the prescribed displacements in order to sustain the unit displacement in direction j, forms the j th column of the stiffness matrix.

For a prismatic plane frame element the prescribed displacement and force directions, termed as "degrees-of-freedom," are shown in Fig. II.1

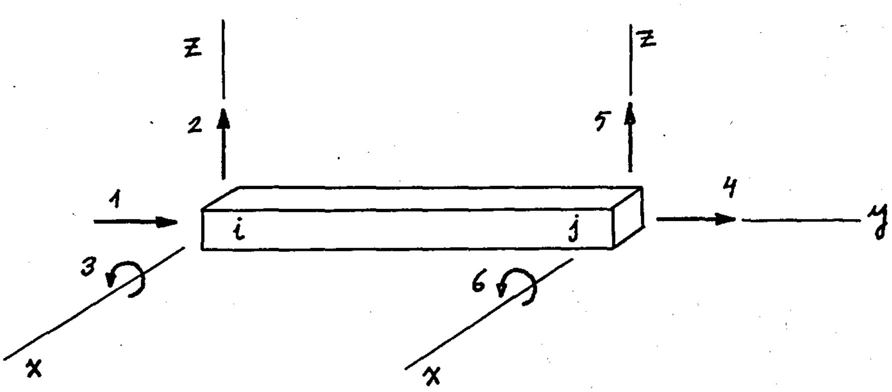


FIGURE II.1 Prescribed Degrees of Freedom and the Local Coordinate System for a Prismatic Plane Frame Element

The equilibrium equations for a general straight plane frame element, in matrix notation, is

$$\begin{Bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{Bmatrix} = \begin{bmatrix} S & 0 & 0 & -S & 0 & 0 \\ 0 & D & C_i & 0 & -D & C_j \\ 0 & C_i & A_i & 0 & -C_i & B \\ -S & 0 & 0 & S & 0 & 0 \\ 0 & -D & -C_i & 0 & D & -C_j \\ 0 & C_j & B & 0 & -C_j & A_j \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{Bmatrix} + \begin{Bmatrix} \bar{f}_1 \\ \bar{f}_2 \\ \bar{f}_3 \\ \bar{f}_4 \\ \bar{f}_5 \\ \bar{f}_6 \end{Bmatrix} \quad \text{(II.1a)}$$

or in compact form

$$\{P\} = [k]\ \{d\} + \{\bar{f}\} \qquad \text{(II.1b)}$$

where

$\{P\}$: vector of end forces

$\{d\}$: displacement vector

$[k]$: element stiffness matrix

$\{\bar{f}\}$: vector of fixed end forces.

For a general straight plane frame element, the stiffness coefficients are

$$S = s\frac{EA}{L} \qquad C_i = \frac{A_i + B}{L}$$

$$A_i = a_i\frac{EI}{L} \qquad C_j = \frac{A_j + B}{L} \qquad \text{(II.1c)}$$

$$A_j = a_j \frac{EI}{L} \qquad D = \frac{C_i + C_j}{L}$$

$$B = b_{ij} \frac{EI}{L}$$

For a constant cross-section straight element; $s=1$, $a_i = a_j = 4$, and $b_{ij} = 2$, therefore Eqs.(II.1c) become

$$S = \frac{EA}{L} \qquad C_i = C_j = \frac{6EI}{L^2}$$

$$A_i = A_j = \frac{4EI}{L}$$

$$D = \frac{12EI}{L^3}$$

$$B = \frac{2EI}{L}$$

where

E : modulus of elasticity

A : cross-sectional area

I : moment of inertia

L : length of the element

The details of the derivation of element stiffness matrices for various types of structural members are given in Refs.(1), (2) and (3)*. The fixed end forces due to loads acting on the element form the vector $\{\bar{f}\}$ (see Fig. II.2). Fixed end forces for most common types of loadings

---

**Paranthetical references placed superior to the line of text refer to the bibliography.*

are given in Refs. (1),(2),(3) and (4).

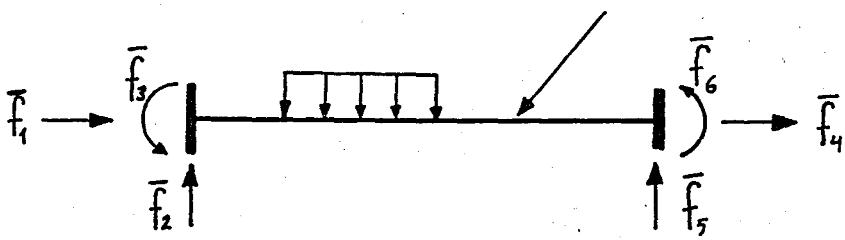


FIGURE II.2 Fixed End Forces Due to Element Loads

For a structural system there are two types of coordinates: the local (member) coordinates, and the global (structure) coordinates. The local and the global coordinate systems are denoted by subscripts xyz and XYZ, and are shown in Figures II.1 and II.3, respectively.

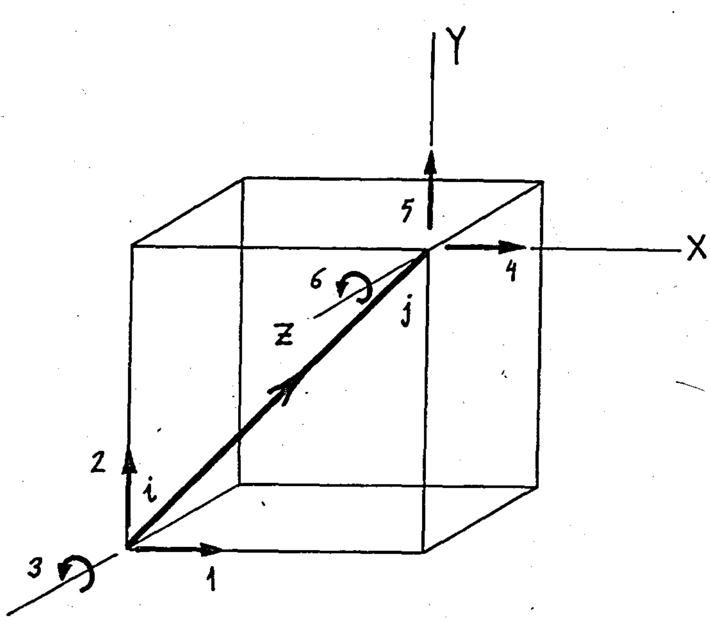


FIGURE II.3 The Global Coordinate System for a Straight Plane Frame Element

Although the forces and displacements can be defined in local coordinates, for the sake of simplicity, they shall be transformed to the global coordinate system in order write the equilibrium equations for the structral system. Any vector quantity, namely the forces and displacements, can be transformed from the global to the local coordinate system by

$$\begin{Bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{Bmatrix}_{xyz} = \begin{bmatrix} m & n & 0 & 0 & 0 & 0 \\ -n & m & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & m & n & 0 \\ 0 & 0 & 0 & -n & m & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{Bmatrix}_{XYZ} \qquad \text{(II.2a)}$$

or

$$\{V\}_{xyz} = [T]\ \{V\}_{XYZ} \qquad \text{(II.2b)}$$

where

$\{V\}$: force or displacement vector in either coordinate system.

The transformation matrix $[T]$ can be used both to transform forces and displacements. This type of transformation is called orthogonal transformation. The columns and the rows of the transformation matrix are normal and orthogonal to each other, therefore its inverse is equal to its transpose.

$$\{V\}_{XYZ} = [T]^{-1} \{V\}_{xyz}$$

or

$$\{V\}_{XYZ} = [T]^{T} \{V\}_{xyz} \tag{II.3}$$

Omitting the fixed end forces in Eq. (II.1b), and writing the force and displacement vectors in the new notation, one obtains the following set of equations.

$$\{P\}_{xyz} = [k]_{xyz} \{d\}_{xyz} \tag{II.4a}$$

$$\{P\}_{xyz} = [T] \{P\}_{XYZ} \tag{II.4b}$$

$$\{d\}_{xyz} = [T] \{d\}_{XYZ} \tag{II.4c}$$

Substituting Eqs. (II.4b) and (II.4c) into (II.4a) it becomes

$$[T] \{P\}_{XYZ} = [k]_{xyz} [T] \{d\}_{XYZ}$$

Multiplying both sides of the above equation by $[T]^{-1}$

from left leads to

$$\{P\}_{XYZ} = [T]^{-1} \; [k]_{xyz} \; [T] \; \{d\}_{XYZ}$$

or

$$\{P\}_{XYZ} = [T]^{T} \; [k]_{xyz} \; [T] \; \{d\}_{XYZ} \qquad \text{(II.5a)}$$

The element stiffness matrix with respect to global coordinate system is

$$[k]_{XYZ} = [T]^{T} \; [k]_{xyz} \; [T] \qquad \text{(II.5b)}$$

Hence, the equilibrium equation for a single element, in global coordinates, becomes

$$\{P\}_{XYZ} = [k]_{XYZ} \; \{d\}_{XYZ} \qquad \text{(II.6)}$$

The coefficients of $[k]_{XYZ}$ are given in Table II.1.

This form of the equilibrium equations should be used in the assembly of the individual element stiffness matrices to form the structure (system) stiffness matrix.

$Dn^2 + Sm^2$	$(S-D)mn$	$-C_i n$	$-Dn^2-Sm^2$	$(D-S)mn$	$-C_j n$
	$Dm^2 + Sn^2$	$C_i m$	$(D-S)mn$	$-Dm^2 - Sn^2$	$C_j m$
		A_i	$C_i n$	$-C_j m$	B
	(SYMMETRIC)		$Dn^2 + Sm^2$	$(S-D)mn$	$C_j n$
				$Dm^2 + Sn^2$	$-C_j m$
					A_j

TABLE II.1 Stiffness matrix of a General Straight Plane Frame Element (Transformed to Global Coordinates)

II.A.2 ASSEMBLY

Code number technique is used to obtain structure stiffness matrix and the load vector(s), via assembling the individual element stiffness matrices, element fixed end forces, and the nodal point loads. With this technique, assembly of stiffness coefficients of the structural members connected to a node is easier and the technique is very suitable for computer programming. Although the method is illustrated on a plane frame system in this section, it is equally applicable in any system consisting of two or three dimensional finite elements.

In the stiffness method, the unknowns are chosen to be the nodal displacements, therefore the order of the structure stiffness matrix is equal to the number of nodal displacements within the system. Every unknown nodal displacement is given an equation number (code number). Forces and displacements belonging to a node are associated with the equation numbers of that node.

Equation (code) numbers for a single element are defined to be the set of equation numbers belonging to the nodes of the element.

The equation numbering scheme is illustrated on a typical plane frame system in Fig. II.4, and the equation numbers for the elements of the system are given in Table II.2. For those directions which no displacement is allowed, a zero equation number is given. The numbers next to the arrows showing the positive displacement directions, are the equation numbers. The numbers enclosed in circles, and the numbers next to the nodes are the element and node numbers, respectively.

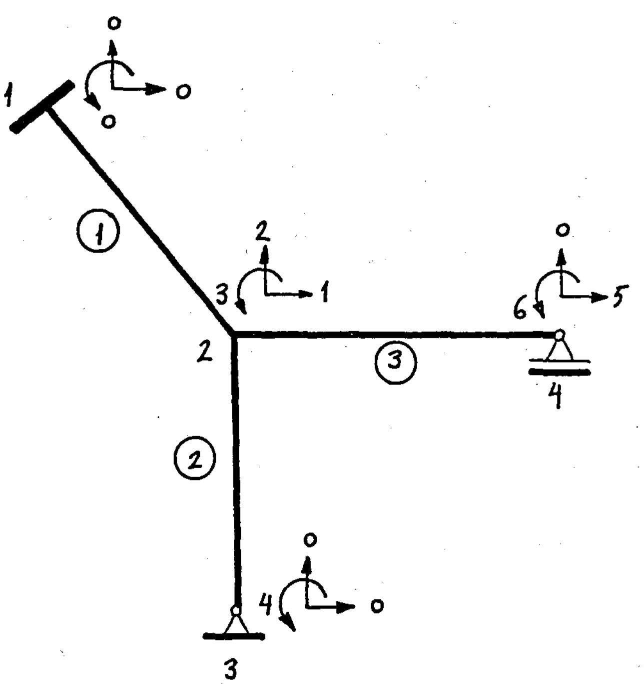


FIGURE II.4 Numbering of the Displacements of a Typical Plane Frame

Element No	Equation Numbers					
1	0	0	0	1	2	3
2	0	0	4	1	2	3
3	1	2	3	5	0	6

TABLE II.2 Equation Numbers for the Elements of the Plane Frame in Figure II.4

In order to have a unit displacement in direction 1 (X direction at node 1), the force that must be applied in that direction is

$$K_{11} = k^{1}_{44} + k^{2}_{44} + k^{3}_{11}$$

and similarly in the other directions are

$$K_{21} = k^{1}_{54} + k^{2}_{54} + k^{3}_{12}$$

$$K_{31} = k^{1}_{64} + k^{2}_{64} + k^{3}_{31}$$

$$K_{41} = k^{2}_{34}$$

$$K_{51} = k^{3}_{41}$$

$$K_{61} = k^{3}_{61}$$

where

K : coefficients of the structure (system) stiffness matrix

k^{i} : stiffness coefficients of member i.

It is clear that the participation of element stiffness coefficients to the structure stiffness are related to the code numbers of the elements. Any element participates only to these equations given its code numbers. The relationship between the element and structure stiffness coefficients is given by the following formula

$$K_{nm} = \sum_{L} k_{ij}^{L} \qquad L = 1, \ldots, NE$$

where

k_{ij}^{L} : the stiffness coefficient at location (i,j) in the stiffness matrix of element L.

K_{nm} : the structure stiffness coefficient at location (n,m)

NE : number of elements in the structural system

n and m are given by

$$n = c(i)$$

$$m = c(j)$$

where c is the vector containing the code numbers of element L.

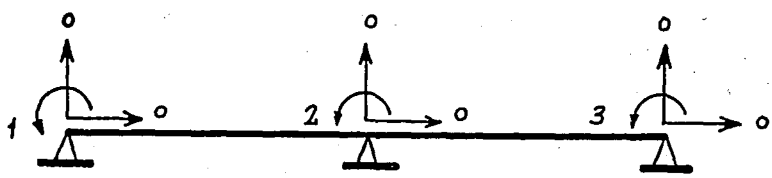
The details of the code number technique is given in Ref.(1).

II.A.3 THE FRONTAL TECHNIQUE

The frontal solution technique (5),based on Gaussian elimination method is an out-of-core solution technique to solve large positive definite matrix equations arising in the finite element method. Although in some applications the finite element equations are unsymmetric, in most of the structural mechanics problems the finite element matrices are proven to be symmetric and banded. Practically there is no limit to the total number of equations to be solved.

Only a small number of equations may be held in the main memory simultaneously at a certain instant, which are the "active" ones. The fully summed equations are eliminated and kept in auxillary (peripheral) memory, hence the available memory "freed" this way may be used for further assemly of the succeeding equations. When finally all equations are eliminated,the solution is obtained by an out-of-core backsubstitution technique.

To illustrate the procedure the three-node, two-element continuous beam shown below will be considered.



$$[K]^{i}_{XYZ} = \begin{bmatrix} k^{i}_{11} & k^{i}_{12} \\ k^{i}_{21} & k^{i}_{22} \end{bmatrix}$$

number of unknowns = 3
half bandwidth = 2
number of load vectors = 1

FIGURE II.6 Three-Node, Two-Element Continuous Beam

After the assembly of the first element equations, the state of the system stiffness matrix is as follows :

$$\begin{bmatrix} k^{I}_{11} & k^{I}_{12} & 0 \\ k^{I}_{21} & k^{I}_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} \begin{Bmatrix} b^{I}_{1} \\ b^{I}_{2} \\ 0 \end{Bmatrix}$$

where the superscript denotes the contributing element's number.

The elimination of equation one can occur at this stage, because there will be no contribution from the second element to this equation. However, elimination of equation two can occur only when the contribution of the coefficients from element two is completed. This point is not reached before the assembly of element two takes place. After the elimination of equation one, the structure stiffness matrix becomes

$$\begin{bmatrix} k_{11}^{I} & k_{12}^{I} & 0 \\ 0 & k_{22}^{I} - \dfrac{k_{21}^{I}}{k_{11}^{I}} * k_{12}^{I} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} \quad \begin{Bmatrix} b_1^{I} \\ b_2^{I} - \dfrac{k_{21}^{I}}{k_{11}^{I}} * b_1^{I} \\ 0 \end{Bmatrix}$$

From the above table it may be observed that although equation two is altered before it was fully summed, the terms subtracted involve only that component involving equation one which was already fully summed. For frontal routines each equation can be eliminated at an earlier stage than band routines-as soon as it is complete. After the elimination of equation one, there is no need to keep it in the main memory any more, therefore it may be written on the peripheral memory and that part of the main memory "freed" may be used for storing the succeding equations' coefficients. Therefore, before continuing with the assembly process, the incomplete equations in the main memory must be shifted above one line to free the very bottom line of the matrix so that it may be used for the assembly of the next equations.

Assembling the stiffness matrix of element two, the system stiffness matrix becomes

$$\begin{bmatrix} k_{11}^{I} & k_{12}^{I} & 0 \\ 0 & k_{22}^{I} - \dfrac{k_{21}^{I}}{k_{11}^{I}} * k_{12}^{I} + k_{11}^{II} & k_{12}^{II} \\ 0 & k_{21}^{II} & k_{22}^{II} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} \begin{Bmatrix} b_1^{I} \\ b_2^{I} - \dfrac{k_{21}^{I}}{k_{11}^{I}} * b_1^{I} + b_1^{II} \\ b_2^{II} \end{Bmatrix}$$

Making use of the symmetry and banding properties of the system stiffness matrix, a square matrix of (2x3) would be sufficient to solve this set of linear simultaneous equations, whereas it would actually need a (3x3) matrix if a similar but in-core-solution algorithm was used.

Backsubstitution process may be performed using a similar way; the equations and the right-hand-side vectors are read into the main memory in smaller pieces. Those right-hand-side coefficients which backsubstitution is complete are written back to the peripheral memory and those equations which the backsubstitution is not complete are shifted below to read new equations into the "freed" memory. It is extremely important to keep in mind that peripheral access is slower than access to main memory. Therefore, instead of writing the equations to the peripheral memory as soon as they are complete and eliminated, a better algorithm would be to eliminate and write the equations in groups, which will decrease the total access time significantly.

The frontal solution technique has proved a very effective means for solving the positive definite matrix equations arising in the finite element method. Using the symmetry and banding properties of the system stiffness matrix, it is possible to use either a 'band' or a 'frontal' out-of-core method to analyze systems with very large stiffness matrices. However, in spite of their slightly greater complexity, frontal routines may be preferred, since they are in general faster and require less core than band routines. In addition, it is not necessary to apply a 'stringent' node numbering scheme, because the size of the front width (equivalent to half bandwidth in band routines) depends on the element numbering scheme.

II.B UNDAMPED FREE VIBRATIONS

The coupled equations of motion for a freely vibrating undamped system is

$$[M]\{\ddot{y}\} + [\bar{K}]\{y\} = \{0\} \tag{II.8}$$

where

$[M]$: mass matrix containing the mass influence coefficients

$[\bar{K}]$: reduced stiffness matrix

$\{y\},\{\ddot{y}\}$: relative displacement and acceleration vectors, respectively.

In this equation, damping effects and externally applied loads are omitted. Assuming that the free vibration motion is harmonic, the solution to Eq.(II.8) is

$$\{y\} = \{Y\} \sin(\omega t+\theta) \tag{II.9}$$

where $\{Y\}$ represents the time independent shape of the system, θ is the phase angle, and ω is the system frequency.

Differentiating Eq.(II.9) with respect to time, the velocity and acceleration vectors are

$$\{\dot{y}\} = \omega \{Y\} \cos(\omega t+\theta) \tag{II.10a}$$

and

$$\{\ddot{y}\} = -\omega^2 \{Y\} \sin(\omega t+\theta) \qquad \text{(II.10b)}$$

Substituting Eqs.(II.9a) and (II.10b) into Eq.(II.8) gives

$$-\omega^2 [M]\{Y\} \sin(\omega t+\theta) + [\bar{K}]\{Y\} \sin(\omega t+\theta) = \{0\} \qquad \text{(II.11)}$$

Omitting the arbitrary sine terms, Eq. (II.11) may be written

$$([\bar{K}] - \omega^2 [M])\{Y\} = \{0\} \qquad \text{(II.12)}$$

For a nontrivial solution to exist, the determinant of $([\bar{K}] - \omega^2 [M])$ should vanish, i.e.

$$\det([\bar{K}] - \omega^2 [M]) = 0 \qquad \text{(II.13)}$$

There exists N roots for the equation obtained by expanding the determinant for a system having N dynamic degrees of freedom. The N roots of this equation (ω_i^2; i= 1,...,N) represent the free vibration frequencies of the system.

Another possible formulation of the problem may be derived by premultiplying Eq.(II.12) by $[\bar{K}]^{-1}$ and dividing by ω^2 :

$$\left(\frac{1}{\omega^2} [\bar{K}]^{-1} [\bar{K}] - [\bar{K}]^{-1} [M]\right)\{Y\} = \{0\} \qquad \text{(II.14)}$$

Defining the reduced flexibility matrix to be

$$[\bar{F}]^{-1} = [\bar{K}]$$

and rearranging, Eq.(II.14) becomes

$$([\bar{F}][M] - \frac{1}{\omega^2}[I])\{Y\} = \{0\} \qquad \text{(II.15)}$$

where $[I]$ is the identity matrix and the product $[\bar{F}][M]$ is the dynamic matrix.

The mode having the smallest (highest) frequency is called the first mode, the second highest frequency is called the second mode, etc.

By substituting any one of the frequencies into Eq.(II.15) a homogenous set of linear equations is obtained. The shape of the freely vibrating system called the modal shape, can be determined in terms of any one coordinate. Thus,

$$\{y\}_k = \begin{Bmatrix} Y_{1k} \\ Y_{2k} \\ \vdots \\ Y_{Nk} \end{Bmatrix} \equiv \frac{1}{Y_{mk}} \begin{Bmatrix} Y_{1k} \\ Y_{2k} \\ \vdots \\ Y_{Nk} \end{Bmatrix}$$

$$(k=1,\ldots,N), \qquad (1 \leqslant m \leqslant N)$$

where Y_{mk} is the reference component.

The free vibration mode shapes $\{Y\}_k$ have certain special properties, called the orthogonality relationships, which are very useful in structural dynamic analysis. The basic relationship "mass ortogonality condition" can be demonstrated applying Betti's reciprocal theorem as follows:

$$- \{Y\}_n^T \omega_m^2 [M] \{Y\}_m = - \{Y\}_m^T \omega_n^2 [M]\{Y\}_n \qquad (II.16)$$

where $\omega^2 [M]\{Y\}$ are the applied inertia load vectors, $\{Y\}_n$ and $\{Y\}_m$ represent the resulting displacements due to these two different applied load systems. Taking into account the symmetry of $[M]$, and noting that the matrix products in Eq. (II.16) are scalars, they can be transposed arbitrarily. Therefore Eq.(II.16) may be written

$$(\omega_n^2 - \omega_m^2) \{Y\}_m^T [M]\{Y\}_n = 0 \qquad (II.17a)$$

which implies that

$$\{Y\}_m^T [M] \{Y\}_n = 0 \qquad m \neq n \qquad (II.17b)$$

For the case m=n a scalar factor maybe defined as

$$C_m^2 = \{Y\}_m^T [M]\{Y\}_m \qquad (II.18)$$

which may be used to normalize the mode shape vectos as follows

$$\{\phi\}_m = \frac{1}{C_m} \{Y\}_m \qquad (II.19)$$

where $\{\phi\}_m$ is the mass-orthonormalized mode shape vector for mode m.

A consequence of this type of normalization, together with the modal orthogonality relationships relative to the mass matrix [Eq.(II.17b)] is that

$$[\phi]^T[M]\ [\phi] = [I] \qquad \text{(II.20)}$$

where $[\phi]$ is the complete set of N normalized mode shapes and $[I]$ is the identity matrix.

Furthermore, it is clear that replacing {Y} by {ϕ} also satisfies Eq.(II.12), hence

$$[\bar{K}]\ \{\phi\}_n = \omega_n^2\ [M]\{\phi\}_n \qquad \text{(II.21)}$$

A second orthogonality condition can be derived directly from this by premultiplying Eq.(II.20b) by $\{\phi\}_m^T$; thus

$$\{\phi\}_m^T\ [\bar{K}]\{\phi\}_n = \omega_k^2\ \{\phi\}_m^T\ [M]\{\phi\}_n$$

When the mass orthogonality condition is applied to the right-hand side it is clear that

$$\{\phi\}_m^T\ [\bar{K}]\{\phi\}_n = 0 \qquad m \neq n \qquad \text{(II.22)}$$

For the case m=n

$$\{\phi\}_n^T \; [\bar{K}]\{\phi\}_n = \omega_n^2 \qquad \text{(II.23a)}$$

Representing this orthogonality condition in compact form it follows that

$$[\phi]^T [K] \; [\phi] = [\omega^2] \qquad \text{(II.23b)}$$

where $[\omega^2]$ is the frequency matrix having the free vibration frequencies of the system on its diagonal.

Numerous methods exist to solve the eigenvalue problem given in Eqs.(II.12) and (II.15). Vianello Stodola (power iteration) method can conveniently be used for the vibration analysis of small systems [Refs. (6),(7),(8), and (9)].

In Stodola method, an initial assumption is made of the vibration mode shape, and it is adjusted iteratively until an adequate approximation to the true mode shape has been achieved. In the following paragraphs the Vianello Stodola method is explained.

Let the eigenvalues and vectors of a given matrix $[A]$ be $\lambda_1,\ldots.,\lambda_n$ and $\{e\}_1,\ldots,\{e\}_n$, and let the eigenvalues satisfy the eigenvalues satisfy the condition

$$|\lambda_1| > |\lambda_2| \geqslant \ldots \geqslant |\lambda_n|$$

If the first iteration is carried out by a starting vector

$$\{V\} = C_1 \; \{e\}_1 + \ldots.+ C_n \; \{e\}_n$$

one gets

$$[A]\{V\} = C_1 [A]\{e\}_1 + \ldots + C_n [A]\{e\}_n$$

$$= C_1\lambda_1\{e\}_1 + \ldots + C_n\lambda_n\{e\}_n$$

$$= \lambda_1 \quad C_1\{e\}_1 + C_2(\frac{\lambda_2}{\lambda_1})\{e\}_2 + \ldots + C_n(\frac{\lambda_n}{\lambda_1})\{e\}_n$$

Repeating the iteration p times

$$[A]^p\{V\} = \lambda_1^p \quad C_1\{e\}_1 + \ldots + C_n(\frac{\lambda_n}{\lambda_1})^p\{e\}_n$$

Knowing that $\lambda_i/\lambda_1<1$ $(i = 2, \ldots, n)$,

$$\lim_{p\to\infty} [A]^p\{V\} = C_1\lambda_1^p \{e\}_1, \text{ since } \lim_{p\to\infty} (\lambda_i/\lambda_1)^p = 0.$$

Practically, if p is large enough, $(\lambda_i/\lambda_1)^p$ will vanish. The eigenvalue is the largest coefficient of the last iteration vector computed, and the vector obtained by dividing the last iteration vector by the eigenvalue is the corresponding eigenvector. The frequency can be computed by

$$\omega_1 = \sqrt{1/\lambda_1} \qquad \text{(II.24)}$$

and the orthonormalized mode shape vector can be computed by Eqs.(II.18) and (II.19).

The Stodola iteration process can also be used to compute lower modes as well. Unless the first mode components are eliminated from the dynamic matrix [Eq.(II.15)], any given trial vector will always converge towards the first mode eigenpair. Equation (II.17b) states that any two distinct eigenvectors are orthogonal relative to the mass matrix. Rewriting Eq.(II.17b) for the first mode shape vector and any other mode shape vector j

$$\{Y\}_1^T [M] \{Y\}_j = 0 \qquad j = 2,\ldots,n$$

or

$$Y_{11}\, m_1\, Y_{1j} + \ldots + Y_{n1}\, m_n\, Y_{nj} = 0 \qquad \text{(II.25)}$$

where Y_{i1} are the coefficients of the first mode shape vector, Y_{ij} are the coefficients of the other mode shape vectors, and m_i are the coefficients of the diagonal mass matrix.

Solving for Y_{1j} from Eq.(II.25), substituting into dynamic matrix, and omitting the first equation, a new dynamic matrix of order (n-1)x(n-1) is obtained. This new matrix does not contain any first mode components, therefore any non-zero trial shape will converge to the highest mode of this new dynamic matrix, which is, in fact, the second mode of the original dynamic matrix. The elimination process described above is called "sweeping."

Computing the (n-1) elements of the new mode shape vector by Stodola method, and using Eq.(II.25)

Y_{12} may be computed.

From the orthogonality condition

$$\{Y\}_2^T [M]\{Y\}_j = 0 \qquad j = 3,\ldots,n$$

or

$$Y_{12}\, m_1\, Y_{1j} + Y_{22}\, m_2\, Y_{2j} + \ldots + Y_{n2}\, m_n\, Y_{nj} = 0$$

Y_{1j} may be solved and by substituting its value to (II.25) a new equation of order 1x(n-1) is obtained:

$$\{Y\}_1^T [M]\{Y\}_j = 0 \qquad j = 3,\ldots,n \qquad \text{(II.26)}$$

Solving for Y_{2j}, substituting its value into the dynamic matrix of order (n-1)x(n-1), and omitting the second equation a new dynamic matrix of order (n-2)x(n-2) is obtained.

The procedure can be repeated the same way to compute the frequencies and mode shapes for the third and lower modes, except for the last mode.

The order of the dynamic matrix reduces to one for the last (lowest) mode:

$$(FM)_{nn} - \frac{1}{\omega_n^2} Y_{nn} = 0$$

The choice for Y_{nn} is arbitrary and ω_n can be computed from

$$\omega_n = \frac{1}{\sqrt{(FM)_{nn}}}$$

Using all the previously stated orthogonality conditions, the highest mode shape vector may be constructed such that it is orthogonal to all the higher mode shape vectors.

The mode shapes of different building type plane frames are quite similar, hence it is easier to make a guess on the starting iteration vectors for the highest few modes (10). The mode shapes of arch type plane frames depend very much on their slenderness and aspect ratios, therefore a choice on the starting iteration vectors is very difficult to make (11). The properties of the mode shapes of arches and building frames will be discussed later in Chapter IV.

II.C SINGLE DEGREE-OF-FREEDOM VIBRATING SYSTEMS

The equation of motion for a single degree-of-freedom vibrating system is

$$m\ddot{y} + c\dot{y} + ky = f(t) \tag{II.27a}$$

Where

m : mass

c : damping

k : stiffness

f(t) : applied load function

$y, \dot{y}, \ddot{y}$: displacement, velocity and acceleration, of the moving body, respectively.

The idealized single degree of freedom system and its free body diagram is shown in Figure II.5.

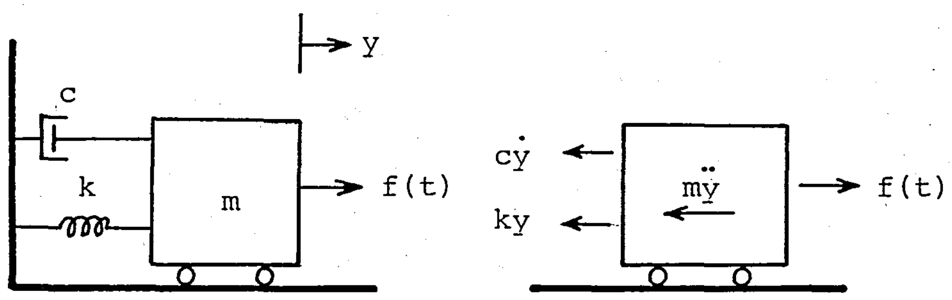


FIGURE II.5 Idealized Single Degree-of-Freedom System

It is more convenient to express Eq.(II.27a) in the following form

$$\ddot{y} + \frac{c}{m}\dot{y} + \frac{k}{m}y = f(t)/m \tag{II.27b}$$

or

$$\ddot{y} + 2\zeta\omega\dot{y} + \omega^2 y = a(t) \tag{II.27c}$$

where

ζ : damping ratio (ratio of damping to critical damping value),

ω : angular frequency of the system,

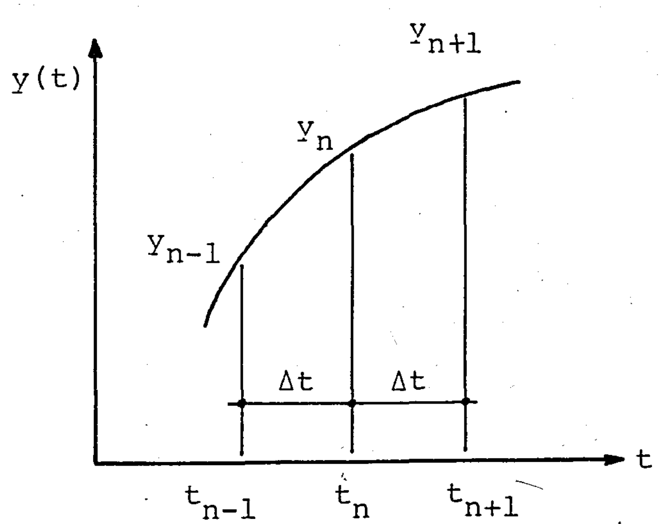
$a(t)$: input acceleration function.

In the case when the applied load represents an eartquake base excitation, a(t) is a complex function, therefore the solution to Eqs.(II.27) can not be obtained in closed form. Instead, a numerical integration scheme has to be employed.

The single step linear acceleration method is used to integrate the equation of motion for a single degree of freedom system, which assumes that the acceleration is linearly varying within a time interval Δt.

Taylor series expansion of the displacement function y(t) is

$$y(t_{n+1}) = \sum_{k=0}^{\infty} y^{(k)}(t_n) \frac{(t_{n-1}-t_n)^k}{k!} \tag{II.28}$$



Defining $\Delta t = t_{n+1} - t_n$ and knowing that the fourth and the higher order derivatives of Eq.(II.28) are equal to zero

$$y_{n+1} = y_n + \dot{y}_n \Delta t + \ddot{y}_n \frac{(\Delta t)^2}{2} + \dddot{y}_n \frac{(\Delta t)^3}{6} \tag{II.29a}$$

Differentiating Eq.(II.29a) with respect to time twice, one gets

$$\dot{y}_{n+1} = \dot{y}_n + \ddot{y}_n \Delta t + \dddot{y}_n \frac{(\Delta t)^2}{2} \tag{II.29b}$$

$$\ddot{y}_{n+1} = \ddot{y}_n + \dddot{y}_n \Delta t \tag{II.29c}$$

Substituting the value of $\dddot{y}_n$ from Eq.(II.29c) into Eqs.(II.29a) and (II.29b) the following set of equations is obtained

$$\dot{y}_{n+1} = \dot{y}_n + \frac{1}{2} (\ddot{y}_{n+1} + \ddot{y}_n) \Delta t \tag{II.30a}$$

$$y_{n+1} = y_n + \dot{y}_n \Delta t + (2\ddot{y}_n + \ddot{y}_{n+1}) \frac{(\Delta t)^2}{6} \tag{II.30b}$$

The Eqs. (II.30a) and (II.30b) can be substituted into Eq.(II.27c) and rearranging these three equations the following recurrence relationships are obtained

$$\ddot{y}_{n+1} = (a(t) - 2\zeta\omega A_n + \omega^2 B_n)/e$$

$$\dot{y}_{n+1} = A_n + \ddot{y}_{n+1} \frac{\Delta t}{2} \tag{II.31a}$$

$$y_{n+1} = B_n + \ddot{y}_{n+1} \frac{(\Delta t)^2}{6}$$

where

$$e = 1 + 2\zeta\omega \frac{\Delta t}{2} + \omega^2 \frac{(\Delta t)^2}{6}$$

$$A_n = \dot{y}_n + \ddot{y}_n \frac{\Delta t}{2} \tag{II.31b}$$

$$B_n = y_n + \dot{y}_n \Delta t + \ddot{y}_n \frac{(\Delta t)^2}{3}$$

Knowing the characteristics of system, and the acceleration, velocity and displacements at any time t_n, a new set of $y, \dot{y}$ and $\ddot{y}$ values can be computed at time t_{n+1}, using Eqs.(II.31) recursively. When the initial conditions $\dot{y}_o$ and y_o are known, the initial acceleration of the system can be computed by

$$\ddot{y}_o = a(t_o) - 2\zeta\omega\dot{y}_o - \omega^2 y_o \tag{II.32}$$

An important observation was that the cost of this integration analysis (i.e., the number of operations required) is directly proportional to the number of steps required for solution. It follows that the selection of an appropriate solution time step must be small enough so that accuracy is obtained in the solution. But on the other hand, the solution time step must not be smaller than necessary, because this would mean that the solution is more costly than actually required. Also, very small time step values may cause error accumulation in low precision computers.

The effect of the selection of solution time step on accuracy and stability of the integration, will be discussed in Chapter IV.

II.D MODE SUPERPOSITION METHOD

The generalized coordinate transformation, which serves to change the set of N coupled equations of motion of a multi degree-of-freedom system into a set of N uncoupled equations is the basis of the mode superposition of dynamic analysis. This method can be used to evaluate the dynamic response of any linear structure for which the displacements have been expressed in terms of a set of N discrete coordinates, and where the damping can be expressed by modal damping ratios. The procedure consists of the following steps.

STEP 1 : The equation of motion for a linear multi-degree-of-freedom system is expressed as

$$[M]\{\ddot{y}\} + [C]\{\dot{y}\} + [\bar{K}]\{y\} = - [M]\{U\}\ \ddot{d}_x \qquad \text{(II.33)}$$

where

$[M]$: mass matrix

$[C]$: damping matrix

$[\bar{K}]$: reduced stiffness matrix

$\ddot{d}_x$: input ground acceleration in the horizontal direction

$\{U\}$: a vector used to couple the mass coefficients representing the inertia of the system in horizontal direction with the input base acceleration

$\{\ddot{y}\},\{\dot{y}\},\{y\}$: the acceleration, velocity and displacement vectors representing the motion of dynamic degrees of freedom.

STEP 2 : For undamped free vibrations, the matrix equation (II.33) can be reduced to the eigenvalue equation

$$([\bar{K}] - \omega^2 [M]) \{Y\} = \{0\}$$

or

$$([\bar{F}][M] - \frac{1}{\omega^2} [I])\{Y\} = \{0\}$$

from which the free vibration mode shape vectors and frequencies can be computed.

STEP 3 : The state of the vibration of a multi-degree-of-freedom system can be expressed in terms of the orthonormalized mode shapes by the transformation

$$\{y\} = [\phi]\{\bar{y}\} \qquad \text{(II.34a)}$$

where $\{\bar{y}\}$ is the vector containing the modal amplitudes in generalized coordinates, and modal matrix $[\phi]$ serves to transform from the generalized coordinates $\{\bar{y}\}$ to the geometric coordinates $\{y\}$. Differentiating Eq.(II.34a) twice with respect to time, the velocity and acceleration vectors are found.

$$\{\dot{y}\} = [\phi]\{\dot{\bar{y}}\} \qquad \text{(II.34b)}$$

$$\{\ddot{y}\} = [\phi]\{\ddot{\bar{y}}\} \qquad \text{(II.34c)}$$

Substituting Eqs.(II.34) into Eq.(II.33)

$$[M][\phi]\{\ddot{\bar{y}}\} + [C][\phi]\{\dot{\bar{y}}\} + [\bar{K}][\phi]\{\bar{y}\} = -[M]\{u\}\,\ddot{d}_x$$

premultiplying the above equation by $[\phi]^T$, it becomes

$$[\phi]^T[M][\phi]\{\ddot{\bar{y}}\} + [\phi]^T[C][\phi]\{\dot{\bar{y}}\} + [\phi]^T[\bar{K}][\phi]\{\bar{y}\}$$
$$= -[\phi]^T[M]\{u\}\,\ddot{d}_x \qquad \text{(II.35)}$$

With the use of Eq.(II.20) and Eq.(II.23b), Eq.(II.35) simplifies to

$$\{\ddot{\bar{y}}\} + [\phi]^T[C][\phi]\{\dot{\bar{y}}\} + [\omega^2]\{\bar{y}\} = -[\phi]^T[M]\{u\}\,\ddot{d}_x$$

Practically, the matrix $[\phi]^T[C][\phi]$ is assumed to be diagonal, and the diagonal coefficients are represented by a fraction of the modal critical damping ratios. Matrix $[\omega^2]$ is also a diagonal matrix having the squares of the free vibration frequencies of the system as its diagonal elements.

Hence, the coupled set of linear differential equations given by Eq.(II.33) is reduced to a set of N uncoupled linear differential equations of order two, which are of the form

$$\ddot{\bar{y}}_k + 2\zeta_k\,\omega_k\,\dot{\bar{y}}_k + \omega_k^2\,\bar{y}_k = -r_k\,\ddot{d}_x \qquad \text{(II.36)}$$

where the subscript k denotes the mode number, and r_k is the participation factor for mode k which is defined to be

$$r_k = \{\phi\}_k[M]\{u\} \qquad \text{(II.37)}$$

STEP 4 : The resulting independent single-degree-of-freedom equations can be integrated by any suitable method depending on the type of loading.

STEP 5 : When response for each mode $\bar{y}_k(t)$ has been determined, the displacements, expressed in geometric coordinates, are computed by Eq.(II.34a) :

$$\{y\}_t = [\phi]\{\bar{y}\}_t$$

This equation may also be written in the following form

$$\{y\}_t = \{\phi\}_1\ \bar{y}_1(t) + \{\phi\}_2\ \bar{y}_2(t) + \ldots. \qquad \text{(II.38)}$$

It should be noted that for most types of loadings the contributions of the various modes generally are greatest for the highest frequencies and tend to decrease for the lower frequencies. Consequently, it usually is not necessary to include all the lower modes of vibration in the superposition process given by Eq.(II.38). The series can be truncated when the response has been obtained to any desired degree of accuracy. Moreover, it should be kept in mind that the mathematical idealization of any complex structural system also tends to be less reliable in predicting the higher modes of vibration; for this reason, too, it is desired to limit the number of modes considered in a dynamic response analysis.

II.E DIRECT STEP-BY-STEP INTEGRATION

Step-by-step integration procedure is an effective way of integrating the uncoupled equations of motion given by Eq.(II.33). Although the computation time is longer than the mode superposition method, in some cases it may be advantageous to use this method, since it does not require the evaluation of the vibration mode shapes and frequencies, which is a very time consuming computational task in systems with many degrees-of-freedom. In general, direct step-by-step integration tends to be most useful in evaluating the response of large, complex structures to short duration loads which tend to excite many modes of vibration but which requires that only a shorter response history is to be evaluated.

One potential difficulty in the step-by-step integration is that the damping matrix must be defined explicitly rather than in terms of modal damping ratios. It is very difficult to estimate the coefficients of a complete damping matrix. The most effective means for deriving a suitable damping matrix is to assume appropriate values for modal damping ratios which are considered to be important, and than to compute an orthogonal damping matrix described in the preceeding section.

On the other hand, defining the damping matrix explicity rather than by modal damping ratios, may be adventageous because it increases the generality of the step-by-step integration method.

The choice of the damping matrix and modal damping ratios is discussed in detail in Refs. (2),(12),(13) and (14).

In the following paragraphs the derivation of the recurrence equations for direct step-by-step integration procedure is given. It is assumed that acceleration varies linearly within the specified time interval Δt.

The velocity and displacement vectors in the general equation of motion [Eq.(II.33)], by analogy with the single-degree-of-freedom expressions [Eqs.(II.31)], can be expressed as

$$\{\dot{y}\}_{n+1} = \{\dot{y}\}_n + (\{\ddot{y}\}_{n+1} + \{\ddot{y}\}_n)\,\frac{\Delta t}{2}$$

and

$$\{y\}_{n+1} = \{y\}_n + \{\dot{y}\}_n\,\Delta t + (2\{\ddot{y}\}_n + \{\ddot{y}\}_{n+1})\,\frac{(\Delta t)^2}{6} \qquad \text{(II.39)}$$

Substituting Eqs.(II.39) into Eq. (II.33) and rearranging one obtains

$$[E]\{\ddot{y}\}_{n+1} + [C]\{A\}_n + [\bar{K}]\{B\}_n = -\,[M]\{u\}\,\ddot{d}_x(t_{n+1})$$

or

$$\{\ddot{y}\}_{n+1} = [E]^{-1}\,(-\,[M]\{u\}\,\ddot{d}_x(t_{n+1}) - [C]\{A\}_n - [\bar{K}]\{B\}_n) \qquad \text{(II.40a)}$$

where

$$[E] = [M] + [C]\,\frac{\Delta t}{2} + [\bar{K}]\,\frac{(\Delta t)^2}{6}$$

$$\{A\}_n = \{\dot{y}\}_n + \{\ddot{y}\}_n \frac{\Delta t}{2}$$

$$\{B\}_n = \{y\}_n + \{\dot{y}\}\Delta t + \{\ddot{y}\}_n \frac{(\Delta t)^2}{3}$$

The velocity and displacement recurrence relationships are:

$$\{\dot{y}\}_{n+1} = \{A\}_n + \{\ddot{y}\}_n \frac{\Delta t}{2} \tag{II.40b}$$

$$\{y\}_{n+1} = \{B\}_n + \{\ddot{y}\}_n \frac{(\Delta t)^2}{6} \tag{II.40c}$$

Knowing the mass, damping and stiffness properties of the system; acceleration, velocity and displacements for the dynamic degrees-of-freedom of the system can be computed by Eqs.(II.40) for any given base acceleration. The initial acceleration of the system can be computed by

$$\{\ddot{y}\}_o = [M]^{-1} (-[M]\{u\}\ \ddot{d}_x(t_o) - [C]\{\dot{y}\}_o - [\bar{K}]\{y\}_o) \tag{II.41}$$

where $\{\dot{y}\}_o$ and $\{y\}_o$ are the initial velocity and the displacement vectors, respectively.

Appropriate values for the solution time step Δt shall be selected considering the cost and accuracy of the solution. Some examples are given for building and arch type plane frames in Chapter IV.

II. F RESPONSE SPECTRA ANALYSIS

Numerical integration of the uncoupled linear differential equations of motion for every mode is a very time consuming operation. In engineering, generally, the maximum values are of interest. Therefore, for each mode to be superposed only the maximum values, namely displacements, forces, stresses, etc., are computed and superposed depending on a certain probabilistic approach. The maximum acceleration value detected during the integration of a single-degree-of-freedom system with a given frequency, subject to a certain earthquake motion is called to be the spectral acceleration value, S_a, at that frequency. Assuming the motion to be harmonic, the maximum velocity and maximum displacement can approximately be computed by

$$S_v = S_a/\omega \qquad \text{(II.42a)}$$

$$S_d = S_a/\omega^2 \qquad \text{(II.42b)}$$

For small damping ratios the maximum velocity and displacement values computed by Eqs.(II.42a) and (II.42b) are very close to the real maximum velocity and displacement values. For this reason they are called fictitious-spectral velocity and fictitious-spectral displacement.

When the spectral values for an earthquake motion are known, the spectral displacements belonging to a certain mode j can be computed by

$$\{y\}_j = \{\phi\}_j \, r_j \, S_{aj}/\omega_j^2 \qquad \text{(II.43a)}$$

or $$\{y\}_j = \{\phi\}_j \, r_j \, S_{vj}/\omega_j \qquad \text{(II.43b)}$$

or $$\{y\}_j = \{\phi\}_j \, r_j \, S_{dj} \qquad \text{(II.43c)}$$

Other response values for the structure can also be computed using similar formulas to Eqs.(II.43) for each mode.

The response values from each mode could have been superposed directly, if all terms in each summation were to be evaluated at the same instant in time. The response spectrum is a plot of maximum response for a spectrum of periods of vibration. The time at which this response occurs is not recorded. Therefore, the maximum response values defined in Eqs.(II.43) probably will occur at different times for each mode. If the absolute values of the responses for each mode are summed it is assumed that all the response spectrum maximum times are the same, which is probably never the situation.

The structure response values are combined by using approximate summation formulas. If the absolute values of the maximum modal responses are summed, this result gives an upper bound to the answer. Denoting any response in the j^{th} mode as P_j, it follows that the absolute sum response (peak response) is

$$P = |P_1| + |P_2| + \ldots \tag{II.44}$$

Another well known formula is called the root sum square and is given by

$$P = [(P_1)^2 + (P_2)^2 + (P_3)^2 + \ldots]^{1/2} \tag{II.45}$$

A third formula that combines Eqs.(II.44) and (II.45) in the same sense is

$$P = |P_1| + [(P_2)^2 + (P_3)^2 + \ldots]^{1/2} \tag{II.46}$$

There is no universally accepted formula for summing the modal responses.

It is customary not to retain all the terms in Eqs. (II.44) to (II.46), which this additional approximation reduces the computational effort considerably. This reduction in the number of terms to be summed is called modal truncation.

II.G EQUIVALENT STATIC LOADS

Another method to predict the modal responses is to define fictitious static loads that would yield the displaced shape of a given structure. Hence, the dynamic problem reduces to a static problem (10). Rearranging Eq.(II.33), it follows that

$$[M]\{\ddot{y}\} + [C]\{\dot{y}\} + [M]\{u\}\ \ddot{d}_x = -\ [\bar{K}]\{y\}$$

where the term on the left is the earthquake force vector. Therefore

$$\{F\} = -\ [\bar{K}]\{y\}$$

Using Eq. (II.34a), it follows that

$$\{F\} = -\ [\bar{K}][\phi]\{\bar{y}\}$$

Finally, using Eq. (II.21) the earthquake force is

$$\{F\} = -\ [\omega^2][M][\phi]\{\bar{y}\}$$

or

$$\{F\} \doteq -\ \left[\omega_1^2[M]\{\phi\}_1\ \bar{y}_1 + \omega_2^2\ [M]\{\phi\}_2\ \bar{y}_2 + \ldots\right]$$

The interpretation of the above formula is as follows: at any time t, the elastic forces acting on the structure is defined as a combination of the elastic forces due to each mode.

Hence

$$\{F\} = \{F\}_1 + \{F\}_2 + \ldots$$

where

$$\{F\}_k = - \omega_k^2 \; [M]\{\phi\}_k \; \bar{y}_k$$

Since the maximum response is investigated, the maximum modal forces for any mode k can be defined as

$$\max\ (\{F\}_k) = - \omega_k^2 [M]\{\phi\}_k \max(\bar{y}_k) \qquad \text{(II.47)}$$

The maximum value of $\bar{y}_k$ is equal to the participation factor r_k times the spectral displacement(15). In other words

$$\max(\bar{y}_k) = r_k \; S_{dk}$$

or in terms of the spectral acceleration

$$\max\ (\bar{y}_k) = r_k \; S_{ak}/\omega_k^2 \qquad \text{(II.48)}$$

Substituting Eq.(II.48) into Eq.(II.47) the equivalent maximum elastic force representing mode k becomes

$$\max\{F\}_k = -[M]\{\phi\}_k \; r_k \; S_{ak} \qquad \text{(II.49)}$$

Because the response spectrum does not retain the sign of the maximum response, all forces in Eq.(II.49) can be multiplied by a plus or minus sign.

Applying these static loads to the structural system, any response value can be computed using a static analysis program. The response values obtained for each mode can be superposed using one of the summation formulas given by Eqs.(II.44), (II.45) or (II.46).

III. DEVELOPED COMPUTER PROGRAMS

In this chapter, information is given on the computer programs, which were developed throughout this study. The computer programs PFRAME5, PFRAME6, EIG3, SPEC, MODAL and DSSI were developed on CDC Cyber 170/815 system (operating system: NOS 2.2) operating at Computer Center, Boğaziçi University, Istanbul

III.A STATIC ANALYSIS OF PLANE FRAME STRUCTURES : PRAME5 AND PFRAME6

The computer programs PFRAME5 and PFRAME6 were developed primarily for the static analysis of plane frame structures subject to nodal and member loads. In these programs stiffness (displacement) method of analysis is used, where the unknowns were chosen to be the nodal displacements. Fundamentals of the problem are given in Section II.A. For this program, practically, there is no limit to the size of the problem to be solved.

The control information, material properties, element type information and the boundary conditions are read before starting the frontal solution process. The frontal routine reads the connectivity and loading data for each member, generates the equation (code) numbers for each node, computes and element stiffness matrices, assembles the structure (system) stiffness matrix and forms the right-hand-side (load) vectors. The fully summed equations are eliminated before all the element stiffness matrices are assembled, whenever necessary. After all the equations are fully summed and eliminated, the unknown nodal displacements are computed by an out-of-core backsubstitution process. Finally, the end forces are computed for each member.

The generated equation numbers and the computed displacements are saved on disk files. If desired, these nodal displacements may later be used to construct the reduced flexibility matrix in frequency/mode shape computations.

The frontal solution algorithm used here has two basic short cuts: the half bandwidth of the structure stiffness matrix and the last element containing each node (last appearence of each equation) are not computed automatically by the program. Since, for high-rise framed structures the half bandwidth depends on the number of bays, for simplicity, its value is either supplied by the user, or it can be computed when the number of bays is known. The last appearance of every unknown during the assembly process depends on the number of bays, and on the numbering pattern of the elements. Simple formulas are derived for finding the fully summed equations at any stage of the elimination process.

As it was mentioned in Section II.A.2, frontal solution technique is an out-of-core solution algorithm which enables the analysis of structures with very large number of nodal unknowns, where it would be impossible by using an in-core solution algorithm. In the case when micro computers are used, which have limited main memory, but usually have peripheral memory of reasonable size, it becomes obligatory to employ out-of core solution algorithms, even when medium size problems are encountered. The program PFRAME5 can easily be installed to operate on micro computers which have about 256 kilo-bytes of random access memory (256 kBytes RAM).

Practically, there is no limit to the number of equations to be solved, as long as a rectangular array can be declared with number of columns equal to the half bandwidth, and number of rows greater than or equal to the half bandwidth. Clearly for the cases when the number of equations that can be stored simultaneously in the main memory is larger than the half bandwidth, the total computation time will considerably be reduced.

The computer program PFRAME6 is a modified version of PFRAME5. It was modified to analyse towers and arch type plane frames, where the half bandwidth for the problem is supplied by the user. With the necessary modifications it became possible to define nodal concentrated loads, whereas it was not for PFRAME5.

Data input description and a listing of program PFRAME5 are given in Appendix A. The algorithm for PFRAME5 is given below.

1. Read the control information.
2. Read the boundary conditions, and store on file TAPE10.
3. Read the element cross-sectional properties and generate element stiffness properties for each type of element which would give a different element stiffness matrix. Write these element stiffness matrices on file TAPE11.
4. Read the element connectivity and type data.
5. Generate the equation numbers belonging to the nodes of the element.
6. Read the element stiffness matrix from file TAPE11.
7. Assemble the element stiffness matrix into the system stiffness matrix.
8. Read element loading data, compute fixed-end forces in local coordinates, transform the

fixed-end forces to global coordinates and substract from the right-hand side (load) vectors.

9. Write fixed-end forces, element type and code numbers on file TAPE12.
10. If there is sufficient memory left for the further assembly of the next element's equations go to Step 4.
11. Eliminate the fully summed rows
12. Write the fully summed rows of the system stiffness matrix and the right-hand side(load) vectors on files TAPE13 and TAPE14, respectively.
13. Shift all the incomplete equations up, to "free" space in the main memory for further assembly.
14. If there are other elements waiting for assembly, go to Step 4.
15. Perform the backsubstitution process.
16. Read the element equation numbers, type, and fixed-end forces from file TAPE12.
17. Read the element stiffness matrix from file TAPE11.
18. Read element nodal displacements from file TAPE14.
19. Compute and print the end forces for the element.
20. If any element is left, go to Step 16.
21. Stop.

III.B FREQUENCY/MODE SHAPE COMPUTATION: EIG3

The computer program EIG3 was developed to compute the frequencies and orthonormalized modeshape vectors of a given system with known reduced flexibility and diagonal mass matrices.

EIG3 reads the heading of the problem, order of the free vibration problem and massinformation. It forms the massmatrix and constructs the flexibilty matrix by reading the displacements and rotations from file TAPE14 which were previously written by programs PFRAME5 or PFRAME6.

Flexibility and mass matrices are multiplied to get the dynamic matrix given by Eq.(II.15). To compute the eigenvalues and eigenvectors, subroutine EISRG1 of Cern Computer Centre Program Library I is used. EISRG1 is a subroutine which computes all eigenvalues and eigenvectors of a given general real matrix by calling routines in the EISPACK package.

Finally, the frequencies, periods,participation factors are computed and all the mode shape vectors are orthonormalized relative to the mass matrix. The free vibration frequencies, participation factors and orthonormalized mode shape vectors for each mode are written on file TAPE20 which may be used later for time-history analysis by mode superposition method,if desired.

III.C SPECTRAL ACCELERATION COMPUTATION : SPEC

The computer program SPEC was developed to compute the response of a single degree-of-freedom system to an arbitrary loading. Linear acceleration method (Section II.C) was used to integrate the equation of motion given by Eq.(II.27c).

Program SPEC reads the time step, periods and the corresponding damping ratios, and acceleration function definition points. The accelerations, velocities and displacements are computed at every time step, and finally the spectral acceleration values, the step and time at which the maximums were detected are printed for each of the input period and damping ratio pairs. The computed acceleration, velocity and displacement values can also be printed depending on the choice of the user.

Data input description and listing of program SPEC are given in Appendix B.

III.D RESPONSE-HISTORY ANALYSIS BY MODE SUPERPOSITION: MODAL

Computer program MODAL evaluates the dynamic response of any linear structure to any given earthquake accelerogram by mode superposition method. Angular frequencies, participation factors and orthonormalized mode shapes for each mode to be superposed must be known prior to the execution of MODAL.

The program MODAL reads the earthquake accelerogram, integration step size, and the modal damping ratios. Previously computed frequencies, participation factors and mode shape vectors are read from file TAPE20 ; therefore, is must be executed after execution of EIG3. The response of the uncoupled equations of motion to the input earthquake accelerogram is computed by the linear acceleration method. The geometric displacements are computed by the superposition of any desired number of modes, at every time step. The maximum values of the displacements in the directions of each of the prescribed dynamic degrees-of-freedom are detected throughout the integration.

These maximum displacement values and the times at which these maximums have occured are printed at the end of the integration. The computed generalized accelerations, velocities and displacements, along with the displacements in geometric coordinates can also be printed at every integration step, depending on the choice of the user. Also the displacements are printed for those steps required by the user.

The procedure for program MODAL was given in Section II.D. Data input description and program listing of MODAL are given in Appendix C.

III.E RESPONSE HISTORY ANALYSIS BY DIRECT STEP-BY-STEP INTEGRATION : DSSI

The computer program DSSI was developed to compute the response of a structural system to any given earthquake accelerogram, by linear acceleration method. The user shall supply the mass information, earthquake accelerogram, andthe solution time step. DSSI constructs the diagonal mass matrix, forms the reduced flexibility matrix which was previously computed and stored on file TAPE14. Hence, PFRAME5, or PFRAME6 must be executed prior to the execution of DSSI.

It inverts the reduced stiffness matrix, and also computes the damping matrix. Rayleigh damping matrix formulation is used, where the damping matrix is assumed to be a combination of the mass and reducedstiffness matrices (6,12).

Rayleigh showed that a damping matrix of the form

$$[C] = \alpha\ [M] + \beta\ [K] \qquad \text{(III.1)}$$

where α and β are proportionality factors, will satisfy the orthogonality condition given in Section II.D.

The displacements, velocities and accelerations of each mass are computed by the recurrance relationships given by Eqs.(II.40). The initial velocities and displacements are assumed to be equal to zero, and initial accelerations are computed by Eq.(II.41).

The maximum displacements, time and step at which these maximum values are detected are printed at the end of the execution. The computed displacements, velocities and accelerations at every step can also be printed depending on the choice of the user.

The algorithm of the program DSSI is given below.

1. Read the system parameters and control information.
2. Read the earthquake accelerogram.
3. Read the mass coefficients, construct the diagonal mass matrix $[M]$, form the reduced flexibility matrix $[\bar{F}]$, formthe $\{u\}$ vector.
4. Invert $[\bar{F}]$ to obtain the reduced stiffness matrix $[\bar{K}]$.
5. Compute the Rayleigh damping matrix [Eq.(III.1)] .
6. Compute matrix $[E] = [M] + [C]\dfrac{\Delta t}{2} + [\bar{K}]\dfrac{(\Delta t)^2}{6}$ and invert $[E]$ to obtain $[E]^{-1}$.
7. Set the initial displacements and velocities equal to zero, and compute the initial acceleration vector by Eq.(II.41).

8. Increase time ($t_n = t_{n-1} + \Delta t$). If the integration time t_n is greater than maximum integration time t_{max}, than goto Step 15.

9. Compute $\{A\}_n = \{\dot{y}\}_n + \{\ddot{y}\}_n \dfrac{\Delta t}{2}$

$$\{B\}_n = \{y\}_n + \{\dot{y}\}_n \Delta t + \{\ddot{y}\}_n \frac{(\Delta t)^2}{3}$$

10. Compute base acceleration $\ddot{d}_x(t_{n+1})$ by interpolation.

11. Compute the acceleration vector from Eq.(II.40a).

12. Compute the velocity and displacement vectors from Eqs.(II.40b) and (II.40c), respectively.

13. Check whether a maximum displacement value has occured.

14. Go to step 8.

15. Print the maximum displacement values occured at every dynamic degree-of-freedom.

16. Print the displacement vectors at times when these maximum values have occured.

17. Stop.

Data input description and program listing for DSSI are given in Appendix D.

IV. APPLICATIONS

The case studies in this chapter are given in two sections. In the first section, cases analysed for the verification of the computer programs developed throughout this study, are presented.In the second section, other case studies are given, in which, various aspects of the dynamic problem are discussed. The problems were solved on the CDC Cyber 170/815 system operating at the Computer Center, Boğaziçi University, Istanbul. All of the problems were solved using single precision arithmetic. A single precision word is 60 bits long in this system.

IV.A CASES ANALYSED FOR THE VERIFICATION OF THE COMPUTER PROGRAMS

In this section, three problems are given, which were solved to test the performances of the computer programs PFRAME5 and EIG3. The first two are examples on high-rise building frames subjected to lateral fictitious static loads. In the third example, the free vibration periods of an arch type plane frame are computed. The results are compared with those obtained from Refs.(11) and (16). The program was forced to use the out-of-core frontal method when solving the two high-rise building frames given in Sections IV.A.1 and IV.A.2.

IV.A.1 THREE-BAY, 20-STOREY BUILDING FRAME

The plane frame system shown in Fig. IV.A.1 is analysed by the computer program PFRAME5. The static lateral loads, which represent the fictitious earthquake forces, were computed based on a triangular displacement distribution assumption. The modulus of elasticity E was assumed to be 2000000 t/m^2. Three displacements (two translations and one rotation) are allowed at each node, except at the supports. The number of unknown displacements and the half bandwidth are 240 and 15, respectively. The execution time to solve this problem was 14.5 CP (Central processing) seconds.

The horizontal displacement distribution and their values are given in Fig. IV.A.2. This problem was previously solved on a Borroughs 3600 system (16). The results obtained from Ref.(16) are also given in the same figure. The results were about the same; the slight difference is due to the different word size of the computers.

It is clearly seen from Fig.IV.A.2 that, the higher storeys are undergoing "shear-type" deformations, whereas the lower storeys are undergoing "bending-type" deformations.

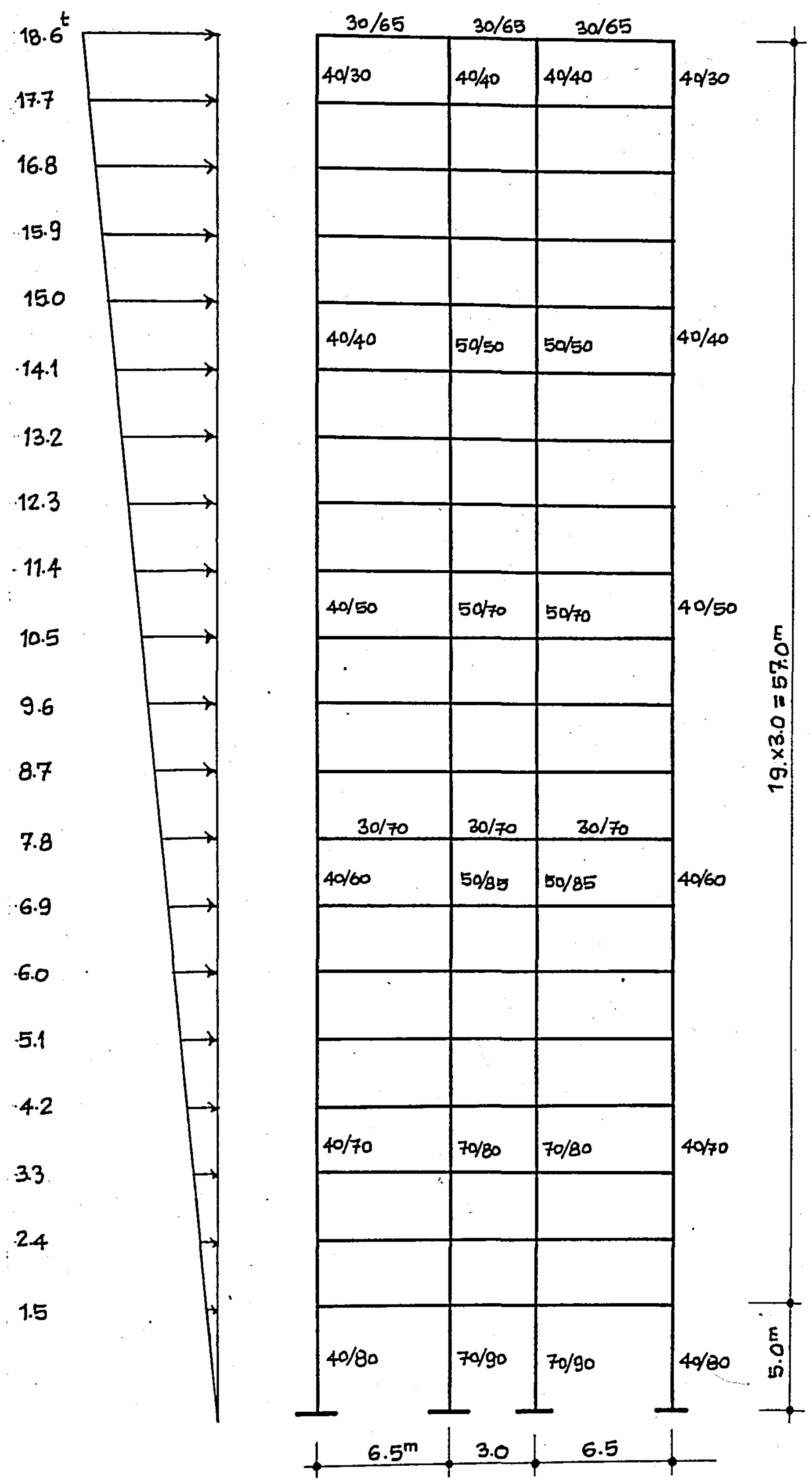


FIGURE IV.A.1 Three-Bay, 20-Storey Building Frame

Storey No	Ref.(16)	PFRAME5
20	0.3887	0.3885
19	0.3744	0.3742
18	0.3558	0.3556
17	0.3331	0.3329
16	0.3065	0.3063
15	0.2855	0.2853
14	0.2627	0.2626
13	0.2384	0.2382
12	0.2126	0.2125
11	0.1917	0.1916
10	0.1703	0.1702
9	0.1486	0.1485
8	0.1277	0.1277
7	0.1103	0.1102
6	0.0931	0.0931
5	0.0764	0.0763
4	0.0602	0.0601
3	0.0454	0.0454
2	0.0312	0.0312
1	0.0177	0.0176

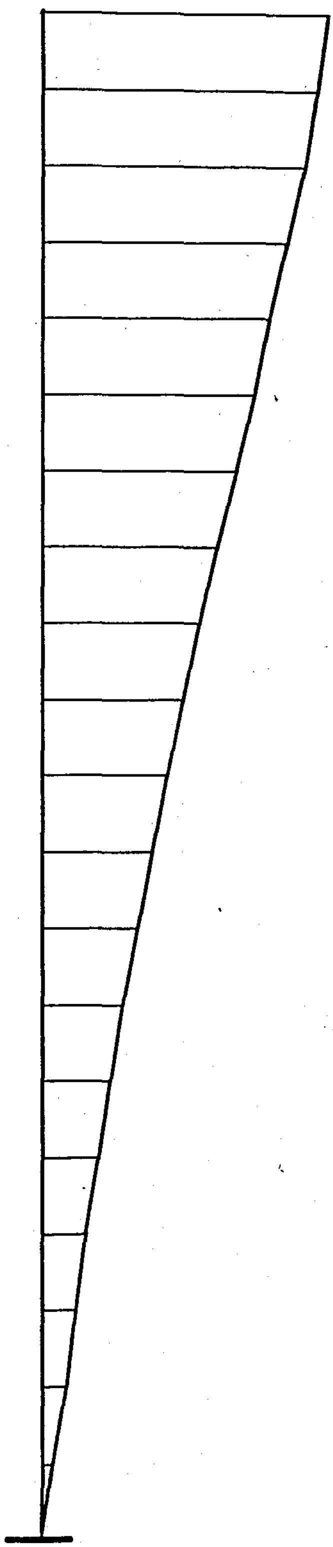


FIGURE IV.A.2 Horizontal Displacement Distribution of the Problem Given in Section VI.A.1

This is due to the different beam-to-column moment of inertia ratios. For the higher storeys; moment of inertias of the beams are larger than the moment of inertias of the colums; however, this situation is just the opposite for the lower storeys. Nevertheless, the triangular displacement distribution assumption seems to be quite reasonable for this case.

IV.A.2 THREE-BAY, 100-STOREY BUILDING FRAME

The hypothetical plane frame system shown in Fig. IV.A.3. is derived from the system given in Section IV.A.1 by extending every storey of that system to five storeys. The variation of the cross-sectional properties of the columns and the beams along the structure is very similar to the system in the preceeding section. The loading pattern of the fictitious earthquake forces is also similar; one at every five storeys. The material properties and the boundary conditions are similar as well. The number of unknowns and the half bandwidth are 1200 and 15, respectively.

The results obtained from Ref.(16) and the solution of PFRAME5 are given in Table IV.A.1. The double precision results from Ref.(16) are very close to the results of PFRAME5, however, the single precision results are not. Since the size of a double precision real variable is twice the size of a single precision variable, it is

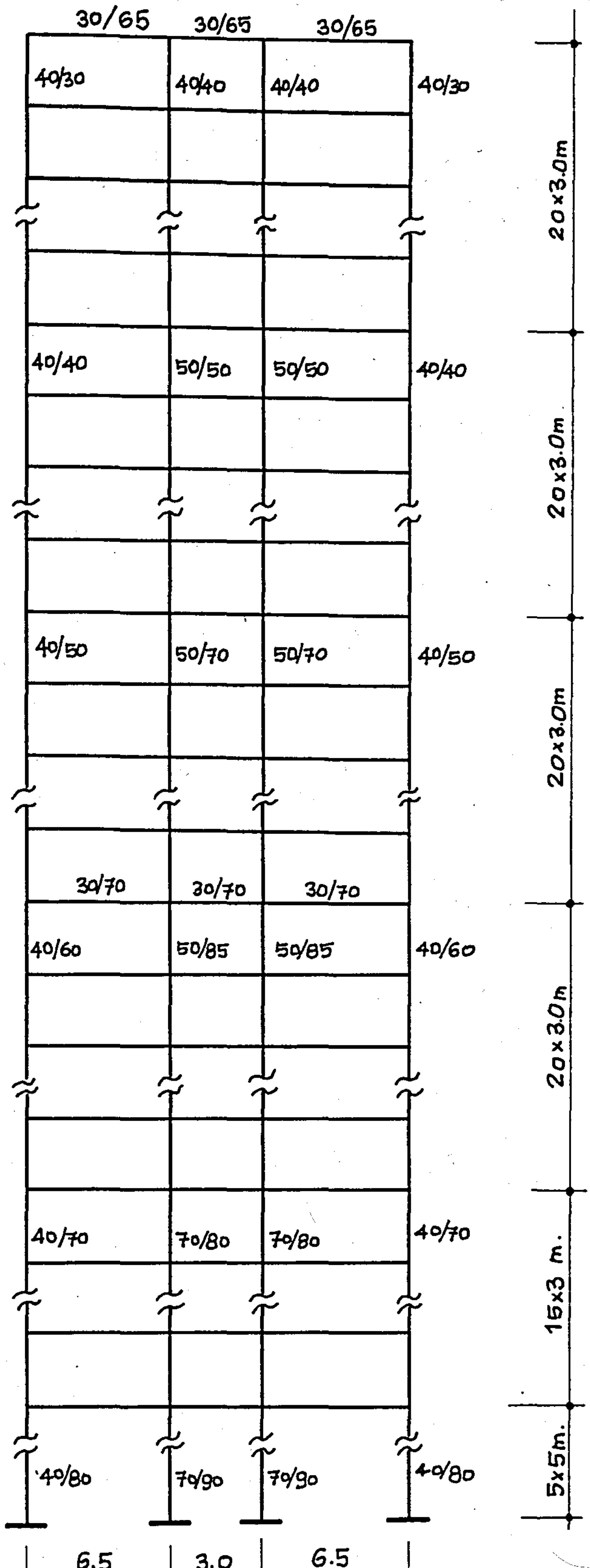
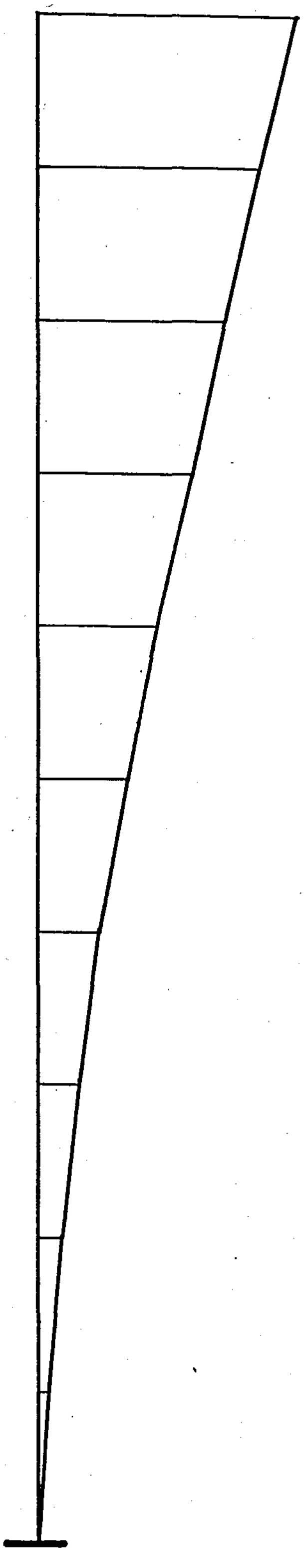


FIGURE IV.A.3 Three-Bay, 100-Storey Building Frame

Storey No.	Ref. (16)		PFRAME5 (m)
	Single Precision (m)	Double Precision (m)	
100	21.21	17.43	17.4888
90	18.37	15.08	15.1335
80	15.48	12.67	12.7190
70	12.67	10.35	10.3923
60	9.93	8.11	8.14181
60	7.43	6.06	6.08652
40	5.16	4.22	4.23270
30	3.27	2.67	2.68399
20	1.76	1.44	1.44836
10	0.71	0.58	0.581719
Execution Time	80 secs.	134 secs.	54.140 CP secs.

TABLE IV.A.1 Horizontal Displacements of the System in Fig.IV.A.3



Storey No	PFRAME5 (m)
100	17.4888
90	15.1335
80	12.7190
70	10.3923
60	8.14181
50	6.08652
40	4.23270
30	2.68399
20	1.44836
10	0.581719

FIGURE IV.A.4 Horizontal Displacement Distribution of the Problem Given in Section IV.A.2.

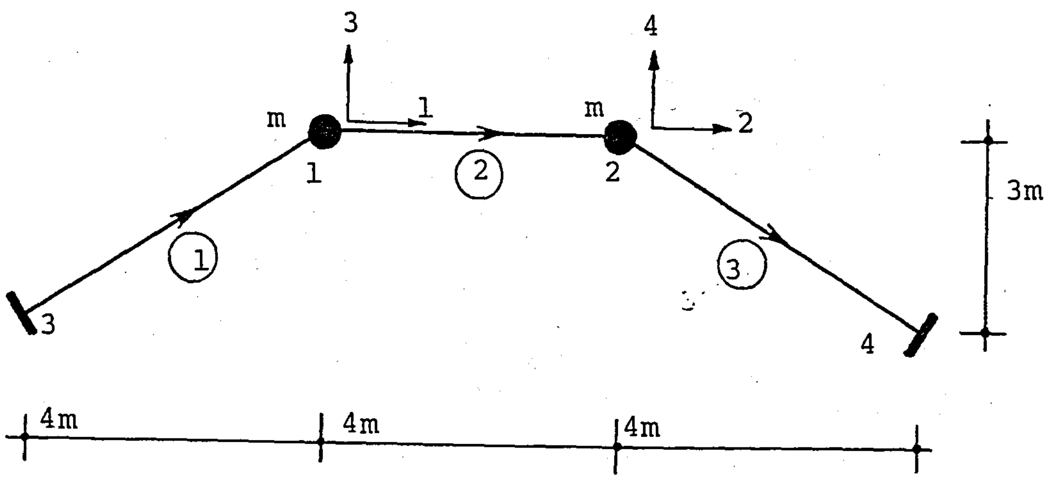
clear that the error accumulation due to rounding is extensively large in the single precision case.

The horizontal displacement distribution of the solution of PFRAME5 is given in Fig. IV.A.4. The lower storeys are undergoing "bending type" deformations; however, the displacement distribution is almost linear for the higher storeys. It is clear that the horizontal displacements are not increasing linearly along the structure; therefore, the triangular displacement distribution assumption is not valid for this case.

IV.A.3 ARCH TYPE PLANE FRAME

The periods of vibration of the arch type plane frame given in Fig.IV.A.5 are computed by the computer program EIG3. The same problem was previously solved on a Borroughs 3600 system(11). Two translational dynamic degrees-of-freedom are selected at every node, except at the supports. The massof the system is assumed to be concentrated at nodes 1 and 2.

The results obtained from Ref.(11) and the results of EIG3 are given in Table IV.A.2. The execution takes less than a CP second on the CDC Cyber 170/815 computer. The computed periods of Ref.(11) and EIG3 are exactly the same up to four digits after the decimal point.



Element no. 1 and 3 : EI = 22680 tm^2, EA = 476280 t.

Element no. 2 : EI = 11340 tm^2, EA = 378000 t.

m = 0.435 t sec^2/m

FIGURE IV.A.5 Arch Type Plane Frame of Section IV.A.3

Mode No.	Ref.(11)	EIG3 (sec)
1	0.1565	0.156484
2	0.0414	0.0413567
3	-	0.0184287
4	-	0.0112753

TABLE IV.A.2 Periods of Vibration of the System Given in Fig.IV.A.5

IV.B OTHER CASE STUDIES

Various case studies are given in this section, which were examined to point out the characteristics of; single-degree-of-freedom and multi-degree-of-freedom vibrating systems subjected to lateral loads. Two types of multi-degree-of-freedom systems are investigated; multi-storey and arch type plane frames.

Several aspects of the free and forced vibration of these systems were considered, including; effect of solution time step and number of modes to be superposed on the accuracy of the solution, and for arch type structures; effect of slenderness and aspect ratios to the symmetry or antimetry of the mode shapes, and to the value of the participation factors. It is usually easier to guess roughly the mode shapes of high-rise buildings; however, more uncertainties are involved with arches. The last two examples are on the response spectra analysis procedure, and on the fictitious static loads approach described in Section II.G.

Latino Americana Tower earthquake accelerogram is used in the forced vibration analyses (17). The structures in all examples are assumed to have no damping. The equations of motion are integrated by linear acceleration method.

IV.B.1 SPECTRAL ACCELERATIONS FOR VARIOUS PERIODS OF VIBRATION

Undamped spectral accelerations were computed for various periods of vibration using the computer program SPEC. Numerical integrations are carried out for several different solution time steps. The spectral accelerations for smaller periods of vibration are given in Table IV.B.1.. Values for larger periods of vibration are given in Table IV.B.2. Numbers enclosed in parantheses are the times (in seconds) at which these maximums are detected.

It has been observed from Table IV.B.1 that, for periods less than one second, a solution time step of about 0.01 second gives results within three per cent accuracy. However, for a time step value of 0.1 second, the error in the solution varies from 45 per cent up to 110 per cent. Hence, for this range of free vibration periods it may be recommended to select time step values less than or equal to 0.01 second.

For free vibration periods greater than or equal to two seconds (Table IV.B.2), solution time steps of 0.1 second or less gives results within 1.2 per cent accuracy. For this range of periods, solution time steps less than 0.1 second can safely be selected.

Δt (sec)	T = 0.20 (sec)	T = 0.40	T = 0.6	T = 0.8	execution time
0.10	0.812037 (30.8)	0.517121 (32.9)	-.0802375 (29.8)	-0.583594 (19.4)	1.669 (CP secs)
0.06	-.506043 (58.74)	-0.443263 (31.68)	0.443487 (36.72)	1.03866 (58.26)	2.127
0.01	0.397492 (46.63)	0.365669 (23.98)	0.769837 (58.21)	-1.53932 (52.80)	7.760
0.005	0.370657 (33.755)	0.361173 (23.970)	0.752674 (58.200)	-1.54433 (52.795)	14.500
0.001	0.386594 (33.952)	0.358933 (23.698)	0.746464 (58.195)	-1.54608 (52.791)	68.418
0.0005	0.387150 (33.7515)	0.358863 (23.9675)	0.746268 (58.1955)	-1.54613 (52.7910)	136.975
0.0001	0.387335 (33.7516)	0.358863 (23.9676)	0.746206 (58.1952)	-1.54614 (52.7912)	682.678

TABLE IV.B.1 Spectral Accelerations for Small Periods of Vibration (m/sec^2)

Δt (sec)	T = 1.10 (sec)	T = 2.00	T = 2.60	T = 3.10	execution time
0.10	0.7813370 (48.9)	1.544301 (52.8)	2.897893 (52.5)	1.429067 (57.9)	1.690 (CP secs)
0.05	0.6528428 (21.95)	1.560772 (52.70)	2.910700 (52.50)	1.427083 (57.90)	2.390
0.01	0.6665296 (21.92)	1.562965 (52.49)	2.907498 (52.49)	1.425107 (57.89)	7.855
0.001	0.6670427 (21.915)	1.563021 (52.662)	2.907426 (52.486)	1.425089 (57.885)	68.911

TABLE IV.B.2 Spectral Accelerations for Larger Periods of Vibration (m/sec^2)

The execution times are found to be increasing as the solution time step decreases. The ratios of the execution times are about equal to the ratios of the solution time steps.

IV.B.2 ONE-BAY, FIVE-STOREY BUILDING FRAME

Free and forced vibration analyses of the building frame (Fig.IV.B.1) are carried out by the computer programs PFRAME5, EIG3, MODAL and DSSI (Chapter III). The cross-sectional properties of the elements of this system are given in Table IV.B.3.

One horizontal dynamic degree-of-freedom is selected at every storey, and the reduced flexibility matrix is computed by loading each dynamic degree-of-freedom by a unit force, seperately. The reduced flexbility, mass and dynamic matrices are given in Table IV.B.4.

The free vibration parameters, namely, frequencies, periods, participation factors, and the orthonormalized mode shapes are given in Table IV.B.5. The mode shapes of this system are shown in Fig.IV.B2. It can be seen from Table IV.B.5 that the participation factors; have non-zero values, and neglecting their signs, are sorted in decreasing order.

The step-by-step integration is carried out for various solution time steps. The maximum displacements

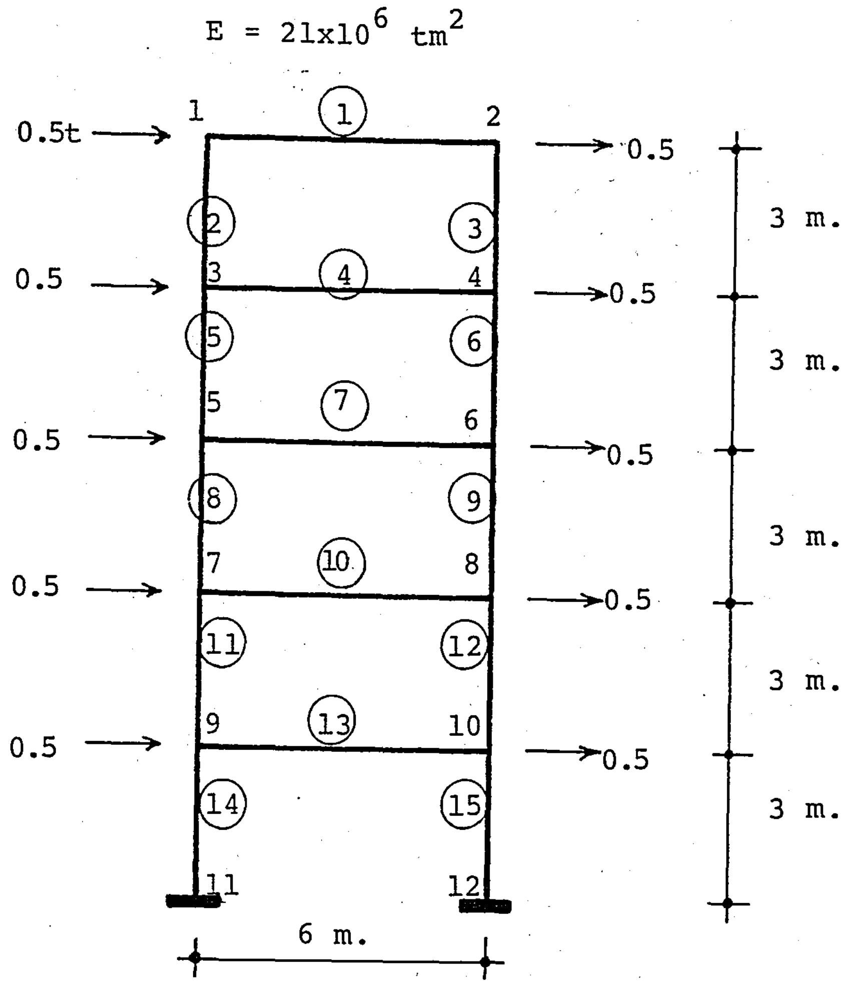


FIGURE IV.B.1 One-Bay, Five-Storey Building Frame

TYPE NO	CROSS SECTION	ELEMENTS	AREA (cm^2)	MOM.INERT (cm^4)	cosα	sinα
1	NP32	1,4	77.7	12510	1.0	0.0
2	NP34	2,3,5,6	86.7	15700	0.0	1.0
3	NP34	7,10	86.7	15700	1.0	0.0
4	NP36	8,9,11,12	97.0	19600	0.0	1.0
5	NP36	13	97.0	19600	1.0	0.0
6	NP38	14,15	107.0	24010	0.0	1.0

TABLE IV.B.3 Cross-Sectional Properties of the Elements of the Frame Given in Fig.IV.B.1

$[\bar{F}] =$ $*10^{-3}$ m/t

7.194359	5.510706	3.676370	2.039281	.6694189
	4.939812	3.538369	2.003419	.6630759
		3.0856696	1.901652	.6476206
(symmetric)			1.532220	.5951565
				.3886265

$[M] =$ t-sec^2/m

7.894	0	0	0	0
	11.841	0	0	0
		11.841	0	0
			11.841	0
(symmetric)				11.841

$[\bar{F}][M] =$ $*10^{-3}$ sec^2

56.79227	65.25227	43.53190	24.14713	7.926589
	58.49231	41.89783	23.72248	7.851481
		36.53772	22.51747	7.668476
(symmetric)			18.14302	7.047248
				4.601726

TABLE IV.B.4 Reduced Flexbility, Mass and Dynamic Matrices of the Five Storey Building Frame

Mode No.	1	2	3	4	5
ω (rad/sec)	2.56427	8.06166	15.2719	23.4906	31.6202
T (sec)	2.45028	.779391	.411421	.267477	.198708
r	6.57153	− 2.60756	1.72528	1.22202	−.896515
{φ}	.205013	.206484	.169649	−.109323	−.0357772
	.177536	.0331116	−.127858	.171675	.0775695
	.133280	−.135303	−.110326	−.131947	−.137114
	.0799418	−.17220	.128967	−.0325827	.175236
	.0275478	−.083460	.141821	.168940	−.167553

TABLE IV.B.5 Free Vibration Parameters of the Five-Storey Building Frame

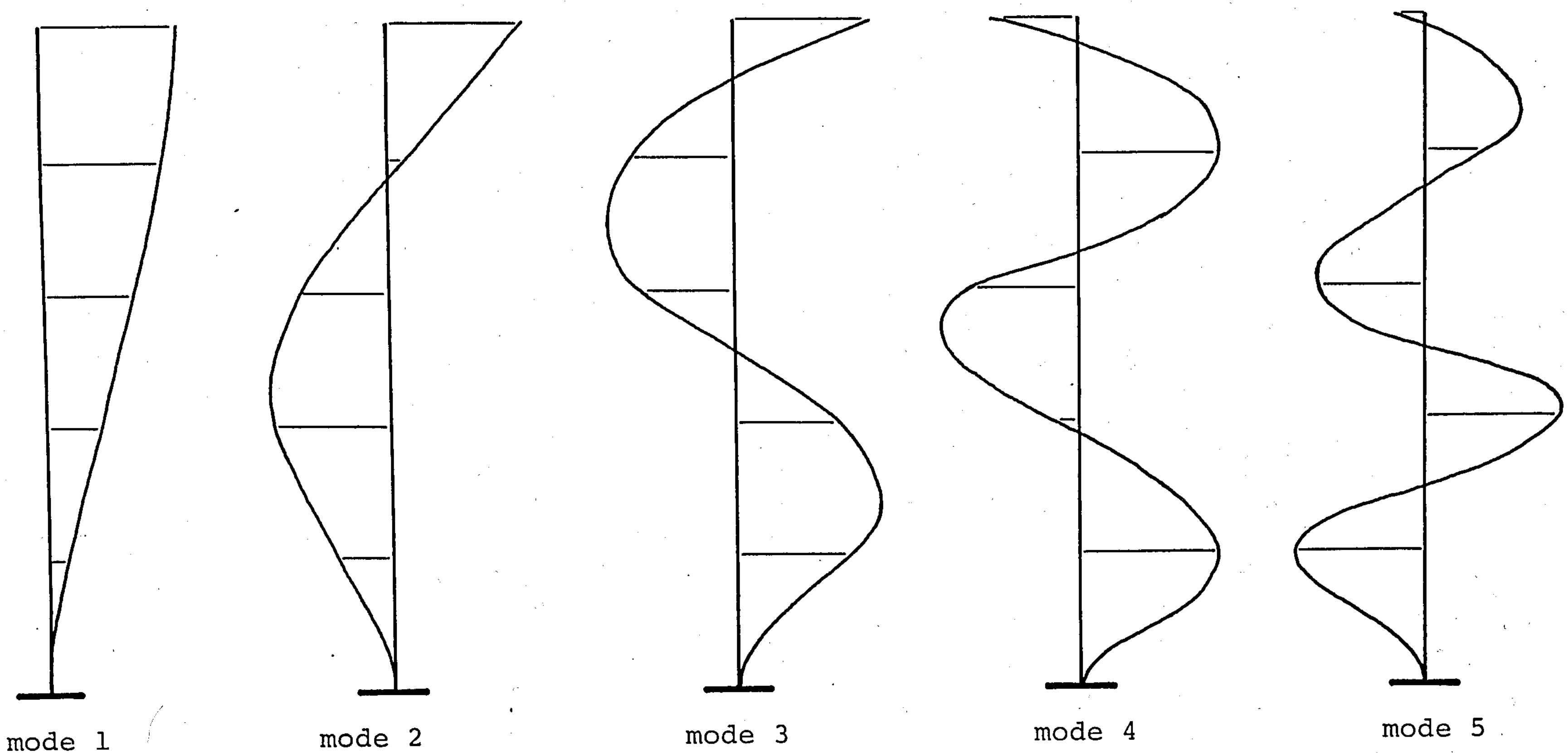


FIGURE IV.B.2 Mode Shapes of the Five-Storey Building Frame computed for each of the dynamic degrees-of-freedom, and the times at which these maximums have occured are given in Table IV.B.6. The weighted errors are computed relative to the results obtained for the smallest solution time step. It can be seen from this table that, as the time step decreases the errors in the solutions also decrease. Nevertheless, the errors are less than 1.5 per cent for time steps less than or equal to 0.05 second.

Maximum displacement values computed by modal analsis are given in Table IV.B.7. It is clearly seen that the accuracy of the results increase rapidly as the number of modes participating to the modal summation increases. Summing the responces of the first two modes, the error in the results decrease to about 0.1 per cent, which is a quite high level accuracy, as far as the uncertainties in the earthquake penomenon are considered. Instead of step-by-step integration, carrying out a modal analysis including only the first two modes reduces the execution time more than 50 per cent.

Displacements at a certain time t=58.385 seconds computed by modal analysis and step-by-step integration are given in Table IV.B.8. The errors in the solution decrease as the number of modes included in the modal summation increases. The horizontal displacement distribution along the structure is shown in Fig. IV.B.3.

time step (sec)	Δt = 0.05		Δt = 0.01		Δt = 0.005		Δt = 0.001	
execution time (CP secs.)	7.842		34.136		67.486		333.710	
Storey	$\{y\}_{max}$ (m)	t (sec)	$\{y\}_{max}$ (m)	t (sec)	$\{y\}_{max}$ (m)	t (sec)	$\{y\}_{max}$ (m)	t (sec)
5	-.341137	58.45	-.337389	58.39	-.336744	58.385	-.336531	58.386
4	-.289691	58.45	-.286230	58.41	-.286016	58.410	-.285960	58.411
3	-.214554	58.40	-.210583	59.69	-.211114	59.695	-.211251	59.693
2	.130289	57.10	.128483	59.73	.128947	59.730	.129138	59.728
1	.0460746	57.10	.0451889	59.75	.0452513	59.735	.0452463	59.734
Weighted Error (%)	1.351		1.235		0.060		-	

TABLE IV.B.6 Maximum Displacements Computed by DSSI for the Five Storey Plane Frame

(Δt = 0.005 sec.)

Modes Superposed	1		1+2		1+2+3		1+2+3+4		1+2+3+4+5		DSSI	
Execution Time (CP sec)	26.143		30.862		36.000		41.352		46.103		67.486	
Storey No.	$\{y\}_{max}$ (m)	t(sec)	$\{y\}_{max}$ (m)	t (sec)	$\{Y\}_{max}$ (m)	t (sec)	$\{Y\}_{max}$ (m)	t (sec)	$\{Y\}_{max}$ (m)	t (sec)	$\{Y\}_{max}$ (m)	t (sec)
5	-.328473	58.415	-.337138	58.385	-.336714	58.385	-.336732	58.385	-.336744	58.385	-.336744	58.385
4	-.284449	"	-.285694	58.410	-.286023	58.410	-.286035	58.410	-.286016	58.410	-.286016	58.410
3	-.213541	"	.211337	59.690	.211130	59.695	.211075	59.695	.211114	59.695	.211114	59.695
2	-.128083	"	.128892	59.730	.128952	59.725	.128946	59.730	.128947	59.730	.128947	59.730
1	-.0441373	"	.0452939	59.755	.0452564	59.740	.0452607	59.735	.0452513	59.735	.0452513	59.735
Weighted Error (%)	1.41289		0.10283		0.00626		0.00798		0.000		-	

TABLE IV.B.7 Maximum Displacements Computed by MODAL for the Five-Storey Plane Frame

(Δt = 0.005 sec.)

Modes Superposed		1	1+2	1+2+3	1+2+3+4	1+2+3+4+5	DSSI
$\{y\}_t$ (m)	5	-.327374	-.337138	-.336714	-.336732	-.336744	-.336744
	4	-.283498	-.285063	-.285383	-.285355	-.285328	-.285328
	3	-.212827	-.206429	-.206705	-.206727	-.206774	-.206774
	2	-.127655	-.119511	-.119188	-.119193	-.119133	-.119133
	1	-.0439896	-.040043	-.0396881	-.0396602	-.0397179	-.0397179
Weighted Error (%)		3.0421	0.1728	0.0242	0.0206	0.0	-

TABLE IV.B.8 Displacements of the Five Storey Plane Frame at t = 58.385 secs.

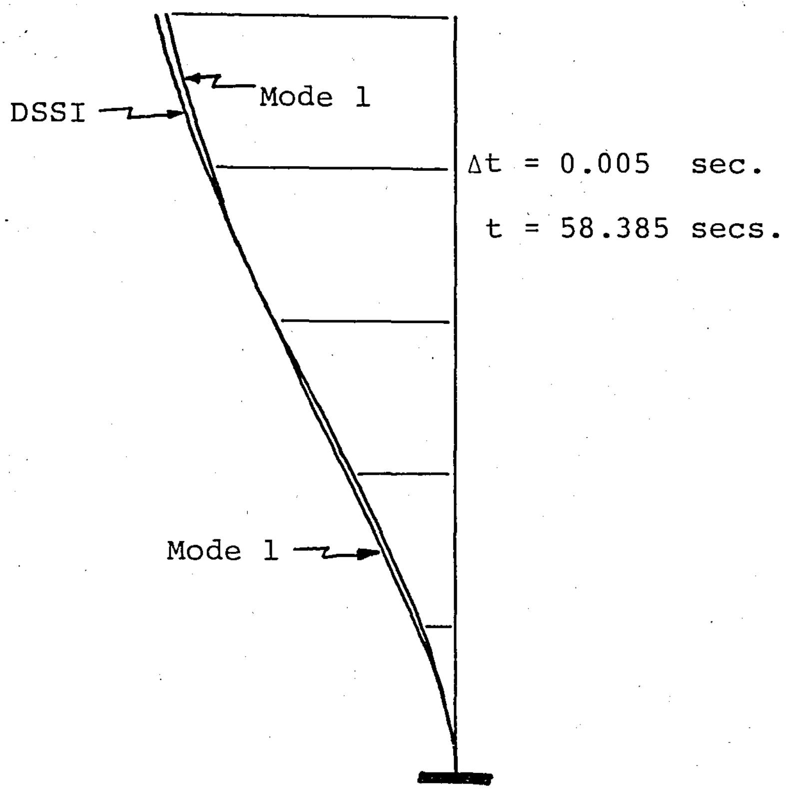


FIGURE IV.B.3 Displacement Distribution Along the Five Storey Plane Frame

IV.B.3 HIGH-RISE TOWER

Free and forced vibration analyses of the high-rise tower given in Fig.IV.B.4 is presented in this section. The program PFRAME6 is used for the computation of the reduced flexibility matrix. Geometric, cross-sectional and material properties, and the mass distribution along the tower are given in Table IV.B.9. Ten horizontal dynamic degrees-of-freedom are selected, and the structure was idealized consisting of 10 constant cross-section plane frame elements. The cross-sectional properties are chosen such that every element represents that part of the tower at which it is located.

The free vibration analysis results are given in TABLE IV.B.10. Mode shapes of the system are shown in Fig. IV.B.5. Neglecting their signs, the participation factors are sorted in decreasing order up to the sixth mode. Participation factors of the seventh and the lower modes are relatively larger than that of the sixth mode.

The step-by-step integration results are given in Table IV.B.11. For sulution time steps greater than about 0.06 second the integration diverges. The $T/\Delta t$ ratios for diverging time steps are given in Table IV.B.12. It is clear that the integration diverges for $T_{10}/\Delta t$ values approximately less than 2.0. The error in the converging solution with the largest solution time step (i.e., the case $\Delta t = 0.006$) is less than 0.15 per cent. The accuracy

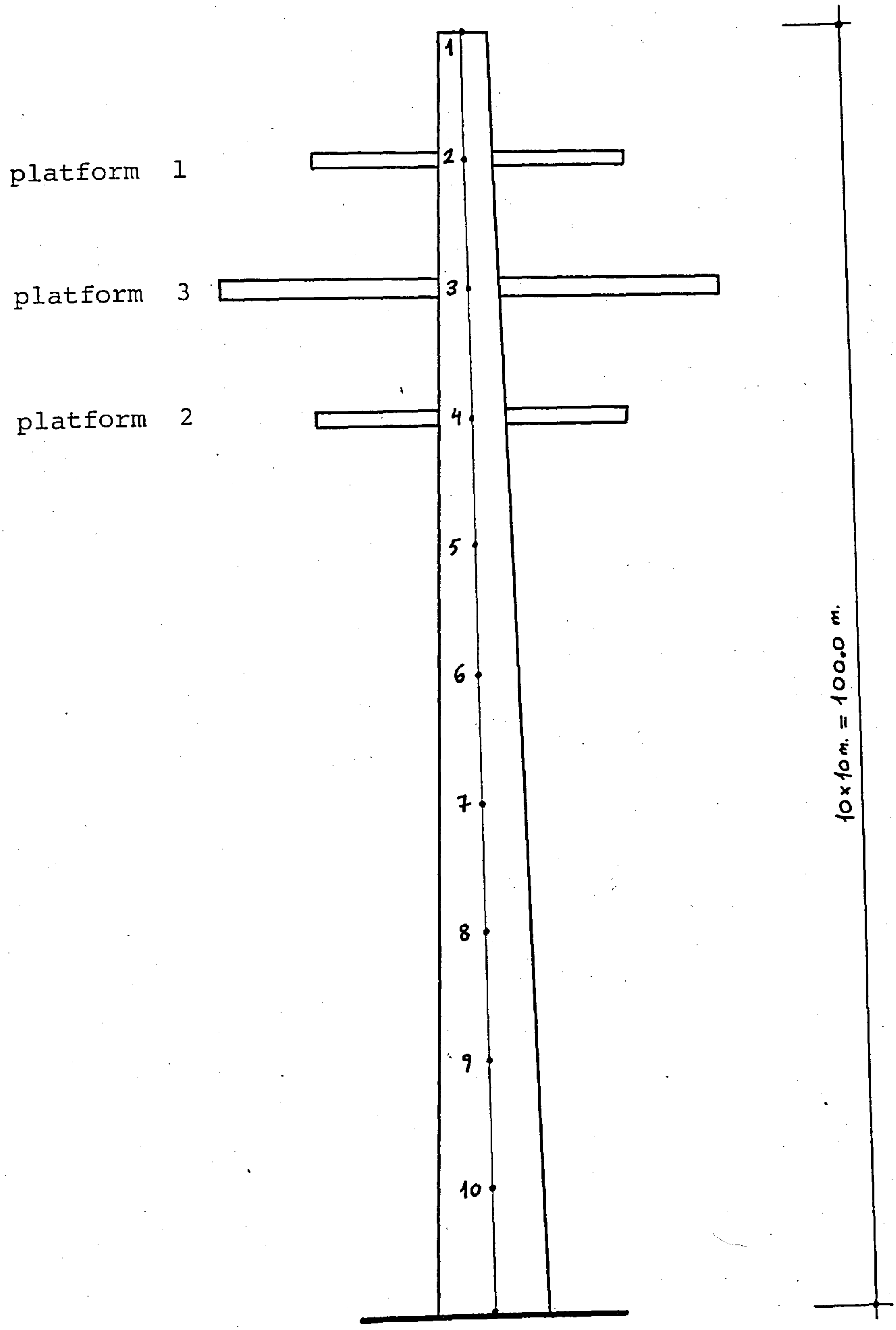


FIGURE IV.B.4 High-Rise Tower

platforms 1,2 : d = 25 m. t = 50 cm.

platform 3 : d = 40 m. t = 50 cm.

live load on the platforms ; $q = 0.75\ t/m^2$

$g = 9.81\ m/sec^2$

$\gamma = 2.4\ T/m^3$

$E = 2100000 t/m^2$

Section No.	Outer Diameter (m)	Thickness (m)	Area (m^2)	Moment of Inertia (m^4)	Mass ($t\text{-}sec^2/m$)
1	4.0	0.40	2.38761	4.32157	2.921
2	4.5	0.42	2.83026	6.52665	104.498
3	5.0	0.44	3.30370	9.45552	257.872
4	5.5	0.46	3.80792	13.24482	106.890
5	6.0	0.48	4.34294	18.04230	10.625
6	6.5	0.50	4.90874	24.00680	12.009
7	7.0	0.52	5.50533	31.30824	13.469
8	7.5	0.54	6.13270	40.12766	15.004
9	8.0	0.56	6.79087	50.65715	16.614
10	8.5	0.58	7.47982	63.09993	18.299
11	9.0	0.60	8.19956	77.67030	10.030

TABLE IV.B.9 Various Properties of the High-Rise Tower

Mode No.		1	2	3	4	5	6	7	8	9	10
ω (rad/sec)		1.13336	9.76900	27.1876	46.7756	92.5948	104.676	176.350	282.144	397.607	519.660
T (sec)		5.54384	.643176	.231105	.134326	.678568	.0600248	.0356291	.0222694	.0158025	.0120909
r		22.1020	-5.22336	3.99065	-2.81374	2.72025	-.912170	2.14723	-1.65434	-1.28864	-1.15602
{φ}	1	.0662972	.141459	.134455	.138993	.200001	.490105	.017609	.00530397	-.00194123	.00035944
	2	.0556022	.0615055	.0337698	.0154532	-.0097685	-.0348352	-.0024461	-.0008993	-.00035064	-.00006699
	3	.0449842	-.0110163	-.0282146	-.0270026	-.0030148	-.0135825	.00338478	.0015203	-.000631606	.000123522
	4	.0347805	-.0568555	.00824332	.0577872	.0274074	-.0173535	-.0178798	-.0103932	.00497243	-.0010623
	5	.0255149	-.0723764	.0736422	.0476819	-.100119	-.0162762	.166442	.170804	-.111912	.0290119
	6	.175603	-.0673833	.117105	-.0264987	-.145229	.0517246	.0525233	-.107038	.151403	-.0553199
	7	.0110866	-.0523939	.122677	-.0877439	-.0371089	.0247356	-.133612	-.0501266	-.135680	.886860
	8	.00613707	-.0335944	.0958811	-.100063	.0919710	-.0259802	-.0479434	.130859	.0544548	-.121471
	9	.00268510	-.0164200	.0536771	-.0683295	.119070	-.0431494	.114456	-.0391564	.0509307	.138190
	10	.000666809	-.00444361	.0159658	-.0228554	.0521439	-.0203153	.0878691	-.110611	-.117662	-.130071

TABLE IV.B.10 Free Vibration Parameters of the Tower

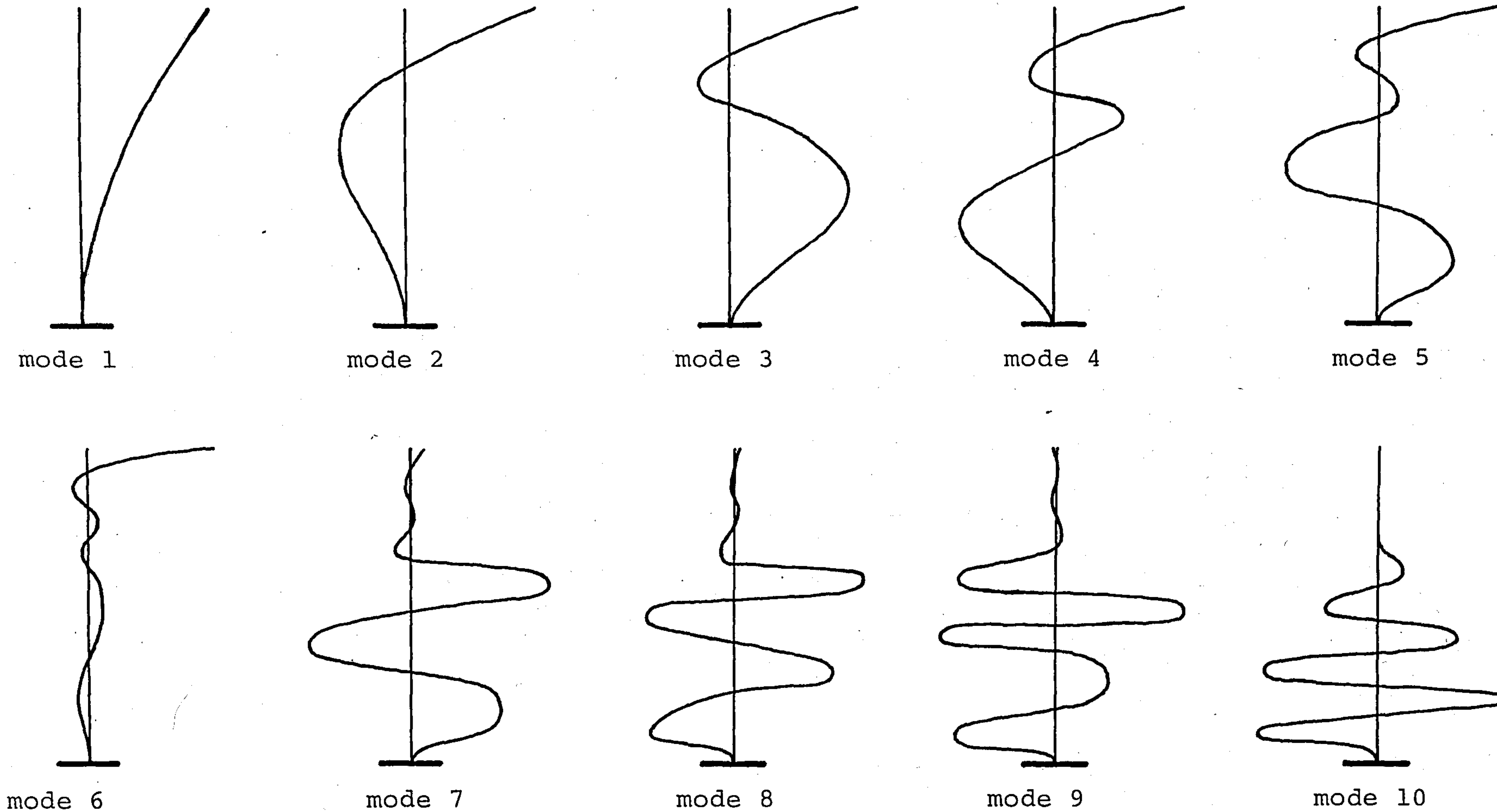


FIGURE IV.B.5 Mode Shapes of the Tower

Time Step (sec)		0.006	0.005	0.0025	0.001
Execution time (CP secs)		158.619	190.001	375.969	936.425
$\{y\}_{max}$ (m)	1	-.112030	-.112112	-.112238	-.112233
	2	-.0918747	-.0919153	-.0919982	-.0920229
	3	-.0761047	-.0760916	-.0760841	-.0760846
	4	-.0612093	-.0612195	-.0612111	-.0612062
	5	-.0464689	-.0464843	-.0464873	-.0464776
	6	-.0329294	-.0329382	-.0329658	-.0329588
	7	-.0213171	-.0213197	-.0213631	-.0213694
	8	-.0120542	-.0120584	-.0120965	-.0121125
	9	-.00537091	-.00537562	-.00539574	-.00541028
	10	-.00135444	-.00135666	-.00136218	-.00136759
Weighted Error (%)		0.1248	0.0923	0.0204	-

TABLE IV.B.11 Maximum Displacements Computed by DSSI for the High Rise Tower

Δt (sec)	$T_1/\Delta t$	$T_{10}/\Delta t$	Status
0.0500	111	0.24	Diverged
0.0100	554	1.21	"
0.0075	739	1.61	"
0.0070	791	1.73	"
0.0065	853	1.86	"
0.060	924	2.02	Converged

TABLE IV.B.12 $T/\Delta t$ Values for Diverging Time Steps further improves for smaller values Δt.

The rusults of the modal analysis are given in Table IV.B.13. The error in the results decrease to about 0.3 per cent for the case which the first three modes are superposed. Further contribution of the lower modes slightly decrease the error; however, for the case when all the modal responses are summed still an error of 0.28 per cent is retained. This is probably due to the computation error in the lower modes introduced from the eigenvalue solution.

The displacements of the system at t = 48.52 seconds are given for various modal contributions in Table IV.B.14. The displacement distribution is shown in Fig. IV.B.6. The per cent error for the case when only first mode is contributing is about 5.7 and decreases to about 0.40 per cent when all modes are contributing.

It can be clearly seen from Figures IV.B.3 and IV.B.6 that the first mode is dominating the structural response .

($\Delta t = 0.0025$)

Modes Superposed		1	1+2	1+2+3	1+2+3+4	1+2+3+4+5	1+2+3+4+5+6+7+8+9+10	DSSI
Execution time (CP secs)		63.267	78.415	93.833	108.077	124.078	195.047	375.969
$\{y\}_{max}$ (m)	1	-.110971	-.111865	-.112140	-.112167	-.112170	-.112172	-.112238
	2	-.0930688	-.0919124	-.0919047	-.0919057	-.0919060	-.0919055	-.0919982
	3	-.0752960	-.0758137	-.0757639	-.0757644	-.0757645	-.0757646	-.0760841
	4	-.0582168	-.0609200	-.0609326	-.0609343	-.0609335	-.0609336	-.0612111
	5	-.0427077	-.0461573	-.0462711	-.0462708	-.0462745	-.0462747	-.0464873
	6	-.0293930	-.0326223	-.0328092	-.0328105	-.0328156	-.0328156	-.0329658
	7	-.0185571	-.0210584	-.0212599	-.0212661	-.0212674	-.0212678	-.0213631
	8	-.0102724	-.0118768	-.0120376	-.0120464	-.0120437	-.0120439	-.0120965
	9	-.00449440	-.00527877	-.00537007	-.00537680	-.00537399	-.00537285	-.00539574
	10	-.00111613	-.00132843	-.00135588	-.00135832	-.00135725	-.00135651	-.00136218
Weighted Error (%)		4.17385	0.51364	0.29425	0.28224	0.28091	0.28078	-

TABLE IV.B.13 Maximum Displacements Computed by MODAL for the High-Rise Tower

(Δt = 0.0025)

Modes Superposed		1	1+2	1+2+3	1+2+3+4	1+2+3+4+5	1+2+3+4+5+6 +7+8+9+10	DSSI
$\{y\}_{max}$ (m)	1	-.110932	-.104161	-.104339	-.104349	-.104343	-.104339	-.104705
	2	-.0930364	-.0900925	-.0901372	-.0901383	-.0901386	-.0901389	-.0904892
	3	-.0752698	-.0757971	-.0757597	-.0757579	-.0757579	-.0757578	-.0760841
	4	-.0581965	-.0609179	-.0609288	-.0609327	-.0609320	-.0609321	-.0612111
	5	-.0426928	-.0461571	-.0462547	-.0462579	-.0462607	-.0462605	-.0464777
	6	-.0293827	-.0326223	-.0327775	-.0327757	-.0327797	-.0327794	-.0329338
	7	-.0185506	-.0210584	-.0212210	-.0212150	-.0212161	-.0212157	-.0213147
	8	-.0102689	-.0118768	-.0120039	-.0119971	-.0119945	-.119947	-.0120497
	9	-.00449284	-.00527877	-.00534991	-.00534526	-.00534196	-.00534250	-.00536654
	10	-.00111574	-.00132843	-.00134959	-.00134803	-.00134659	-.00134687	-.00135282
Weighted Error (%)		5.6628	0.5961	0.4123	0.4132	0.4145	0.4153	-

TABLE IV.B.14 Displacements of the High-Rise Tower at t = 48.52 secs.

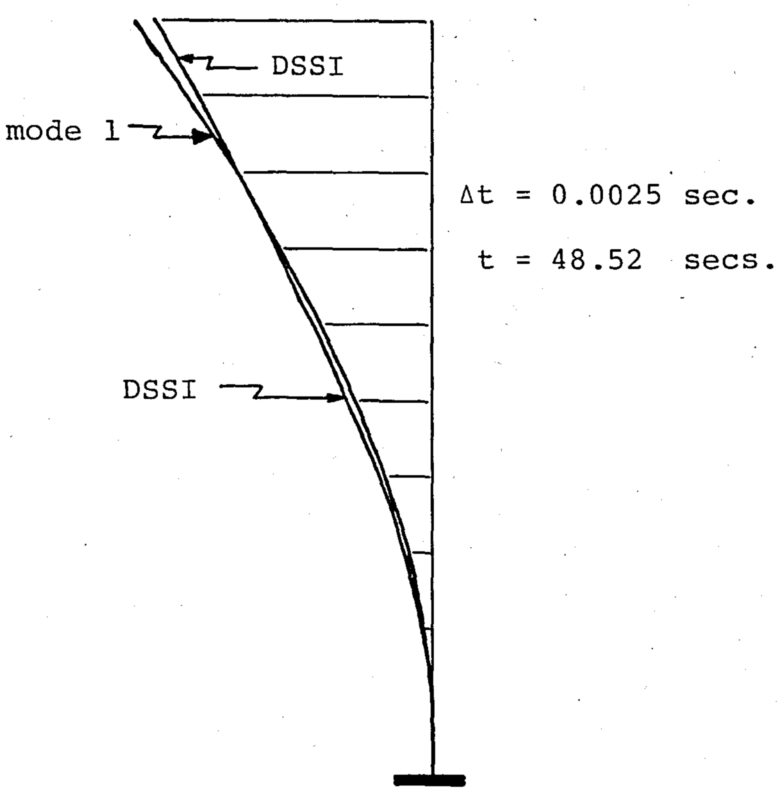


FIGURE IV.B.6 Displacement Distribution Along the Tower

IV.B.4 SYMMETRIC ARCHES

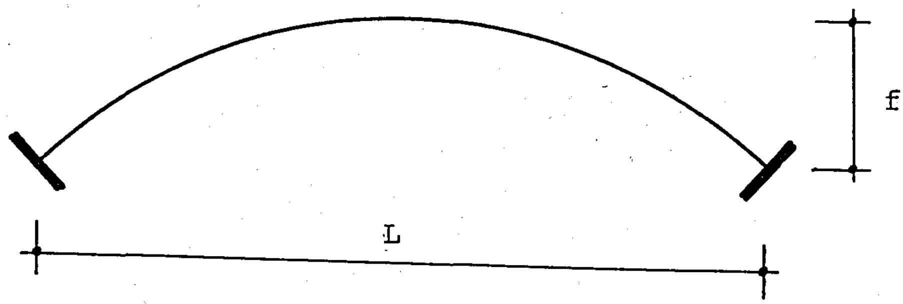
In this section, free and forced vibration analysis results for four arches are given. All of the arches are symmetric in terms of stiffness and mass distribution. A typical constant cross-section, circular arch is shown in Fig. IV.B.7.

The free vibration mode shapes of all arches are either symmetric or antimetric. The participation factors coupling the inertia of the structure to the horizontal component of the earthquake motion are found to be equal to zero for all the symmetric modes. This implies that only the antimetric modes are contributing to the response of the arches, when they are subjected to a horizontal base excitation.

The slenderness and aspect ratios of the arches are affecting the first mode shape being symmetric or antimetric(11).

The free vibration properties (frequencies, periods, participation factors and mode shape vectors) of Arch No.1 are given in TABLE IV.B.15. The geometric characteristics, and stiffness of Arch No.1 are given below:

f = 20 m	r = 0.25 m.
L = 100 m	S/r = 440
S = 110 m	f/L = 0.2
EI = 23668.642 tm^2	
EA = 378408.42 t	



f = height of the arch

L = length of the span

S = arch length

EI = flexural stiffness

EA = axial stiffness

r = radius of gyration $(=\sqrt{EI/EA})$

f/L= aspect ratio

S/r= slenderness ratio

FIGURE IV.B.7 Typical Arch

Mode No		1	2	3	4	5	6	7	8	9	10
w(rad/sec)		2.35231	4.46353	8.06168	11.6869	16.5768	20.631	25.2948	26.3614	28.4266	61.2030
T (Sec)		2.67107	1.40767	.779390	.537625	.379036	.304963	.248399	238348	.221032	.102661
r		-1.66896	0.0	.978132	0.0	.673877	0.0	.418524	0.0	0.0	2.014661
$\{\phi\}^{(y)}$	3	.106730	-.157461	-.255675	-.272095	-.344934	-.224530	-.279882	.267914	.299034	.242278
	5	.300611	-.284160	-.307109	-.0226490	.124419	.446347	.399184	.308728	-.138606	.245412
	7	.391008	-.0610642	.182798	.515578	.348439	.0646694	-.335498	.0946199	.414727	.210560
	9	.275341	.373024	.447601	.145226	-.380343	-.141937	.219209	.515061	-.197822	.117502
	11	0.0	.593748	0.0	-.345002	0.0	.544175	0.0	-.00389329	.460825	0.0
	13	-.275341	.373024	-.447601	.145226	.380343	-.141937	-.219209	.515061	-.197822	-.117502
	15	-.391008	-.0610642	-.182798	.515578	-.348439	.0646694	.335498	.0946199	.414727	-.210560
	17	-.300611	-.284160	.307109	-.0226490	-.124419	.446347	-.399184	.308728	-.138606	-.245412
	19	-.106730	-.157461	.255675	-.272095	.344934	-.224530	.279882	.267914	.299034	-.242278
$\{\phi\}^{(x)}$	3	-.844986	.126169	.204854	.223759	.282430	.202194	.235816	-.164210	-.219469	-.00145865
	5	-.199281	.205088	.243312	.0901558	.0167486	-.178497	-.166098	-.150492	.0653422	.150798
	7	-.237540	.120407	-.0503121	-.127284	-.0846184	-.0250469	.126340	-.0174632	-.135330	.279990
	9	-.212777	.0201110	-.0177069	-.0468673	.0865745	.465285	.00585496	-.0794349	.0295394	.371275
	11	-.192647	0.0	.0116297	0.0	.0683287	0.0	-.0126614	0.0	0.0	.403850
	13	-.212777	-.0201110	-.0177069	.0468673	.0865745	-.0465285	.00585496	.0794349	-.0295396	.371275
	15	-.237540	-.120407	.0503121	.127284	-.0846184	.250469	.126340	.0174632	.135330	.279990
	17	-.199281	-.205088	.243312	-.0901558	.0167486	.178497	-.166098	.150492	-.0653422	.150698
	19	-.0844986	-.126169	.204954	-.223759	.282430	-.202194	.235816	.164210	.219469	-.00145865

TABLE IV.B.15 Free Vibration Properties of Arch No.1

The arch is represented by 20 elements and the mass is assumed to be concentrated at nine nodes. Two dynamic degrees-of-freedom are selected at each of these nodes (in X and Y translational directions). The mass influence coefficients in each dynamic degree-of-freedom is 1.00489 $tsec^2/m$. For this arch, the first mode shape is found to be antimetric (See Fig. IV.B.8). The step-by-step integration is carried out for several time step values, and the maximum displacements are given in Table IV.B.16. The computed maximum displacements from modal analysis are given in Table IV.B.17. Displacements of Arch No.1 at a certain instant are given in Table IV.B.18. The displaced shape at that instant is given in Fig.IV.B.9.

To find the effect of the slenderness ratio on the shape of the first mode shape the S/r is reduced to 25. This is achieved by choosing an axial stiffness, EA, of 1215.0 tons. For this arch (Arch No.2), the first mode shape is found to be symmetric(TABLE IV.B.19). Maximum displacements computed by computer programs DSSI and MODAL, and the displacements at t = 24.48 seconds are given in Tables IV.B.20 to IV.B.22, respectively.

The slenderness ratio of the circular arch is set to a constant value (S/r = 25), and the aspect ratio (f/L) is decreased to 0.05. The geometric and stiffness properties of this new arch, Arch No.3, are given below :

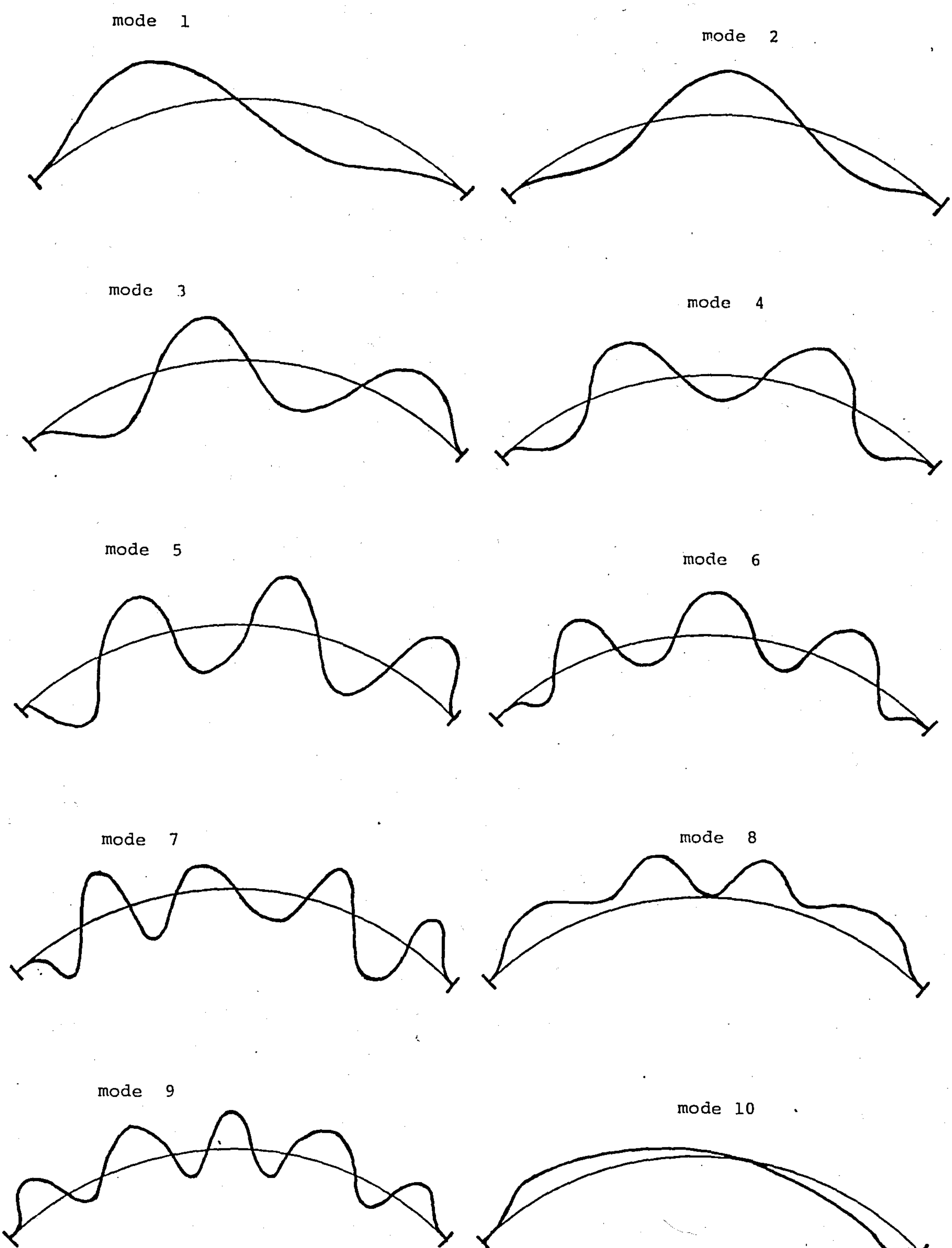


FIGURE IV.B.8 Mode Shapes of Arch No.1

Time Step (sec)		Δt=0.01	Δt=0.005	Δt=0.0025	Δt=0.001
Eecution Time (cP secs)		264.823	516.873	1034.620	2586.792
$\{y\}^{(Y)}_{max}$	3	-.0947778	-.0948353	-.0948015	-.0947958
	5	-.256027	-.256152	-.256232	-.256257
	7	-.324984	-.324634	-.324575	-.324569
	9	-.235744	-.235900	-.235919	-.235924
	11	0.0	0.0	0.0	0.0
	13	.235744	-.235900	.235919	.235924
	15	.324984	.324634	.324575	.324569
	17	.256027	.256152	.256232	.256257
	19	-.0947778	.0948353	.0948015	.0947958
$\{Y\}^{(x)}_{max}$ (m)	3	.0751106	.0751527	.0751270	.0751174
	5	.170768	.170846	.170888	.170897
	7	-.198205	-.198212	-.198220	-.198224
	9	-.177258	-.177281	-.177284	-.177290
	11	-.160657	-.160705	-.160716	-.160721
	13	-.177258	-.177281	-.177284	-.177290
	15	-.198205	-.198212	-.198220	-.198224
	17	.170768	.170846	.170888	.170897
	19	.0751106	.0751527	.0751270	.0751174
Weighted Error (%)		0.066	0.020	0.004	-

TABLE IV.B.16 Maximum Displacements Computed by DSSI for Arch No.1

(Δt = 0.01 sec.)

Modes Superposed		1	1+2	1+2+3	1+2+3+4	1+2+3+4+5	1+2+3+4+5+6 +7	1+2+3+4+5+6 +7+8+9+10	DSSI
Execution Time (CP scs)		22.608	28272	33.743	39.073	44.656	55.264	72.524	264.823
$\{Y\}^{Y}_{max}$ (m)	3	.0889197	.0889197	-.0943309	-.0943309	-.0946843	-.0947805	-.0947782	-.0947778
	5	.250447	.250447	-.256281	-.256281	-.256156	-.256035	-.256026	-.256027
	7	.325758	.325758	-.325139	-.325139	-.324892	-.324981	-.324984	-.324984
	9	.229394	.229394	-.235606	-.235606	-235817	-.235745	-.235744	-.235744
	11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	13	-.229394	-.229394	.235606	.235606	.235817	.235745	.235744	.235744
	15	-.325758	-.325758	.325139	.325139	.324892	.324981	.324984	.324984
	17	-.250447	-.250447	.256281	.256281	.256156	.256035	.256026	.256027
	19	-.0889197	-.0889197	.0943309	.0943309	.0946843	.0947805	.0947782	.0947778
$\{Y\}^{(x)}_{max}$ (m)	3	-.0703979	-.0703979	.0747397	.0747397	.0750292	.0751102	.0751102	.0751106
	5	-.166026	-.166026	.170798	.170798	.170814	.170764	.170769	.170768
	7	-.197901	-.197901	-.198245	-.198245	-.198156	-.198192	-.198205	-.198205
	9	-177270	-.177270	-.177158	-.177158	-.177243	-.177245	-.177257	-.177258
	11	-.160499	-.160499	-.160572	-.160572	-.160638	-.160643	-.160657	-.160657
	13	-.177270	-.177270	-.177158	-.177158	-.177243	-.177245	-.177257	-.177258
	15	-.197901	-.197901	-.198245	-.198245	-.198156	-.198192	-.198205	-.198205
	17	-.166026	-.166026	.170798	.170798	.170814	.170764	.170769	.170768
	19	-.0703979	-.0703979	.0747397	.0747397	.0750292	.0751102	.0751102	.0751106
Weighted Error (%)		1.7612	1.7612	0.0978	0.0978	0.036	0.0032	0.0002	-

TABLE IV.B.17 Maximum Displacements Computed by MODAL for Arch. No.1

(Δt = 0.01 sec)

Modes Superposed		1	1+2	1+2+3	1+2+3+4	1+2+3+4+5	1+2+3+4+5 +6+7	1+2+3+4+5 +6+7 +8+9+10	DSSI
$\{y\}_t^{(Y)}$	3	.0889197	.0889197	.0905272	.0905272	.0908629	.0909547	.0909463	.0909465
	5	.250447	.250447	.252377	.252377	.252256	.252125	.252117	.252117
	7	.325758	.325758	.324609	.324609	.324270	.324380	.324373	.324373
	9	.229394	.229394	.226580	.226580	.226950	.226878	.226874	.226874
	11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	13	-.229394	-.229394	-.226580	-.226580	-.226950	-.226878	-.226874	-.226874
	15	-.325758	-.325758	-.324609	-.324609	-.324270	-.324380	-.324373	-.324373
	17	-.250447	-.250447	-.252377	-.252377	-.252256	-.252125	-.252117	-.252117
	19	-.0889197	-.0889197	-.0905272	-.0905272	-.0908629	-.0909547	-.0909463	-.0909465
$\{y\}_t^{(x)}$	3	-.0703979	-.0703979	-.0716864	-.0716864	-.0719613	-.0720387	-.0720386	-.0720383
	5	-.166026	-.166026	-.167556	-.167556	-.167572	-.167518	-.167523	-.167522
	7	-.197901	-.197901	-.198217	-.198217	-.198135	-.198176	-.198186	-.198186
	9	-.177270	-.177270	-.177158	-.177158	-.177243	-.177245	-.177257	-.177258
	11	-.160499	-.160499	-.160572	-.160572	-.160638	-.160643	-.160657	-.160657
	13	-.177270	-.177270	-.177158	-.177158	-.177243	-.177245	-.177257	-.177258
	15	-.197901	-.197901	-.198217	-.198217	-.198135	-.198176	-.198186	-.198186
	17	-.166026	-.166026	-.167556	-.167556	-.167572	-.167518	-.167523	-.167522
	19	-.0703979	-.0703979	-.0716864	-.0716864	-.0719613	-.0720387	-.0720386	-.0720383
Weighted Error (%)		0.6992	0.6992	0.1113	0.1113	0.0380	0.0039	0.0002	—

TABLE IVB.18 Displacements of Arch No.1 at t= 58.54 secs.

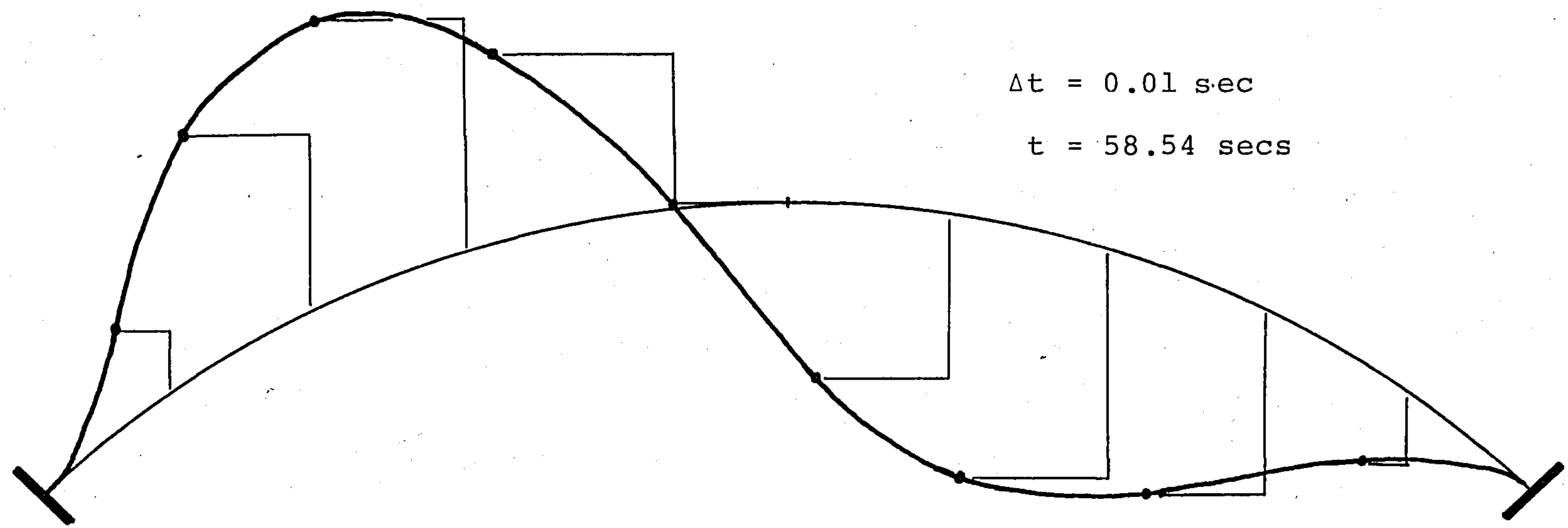


FIGURE IV.B.9 Displaced Shape of Arch No.1 at t = 58.54 secs.

Mode No		1	2	3	4	5	6	7	8	9	10
w (rad/sec)		1.56200	2.23251	3.63608	4.86883	6.67687	8.10044	9.70617	11.8586	12.6835	14.8275
T (sec)		4.02252	2.81440	1.72801	1.29049	.941038	.775660	.647339	.529843	.495384	.423753
r		0.0	-2.16115	-1.81428	0.0	0.0	.597586	.606018	0.0	0.0	.449367
$\{\phi\}^{(Y)}$	3	.0620019	.0785806	-.133016	-.180152	.190775	-.250434	-.262263	.184492	.398358	.116840
	5	.184947	.238258	-.285005	-.376179	.280151	-.320949	-.245281	.100717	.152660	-.0303845
	7	.336534	.315205	-.349868	-.227541	.162295	.120130	.0524175	-.251959	-.333546	-.0552084
	9	.461921	.223100	-.244370	.166292	-0.844289	.378667	.209873	-.0197454	-.0690376	-.176829
	11	.510519	0.0	0.0	.377461	-.213372	0.0	0.0	.310573	.309213	0.0
	13	.461921	-.223100	.244370	.166292	-.0844289	-.378667	-.209873	-.0197454	-.0690376	.0176829
	15	.336534	-.315205	.349868	-.227541	.162295	-.120130	-.0524175	-.251959	-.333546	.0552084
	17	.184947	-.238258	.285005	-.376179	.280151	.320949	.245281	.100717	.152660	.0303845
	19	.0620019	-.0785806	.133016	-.180152	.190775	.250434	.262263	.184492	.398358	-.116840
$\{\phi\}^{(x)}$	3	.00603322	-.986892	-.0828962	.151864	.192827	.216632	-.262176	-.350445	.200187	.418220
	5	-.0177335	-.231060	-.141494	-.290593	.322545	.285699	-.328028	-.252769	.124859	.0289632
	7	-.0367233	-.297381	-.215832	.235352	.355587	.0474402	-.130848	.224267	-.137885	-.447094
	9	-.0296089	-.301381	-.296237	.100186	.242479	-.152621	.219020	.322932	-.277615	-.00450289
	11	0.0	-.293607	-.332535	0.0	0.0	-.199629	.400995	0.0	0.0	.456007
	13	.0296089	-.301381	-.296237	-.100186	-.242479	-.152621	.219020	-.322932	.277615	-.00450289
	15	.0367233	-.297381	-.215832	-.235352	-.355587	.0474402	-.130848	-.224267	.137885	-.447094
	17	.0177335	-.231060	-.141494	-.290593	-.322545	.285699	-.328028	.252769	-.124859	.0289632
	19	-.00603322	-.0986892	-.0828962	-.151864	-.192827	.216632	-.262176	.350445	-.200187	.418220

TABLE IV.B.19 Free Vibration Properties of Arch No.2

Time Step (sec)		Δt=0.1	Δt=0.05	Δt=0.01	Δt=0.005
Execution Time (CP sec)		30.123	55.808	262.050	521.053
$\{y\}^{(Y)}_{max}$ (m)	3	.0627173	.0624591	.0625224	.0624976
	5	.172145	.171427	.171734	.171755
	7	.226700	.227127	.227209	.227194
	9	.161974	.163279	.163284	.163298
	11	0.0	0.0	0.0	0.0
	13	-.161974	-.163279	-.163284	-.163298
	15	-.226700	-.227127	-.227209	-.227194
	17	-.172145	-.171427	-.171734	-.171755
	19	-.0627173	-.0624591	-.0625224	-.0624976
$\{y\}^{(x)}_{max}$ (m)	3	-.0615298	-.0622995	-.0619336	-.0619013
	5	-.134905	-.137509	-.137591	-.137591
	7	-.177199	-.178277	.178340	.178378
	9	-.190771	-.191267	-.192195	-.192212
	11	-.192578	-.193340	-.194359	-.194386
	13	-.190771	-.191267	-.192195	-.192212
	15	-.177199	-.178277	-.178340	.178378
	17	-.134905	.137509	.137591	.137591
	19	-.0615298	-.0622995	-.0619336	-.0619013
Weighted Error (%)		0.697	0.194	0.015	-

TABLE IV.B.20 Maximum Displacements Computed by DSSI for Arch No.2

(Δt = 0.01 sec.)

Modes Superposed		1+2	1+2+3	1+2+3+4+5 +6	1+2+3+4+5 +6+7	1+2+3+4+5 +6+7+8+9+10	DSSI
Execution Time (CP sec)		28.172	33.792	49.948	55.955	72.387	262.050
$\{y\}^{(Y)}_{max}$ (m)	3	-.0413275	.0645780	.0624971	.0625140	.0625437	.0625224
	5	-.125306	.174540	.171501	.171721	.171713	.171734
	7	-.165774	.226023	.227330	.227230	.227213	.227209
	9	-.117334	.159391	.163857	.163294	.163288	.163284
	11	0.0	0.0	0.0	0.0	0.0	0.0
	13	.117334	-.159391	-.163857	-.163294	-.163288	-.163284
	15	.165774	-.226023	-.227330	-.227230	-.227213	-.227209
	17	.125306	-.174540	-.171501	-.171721	-.171713	-.171734
	19	.0413275	-.0645780	-.0624971	-.0625140	-.0625437	-.0625224
$\{y\}^{(x)}_{max}$ (m)	3	.0519031	-.0608325	.0633572	.0619327	-.0619239	-.0619336
	5	.121520	.135369	.139348	.137598	.137578	.137591
	7	.156400	-.178358	-.178768	-.178101	.178344	.178340
	9	.158504	-.192109	-.191066	-.192165	-.192165	-.192195
	11	.154415	-.193635	-.192446	-.194442	-.194400	-.194359
	13	.158504	-.192109	-.191066	-.192165	-.192165	-.192195
	15	.156400	-.178358	-.178768	-.178101	.178344	.178340
	17	.121520	.135369	.139348	.137598	.137578	.137591
	19	.0519031	-.0608325	.0633572	.0619327	-.0619239	-.0619336
Weighted Error (%)		78.37	1.05	0.51	0.03	0.01	-

TABLE IV.B.21 Maximum Displacements Computed by MODAL for Arch No.2

(Δt = 0.01 sec.)

Modes Superposed		1+2	1+2+3	1+2+3+4+5 +6	1+2+3+4+5 +6+7	1+2+3+4+5 +6+7+8+9-10	DSSI
$\{Y\}_t^{(Y)}$ (m)	3	.0385540	.0187230	.0203086	.0216254	.0216228	.0215925
	5	.116897	.0744061	.0764382	.0776697	.0776734	.0777088
	7	.154649	.102488	.101728	.101465	.101466	.101433
	9	.109460	.0730273	.0706297	.0695760	.0695764	.0695939
	11	0.0	0.0	0.0	0.0	0.0	0.0
	13	-.109460	-.0730273	-.0706297	-.0695760	-.0695764	-.0695939
	15	-.154649	-.102488	-.101728	-.101465	-.101466	-.101433
	17	-.116897	-.0744061	-.0764382	-.0776697	-.0776734	-.0777088
	19	-.0385540	-.0187230	-.0203086	-.0216254	-.216228	-.0215925
$\{Y\}_t^{(x)}$ (m)	3	-.0484199	-.0607787	-.0621504	-.0608340	-.0608433	-.0608519
	5	-.113365	-.134460	-.136269	-.134622	-.134623	-.134612
	7	-.145904	-.178082	-.178383	-.177726	-.177716	-.177709
	9	-.147867	-.192032	-.191066	-.192165	-.192165	-.192195
	11	-.144052	-.193629	-.192365	-.194379	-.194389	-.194348
	13	-.147867	-.192032	-.191066	-.192165	-.192165	-.192165
	15	-.145904	-.178082	-.178383	-.177726	-.177716	-.177709
	17	-.113365	-.134460	-.136269	-.134622	-.134623	-.134612
	19	-.0484199	-.0607787	-.621504	-.0608340	-.0608433	-.0608519
Weighted Error (%)		30.4643	1.2629	1.0329	0.0228	0.004	-

TABLE IV.B.22 Displacements of Arch No.2 at t = 24.48 secs.

$f = 5$ m	$r = 4.027$ m.
$L = 100$ m.	$S/r = 25$
$S = 100.67$ m.	$f/L = 0.05$
$EI = 23668.642$ tm^2	
$EA = 1459.801$ t	

The mass at each dynamic degree-of-freedom is 1.00489 tsec^2/m.

The free vibration properties of Arch No.3 are given in Table IV.B.23. The first mode shape is found to be symmetric for this arch. Maximum displacements computed by programs DSSI and MODAL are given in Tables IV.B.24 and IV.B.25, respectively. The displacements computed at a certain instant are given in Table IV.B.26.

Arch No.4 is a parabolic arch with geometric and stiffness properties;

$f = 50$ m.	$r = 4.647$ m.
$L = 50$ m.	$S/r = 25$
$S = 116.17$ m.	$f/L = 1.0$
$EI = 23668.642$ tm^2	
$EA = 1096.146$ t.	

The first mode shape vector is antimetric for Arch No.4. Free vibration properties, maximum displacements computed by step-by-step integration and modal analysis, and displacements at t = 58.175 are given in Tables IV.B.27 to IV.B.30.

Mode No.		1	2	3	4	5	6	7	8	9	10
w (rad/sec)		1.14138	2.92164	3.80040	5.78209	7.44323	9.52884	10.9263	13.9986	14.2430	32.2486
T (sec)		5.50492	2.15057	1.65330	1.08666	.844147	.659386	.575054	.448842	.441141	.194836
r		0.0	-.862017	2.69541	0.0	0.0	.192755	-.855953	0.0	0.0	0.0
	3	.0598015	.135306	.0510247	.237540	-.0584636	-.331356	-.0820760	-.263763	.326028	.221898
	5	.194636	.362016	.115439	.466803	-.0876126	-.406359	-.0788671	-.137825	.152525	-.291582
	7	.345099	.453418	.140160	.269471	-.0502849	.132626	.0167684	-.284693	-.309745	.353367
	9	.459299	.312160	.0961618	-.195749	.258816	.433109	.0667576	.0407668	-.0542850	-.391756
	11	.501601	0.0	0.0	-.437865	.0650228	0.0	0.0	-.311839	-.315016	.404886
	13	.459299	-.312160	-.961618	-.195749	.0258816	-.433109	-.0667576	.0407668	-.0542850	-.311756
$\{\phi\}^{(Y)}$	15	.345099	-.453418	-.140160	.269471	.0502849	-.132626	-.0167684	.284693	-.309745	.353367
	17	.194636	-.362016	-.115439	.466803	.087126	.406359	-.0788671	.137825	.152525	-.291582
	19	.0598015	-.135306	-.0510247	.237540	.0584639	.331356	.082060	-.263763	.326028	.221898
	3	.00960745	-.0413323	.130837	-.0483789	-.256440	.0700908	-.352402	.318923	.277348	-.0327193
	5	-.00689630	-.0911345	.246636	-.0893479	-.415515	.0896496	-.415528	.204921	.166195	.0321727
	7	-.0115766	-.116876	.341977	-.0718432	-.418270	.0154932	-.137051	-.197468	-.172639	-.0261428
	9	-.00862434	-.120336	.406733	-.3039394	-.260378	-.475924	.258228	-.308170	-.291814	.0145383
$\{\phi\}^{(x)}$	11	0.0	-.118465	.429923	0.0	0.0	-.0634657	.441718	0.0	0.0	0.0
	13	.00862434	-.120336	.406733	..0309394	.260378	-.475924	.258228	.308170	.291814	-.0145383
	15	.0115766	-.116876	.341977	.0718432	.418270	.0154932	-.137051	.197468	.172639	.0261428
	17	.00689630	-.0911345	.246636	.0893479	.415515	.0896496	-.415528	-.204921	.166195	-.0321727
	19	-.00960745	-.0413323	.130837	.0483789	.256440	.0700908	-.352402	-.318923	-.277348	-.0327193

TABLE IV.B.23 Free Vibration Properties of Arch No.3

Time Step (sec)		0.05	0.02	0.01	0.005
Execution Time (CP sec)		55.908	131.462	261.615	517.694
	3	-.0315594	-.0316035	-.0317899	-.0317980
	5	-.0769829	-.0773819	-.0774684	-.774899
	7	-.0950240	-.0955715	-.0956103	-.0956182
	9	-.0652556	-.0655848	-.0655481	-.655384
$\{y\}^{(Y)}_{max}$	11	0.0	0.0	0.0	0.0
(m)	13	.652556	.0655848	.655481	.0655384
	15	.0950240	.0955715	.0956103	.0956182
	17	.0769829	.0773819	.0774684	.0774899
	19	.0315594	.0316035	.0317899	.317980
	3	.0664366	.0644721	.0657085	.0659194
	5	.124802	.124157	.124532	.124734
	7	.170793	.173015	.172950	.172953
	9	.200820	.204695	.205065	.205068
	11	.211424	.215605	.216084	.216249
$\{Y\}^{(x)}_{max}$	13	.200820	.204695	.205065	.205068
	15	.170793	.173015	.172950	.172953
(m)	17	.124802	.124157	.124532	.124734
	19	.0664366	.0644721	.0657085	.0659194
Weighted Error (%)		1.16	0.34	0.06	-

TABLE IV.B.24 Maximum Displacements Computed by DSSI for Arch No.3

(Δt = 0.01 sec.)

Modes Superposed		1+2	1+2+3	1+2+3+4+5 +6	1+2+3+4+5 +6+7	DSSI
Execution Time		27.609	33.073	50.600	55.980	261.615
$\{Y\}^{(Y)}_{max}$ (m)	3	-.0177583	-.0321700	-.0316614	-.0317810	-.0317899
	5	-.0475128	-.0773105	-.0778461	-.0774601	-.0774684
	7	-.0595089	-.0956868	-.0955120	-.0955940	-.0956103
	9	-.0409695	-.0657905	-.0654480	-.0655463	-.0655481
	11	0.0	0.0	0.0	0.0	0.0
	13	.0409695	.0657905	.0654480	.0655463	.0655481
	15	.0595089	.0956868	.0955120	.0955940	.0956103
	17	.0475128	.0773105	.0778461	.0774601	.0774684
	19	.0177583	.0321700	.0316614	.0317810	.0317899
$\{y\}^{(x)}_{max}$ (m)	3	.00542466	.0659868	.0660044	.0655114	.0657085
	5	.0119610	.125263	.125286	.124699	.124532
	7	.0153394	.173059	.173063	.172869	.172950
	9	.0157935	.204595	.204583	.204963	.205065
	11	.0155479	.215682	.215666	.216342	.216084
	13	.0157935	.204595	.204583	.204963	.205065
	15	.0153394	.173059	.173063	.172869	.172950
	17	.0119610	.125263	.125286	.124699	.124532
	19	.00542466	.0659868	.0660044	.0655114	.0657085
Weighted Error (%)		76.75	0.28	0.27	0.08	-

TABLE IV.B.25 Maximum Displacements Computed by MODAL for Arch No.3

(Δt = 0.01 sec.)

Modes Superposed		1+2	1+2+3	1+2+3+4+5 +6	1+2+3+4+5+6 +7	DSSI
$\{y\}_t^{(Y)}$	3	-.00903063	.0156229	.0154875	.0153509	.0153609
	5	-.0241617	.0316150	.0314489	.0313176	.0313026
	7	-.0302621	.0374590	.0375132	.0375411	.0375492
	9	-.208342	.0256282	.0258052	.0259163	.0259198
	11	0.0	0.0	0.0	0.0	0.0
	13	.0208342	-.0256282	-.0258052	-.0259163	-.0259198
	15	.0302621	-.0374590	-.0375132	-.0375411	-.0375492
	17	.0241617	-.0316150	-.0314489	-.0313176	-.0313026
	19	.00903063	-.0156229	-.0154875	-.0153509	-.0153609
$\{Y\}_t^{(x)}$	3	.00275861	.0659749	.0660036	.0654170	.0654509
	5	.00608251	.125250	.125286	.124595	.124445
	7	.00780057	.173034	.173040	.172812	.172950
	9	.00803147	.204553	.204553	.204963	.205065
	11	.00790661	.215632	.215606	.216342	.216078
	13	.00803147	.204553	.204553	.204963	.205065
	15	.00780057	.173034	.173040	.172812	.172950
	17	.00608251	.125250	.125286	.124595	.124445
	19	.00275861	.0659749	.0660036	.0654170	.0654509
Weighted Error (%)		107.08	0.39	0.34	0.08	-

TABLE IV.B.26 Displacements of Arch No.3 at t = 36.50 secs.

Mode No		1	2	3	4	5	6	7
w (rad/sec)		1.38814	2.98879	3.70189	3.83912	6.56858	8.349156	9.76586
T (sec)		4.52633	2.10225	1.69729	1.63662	.956652	.752553	.643382
r		2.59726	0.0	-.180436	0.0	.843310	0.0	-.119806
$\{\phi\}^{(Y)}$	3	-.00481319	.216218	.278403	.686071	.100567	.450860	.478503
	5	-.364666	.365288	.400778	.0643866	.211521	.179207	-.0210890
	7	-.0415781	.438287	.335303	-.546878	.247778	-.178972	-.318261
	9	-.0236320	.464222	.180004	-.154749	.141708	-.346231	-.229644
	11	0.0	.470040	0.0	-.184754	0.0	-.389230	0.0
	13	.0236320	.464222	-.180004	-.154749	-.141708	-.346231	.229644
	15	.0415781	.438287	-.335303	-.0546878	-.247778	-.178972	.318261
	17	.0364666	.365288	-.400778	.643866	-.211521	-.179207	.0210890
	19	.00481319	.216218	-.278403	.0686071	-.100567	.450860	-.478503
$\{\phi\}^{(x)}$	3	.185293	-.101633	-.167515	-.399272	-.461690	-.0468338	-.0873044
	5	.383015	-.397749	-.154005	-.467068	.159399	.178481	.189039
	7	.452995	-.0339090	.0803941	-.222289	-.203184	.234634	.641503
	9	.456376	-.0158344	.286711	-.566159	-.266118	.123843	-.219931
	11	.452544	0.0	.360977	0.0	-.257522	0.0	-.334951
	13	.456376	.158344	.286711	.0566159	-.266118	-.123843	-.219931
	15	.452995	.0339090	.0803941	.222289	-.203184	-.234634	.0641503
	17	.383015	.0397749	-.154005	.467068	.159399	-.178481	.189039
	19	.185293	.0101633	-.167515	.399272	.461690	.468338	-.0873044

TABLE IV.B.27 Free Vibration Properties of Arch No.4

Time Step (sec)		0.02	0.01	0.005	0.001
Execution Time (CP secs)		133.141	262.047	517.691	2541.400
$\{y\}^{(Y)}_{max}$ (m)	3	.00741816	.00742277	.00742941	.00743032
	5	.0155891	.0156155	.0157045	.0157332
	7	.0163129	.0163650	.0164123	.0164276
	9	.00921937	.00922209	.00923708	.00924224
	11	0.0	0.0	0.0	0.0
	13	-.00921937	-.00922209	-.00923708	-.00924224
	15	-.0163129	-.0164650	-.0164123	-.0164276
	17	-.0155791	-.0156155	-.0157045	-.0157332
	19	-.00741816	-.00742277	-.00742941	-.00743032
$\{Y\}^{(x)}_{max}$ (m)	3	-.0572873	-.0572615	-.0572482	-.0572439
	5	-.108210	-.108221	-.108243	-.108247
	7	-.129612	.129563	-.129509	.129489
	9	.133773	.133768	.133661	.133626
	11	.133806	.133792	.133665	.133629
	13	.133773	.133768	.133661	.133626
	15	.129612	.129563	.129509	.129489
	17	-.108210	.108221	-.108243	-.108247
	19	-.0572873	-.0572615	-.0572482	-.0572439
Weighted Error (%)		0.13	0.08	0.02	—

TABLE IV.B.28 Maximum Displacements Computed by DSSI for Arch No.4

(Δt = 0.005 sec)

Modes Superposed		1	1+2+3	1+2+3+4+5	1+2+3+4+5 +6+7	DSSI
Execution Time (CP sec)		42.189	64.066	85.955	107.264	517.691
$\{Y\}^{(Y)}_{max}$ (m)	3	-.00136159	-.00675870	.00691346	.741823	.00742941
	5	-.0103159	.065465	.0157051	.0157262	.0157045
	7	-.0117619	.0168491	.0161702	.164263	.0164123
	9	-.00668519	.00940136	.00904949	.00922806	.00923708
	11	0.0	0.0	0.0	0.0	0.0
	13	.00668519	-.00940136	-.00904949	-.00922806	-.00923708
	15	.0117619	-.0168491	-.0161702	-.0164263	-.0164123
	17	.0103159	-.0165465	-.0157051	-.0157262	-.0157045
	19	.00136159	.00675870	-.00691346	-.00741823	-.00742941
$\{Y\}^{(x)}_{max}$ (m)	3	.0524169	-.0531308	-.0573253	-.0572995	-.0572482
	5	.108350	-.106662	-.108045	-.108158	-.108243
	7	.128146	-.129436	-.129419	.129488	.129509
	9	.129103	.133734	.133924	.133685	.133661
	11	.128019	.133860	.134071	.133706	.133665
	13	.129103	.133734	.133924	.133685	.133661
	15	.128146	.129436	.129419	.129488	.129509
	17	.108350	-.106662	-.108045	-.108158	-.108243
	19	.0524169	-.0531308	-.0573253	-.0572995	-.0572482
Weighted Error (%)		5.94	1.48	0.33	0.05	-

TABLE IV.B.29 Maximum Displacements Computed by MODAL for Arch No.4

(Δt = 0.005 sec.)

Modes Superposed		1	1+2+3	1+2+3+4+5	1+2+3+4+5 +6+7	DSSI
$\{y\}_t^{(Y)}$	3	-.00135614	.00330580	.00310290	.00362363	.00362587
	5	-.0102746	-.00356352	-.00399028	-.00401323	-.00404117
	7	-.0117148	-.00610011	-.00660001	-.00694636	-.00689517
	9	-.00665844	-.00364423	-.00393013	-.00418004	-.00415134
	11	0.0	0.0	0.0	0.0	0.0
	13	.00665844	.00364423	.00393013	.00418004	.00415134
	15	.0117148	.00610011	.00660001	.00694636	.00689517
	17	.0102746	.00356352	.00399028	.00401323	.00404117
	19	.00135614	-.00330580	-.00310290	-.00362363	-.00362587
$\{Y\}_t^{(x)}$	3	.0522072	.0494021	.0484706	.0483756	.0483549
	5	.107916	.105337	.105016	.105222	.105246
	7	.127633	.128980	.129390	.129459	.129478
	9	.128586	.133387	.133924	.133685	.133661
	11	.127506	.133551	.134071	.133706	.133665
	13	.128586	.133387	.133924	.133685	.133661
	15	.127633	.128980	.129390	.129459	.129478
	17	.107916	.105337	.105016	.105222	.105246
	19	.0522072	.0494021	.0484706	.0483756	.0483549
Weighted Error (%)		6.98	0.81	0.40	0.04	-

TABLE IV.B.30 Displacements of Arch No.4 at t = 58.175 secs.

It can be seen from the tables that, the mode shapes for all of the arches are either symmetric or antimetric. The participation factors for all symmetric modes are equal to zero. For step-by-step integration analysis, as the solution time step, Δt, decreases, the accuracy of the solution increases. For Δt values below 0.01 the error in the solution is below 0.1 per cent.

Since the participation factors for symmetric modes are equal to zero, they shall not be included in modal analysis. This will decrease the execution time. Contribution of a single mode may give very in accurate results; therefore, the superposition of more than one mode is generally required in modal analysis. The weighted errors of the maximum displacements for arches 1,2,3 and 4, when a single mode is included in modal summation, are 1.76, 7837, 76.55 and 5.94 per cent respectively.

IV.B.5 RESPONSE SPECTRA ANALYSIS OF THE FIVE-STOREY PLANE FRAME

The five storey plane frame given in Fig. IV.B.1 is analysed by response spectra analysis procedure. The maximum displacements are computed by peak response formulation given in Section II.F. The results of the computations are summarized in Tables IV.B.31 and IV.B.32. As the number of modes involved in modal summation increases, the error increases slightly, however the errors in the solution does not go beyond 3 per cent.

mode	ω (rad/sec)	S_a (m/sec^2)	r	$r\frac{S_a}{\omega^2}$
1	2.5643	1.5891	6.5715	1.5881
2	8.0617	1.7708	-2.6076	-0.071049
3	15.2719	0.4434	1.7253	0.0032800
4	23.4907	0.2375	1.2220	0.00052595
5	31.6202	0.4132	-0.8965	-0.00037049

TABLE IV.B.31 Multiplication Factors for 5-Storey Plane Frame to be Used in Response Spectra Analysis Procedure

Modes Superposed		1	1+2	1+2+3	1+2+3+4	1+2+3+4+5	DSSI
$\{y\}_{max}$ (m)	5	.32558	.34025	.34081	.34086	.34088	.33674
	4	.28195	.28430	.28472	.28481	.28484	.28602
	3	.21166	.22127	.22163	.22170	.22175	.21111
	2	.12696	.13920	.13962	.13964	.13970	.12895
	1	.043749	.049679	.050143	.050233	.050295	.045251
Weighted Error (%)		1.912	2.983	3.120	3.134	3.150	-

TABLE IV.B.32 Maximum Displacements Computed by Response Spectra Analysis for 5-Storey Plane Frame

IV.B.6 RESPONSE OF THE FIVE STOREY BUILDING FRAME TO FICTITIOUS STATIC LOADS

The response of the five storey building frame given in Fig.IV.B.1 to the fictitious static loads (Section II.G), which depend on the dynamic characteristics of the structure and the earthquake motion, are computed. The equivalent fictitious loads representing the responses of the modes of this structure are given in Table IV.B.33. The maximum displacements computed by the step-by-step integration method are compared with the peak response of the displacements (Table IV.B.34). The errors in the peak response increase as the number of modes involved in the modal summation increases.

Mode		1	2	3	4	5
$\{F\}_{max}$ (t)	5	16.9001	-7.5263	1.0245	-0.2504	0.1046
	4	21.9534	-1.8104	-1.1582	0.5899	-0.3402
	3	16.4805	7.3976	-0.9994	-0.4534	0.6014
	2	9.8851	9.4163	1.16826	-0.1120	-0.7686
	1	3.4064	4.5632	1.2847	0.5805	0.7349

TABLE IV.B.33 Fictitious Static Loads for 5-Storey Plane Frame

Modes Superposed		1	1+2	1+2+3	1+2+3+4	1+2+3+4+5	DSSI
$\{y\}_{max}$ (m)	5	.32559	.34026	.34082	.34088	.34089	.33674
	4	.28195	.28430	.298472	.28481	.28484	.28602
	3	.21167	.22128	.22164	.22171	.22176	.21111
	2	.12696	.1392	.13962	.13964	.13971	.12895
	1	.04375	.04968	.05015	.05024	.05030	.045251
Weighted Error (%)		1.910	3.057	3.123	3.136	3.153	-

TABLE IV.B.34 Maximum Displacements Computed by Equivalent Fictitious Static Loads Approach for 5-Storey Plane Frame

V. CONCLUSIONS

The frontal solution technique, which is used in the static analysis program, is found to be effective in solving large sets of linear equations arising in structural analysis. The solution accuracy is significantly affected by the word size of the computer and the solution algorithm used to solve the equilibrium equations.

The selection of the solution time step affects both the solution accuracy, and the execution time. Smaller values of the time step gives more accurate results;however, the required solution time also increases. For every small solution time step values, the number of arithmetic operations done at a certain entry in the main memory of the computer is very large, hence,closer to the end of the integration process the response values may be inaccurate. Therefore, in low precision computers, selection of very small time steps shall be avoided.

Linear acceleration method is not an unconditionally stable method. For some of the cases given in Section IV.B, the numerical integrations were observed to be diverging for relatively large solution time steps. A basis for selecting appropriate time step values which will always give converging results can not be stated for this method.

It has been observed that superposing the responses of a few of the highest modes gives satisfactory results. For arch type structures, superposing the response of the first mode which has a non-zero participation factor, may give very inaccurate results. When lower modes of vibration are superposed, the errors in the results may increase. This is due to the computation errors made in the lower modes of vibration.

Free vibration characteristics of arches with symmetric mass and stiffness distribution depends on their slenderness and aspect ratios. The shape of the first mode-being symmetric or antimetric-is depending on these ratios. Participation factors for symmetric mode shapes are equal to zero. Therefore, in modal analysis, superposing only those modes with none-zero participation factors will further reduce the computation time.

APPENDICES

APPENDIX A - COMPUTER PROGRAM PFRAME5

DATA INPUT TO PFRAME5

I. TITLE CARD (20A4)

Input List : TITLE

Explanation :

TITLE : Character array containing the title for the problem.

The rest of the cards are free-formatted.

II. CONTROL CARD

Input List : NNODES, NELTS, NLC, NETYPS

Explanation:

NNODES : total number of nodal points

NELTS : total number of plane frame elements

NLC : number of structure loading cases

NETYPS : number of different element types which would yield different element stiffness matrices.

III. STRUCTURE GEOMETRY CARD

Input List : NBAY, NSTRY

Explanation :

NBAY : number of bays

NSTRY : number of storeys

IV. MATERIAL PROPERTY CARDS

Input List : E

Explanation :

E : modulus of elasticity

V. OPTIONS CARD

Input List : KOPT1, KOPT2, KOPT3

Explanation :

KOPT1 : execution mode option;

= 0 ; prints both displacements and end forces

= 1 ; prints end foces only

= 2 ; prints displacements only

KOPT2 : option for printing the details of the problem solution ;

= 0 ; does not print details

≠ 0 ; prints the element stiffness

matrices, fixed end forces in local coordinates, and details of the frontal solution process

KOPT3 : option for comparing the computed horizontal displacement distribution to triangular displacement distribution.
= 0 ; do not compare
≠ 0 ; compare

VI. ELEMENT TYPE CARDS

Input List : HEAD

Explanation :

MTYPE : element type number

A : cross sectional area

XI : moment of inertia

EM,EM : cosines of the angles between the local y axis and the global X and Y axes, respectively.
(See Figures II.1 and II.3)

Note : This card must be repeated NETYPS times.

VII. BOUNDARY CONDITIONS CARDS

Input List : NNUM, (NDC(I), I=1,3), ISTOP

Explanation :

NNUM : node number at which fixed boundary conditions are specified

NDC : Array containing the boundary condition codes ;

NDC(1) ; X-translation boundary condition code

NDC(2) ; Y-translation boundary condition code

NDC(3) ; Z-rotation boundary condition code

= 0 ; free (displacements and loads allowed)

≠ 0 ; fixed (no displacements or loads allowed)

ISTOP : parameter to stop the boundary condition data input;

= 0 ; a new boundary condition card will follow

≠ 0 ; end of boundary condition cards

Note : This set of cards must be repeated until ISTOP ≠ 0. Any number of cards can be given to describe the fixed boundary conditions. The rest of the nodal boundary conditions are set to be free.

VIII. ELEMENT DATA CARDS

VIII. A ELEMENT CONNECTIVITY AND TYPE DATA CARDS

Input List : NEL, JNO1, JNO2, MTYP, LDCOD

Explanation :

NEL	:	element number
JNO1	:	node number I
JNO2	:	node number J (see Fig III.2 and III.3)
MTYP	:	element type number
LDCOD	:	parameter for the existance of element loads; = 0 ; no element loads exist ≠ 0 ; element loading cards will follow this card.

VIII. B ELEMENT LOADING CARDS

Input List : LD, LDTP, Q, X, Y, ESTOP

Explanation :

LD	:	structural load case number, which the element load given by this card is acting
LDTP	:	element loading type number; = 1 ; uniformly distributed transverse load

= 2 ; concentrated transverse load

= 3 ; uniformly distributed axial load

= 4 ; concentrated axial load

X : left margin

Y : right margin

Q : magnitude of the load

ISTOP : parameter to stop the element loadings for element number NEL;

= 0 ; a new element loading card acting on element number NEL will follow this card

≠ 0 ; end of element loading cards for this element

Card(s) VIII.B must be skipped if LDCOD=0.

Note : The element loading parameters must be defined in local coordinates. Concentrated joint loads can be defined as concentrated element loads acting at nodes I or J of the element. Positive directions of the element loads are given in Fig. A.1.

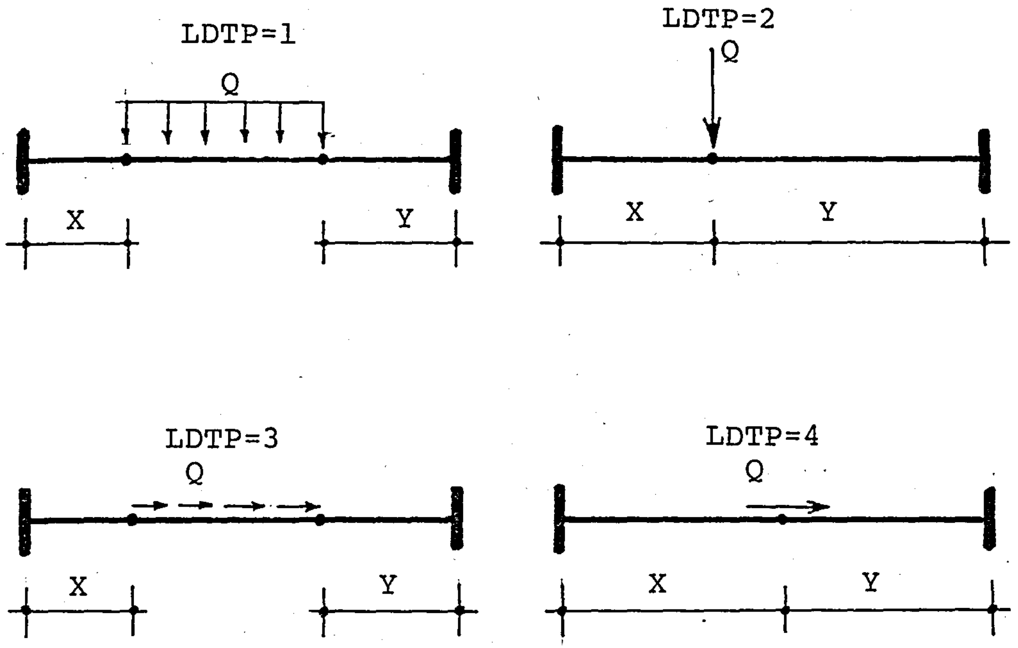


FIGURE A.1 Element Load types for PFRAME5.

Note : Cards VIII.A and VIII.B must be repeated NELTS times

IX. ACTUAL TO TRIANGULAR DISPLACEMENT DISTRIBUTION COMPARISON CARDS

XI.A

Input List : NND

Explanation:

NND : number of nodal point for which the comparison will be performed

IX.B

Input List : NODE(I),Y(I),I=1,NND)

Explanation:

NODE(I) : Node number

Y(I) : Elevation of node number NODE(I) measured from foundation level

Note : Node number NODE(I) must be a node located at the top-most storey.

```
      OPEN(UNIT=8,STATUS='NEW',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     FILE='TAPE8',RECL=NLC)
C
C     (TAPE 9 CONTAINS SYSTEM STIFFNESS MATRIX IN ORIGINAL FORM)
C
      OPEN(UNIT=9,STATUS='NEW',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     FILE='TAPE9',RECL=NBAND)
C
C     (TAPE 10 CONTAINS NODE INFORMATION)
C
      OPEN(UNIT=10,STATUS='NEW',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     FILE='TAPE10',RECL=1+NUN)
C
C     (TAPE 11 CONTAINS ELEMENT TYPE INFORMATION)
C
      OPEN(UNIT=11,STATUS='SCRATCH',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     RECL=3+21)
C
C     (TAPE 12 CONTAINS ELEMENT INFORMATION)
C
      OPEN(UNIT=12,STATUS='SCRATCH',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     RECL=1+NUE+NUE*NLC)
C
C     (TAPE 13 CONTAINS SYSTEM STIFFNESS MATRIX IN TRIANGULIZED FORM)
C
      OPEN(UNIT=13,STATUS='SCRATCH',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     RECL=NBAND)
C
C     (TAPE 14 CONTAINS RIGHT HAND SIDE MATRIX)
C
      OPEN(UNIT=14,STATUS='NEW',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     FILE='TAPE14',RECL=NLC)
C
      CALL ELTYPE(NOESM)
      CALL NODDAT
      CALL SOLVE(*50)
      IF(KOPT1.EQ.1) GO TO 10
      CALL PRTDISP
      IF(KOPT3.NE.0) CALL CDTR(NUN,NNODES)
      IF(KOPT1.EQ.2) GO TO 50
   10 CALL ENDFOR
C
   50 WRITE(6,303)
      STOP
C
C *****************************************************************
C     FORMATS
C *****************************************************************
C
  201 FORMAT(20A4)
  301 FORMAT(1H1,///20X,20A4,/)
  302 FORMAT(20X,'NUMBER OF NODES          = ',I5,/
     1       20X,'NUMBER OF ELEMENTS       = ',I5,/
     2       20X,'NUMBER OF LOADING CASES  = ',I5,/
     3       20X,'NUMBER OF ELEMENT TYPES  = ',I5,/
     4       20X,'MODULUS OF ELASTICITY    = ',G13.7,/
     5       20X,'EXECUTION MODE OPTION    = ',I5,/
```

```
      6        20X,'PRINT OPTION            = ',I5,/
      7        20X,'TRIANG. COMPARISON OPT. = ',I5,)
  303 FORMAT(1H1,//)
  304 FORMAT(/20X,'NUMBER OF BAYS    = ',I5,/
     1         20X,'NUMBER OF STORYS = ',I5,/
     2         20X,'HALF BANDWIDTH   = ',I5,//)
  305 FORMAT(//10X,'MAXNEM MUST BE .GE. MAXBAN',
     1        /20X,'(MAXNEM = ',I5,
     2        /20X,' MAXBAN = ',I5,' )')
  306 FORMAT(//20X,'NBAND MUST BE .LE. MAXBAN',
     1        /20X,'(MAXBAN = ',I5,' )')
  307 FORMAT(//20X,'NLC MUST BE .LE. MNLC',
     1        /20X,'(MNLC = ',I5,' )')
      END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	1B	/MEMBER/		REAL	
E	0B	/MEMBER/		REAL	
EM	4B	/MEMBER/		REAL	
EN	5B	/MEMBER/		REAL	
FEML	212B	/MEMBER/		REAL	30
KOPT1	0B	/OPTION/		INTEGER	
KOPT2	1B	/OPTION/		INTEGER	
KOPT3	532B			INTEGER	
LUBR	6B	/SYSVAL/		INTEGER	
MAXBAN	6B	/MAXVAL/		INTEGER	
MAXNEM	5B	/MAXVAL/		INTEGER	
MAXNES	7B	/MAXVAL/		INTEGER	
MESM	204B	/MEMBER/		INTEGER	6
MNLC	4B	/MAXVAL/		INTEGER	
NBAND	5B	/SYSVAL/		INTEGER	
NBAY	7B	/SYSVAL/		INTEGER	
NDC	0B	/JOINT/		INTEGER	6
NELTS	1B	/SYSVAL/		INTEGER	
NETYPS	3B	/SYSVAL/		INTEGER	
NLC	2B	/SYSVAL/		INTEGER	
NNE	0B	/MAXVAL/		INTEGER	
NNODES	0B	/SYSVAL/		INTEGER	
NOESM	3B	/MAXVAL/		INTEGER	
NSTRY	531B			INTEGER	
NUE	2B	/MAXVAL/		INTEGER	
NUN	1B	/MAXVAL/		INTEGER	
NUS	4B	/SYSVAL/		INTEGER	
RHS	1356B	/SYSTEM/		REAL	250
RHSS	3326B	/SYSTEM/		REAL	250
S	0B	/SYSTEM/		REAL	750
SM	6B	/MEMBER/		REAL	126
SS	1750B	/SYSTEM/		REAL	750
TITLE	0B	/HEAD/		REAL	20
XI	2B	/MEMBER/		REAL	
XL	3B	/MEMBER/		REAL	

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
CDTR		2	SUBROUTINE
ELTYPE		1	SUBROUTINE
ENDFOR		0	SUBROUTINE
NODDAT		0	SUBROUTINE
PRTDISP		0	SUBROUTINE
SOLVE		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

LABEL	ADDRESS	PROPERTIES	DEF
10	167B		96
50	171B		98
201	223B	FORMAT	105
301	225B	FORMAT	106
302	230B	FORMAT	107
303	270B	FORMAT	115
304	272B	FORMAT	116
305	305B	FORMAT	119
306	317B	FORMAT	122
307	327B	FORMAT	124

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PFRAMES    20B     0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------     -NAME--- PROPERTIES-------------     -NAME--- PROPERTIES-------------

  TAPE10  AUX                          TAPE13  AUX                          TAPE6   FMT/SEQ
  TAPE11  AUX                          TAPE14  AUX                          TAPE8   AUX
  TAPE12  AUX                          TAPE5   FMT/SEQ                      TAPE9   AUX

--STATISTICS--

  PROGRAM-UNIT LENGTH             533B =   347
  SCM LABELLED COMMON LENGTH     4244B =  2212
  SCM STORAGE USED              63700B = 26560
  COMPILE TIME                  0.322 SECONDS
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=B,L=LPFR5.

```
      SUBROUTINE NODDAT
C
      COMMON /MAXVAL/ NNE,NUN,NUF,NOESM,MNLC,MAXNEM,MAXBAN,MAXNES
      COMMON /SYSVAL/ NNODES,NELTS,NLC,NETYPS,NUS,NBAND,LUBR,NBAY
      COMMON /JOINT / NDC(6)
C
      JFLG=0
      WRITE(6,300)
C
      DO 10 I=1,NUN
   10   NDC(I)=0
      DO 20 NNUM=1,NNODES
   20   WRITE(10,REC=NNUM) JFLG,(NDC(I),I=1,NUN)
C
   40 READ(5,*) NNUM,(NDC(I),I=1,NUN),ISTOP
      DO 30 I=1,NUN
        IF(NDC(I).NE.0) NUS=NUS-1
   30 CONTINUE
      WRITE(6,301) NNUM,(NDC(I),I=1,NUN)
      WRITE(10,REC=NNUM)JFLG,(NDC(I),I=1,NUN)
      IF(ISTOP.EQ.0) GO TO 40
      LUBR=NUS-NBAND+1
C
      WRITE(6,302) NUS,LUBR
      RETURN
C ***********************************************************************
C     FORMATS
C ***********************************************************************
  300 FORMAT(//20X,'NODAL DATA ( FIXED DIRECTIONS ONLY )',
     1       /19X,38(1H-),//5X,'NODE',3X,'    DEFORMATION CODES  ',
     2           /4X,6(1H-),2X,22(1H-))
  301 FORMAT(4X,I5,3X,3(I6,2X))
  302 FORMAT(//20X,'NUMBER OF UNKNOWNS FOR THE SYSTEM      = ',I5,
     1           /20X,'LENGTH OF THE UNIFORMLY BANDED REGION = ',I5,/)
      END
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
I	213B			INTEGER	
ISTOP	217B			INTEGER	
JFLG	212B			INTEGER	
LUBR	6B	/SYSVAL/		INTEGER	
MAXBAN	6B	/MAXVAL/		INTEGER	
MAXNEM	5B	/MAXVAL/		INTEGER	
MAXNES	7B	/MAXVAL/		INTEGER	
MNLC	4B	/MAXVAL/		INTEGER	
NBAND	5B	/SYSVAL/		INTEGER	
NBAY	7B	/SYSVAL/		INTEGER	
NDC	0B	/JOINT/		INTEGER	6
NELTS	1B	/SYSVAL/		INTEGER	
NETYPS	3B	/SYSVAL/		INTEGER	
NLC	2B	/SYSVAL/		INTEGER	
NNE	0B	/MAXVAL/		INTEGER	
NNODES	0B	/SYSVAL/		INTEGER	
NNUM	215B			INTEGER	
NOESM	3B	/MAXVAL/		INTEGER	
NUF	2B	/MAXVAL/		INTEGER	
NUN	1B	/MAXVAL/		INTEGER	
NUS	4B	/SYSVAL/		INTEGER	

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM         11           300     123B     FORMAT          29
    20  INACTIVE   DO-TERM         13           301     141B     FORMAT          32
    30  INACTIVE   DO-TERM         18           302     144B     FORMAT          33
    40     43B                     15

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  NODDAT      4B      0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE10  BIN/DIR
  TAPE5   FMT/SEQ
  TAPE6   FMT/SEQ

--STATISTICS--

  PROGRAM-UNIT LENGTH              221B =  145
  SCM LABELLED COMMON LENGTH        26B =  22
  SCM STORAGE USED               61700B =  25536
  COMPILE TIME                   0.364 SECONDS
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=3,L=LPFR5.
```

```
      SUBROUTINE ELTYPE(NOESM)
C
      COMMON /SYSVAL/ NNODES,NELTS,NLC,NETYPS,NUS,NBAND,LUBR,NBAY
      COMMON /MEMBER/ E,A,XI,XL,EM,EN,SM(6,21),MESM(6),FEML(6,5)
C
      WRITE(6,301)
      DO 10 MT=1,NETYPS
        READ(5,*) MTYPE,A,XI,XL,EM,EN
        WRITE(6,302) MTYPE,A,XI,XL,EM,EN
        IF(ABS(EN).LT.1.0E-10) THEN
          CALL STIFFH(1)
        ELSE IF(ABS(EM).LT.1.0E-10) THEN
               CALL STIFFV(1)
             ELSE
               CALL STIFF(1)
             ENDIF
        WRITE(11,REC=MTYPE) XL,EM,EN,(SM(1,J),J=1,NOESM)
   10 CONTINUE
      RETURN
C ***********************************************************************
C     FORMATS
C ***********************************************************************
  301 FORMAT(///10X,'ELEMENT TYPE INFORMATION',
     1       /9X,26(1H-),//5X,' TYPE ',2X,'     AREA      ',2X,
     2       ' MOM. INERT. ',2X,'   LENGTH     ',2X,
     3       '      DIRECTION COSINES        ',/4X,6(1H-),3(2X,13(1H-)),
     4       2X,28(1H-))
  302 FORMAT(5X,I4,3(2X,G13.7),2(3X,G13.7))
      END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	1B	/MEMBER/		REAL	
E	0B	/MEMBER/		REAL	
EM	4B	/MEMBER/		REAL	
EN	5B	/MEMBER/		REAL	
FEML	212B	/MEMBER/		REAL	30
J	200B			INTEGER	
LUBR	6B	/SYSVAL/		INTEGER	
MESM	204B	/MEMBER/		INTEGER	6
MT	175B			INTEGER	
MTYPE	177B			INTEGER	
NBAND	5B	/SYSVAL/		INTEGER	
NBAY	7B	/SYSVAL/		INTEGER	
NELTS	1B	/SYSVAL/		INTEGER	
NETYPS	3B	/SYSVAL/		INTEGER	
NLC	2B	/SYSVAL/		INTEGER	
NNODES	0B	/SYSVAL/		INTEGER	
NOESM	1	DUMMY-ARG		INTEGER	
NUS	4B	/SYSVAL/		INTEGER	
SM	6B	/MEMBER/		REAL	126
XI	2B	/MEMBER/		REAL	
XL	3B	/MEMBER/		REAL	

--PROCEDURES--(LO=A)

NAME	TYPE	ARGS	CLASS
ABS	GENERIC	1	INTRINSIC
STIFF		1	SUBROUTINE
STIFFH		1	SUBROUTINE
STIFFV		1	SUBROUTINE

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS------PROPERTIES----DEF

     10  INACTIVE    DO-TERM         18
    301     102B     FORMAT          23
    302     127B     FORMAT          28

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ELTYPE       5B     1

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE11  BIN/DIR
  TAPE5   FMT/SEQ
  TAPE6   FMT/SEQ

--STATISTICS--

   PROGRAM-UNIT LENGTH              204B =  132
   SCM LABELLED COMMON LENGTH       260B =  176
   SCM STORAGE USED               61700B =  25536
   COMPILE TIME                   0.329 SECONDS
```

```
      SUBROUTINE SOLVE(*)
C
      COMMON /MAXVAL/ NNE,NUN,NUE,NOESM,MNLC,MAXNEM,MAXBAN,MAXNES
      COMMON /SYSVAL/ NNODES,NELTS,NLC,NETYPS,NUS,NBAND,LUBR,NBAY
      COMMON /SYSTEM/ S(50,15),RHS(50,5),SS(50,15),RHSS(50,5)
      COMMON /MEMBER/ E,A,XI,XL,EM,EN,SM(6,21),MESM(6),FEML(6,5)
      COMMON /JOINT / NDC(6)
      COMMON /OPTION/ KOPT1,KOPT2
C
      DIMENSION T(6,6)
      LOCN(I,J)=NUE*I-I*(I-1)/2-(NUE-J)
C
C
      WRITE(6,301)
      DET=1.0
      IPOW=0
      NCRIT=MAXNEM-NUE
      NEQ=0
      NEQ1=0
      NESM=0
      NBEG=1
C
C
      DO 10 ME=1,NELTS
C
C     COMPUTE CODE NUMBERS
C
      READ(5,*) NEL,JNO1,JNO2,MTYP,LDCOD
      READ(10,REC=JNO1) JFLG1,(NDC(I),I=1,NUN)
      READ(10,REC=JNO2) JFLG2,(NDC(I),I=1+NUN,NUE)
      IF(JFLG1.NE.0) GO TO 20
      JFLG1=1
      DO 30 I=1,NUN
        IF(NDC(I).EQ.0) THEN
          NEQ=NEQ+1
          NDC(I)=NEQ
        ELSE
          NDC(I)=0
        ENDIF
   30 CONTINUE
      WRITE(10,REC=JNO1) JFLG1,(NDC(I),I=1,NUN)
   20 IF(JFLG2.NE.0) GO TO 40
      JFLG2=1
      DO 50 I=1+NUN,NUE
        IF(NDC(I).EQ.0) THEN
          NEQ=NEQ+1
          NDC(I)=NEQ
        ELSE
          NDC(I)=0
        ENDIF
   50 CONTINUE
      WRITE(10,REC=JNO2) JFLG2,(NDC(I),I=1+NUN,NUE)
   40 WRITE(6,302) NEL,JNO1,JNO2,MTYP,(NDC(I),I=1,NUE),LDCOD
C
C     OBTAIN ELEMENT STIFFNESS MATRIX EITHER FROM MAIN MEMORY
```

```
      C     OR READ FROM TAPE11, COMPUTE THE TRANSPOSE OF THE
      C     TRANSFORMATION MATRIX
      C
            DO 60 I=1,MAXNES
              IF(MTYP.EQ.MESM(I)) THEN
                KSM=I
                IF(LDCOD.NE.0) THEN
                  READ(11,REC=MTYP)XL,EM,EN
                  CALL TRANS(EM,EN,T)
                  T(1,2)=-T(1,2)
                  T(2,1)=-T(2,1)
                  T(4,5)=-T(4,5)
                  T(5,4)=-T(5,4)
                ENDIF
                GO TO 70
              ENDIF
         60 CONTINUE
            NESM=NESM+1
            IF(NESM.GT.MAXNES) NESM=1
            KSM=NESM
            MESM(KSM)=MTYP
            READ(11,REC=MTYP) XL,EM,EN,(SM(KSM,I),I=1,NOESM)
              IF(LDCOD.NE.0) THEN
                CALL TRANS(EM,EN,T)
                T(1,2)=-T(1,2)
                T(2,1)=-T(2,1)
                T(4,5)=-T(4,5)
                T(5,4)=-T(5,4)
              ENDIF
         70 IF(KOPT2.NE.0) WRITE(6,305) (SM(KSM,I),I=1,NOESM)
      C
      C     ASSEMBLE THE ELEMENT STIFFNESS MATRIX INTO
      C     THE SYSTEM STIFFNESS MATRIX
      C
            DO 80 I=1,NUE
              IF(NDC(I).EQ.0) GO TO 80
              IS=NDC(I)-NEQ1
              DO 90 J=1,NUE
                IF(NDC(J).EQ.0) GO TO 90
                IF(NDC(J).LT.NDC(I)) GO TO 90
                JS=NDC(J)-NDC(I)+1
                IE=I
                JE=J
                IF(JE.LT.IE) THEN
                  IE=J
                  JE=I
                ENDIF
                IJE=LOCN(IE,JE)
                S(IS,JS)=S(IS,JS)+SM(KSM,IJE)
                SS(IS,JS)=SS(IS,JS)+SM(KSM,IJE)
         90   CONTINUE
         80 CONTINUE
            DO 120 I=1,NUE
            DO 120 J=1,NLC
        120   FEML(I,J)=0.0
            IF(LDCOD) 100,110,100
      C
```

```
  100 IF(KOPT2.NE.0) WRITE(6,306) ((T(I,J),J=1,NUE),I=1,NUE)
C
C     COMPUTE THE FIXED END FORCES IN LOCAL COORDINATES
C
  130 READ(5,*) LD,LDTP,Q,X,Y,ISTOP
      WRITE(6,304) LD,LDTP,Q,X,Y
      CALL LDTYPS(NEL,LD,LDTP,Q,X,Y,*500)
      IF(ISTOP.EQ.0) GO TO 130
C
      IF(KOPT2.NE.0) THEN
        WRITE(6,307)
        DO 180 I=1,NUE
  180     WRITE(6,308) (FEML(I,J),J=1,NLC)
        WRITE(6,309)
      ENDIF
C
C     TRANSFORM THE FIXED END FORCES TO GLOBAL COORDINATES
C     AND SUBTRACT FROM THE RIGHT HAND SIDE MATRIX
C
      DO 140 J=1,NLC
        DO 150 I=1,NUE
          IF(NDC(I).EQ.0) GO TO 150
          SUM=0.0
          DO 160 K=1,NUE
  160       SUM=SUM+T(I,K)*FEML(K,J)
        IS=NDC(I)-NEQ1
        RHS(IS,J)=RHS(IS,J)-SUM
        RHSS(IS,J)=RHSS(IS,J)-SUM
  150   CONTINUE
  140 CONTINUE
C
  110 WRITE(12,REC=NEL) MTYP,NDC,((FEML(I,J),J=1,NLC),I=1,NUE)
      IF(KOPT2.NE.0) WRITE(6,310)
C
C     DECIDE WHETHER CONTINUE TO THE ASSEMBLY OR TO ELIMINATE
C
      IF(NEQ.EQ.NUS) GO TO 10
      IF(NEQ-NEQ1 .LE. NCRIT) GO TO 10
      CALL ELMNTE(NEL,ME,NEQ,NEQ1,NBEG,DET,IPOW,*500)
C
   10 CONTINUE
      ME=NELTS
      IF(NEQ1.LT.NUS-1) THEN
        CALL ELMNTE(NEL,ME,NEQ,NEQ1,NBEG,DET,IPOW,*500)
      ENDIF
         DET=DET*S(NEQ-NEQ1,1)
         IEX=ALOG10(DET)
         DET=DET/10.0**IEX
         IPOW=IPOW+IEX
      WRITE(9,REC=NUS) (SS(NUS-NEQ1,J),J=1,NBAND)
      WRITE(13,REC=NUS) (S(NUS-NEQ1,J),J=1,NBAND)
      WRITE(14,REC=NUS) (RHS(NUS-NEQ1,J),J=1,NLC)
      WRITE(8,REC=NUS) (RHSS(NUS-NEQ1,J),J=1,NLC)
C
      WRITE(6,311) DET,IPOW
C
C     BACKSUBSTITUTE
```

```
      C
            CALL BACSUB
      C
            RETURN
        500 RETURN 1
      C ******************************************************************
      C     FORMATS
      C ******************************************************************
        301 FORMAT(1H1,//10X,'SOLUTION',/9X,10(1H-),
           1        /5X,' ELT. ',2X,'NODE 1 ',2X,'NODE 2 ',2X,'TYPE',
           2        2X,'           CODE NUMBERS           ',2X,'LDCOD',
           3        /5X,6(1H-),2(2X,7(1H-)),2X,4(1H-),2X,36(1H-),2X,5(1H-))
        302 FORMAT(5X,I6,3X,I5,4X,I5,3X,I4,2X,6(I4,2X),2X,I5)
        304 FORMAT(11X,'LOADING CASE = ',I3,2X,'TYPE = ',I2,2X,'MAGNITUDE = ',
           1        G13.7,2X,'L.MAR.= ',G13.7,2X,'R.MAR.= ',G13.7)
        305 FORMAT(/20X,'ELEMENT STIFFNESS MATRIX',/,(8(2X,G13.7)))
        306 FORMAT(/20X,'TRANSPOSE OF TRANSFORMATION MATRIX',
           1        /,6(6(2X,G13.7),/))
        307 FORMAT(/20X,'FIXED END FORCES IN LOCAL COORDINATES')
        308 FORMAT(8(2X,G13.7))
        309 FORMAT(1X)
        310 FORMAT(1X,120(1H-))
        311 FORMAT(/20X,'DETERMINANT = ',G13.7,'  POWER = ',I10,/)
            END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	1B	/MEMBER/		REAL	
DET	1577B			REAL	
E	0B	/MEMBER/		REAL	
EM	4B	/MEMBER/		REAL	
EN	5B	/MEMBER/		REAL	
FEML	212B	/MEMBER/		REAL	30
I	1575B			INTEGER	
IE	1633B			INTEGER	
IEX	1661B			INTEGER	
IJE	1635B			INTEGER	
IPOW	1600B			INTEGER	
IS	1630B			INTEGER	
ISTOP	1647B			INTEGER	
J	1576B			INTEGER	
JE	1634B			INTEGER	
JFLG1	1615B			INTEGER	
JFLG2	1616B			INTEGER	
JNO1	1611B			INTEGER	
JNO2	1612B			INTEGER	
JS	1632B			INTEGER	
K	1655B			INTEGER	
KOPT1	0B	/OPTION/		INTEGER	
KOPT2	1B	/OPTION/		INTEGER	
KSM	1624B			INTEGER	
LD	1642B			INTEGER	
LDCOD	1614B			INTEGER	
LDTP	1643B			INTEGER	
LUBR	6B	/SYSVAL/		INTEGER	

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
MAXBAN	6B	/MAXVAL/		INTEGER	
MAXNEM	5B	/MAXVAL/		INTEGER	
MAXNES	7B	/MAXVAL/		INTEGER	
ME	1606B			INTEGER	
MESM	204B	/MEMBER/		INTEGER	6
MNLC	4B	/MAXVAL/		INTEGER	
MTYP	1613B			INTEGER	
NBAND	5B	/SYSVAL/		INTEGER	
NBAY	7B	/SYSVAL/		INTEGER	
NBEG	1605B			INTEGER	
NCRIT	1601B			INTEGER	
NDC	0B	/JOINT/		INTEGER	6
NEL	1610B			INTEGER	
NELTS	1B	/SYSVAL/		INTEGER	
NEQ	1602B			INTEGER	
NEQ1	1603B			INTEGER	
NESM	1604B			INTEGER	
NETYPS	3B	/SYSVAL/		INTEGER	
NLC	2B	/SYSVAL/		INTEGER	
NNE	0B	/MAXVAL/		INTEGER	
NNODES	0B	/SYSVAL/		INTEGER	
NOESM	3B	/MAXVAL/		INTEGER	
NUE	2B	/MAXVAL/		INTEGER	
NUN	1B	/MAXVAL/		INTEGER	
NUS	4B	/SYSVAL/		INTEGER	
Q	1644B			REAL	
RHS	1356B	/SYSTEM/		REAL	250
RHSS	3326B	/SYSTEM/		REAL	250

```
      SUBROUTINE SOLVE       74/176  OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+577          85/09/19. 17.33.06         PAGE    5
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE----------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  S            0B  /SYSTEM/                    REAL          750      X        1645B                                REAL
  SM           6B  /MEMBER/                    REAL          126      XI          2B  /MEMBER/                      REAL
  SS       1750B   /SYSTEM/                    REAL          750      XL          3B  /MEMBER/                      REAL
  SUM      1654B                               REAL                   Y        1646B                                REAL
  T        1531B                               REAL           36


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----        -NAME------TYPE--------ARGS------CLASS-----

  ALOG10    REAL         1       INTRINSIC          LDTYPS                    7       SUBROUTINE
  BACSUB                 0       SUBROUTINE         LOCN      INTEGER         2       STAT FUNC
  ELMNTE                 8       SUBROUTINE         TRANS                     3       SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF          -LABEL-ADDRESS-----PROPERTIES----DEF          -LABEL-ADDRESS-----PROPERTIES----DEF

     10     755B      DO-TERM       153           110      700B                     144           304     1212B      FORMAT        183
     20     121B                     42           120   INACTIVE     DO-TERM        110           305     1226B      FORMAT        185
     30   INACTIVE    DO-TERM        40           130      535B                     117           306     1234B      FORMAT        186
     40     174B                     53           140   INACTIVE     DO-TERM        142           307     1244B      FORMAT        188
     50   INACTIVE    DO-TERM        51           150      666B      DO-TERM        141           308     1252B      FORMAT        189
     60   INACTIVE    DO-TERM        72           160   INACTIVE     DO-TERM        137           309     1255B      FORMAT        190
     70     316B                     85           180   INACTIVE     DO-TERM        125           310     1257B      FORMAT        191
     80     443B      DO-TERM       107           301     1157B      FORMAT         178           311     1262B      FORMAT        192
     90     436B      DO-TERM       106           302     1204B      FORMAT         182           500     1140B                    174
    100   INACTIVE                  113


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  SOLVE       5B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------       -NAME--- PROPERTIES-------------       -NAME--- PROPERTIES-------------

  TAPE10  BIN/DIR                        TAPE13  BIN/DIR                        TAPE6   FMT/SEQ
  TAPE11  BIN/DIR                        TAPE14  BIN/DIR                        TAPE8   BIN/DIR
  TAPE12  BIN/DIR                        TAPE5   FMT/SEQ                        TAPE9   BIN/DIR


--STATISTICS--

  PROGRAM-UNIT LENGTH              1666B =   950
  SCM LABELLED COMMON LENGTH      4220B =   2192
  SCM STORAGE USED               63700B =   26560
  COMPILE TIME                   2.091 SECONDS
```

```
      SUBROUTINE ELMNTE(NEL,ME,NEQ,NEQ1,NBEG,DET,IPOW,*)
C
      COMMON /MAXVAL/ NNE,NUM,NUE,NOESM,MNLC,MAXNEM,MAXBAN,MAXNES
      COMMON /SYSVAL/ NNODES,NELTS,NLC,NETYPS,NUS,NBAND,LUBR,NBAY
      COMMON /SYSTEM/ S(50,15),RHS(50,5),SS(50,15),RHSS(50,5)
      COMMON /OPTION/ KOPT1,KOPT2
C
C     COMPUTE THE NUMBER OF EQUATIONS FULLY SUMMED SO FAR
C
      IGR=2*NBAY+1
      NFSEQ=ME/IGR*(NBAY+1)*NUN
      NEAS=ME-ME/IGR*IGR
      IF(NEAS.GT.NBAY) NFSEQ=NFSEQ+(NEAS-NBAY)*NUN
C
C     CHECK IF THERE IS ANY EQUATION TO BE ELIMINATED
C
      IF(NFSEQ.LE.NEQ1) GO TO 10
C
C     SET THE RANGE OF EQUATIONS TO BE ELIMINATED (INCLUSIVE)
C
      NEND=NFSEQ-NEQ1
      IF(NEND.GT.MAXNEM-NBAND+1) NEND=MAXNEM-NBAND+1
      IF(ME.EQ.NELTS) NEND=NUS-1-NEQ1
C
      IF(KOPT2.NE.0) THEN
        WRITE(6,302) NEL,ME,NEQ,NEQ1,IGR,NEAS,NFSEQ,NBEG,NEND
        WRITE(6,303)
        DO 20 I=1,MAXNEM
   20     WRITE(6,304) I,(S(I,J),J=1,MAXBAN)
        WRITE(6,305)
        DO 30 I=1,MAXNEM
   30     WRITE(6,304) I,(RHS(I,J),J=1,MNLC)
      ENDIF
C
C *** PERFORM THE ELIMINATION FROM NBEG TO NEND ***
C
      K1=NBEG
      K2=NEND
      IF(K1.GT.LUBR-NEQ1) GO TO 80
      IF(K2.GT.LUBR-NEQ1) K2=LUBR-NEQ1
C
      IF(KOPT2.NE.0) WRITE(6,309) K1,K2
      DO 50 K=K1,K2
C
C       WRITE THE FULLY SUMMED EQUATIONS ON TAPE9 AND TAPE8
C
        WRITE(9,REC=K+NEQ1) (SS(K,J),J=1,NBAND)
        WRITE(8,REC=K+NEQ1) (RHSS(K,J),J=1,NLC)
C
C       COMPUTE THE DETERMINANT
C
        DET=DET*S(K,1)
        IEX=ALOG10(DET)
        DET=DET/10.0**IEX
```

```
      C
            DO 60 I=K+1,K+NBAND-1
              JJ=I-K+1
              R=S(K,JJ)/S(K,1)
      C
      C       ELIMINATE THE SYSTEM STIFFNESS MATRIX
      C
              DO 70 J=I,K+NBAND-1
                JJ=J-I+1
                JJ1=J-K+1
         70     S(I,JJ)=S(I,JJ)-R*S(K,JJ1)
      C
      C       ELIMINATE THE RIGHT HAND SIDE MATRIX
      C
              DO 60 J=1,NLC
         60     RHS(I,J)=RHS(I,J)-R*RHS(K,J)
      C
      C     WRITE THE FULLY SUMMED AND ELIMINATED EQUATIONS ON TAPES 13 AND 14
      C
            WRITE(13,REC=K+NEQ1) (S(K,J),J=1,NBAND)
            WRITE(14,REC=K+NEQ1) (RHS(K,J),J=1,NLC)
         50 CONTINUE
      C
          IF(K2.EQ.NEND) GO TO 130
          K1=K2+1
          K2=NEND
          IF(KOPT2.NE.0) WRITE(6,309) K1,K2
         80 DO 90 K=K1,K2
      C
      C     WRITE THE FULLY SUMMED EQUATIONS ON TAPE 9 AND TAPE 8
      C
            WRITE(9,REC=K+NEQ1) (SS(K,J),J=1,NBAND)
            WRITE(8,REC=K+NEQ1) (RHSS(K,J),J=1,NLC)
      C
      C     COMPUTE THE DETERMINANT
      C
            DET=DET*S(K,1)
            IEX=ALOG10(DET)
            DET=DET/10.0**IEX
            IPOW=IPOW+IEX
      C
            DO 100 I=K+1,NUS-NEQ1
              JJ=I-K+1
              R=S(K,JJ)/S(K,1)
      C
      C       ELIMINATE THE SYSTEM STIFFNESS MATRIX
      C
              DO 110 J=I,NUS-NEQ1
                JJ=J-I+1
                JJ1=J-K+1
        110     S(I,JJ)=S(I,JJ)-R*S(K,JJ1)
      C
      C       ELIMINATE THE RIGHT HAND SIDE
      C
              DO 100 J=1,NLC
        100     RHS(I,J)=RHS(I,J)-R*RHS(K,J)
      C
```

```
C       WRITE THE FULLY SUMMED AND ELIMINATED EQUATIONS ON TAPES 13 AND 14
C
        WRITE(13,REC=K+NEQ1) (S(K,J),J=1,NBAND)
        WRITE(14,REC=K+NEQ1) (RHS(K,J),J=1,NLC)
   90 CONTINUE
C
  130 IF(KOPT2.NE.0) THEN
        WRITE(6,306)
        DO 120 I=1,MAXNEM
  120     WRITE(6,304) I,(S(I,J),J=1,MAXBAN)
        WRITE(6,305)
        DO 125 I=1,MAXNEM
  125     WRITE(6,304) I,(RHS(I,J),J=1,MNLC)
      ENDIF
C
      IF(ME.EQ.NELTS) GO TO 140
C
C     SHIFT THE EQUATIONS ABOVE
C
      IDIFR=NEND-NBEG+1
      DO 150 K=NEND+1,NEQ-NEQ1
        DO 160 J=1,NBAND
          SS(K-IDIFR,J)=SS(K,J)
  160     S(K-IDIFR,J)=S(K,J)
        DO 150 J=1,NLC
          RHSS(K-IDIFR,J)=RHSS(K,J)
  150     RHS(K-IDIFR,J)=RHS(K,J)
C
C     SET THE REST OF THE EQUATIONS EQUAL TO ZERO
C
      DO 170 K=(NEQ-NEQ1)-IDIFR+1,NEQ-NEQ1
        DO 180 J=1,NBAND
          SS(K,J)=0.0
  180     S(K,J)=0.0
        DO 170 J=1,NLC
          RHSS(K,J)=0.0
  170     RHS(K,J)=0.0
C
      IF(KOPT2.NE.0) THEN
        WRITE(6,307)
        DO 190 I=1,MAXNEM
  190     WRITE(6,304) I,(S(I,J),J=1,MAXBAN)
        WRITE(6,305)
        DO 200 I=1,MAXNEM
  200     WRITE(6,304) I,(RHS(I,J),J=1,MNLC)
      ENDIF
C
C     SET THE POINTER TO ITS NEW POSITION
C
      NEQ1=NEND+NEQ1
C
  140 WRITE(6,308)
      RETURN
C
   10 WRITE(6,301) NEL,ME,NEQ,NEQ1
      RETURN 1
C ***********************************************************************
```

```
      C     FORMATS
      C ******************************************************************
        301 FORMAT(/20X,'THERE ARE NO FULLY SUMMED EQUATIONS IN THE MAIN ',
           1                'MEMORY .',/20X,'FURTHER ASSEMBLY CAN NOT PROCEED.',
           2         /20X,'(LAST ELEMENT ASSEMBLED  = ',I6,
           3         /20X,' ORDER OF THE ASSEMBLAGE = ',I6,
           4         /20X,' LAST EQUATION NUMBER REFERRED   = ',I6,
           5         /20X,' LAST EQUATION NUMBER ELIMINATED = ',I6,' )',/)
        302 FORMAT(/20X,'LAST ELEMENT ASSEMBLED                   = ',I6,
           1         /20X,'ORDER OF ASSEMBLAGE                      = ',I6,
           2         /20X,'LAST EQUATION NUMBER REFERRED            = ',I6,
           3         /20X,'LAST EQUATION NUMBER ELIMINATED          = ',I6,
           4        //20X,'NUMBER OF ELEMENTS AT EVERY STORY        = ',I6,
           5         /20X,'NEAS                                     = ',I6,
           6         /20X,'NUMBER OF FULLY SUMMED EQUATIONS SO FAR = ',I6,
           7        //20X,'ELIMINATION BEGINS FROM ROW              = ',I6,
           8         /20X,'ELIMINATION ENDS   AT   ROW              = ',I6,/)
        303 FORMAT(/20X,'SYSTEM STIFFNESS MATRIX (BEFORE ELIMINATION)',/)
        304 FORMAT(1X,'ROW=',I3,(8(2X,G13.7)))
        305 FORMAT(/20X,'SYSTEM LOAD MATRIX',/)
        306 FORMAT(1X,125(1H-),/20X,'SYSTEM STIFFNESS MATRIX (AFTER ',
           1          'ELIMINATION)',/)
        307 FORMAT(1X,125(1H-),/20X,'SYSTEM STIFFNESS MATRIX (AFTER ',
           1          'SHIFTING)',/)
        308 FORMAT(/1X,120(1H-))
        309 FORMAT(/20X,'K1 = ',I6,/20X,'K2 = ',I6,/)
            END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
DET	6	DUMMY-ARG		REAL	
I	1626B			INTEGER	
IDIFR	1667B			INTEGER	
IEX	1642B			INTEGER	
IGR	1622B			INTEGER	
IPOW	7	DUMMY-ARG		INTEGER	
J	1630B			INTEGER	
JJ	1644B			INTEGER	
JJ1	1647B			INTEGER	
K	1636B			INTEGER	
KOPT1	0B	/OPTION/		INTEGER	
KOPT2	1B	/OPTION/		INTEGER	
K1	1634B			INTEGER	
K2	1635B			INTEGER	
LUBR	6B	/SYSVAL/		INTEGER	
MAXBAN	6B	/MAXVAL/		INTEGER	
MAXNEM	5B	/MAXVAL/		INTEGER	
MAXNES	7B	/MAXVAL/		INTEGER	
ME	2	DUMMY-ARG		INTEGER	
MNLC	4B	/MAXVAL/		INTEGER	
NBAND	5B	/SYSVAL/		INTEGER	
NBAY	7B	/SYSVAL/		INTEGER	
NBEG	5	DUMMY-ARG		INTEGER	
NEAS	1624B			INTEGER	
NEL	1	DUMMY-ARG		INTEGER	
NELTS	1B	/SYSVAL/		INTEGER	
NEND	1625B			INTEGER	
NEQ	3	DUMMY-ARG		INTEGER	
NEQ1	4	DUMMY-ARG		INTEGER	
NETYPS	3B	/SYSVAL/		INTEGER	
NFSEQ	1623B			INTEGER	
NLC	2B	/SYSVAL/		INTEGER	
NNE	0B	/MAXVAL/		INTEGER	
NNODES	0B	/SYSVAL/		INTEGER	
NOESM	3B	/MAXVAL/		INTEGER	
NUE	2B	/MAXVAL/		INTEGER	
NUN	1B	/MAXVAL/		INTEGER	
NUS	4B	/SYSVAL/		INTEGER	
R	1645B			REAL	
RHS	1356B	/SYSTEM/		REAL	250
RHSS	3326B	/SYSTEM/		REAL	250
S	0B	/SYSTEM/		REAL	750
SS	1750B	/SYSTEM/		REAL	750

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ALOG10    REAL           1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF      -LABEL-ADDRESS-----PROPERTIES----DEF      -LABEL-ADDRESS-----PROPERTIES----DEF

    10   1313B                    167        120  INACTIVE   DO-TERM     122         301   1327B     FORMAT        172
    20  INACTIVE   DO-TERM         29        125  INACTIVE   DO-TERM     125         302   1365B     FORMAT        178
    30  INACTIVE   DO-TERM         32        130    737B                 119         303   1450B     FORMAT        187
    50  INACTIVE   DO-TERM         77        140   1310B                 164         304   1457B     FORMAT        188
    60  INACTIVE   DO-TERM         71        150  INACTIVE   DO-TERM     139         305   1463B     FORMAT        189
    70  INACTIVE   DO-TERM         66        160  INACTIVE   DO-TERM     136         306   1467B     FORMAT        190
    80    461B                     83        170  INACTIVE   DO-TERM     149         307   1500B     FORMAT        192
    90  INACTIVE   DO-TERM        117        180  INACTIVE   DO-TERM     146         308   1510B     FORMAT        194
   100  INACTIVE   DO-TERM        111        190  INACTIVE   DO-TERM     154         309   1513B     FORMAT        195
   110  INACTIVE   DO-TERM        106        200  INACTIVE   DO-TERM     157

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ELMNTE      65     7

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------

  TAPE13  BIN/DIR
  TAPE14  BIN/DIR
  TAPE6   FMT/SEQ
  TAPE8   BIN/DIR
  TAPE9   BIN/DIR

--STATISTICS--

  PROGRAM-UNIT LENGTH            1704B =  964
  SCM LABELLED COMMON LENGTH     3742B =  2018
  SCM STORAGE USED              63700B =  26560
  COMPILE TIME                  2.128 SECONDS
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=B,L=LPFR5.
```

```
      SUBROUTINE STIFFH(KSM)
C
      COMMON /MEMBER/ E,A,XI,XL,EM,EN,SH(6,21),MESM(6),FEML(6,5)
C
      BXL=1.0/XL
      S=A*E*BXL
      AI=4.0*E*XI*BXL
      BI=0.5*AI
      CI=(AI+BI)*BXL
      D=2.0*CI*BXL
      SH(KSM,1)=S
      SH(KSM,2)=0.0
      SH(KSM,3)=0.0
      SH(KSM,4)=-S
      SH(KSM,5)=0.0
      SH(KSM,6)=0.0
      SH(KSM,7)=D
      SH(KSM,8)=CI*EM
      SH(KSM,9)=0.0
      SH(KSM,10)=-D
      SH(KSM,11)=SH(KSM,8)
      SH(KSM,12)=AI
      SH(KSM,13)=0.0
      SH(KSM,14)=-SH(KSM,8)
      SH(KSM,15)=BI
      SH(KSM,16)=S
      SH(KSM,17)=0.0
      SH(KSM,18)=0.0
      SH(KSM,19)=D
      SH(KSM,20)=SH(KSM,14)
      SH(KSM,21)=AI
      RETURN
      END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	1B	/MEMBER/		REAL	
AI	115B			REAL	
BI	116B			REAL	
BXL	113B			REAL	
CI	117B			REAL	
D	120B			REAL	
E	0B	/MEMBER/		REAL	
EM	4B	/MEMBER/		REAL	
EN	5B	/MEMBER/		REAL	
FEML	212B	/MEMBER/		REAL	30
KSM	1	DUMMY-ARG		INTEGER	
MESM	204B	/MEMBER/		INTEGER	6
S	114B			REAL	
SH	6B	/MEMBER/		REAL	126
XI	2B	/MEMBER/		REAL	
XL	3B	/MEMBER/		REAL	

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  STIFFH       5B      1

--STATISTICS--

  PROGRAM-UNIT LENGTH              123B =   83
  SCM LABELLED COMMON LENGTH       250B =   168
  SCM STORAGE USED               61700B =   25536
  COMPILE TIME                   0.293 SECONDS
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=B,L=LPFR5.
```

```
      SUBROUTINE STIFFV(KSM)
C
      COMMON /MEMBER/ E,A,XI,XL,EM,EN,SM(6,21),MESM(6),FEML(6,5)
C
      BXL=1.0/XL
      S=A*E*BXL
      AI=4.0*E*XI*BXL
      BI=0.5*AI
      CI=(AI+BI)*BXL
      D=2.0*CI*BXL
      SM(KSM,1)=D
      SM(KSM,2)=0.0
      SM(KSM,3)=-CI*EN
      SM(KSM,4)=-D
      SM(KSM,5)=0.0
      SM(KSM,6)=SM(KSM,3)
      SM(KSM,7)=S
      SM(KSM,8)=0.0
      SM(KSM,9)=0.0
      SM(KSM,10)=-S
      SM(KSM,11)=0.0
      SM(KSM,12)=AI
      SM(KSM,13)=-SM(KSM,3)
      SM(KSM,14)=0.0
      SM(KSM,15)=BI
      SM(KSM,16)=D
      SM(KSM,17)=0.0
      SM(KSM,18)=SM(KSM,13)
      SM(KSM,19)=S
      SM(KSM,20)=0.0
      SM(KSM,21)=AI
      RETURN
      END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS-	-BLOCK-	----PROPERTIES-	-------TYPE-	--------SIZE
A	1B	/MEMBER/		REAL	
AI	115B			REAL	
BI	116B			REAL	
BXL	113B			REAL	
CI	117B			REAL	
D	120B			REAL	
E	0B	/MEMBER/		REAL	
EM	4B	/MEMBER/		REAL	
EN	5B	/MEMBER/		REAL	
FEML	212B	/MEMBER/		REAL	30
KSM	1	DUMMY-ARG		INTEGER	
MESM	204B	/MEMBER/		INTEGER	6
S	114B			REAL	
SM	6B	/MEMBER/		REAL	126
XI	2B	/MEMBER/		REAL	
XL	3B	/MEMBER/		REAL	

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  STIFFV      5B      1


--STATISTICS--

  PROGRAM-UNIT LENGTH            123B =  83
  SCM LABELLED COMMON LENGTH     250B =  168
  SCM STORAGE USED             61700B =  25536
  COMPILE TIME                 0.296 SECONDS
```

```
      SUBROUTINE STIFF(KSM)
C
      COMMON /MEMBER/ E,A,XI,XL,EM,EN,SM(6,21),MESM(6),FEML(6,5)
C
      BXL=1.0/XL
      S=A*E*BXL
      AI=4.0*E*XI*BXL
      BI=0.5*AI
      CI=(AI+BI)*BXL
      D=2.0*CI*BXL
      SM(KSM,1)=D*EN*EN+S*EM*EM
      SM(KSM,2)=(S-D)*EM*EN
      SM(KSM,3)=-CI*EN
      SM(KSM,4)=-SM(KSM,1)
      SM(KSM,5)=-SM(KSM,2)
      SM(KSM,6)=SM(KSM,3)
      SM(KSM,7)=D*EM*EM+S*EN*EN
      SM(KSM,8)=CI*EM
      SM(KSM,9)=SM(KSM,5)
      SM(KSM,10)=-SM(KSM,7)
      SM(KSM,11)=SM(KSM,8)
      SM(KSM,12)=AI
      SM(KSM,13)=-SM(KSM,3)
      SM(KSM,14)=-SM(KSM,8)
      SM(KSM,15)=BI
      SM(KSM,16)=SM(KSM,1)
      SM(KSM,17)=SM(KSM,2)
      SM(KSM,13)=SM(KSM,13)
      SM(KSM,19)=SM(KSM,7)
      SM(KSM,20)=SM(KSM,14)
      SM(KSM,21)=AI
      RETURN
      END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS-	-BLOCK-	----PROPERTIES----	----TYPE----	----SIZE
A	1B	/MEMBER/		REAL	
AI	123B			REAL	
BI	124B			REAL	
BXL	121B			REAL	
CI	125B			REAL	
D	126B			REAL	
E	0B	/MEMBER/		REAL	
EM	4B	/MEMBER/		REAL	
EN	5B	/MEMBER/		REAL	
FEML	212B	/MEMBER/		REAL	30
KSM	1	DUMMY-ARG		INTEGER	
MESM	204B	/MEMBER/		INTEGER	6
S	122B			REAL	
SM	6B	/MEMBER/		REAL	126
XI	2B	/MEMBER/		REAL	
XL	3B	/MEMBER/		REAL	

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  STIFF        5B      1


--STATISTICS--

  PROGRAM-UNIT LENGTH          131B =   89
  SCM LABELLED COMMON LENGTH    250B =   168
  SCM STORAGE USED            61700B =  25536
  COMPILE TIME                0.347 SECONDS
```

```
SUBROUTINE TRANS        74/176   OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+577         85/09/19. 17.33.06          PAGE      1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=B,L=LPFR5.

                   SUBROUTINE TRANS(EM,EN,T)
             C
                   DIMENSION T(6,6)
             C
                   T(1,1)=EM
                   T(1,2)=EN
                   T(2,1)=-EN
                   T(2,2)=EM
                   T(3,3)=1.0
                   T(4,4)=EM
                   T(4,5)=EN
                   T(5,4)=-EN
                   T(5,5)=EM
                   T(6,6)=1.0
                   RETURN
                   END

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  EM          1   DUMMY-ARG                  REAL
  EN          2   DUMMY-ARG                  REAL
  T           3   DUMMY-ARG                  REAL            36

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  TRANS       5B      3

--STATISTICS--

  PROGRAM-UNIT LENGTH           45B =  37
  SCM STORAGE USED           61700B =  25536
  COMPILE TIME               0.116 SECONDS
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=3,L=LPFR5.

```
      SUBROUTINE LDTYPS(NEL,LD,LDTP,Q,X,Y,*)
C
      COMMON /MEMBER/ E,A,XI,XL,EM,EN,SM(6,21),MESM(6),F(6,5)
C
      GO TO (10,20,30,40),LDTP
      WRITE(6,301)LDTP,NEL,LD
      RETURN 1
C
C     UNIFORMLY DISTRIBUTED LOAD
C
C     X= LEFT MARGIN
C     Y= RIGHT MARGIN
C
   10 Z=XL-(X+Y)
      IF(Z.LE.0.0) THEN
        WRITE(6,303) NEL,LD,LDTP,XL,X,Y,Z
        RETURN 1
      ENDIF
      F(1,LD)=0.0
      F(4,LD)=0.0
      F(3,LD)=Q/(12.0*XL*XL)*((XL-X)**3*(XL+3.0*X)-Y**3*(4.0*XL-3.0*Y))
      F(6,LD)=-Q/(12.0*XL*XL)*((XL-Y)**3*(XL+3.0*Y)-X**3*(4.0*XL-3.0*X))
      F(5,LD)=(Q*Z*(X+0.5*Z)-F(3,LD)-F(6,LD))/XL
      F(2,LD)=Q*Z-F(5,LD)
      GO TO 50
C
C     VERTICAL POINT LOAD
C
   20 IF(X+Y.NE.XL) THEN
        WRITE(6,302) NEL,LD,LDTP,XL,X,Y
        RETURN 1
      ENDIF
      F(1,LD)=0.0
      F(4,LD)=0.0
      F(3,LD)=Q*X*Y*Y/(XL*XL)
      F(6,LD)=-Q*X*X*Y/(XL*XL)
      F(5,LD)=(Q*X-F(3,LD)-F(6,LD))/XL
      F(2,LD)=Q-F(5,LD)
      GO TO 50
C
C     UNIFORMLY DISTRIBUTED AXIAL LOAD
C
   30 Z=XL-X-Y
      IF(Z.LE.0.0) THEN
        WRITE(6,303) NEL,LD,LDTP,XL,X,Y,Z
        RETURN 1
      ENDIF
      F(2,LD)=0.0
      F(3,LD)=0.0
      F(5,LD)=0.0
      F(6,LD)=0.0
      F(1,LD)=-Q*Z/XL*(0.5*Z+Y)
      F(4,LD)=F(1,LD)-Q*Z
      GO TO 50
C
```

```
C      CONCENTRATED AXIAL FORCE '
C
    40 IF(X+Y.NE.XL) THEN
         WRITE(6,302) NEL,LD,LDTP,XL,X,Y
         RETURN 1
       ENDIF
       F(2,LD)=0.0
       F(3,LD)=0.0
       F(5,LD)=0.0
       F(6,LD)=0.0
C
       IF(X/XL.LE.XL/100.0) THEN
         F(4,LD)=0.0
         F(1,LD)=-Q
         GO TO 50
       ENDIF
C
       F(4,LD)=-Q/(1.0+Y/X)
       F(1,LD)=-(Q+F(4,LD))
C
    50 RETURN
C ********************************************************************
C      FORMATS
C ********************************************************************
   301 FORMAT(/20X,'LOADING TYPE = ',I6,' DOES NOT EXIST .',
     1         /20X,'(IN ELEMENT ',I6,' , IN LOADING CASE ',I6,/)
   302 FORMAT(/20X,'LEFT AND RIGHT MARGINS DO NOT ADD UP TO LENGTH.',
     1         /20X,'(ELEMENT NO.  = ',I6,
     2         /20X,' LOADING CASE = ',I6,
     3         /20X,' LOADING TYPE = ',I6,
     4         /20X,' LENGTH       = ',G13.7,
     5         /20X,' LEFT MARGIN  = ',G13.7,
     6         /20X,' RIGHT MARGIN = ',G13.7,' )',/)
   303 FORMAT(/20X,'NEGATIVE VALUE FOR THE LOADED REGION IS NOT VALID.',
     1         /20X,'(ELEMENT NO.   = ',I6,
     2         /20X,' LOADING CASE  = ',I6,
     3         /20X,' LOADING TYPE  = ',I6,
     4         /20X,' LENGTH        = ',G13.7,
     5         /20X,' LEFT MARGIN   = ',G13.7,
     6         /20X,' RIGHT MARGIN  = ',G13.7,
     7         /20X,' LOADED LENGTH = ',G13.7,' )',/)
       END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	1B	/MEMBER/		REAL		NEL	1	DUMMY-ARG		INTEGER	
E	0B	/MEMBER/		REAL		Q	4	DUMMY-ARG		REAL	
EM	4B	/MEMBER/		REAL		SM	6B	/MEMBER/		REAL	126
EN	5B	/MEMBER/		REAL		X	5	DUMMY-ARG		REAL	
F	212B	/MEMBER/		REAL	30	XI	2B	/MEMBER/		REAL	
LD	2	DUMMY-ARG		INTEGER		XL	3B	/MEMBER/		REAL	
LDTP	3	DUMMY-ARG		INTEGER		Y	6	DUMMY-ARG		REAL	
MESM	204B	/MEMBER/		INTEGER	6	Z	434B			REAL	

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS------PROPERTIES----DEF         -LABEL-ADDRESS------PROPERTIES----DEF

    10      25B                     14              50     264B                      76
    20     114B                     29             301     276B      FORMAT          80
    30     156B                     43             302     312B      FORMAT          82
    40     215B                     58             303     344B      FORMAT          89

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LDTYPS       6B     6

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6    FMT/SEQ

--STATISTICS--

   PROGRAM-UNIT LENGTH             437B =   287
   SCM LABELLED COMMON LENGTH      250B =   168
   SCM STORAGE USED              61700B =   25536
   COMPILE TIME                  0.829 SECONDS
```

```
      SUBROUTINE BACSUB
C
      COMMON /MAXVAL/ NNE,NUN,NUE,NOESM,MNLC,MAXNEM,MAXBAN,MAXNES
      COMMON /SYSVAL/ NNODES,NELTS,NLC,NETYPS,NUS,NBAND,LUBR,NBAY
      COMMON /SYSTEM/ S(50,15),RHS(50,5),SS(50,15),RHSS(50,5)
      COMMON /OPTION/ KOPT1,KOPT2
C
      NEQ1=NUS-MAXNEM
      NEND=NBAND
      KREAD1=1
      KREAD2=MAXNEM
   10 IF(NEQ1.LT.0) KREAD1=-NEQ1+1
C
      IF(KOPT2.NE.0) WRITE(6,301) NEQ1,NEND,KREAD1,KREAD2
C
      DO 30 I=KREAD1,KREAD2
        READ(13,REC=NEQ1+I) (S(I,J),J=1,NBAND)
   30   READ(14,REC=NEQ1+I) (RHS(I,J),J=1,NLC)
C
      IF(KOPT2.NE.0) THEN
        WRITE(6,302)
        DO 35 I=1,MAXNEM
          WRITE(6,303) I,(S(I,J),J=1,MAXBAN)
   35     WRITE(6,303) I,(RHS(I,J),J=1,MNLC)
      ENDIF
C
      K1=NBAND
      K2=MAXNEM
      IF(NEQ1.LT.1) THEN
        K1=-NEQ1+NBAND
        NEND=-NEQ1+2
      ENDIF
C
      IF(KOPT2.NE.0) WRITE(6,304) K2,K1
C
C     BACKSUBSTITUTION
C
      DO 100 K=K2,K1,-1
        DO 110 J=1,NLC
          RHS(K,J)=RHS(K,J)/S(K,1)
          DO 110 I=K-NBAND+1,K-1
            JJ=K-I+1
  110       RHS(I,J)=RHS(I,J)-RHS(K,J)*S(I,JJ)
  100   WRITE(14,REC=NEQ1+K) (RHS(K,J),J=1,NLC)
C
      IF(NEQ1.LT.1) THEN
        K2=K1-1
        K1=NEND
      ELSE
        GO TO 180
      ENDIF
C
      IF(KOPT2.NE.0) WRITE(6,304) K2,K1
C
```

```
      C
            DO 130 K=K2,K1,-1
              DO 140 J=1,NLC
              RHS(K,J)=RHS(K,J)/S(K,1)
              DO 140 I=-NEQ1+1,K-1
                JJ=K-I+1
        140     RHS(I,J)=RHS(I,J)-RHS(K,J)*S(I,JJ)
        130   WRITE(14,REC=NEQ1+K) (RHS(K,J),J=1,NLC)
      C
            DO 145 J=1,NLC
        145   RHS(NEND-1,J)=RHS(NEND-1,J)/S(NEND-1,1)
            WRITE(14,REC=1) (RHS(NEND-1,J),J=1,NLC)
      C
            IF(KOPT2.NE.0) THEN
              WRITE(6,305)
              DO 190 I=1,MAXNEM
                WRITE(6,303) I,(S(I,J),J=1,MAXBAN)
        190     WRITE(6,303) I,(RHS(I,J),J=1,MNLC)
            ENDIF
      C
            GO TO 200
      C
      C     SHIFT THE EQUATIONS DOWN
      C
        180 IDIF=MAXNEM-(NBAND-1)
            DO 150 I=NBAND-1,1,-1
              DO 160 J=1,NBAND
        160     S(IDIF+I,J)=S(I,J)
              DO 150 J=1,NLC
        150     RHS(IDIF+I,J)=RHS(I,J)
      C
      C     SET THE POINTERS
      C
            NEQ1=NEQ1-MAXNEM+(NBAND-1)
            KREAD2=MAXNEM-(NBAND-1)
      C
            IF(KOPT2.NE.0) THEN
              WRITE(6,305)
              DO 210 I=1,MAXNEM
                WRITE(6,303) I,(S(I,J),J=1,MAXBAN)
        210     WRITE(6,303) I,(RHS(I,J),J=1,MNLC)
            ENDIF
      C
            GO TO 10
      C
        200 RETURN
      C ********************************************************************
      C     FORMATS
      C ********************************************************************
        301 FORMAT(/20X,'NEQ1   = ',I6,
           1        /20X,'NEND   = ',I6,
           2        /20X,'KREAD1 = ',I6
           3        /20X,'KREAD2 = ',I6,/)
        302 FORMAT(/20X,'STIFFNESS AND LOAD MATRIX (BEFORE BACSUB)',/)
        303 FORMAT(1X,'ROW=',I3,(8(2X,G13.7)))
        304 FORMAT(/20X,'BACSUB STARTS FROM ROW = ',I6
           1        /20X,'BACSUB ENDS AT ROW     = ',I6,/)
```

```
   113             305 FORMAT(/20X,'STIFFNESS AND LOAD MATRIX (AFTER BACSUB '
   114                1             ,'AND SHIFTING)',/)
   115                 END

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  I         1105B                             INTEGER                NBAY        7B  /SYSVAL/                   INTEGER
  IDIF      1136B                             INTEGER                NELTS       1B  /SYSVAL/                   INTEGER
  J         1107B                             INTEGER                NEND     1102B                             INTEGER
  JJ        1123B                             INTEGER                NEQ1     1101B                             INTEGER
  K         1117B                             INTEGER                NETYPS      3B  /SYSVAL/                   INTEGER
  KOPT1        0B  /OPTION/                   INTEGER                NLC         2B  /SYSVAL/                   INTEGER
  KOPT2        1B  /OPTION/                   INTEGER                NNE         0B  /MAXVAL/                   INTEGER
  KREAD1    1103B                             INTEGER                NNODES      0B  /SYSVAL/                   INTEGER
  KREAD2    1104B                             INTEGER                NOESM       3B  /MAXVAL/                   INTEGER
  K1        1115B                             INTEGER                NUE         2B  /MAXVAL/                   INTEGER
  K2        1116B                             INTEGER                NUN         1B  /MAXVAL/                   INTEGER
  LUBR         6B  /SYSVAL/                   INTEGER                NUS         4B  /SYSVAL/                   INTEGER
  MAXBAN       6B  /MAXVAL/                   INTEGER                RHS      1356B  /SYSTEM/                    REAL          250
  MAXNEM       5B  /MAXVAL/                   INTEGER                RHSS     3326B  /SYSTEM/                    REAL          250
  MAXNES       7B  /MAXVAL/                   INTEGER                S           0B  /SYSTEM/                    REAL          750
  MNLC         4B  /MAXVAL/                   INTEGER                SS       1750B  /SYSTEM/                    REAL          750
  NBAND        5B  /SYSVAL/                   INTEGER

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

   10     16B                      12         145  INACTIVE    DO-TERM     66          210  INACTIVE    DO-TERM     96
   30  INACTIVE    DO-TERM         18         150  INACTIVE    DO-TERM     85          301     760B     FORMAT     105
   35  INACTIVE    DO-TERM         24         160  INACTIVE    DO-TERM     83          302     772B     FORMAT     109
  100  INACTIVE    DO-TERM         44         180     603B                 80          303    1001B     FORMAT     110
  110  INACTIVE    DO-TERM         43         190  INACTIVE    DO-TERM     73          304    1005B     FORMAT     111
  130  INACTIVE    DO-TERM         63         200     751B                101          305    1016B     FORMAT     113
  140  INACTIVE    DO-TERM         62

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  BACSUB      4B     0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE13   BIN/DIR
  TAPE14   BIN/DIR
  TAPE6    FMT/SEQ
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH            1145B =  613
  SCM LABELLED COMMON LENGTH     3742B =  2018
  SCM STORAGE USED              63700B =  26560
  COMPILE TIME                  1.422 SECONDS
```

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=B,L=LPFR5.
```

```
      SUBROUTINE PRTDISP
C
      COMMON /MAXVAL/ NNE,NUN,NUE,NOESM,MNLC,MAXNEM,MAXBAN,MAXNES
      COMMON /SYSVAL/ NNODES,NELTS,NLC,NETYPS,NUS,NBAND,LUBR,NBAY
      COMMON /SYSTEM/ DISP(3,5),DUMMY(1985)
      COMMON /JOINT / NDC(6)
C
      WRITE(6,301)
      DO 10 NNUM=1,NNODES
        READ(10,REC=NNUM) JFLG,(NDC(K),K=1,NUN)
        DO 20 K=1,NUN
          IF(NDC(K).EQ.0) THEN
            DO 30 J=1,NLC
 30           DISP(K,J)=0.0
            GO TO 20
          ELSE
            READ(14,REC=NDC(K)) (DISP(K,J),J=1,NLC)
          ENDIF
 20     CONTINUE
        WRITE(6,302) NNUM,(DISP(K,1),K=1,NUN),(NDC(J),J=1,NUN)
        IF(NLC.EQ.1) GO TO 10
        DO 40 J=2,NLC
 40     WRITE(6,303) J,(DISP(K,J),K=1,NUN)
 10   CONTINUE
C ********************************************************************
C      FORMATS
C ********************************************************************
 301  FORMAT(1H1,//20X,'NODAL DISPLACEMENTS',/19X,21(1H-),
     1        / 5X,' NODE ',2X,' LOAD ',2X,'   DISP-X     ',2X,
     2        '   DISP-Y     ',2X,' ROTATION-Z ',/5X,
     3        2(6(1H-),2X),3(13(1H-),2X))
 302  FORMAT(5X,I5,4X,'    1',3X,3(G13.7,2X),3(I6,2X))
 303  FORMAT(14X,I4,3X,3(G13.7,2X))
      END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
DISP	0B	/SYSTEM/		REAL	15
DUMMY	17B	/SYSTEM/		REAL	1985
J	265B			INTEGER	
JFLG	262B		*S*	INTEGER	
K	263B			INTEGER	
LUBR	6B	/SYSVAL/		INTEGER	
MAXBAN	6B	/MAXVAL/		INTEGER	
MAXNEM	5B	/MAXVAL/		INTEGER	
MAXNES	7B	/MAXVAL/		INTEGER	
MNLC	4B	/MAXVAL/		INTEGER	
NBAND	5B	/SYSVAL/		INTEGER	
NBAY	7B	/SYSVAL/		INTEGER	
NDC	0B	/JOINT/		INTEGER	6
NELTS	1B	/SYSVAL/		INTEGER	
NETYPS	3B	/SYSVAL/		INTEGER	
NLC	2B	/SYSVAL/		INTEGER	
NNE	0B	/MAXVAL/		INTEGER	
NNODES	0B	/SYSVAL/		INTEGER	
NNUM	260B			INTEGER	
NOESM	3B	/MAXVAL/		INTEGER	
NUE	2B	/MAXVAL/		INTEGER	
NUN	1B	/MAXVAL/		INTEGER	
NUS	4B	/SYSVAL/		INTEGER	

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF

    10    154B      DO-TERM         24           301     166B      FORMAT          28
    20    100B      DO-TERM         19           302     210B      FORMAT          32
    30  INACTIVE    DO-TERM         14           303     216B      FORMAT          33
    40  INACTIVE    DO-TERM         23


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PRTDISP     4B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------

  TAPE10  BIN/DIR
  TAPE14  BIN/DIR
  TAPE6   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH           271B =  185
  SCM LABELLED COMMON LENGTH  37463B =  2022
  SCM STORAGE USED           61700B =  25536
  COMPILE TIME               0.467 SECONDS
```

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=B,L=LPFR5.

```
      SUBROUTINE ENDFOR
C
      COMMON /MAXVAL/ NNE,NUN,NUE,NOESM,MNLC,MAXNEM,MAXBAN,MAXNES
      COMMON /SYSVAL/ NNODES,NELTS,NLC,NETYPS,NUS,NBAND,LUBR,NBAY
      COMMON /SYSTEM/ T(6,6),DG(6,5),EFG(6,5),DUMMY(1904)
      COMMON /MEMBER/ E,A,XI,XL,EM,EN,SM(6,21),MESM(6),FEML(6,5)
      COMMON /JOINT / NDC(6)
      COMMON /OPTION/ KOPT1,KOPT2
C
      LOCN(I,J)=NUE*I-I*(I-1)/2-(NUE-J)
C
      NESM=0
      DO 1 I=1,NUE
       DO 1 J=1,NUE
    1    T(I,J)=0.0
      DO 2 I=1,NUE
        DO 2 J=1,MNLC
          DG(I,J)=0.0
    2     EFG(I,J)=0.0
C
      WRITE(6,301) (I,I=1,NUE)
C
      DO 10 ME=1,NELTS
C
        READ(12,REC=ME) MTYP,NDC,((FEML(I,J),J=1,NLC),I=1,NUE)
        IF(KOPT2.NE.0) THEN
          PRINT*,' ELEMENT = ',ME,' MTYP = ',MTYP,' NDC = ',NDC
        ENDIF
C
C       OBTAIN ELEMENT DISPLACEMENTS FROM FILE
C
        DO 20 I=1,NUE
          IF(NDC(I).EQ.0) THEN
            DO 30 J=1,NLC
   30         DG(I,J)=0.0
          ELSE
            READ(14,REC=NDC(I))(DG(I,J),J=1,NLC)
          ENDIF
   20   CONTINUE
        IF(KOPT2.NE.0) THEN
          PRINT*,' ELEMENT DISPLACEMENTS'
          DO 21 I=1,NUE
   21       PRINT*,' ',(DG(I,J),J=1,NLC)
        ENDIF
C
C       OBTAIN ELEMENT STIFFNESS MATRIX EITHER FROM MAIN MEMORY
C       OR GENERATE USING SUBROUTINES, GENERATE THE TRANSFORMATION MATRIX.
C
        DO 50 I=1,MAXNES
          IF(MTYP.EQ.MESM(I)) THEN
            KSM=I
            READ(11,REC=MTYP) XL,EM,EN
            CALL TRANS(EM,EN,T)
            GO TO 60
          ENDIF
```

```
   50   CONTINUE
        NESM=NESM+1
        IF(NESM.GT.MAXNES) NESM=1
        KSM=NESM
        MESM(KSM)=MTYP
        READ(11,REC=MTYP) XL,EM,EN,(SM(KSM,I),I=1,NOESM)
        CALL TRANS(EM,EN,T)
C
C       COMPUTE END FORCES IN GLOBAL COORDINATES
C
   60   DO 80 I=1,NUE
          DO 80 J=1,NLC
            SUM=0.0
            DO 90 K=1,NUE
              IE=I
              JE=K
              IF(JE.LT.IE) THEN
                IE=K
                JE=I
              ENDIF
              IJE=LOCN(IE,JE)
   90         SUM=SUM+SM(KSM,IJE)*DG(K,J)
   80     EFG(I,J)=SUM
C
        IF(KOPT2.NE.0) THEN
          PRINT*,' ELEMENT STIFFNESS MATRIX'
          PRINT*,' ',(SM(KSM,I),I=1,NOESM)
          PRINT*,' ELEMENT TRANSFORMATION MATRIX'
          DO 81 I=1,NUE
   81       PRINT*,' ',(T(I,J),J=1,NUE)
        ENDIF
C
C       TRANSFORM THE ENDFORCES TO LOCAL COORDINATES AND ADD UP WITH
C       THE FIXED-END-FORCES
C
        DO 100 I=1,NUE
          DO 100 J=1,NLC
            SUM=0.0
            DO 110 K=1,NUE
  110         SUM=SUM+T(I,K)*EFG(K,J)
  100     FEML(I,J)=FEML(I,J)+SUM
        WRITE(6,302) ME,(FEML(J,1),J=1,NUE)
        IF(NLC.LT.2) GO TO 10
        DO 130 I=2,NLC
  130     WRITE(6,303) I,(FEML(J,I),J=1,NUE)
   10 CONTINUE
      RETURN
C *********************************************************************
C     FORMATS
C *********************************************************************
  301 FORMAT(1H1,///5X,'ELEM. ',2X,'LOADING',2X,6(6X,I1,8X),/5X,
     1        6(1H-),2X,7(1H-),2X,6(13(1H-),2X),/)
  302 FORMAT(6X,I4,5X,' 1',3X,6(2X,G13.7))
  303 FORMAT(14X,I3,3X,6(2X,G13.7))
      END
```

--VARIABLE MAP--(LO=A)

-NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	-NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	1B	/MEMBER/		REAL		MESM	204B	/MEMBER/		INTEGER	6
DG	44B	/SYSTEM/		REAL	30	MNLC	4B	/MAXVAL/		INTEGER	
DUMMY	140B	/SYSTEM/		REAL	1904	MTYP	1060B			INTEGER	
E	0B	/MEMBER/		REAL		NBAND	5B	/SYSVAL/		INTEGER	
EFG	102B	/SYSTEM/		REAL	30	NBAY	7B	/SYSVAL/		INTEGER	
EM	4B	/MEMBER/		REAL		NDC	0B	/JOINT/		INTEGER	6
EN	5B	/MEMBER/		REAL		NELTS	1B	/SYSVAL/		INTEGER	
FEML	212B	/MEMBER/		REAL	30	NESM	1050B			INTEGER	
I	1046B			INTEGER		NETYPS	3B	/SYSVAL/		INTEGER	
IE	1100B			INTEGER		NLC	2B	/SYSVAL/		INTEGER	
IJE	1102B			INTEGER		NNE	0B	/MAXVAL/		INTEGER	
J	1047B			INTEGER		NNODES	0B	/SYSVAL/		INTEGER	
JE	1101B			INTEGER		NOESM	3B	/MAXVAL/		INTEGER	
K	1076B			INTEGER		NUE	2B	/MAXVAL/		INTEGER	
KOPT1	0B	/OPTION/		INTEGER		NUN	1B	/MAXVAL/		INTEGER	
KOPT2	1B	/OPTION/		INTEGER		NUS	4B	/SYSVAL/		INTEGER	
KSM	1071B			INTEGER		SM	6B	/MEMBER/		REAL	126
LUBR	6B	/SYSVAL/		INTEGER		SUM	1075B			REAL	
MAXBAN	6B	/MAXVAL/		INTEGER		T	0B	/SYSTEM/		REAL	36
MAXNEM	5B	/MAXVAL/		INTEGER		XI	2B	/MEMBER/		REAL	
MAXNES	7B	/MAXVAL/		INTEGER		XL	3B	/MEMBER/		REAL	
ME	1056B			INTEGER							

--PROCEDURES--(LO=A)

-NAME	TYPE	ARGS	CLASS
LOCN	INTEGER	2	STAT FUNC
TRANS		3	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL	ADDRESS	PROPERTIES	DEF	-LABEL	ADDRESS	PROPERTIES	DEF	-LABEL	ADDRESS	PROPERTIES	DEF
1	INACTIVE	DO-TERM	15	50	INACTIVE	DO-TERM	56	110	INACTIVE	DO-TERM	95
2	INACTIVE	DO-TERM	19	60	363B		66	130	INACTIVE	DO-TERM	100
10	660B	DO-TERM	101	80	INACTIVE	DO-TERM	78	301	712B	FORMAT	106
20	INACTIVE	DO-TERM	39	81	INACTIVE	DO-TERM	85	302	724B	FORMAT	108
21	INACTIVE	DO-TERM	43	90	INACTIVE	DO-TERM	77	303	730B	FORMAT	109
30	INACTIVE	DO-TERM	35	100	INACTIVE	DO-TERM	96				

--ENTRY POINTS--(LO=A)

-NAME	ADDRESS	ARGS
ENDFOR	4B	0

```
--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE11  BIN/DIR
  TAPE12  BIN/DIR
  TAPE14  BIN/DIR
  TAPE6   FMT/SEQ

--STATISTICS--

  PROGRAM-UNIT LENGTH           1112B =  586
  SCM LABELLED COMMON LENGTH    4220B =  2192
  SCM STORAGE USED             63700B =  26560
  COMPILE TIME                 1.356 SECONDS
```

```
SUBROUTINE CDTR        74/176   OPT=0,ROUND= A/ S/ M/-D,-DS      FTN 5.1+577          85/09/19. 17.33.06        PAGE     1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=PFR5,B=B,L=LPFR5.

      SUBROUTINE CDTR(NUN,NNODES)
C
      DIMENSION Y(400),YN(400),NODE(400),NDC(3)
C
      WRITE(6,3001)
C
      READ(5,*) NND
      READ(5,*) (NODE(I),Y(I),I=1,NND)
C
      DO 10 I=1,NND
   10   YN(I)=Y(I)/Y(1)
      DO 20 I=1,NND
        READ(10,REC=NODE(I)) JFLG,(NDC(I),J=1,NUN)
   20   NODE(I)=NDC(1)
      READ(14,REC=NODE(1)) DMAX
      SUM=0.0
      SUMDN=0.0
      DO 30 I=1,NND
        READ(14,REC=NODE(I)) DISP
        DN=DISP/DMAX
        SUM=SUM+ABS((DN-YN(I))*DN)
        SUMDN=SUMDN+ABS(DN)
   30   WRITE(6,3002) I,Y(I),DISP,DN,YN(I)
      WPERR=SUM/SUMDN*100.0
      WRITE(6,3003) WPERR
      RETURN
C ********************************************************************
C     FORMATS
C ********************************************************************
 3001 FORMAT(//20X,'COMPARISON OF TRIANGULAR DISPLACEMENT DISTRIBUTION',
     1 /20X,' AND REAL DISPALACEMENT DISTRIBUTION (HORIZONTALS ONLY)'
     2 //20X,' DOF ',2X,'  HEIGHT OF THE NODE  ',2X,
     3       '     REAL DISPLACEMENT  ',2X,'      DISPI/DMAX        ',2X,
     4       'NORMALIZED TRIANG.DISP',/20X,5(1H-),4(2X,22(1H-)))
 3002 FORMAT(20X,I5,4(2X,G22.16))
 3003 FORMAT(//25X,'WEIGHTED PERCENT ERROR = ',G13.7)
      END

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 DISP     2624B                               REAL                    NNODES      2   DUMMY-ARG UNUSED/*S*       INTEGER
 DMAX     2620B                               REAL                    NODE    1765B                              INTEGER       400
 DN       2625B                               REAL                    NUN         1   DUMMY-ARG                  INTEGER
 I        2611B                               INTEGER                 SUM     2621B                              REAL
 J        2616B                               INTEGER                 SUMDN   2622B                              REAL
 JFLG     2615B              *S*              INTEGER                 WPERR   2626B                              REAL
 NDC      2605B                               INTEGER          3      Y        325B                              REAL          400
 NND      2610B                               INTEGER                 YN      1145B                              REAL          400
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE---------ARGS------CLASS-----

  ABS        GENERIC        1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        11          3001    202B     FORMAT          30
    20  INACTIVE   DO-TERM        14          3002    237B     FORMAT          35
    30  INACTIVE   DO-TERM        23          3003    243B     FORMAT          36


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  CDTR        5B     2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------

  TAPE10  BIN/DIR
  TAPE14  BIN/DIR
  TAPE5   FMT/SEQ
  TAPE6   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH           2631B =  1433
  SCM STORAGE USED             61700B =  25536
  COMPILE TIME                 0.455 SECONDS
 17.52.59.UCLP, AA, P04           1.927KLNS.
```

APPENDIX B - COMPUTER PROGRAM SPEC

DATA INPUT TO SPEC

I. HEADING CARD (20A4)

Input List : HEAD

Explanation :

HEAD : character array containing the title for the problem

II. CONTROL CARD

Input List : NSPEC, H, TMAX, IPRT1

Explanation :

NSPEC : number of equations to be integrated

H : time step, Δt

TMAX : finish time

IPRT1 : option for printing the computed accelerations, velocities and displacemnts at every integration point;

= 0 ; prints

= 0 ; does not print

Note : integration will be carried out for $t \leq TMAX$.

III. PERIOD CARD(S)

Input List : (PERIOD (I), I=1, NSPEC)

Explanation :

PERIOD : array containing the periods of the single degree-of-freedom equations to be integrated

IV. DAMPING RATIO CARD(S)

Input List : (D(I), I=1, NSPEC)

Explanation :

D : array containing the damping ratios of the single degree-of-freedom equations to be integrated.

Note : D(1) corresponds to PERIOD(1), etc.

V. ACCELERATION FUNCTION DEFINITION CARDS

V. A CONTROL CARD

Input List : NFRC, SCATIM, SCACC, IPRT2

Explanation :

NFRC : number of acceleration function definition points

SCATIM : time scale

SCACC : acceleration scale

IPRT2 : option for printing the input acceleration function definition values ;

= 0 ; does not print

$\neq$ 0 ; prints

Note : The input time and acceleration values are multiplied by SCATIM and SCACC, respectively.

V.B TIME v.s. ACCELERATION CARD(S)

Input List : (T(I), F(I), I=1, NFRC)

Explanation :

T : array containing the acceleration values

Note : F(1) corresponds to T(1), etc. Pairs must be input in the order of ascending time value.

```
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=SPEC,B=B,L=LSPEC.

      PROGRAM SPEC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      PARAMETER (N =10)
      PARAMETER (NF=1000)
C
      DIMENSION HEAD(20)
      DIMENSION F(NF),T(NF)
      DIMENSION PERIOD(N),W(N),D(N)
      DIMENSION A(N),B(N),E(N),D0(N),V0(N),A0(N),D1(N),V1(N),A1(N)
      DIMENSION SPECA(N),KSPEC(N),TSPEC(N)
C
      READ(5,'(20A4)') HEAD
      WRITE(6,3000) HEAD
      READ(5,*) NSPEC,H,TMAX,IPRT1
      READ(5,*) (PERIOD(I),I=1,NSPEC)
      READ(5,*) (D(I),I=1,NSPEC)
C
      READ(5,*) NFRC,SCATIM,SCACC,IPRT2
      READ(5,*) (T(I),F(I),I=1,NFRC)
C
      WRITE(6,3002) NSPEC,H,TMAX,IPRT1
C
      WRITE(6,3010)
      DO 5 I=1,NSPEC
        W(I)=2.0*3.141592654/PERIOD(I)
        A(I)=2.0*D(I)*W(I)
        B(I)=W(I)*W(I)
        E(I)=1.0+A(I)*H/2.0+B(I)*H*H/6.0
        WRITE(6,3015) I,PERIOD(I),W(I),D(I)
    5 CONTINUE
C
      DO 20 I=1,NFRC
        T(I)=T(I)*SCATIM
   20   F(I)=F(I)*SCACC
      IF(IPRT2.NE.0) WRITE(6,3001) NFRC,SCATIM,SCACC,(I,T(I),F(I),
     1                                   I=1,NFRC)
C
      WRITE(6,3035)
      TT=0.0
      K=0
      K1=1
      K2=2
C
      DO 15 I=1,NSPEC
        SPECA(I)=0.0
        KSPEC(I)=0
   15   TSPEC(I)=0.0
C
      IF(TT.GT.TMAX) GO TO 200
   30 IF(K2.GT.NFRC) THEN
        ACC=0.0
        GO TO 40
      ENDIF
      IF(TT.LE.T(K2)) THEN
        ACC=F(K1)+(F(K2)-F(K1))/(T(K2)-T(K1))*(TT-T(K1))
```

```
      ELSE
        K1=K2
        K2=K2+1
        GO TO 30
      ENDIF
C
   40 DO 45 I=1,NSPEC
   45   A0(I)=-ACC-A(I)*V0(I)-B(I)*D0(I)
C
  100 TT=TT+H
      K=K+1
      IF(TT.GT.TMAX) GO TO 200
   50 IF(K2.GT.NFRC) THEN
        ACC=0.0
        GO TO 70
      ENDIF
      IF(TT.LE.T(K2)) THEN
        ACC=F(K1)+(F(K2)-F(K1))/(T(K2)-T(K1))*(TT-T(K1))
      ELSE
        K1=K2
        K2=K2+1
        GO TO 50
      ENDIF
C
   70 DO 75 I=1,NSPEC
        AA=V0(I)+A0(I)*H/2.0
        BB=D0(I)+V0(I)*H+A0(I)*H*H/3.0
        A1(I)=(-ACC-A(I)*AA-B(I)*BB)/E(I)
        V1(I)=AA+A1(I)*H/2.0
   75   D1(I)=BB+A1(I)*H*H/6.0
C
      DO 90 I=1,NSPEC
        A0(I)=A1(I)
        V0(I)=V1(I)
        D0(I)=D1(I)
   90 CONTINUE
C
      IF(IPRT1.NE.0) GO TO 220
      WRITE(6,3020) K,TT,ACC
      WRITE(6,3025)
      WRITE(6,3004) (A1(I),I=1,NSPEC)
      WRITE(6,3004) (V1(I),I=1,NSPEC)
      WRITE(6,3004) (D1(I),I=1,NSPEC)
C
  220 DO 110 I=1,NSPEC
        IF(ABS(A1(I)).GT.ABS(SPECA(I))) THEN
          SPECA(I)=A1(I)
          KSPEC(I)=K
          TSPEC(I)=TT
        ENDIF
  110 CONTINUE
C
      GO TO 100
C
  200 WRITE(6,3040)
      DO 210 I=1,NSPEC
  210   WRITE(6,3045) I,PERIOD(I),KSPEC(I),TSPEC(I),SPECA(I)
```

```
      C
            WRITE(6,3006)
      C
            STOP
      C
      C ****************************************************************
      C     FORMATS
      C ****************************************************************
      C
       3000 FORMAT(1H1,///25X,20A4,/)
       3001 FORMAT(/25X,'NUMBER OF ACCELERATION ORDINATES = ',I5,
           1       /25X,'TIME          SCALE              = ',G13.7,
           2       /25X,'ACCELERATION SCALE               = ',G13.7,
           3      //25X,'  NO  ',2X,'    TIME     ',2X,'ACCELERATION ',
           4       /25X,6(1H-),2(2X,13(1H-)),/(25X,I5,3X,2(G13.7,2X)))
       3002 FORMAT(//30X,'NSPEC        = ',I5,
           1        /30X,'TIME STEP   = ',G13.7,
           2        /30X,'FINISH TIME = ',G13.7,
           3        /30X,'PRINT OPT.1 = ',I5,/)
       3004 FORMAT(1X,8(G15.9,1X))
       3006 FORMAT(1H1,6(/))
       3010 FORMAT(/30X,'SPEC.   INFORMATION',/30X,19(1H-),
           1       /30X,' NO.',2X,'      PERIOD      ',2X,' ANGULAR FREQ. ',
           2          2X,'CRIT.DAMP.RATIO',
           3       /30X,4(1H-),3(2X,15(1H-)))
       3015 FORMAT(30X,I3,3X,3(G15.9,2X))
       3025 FORMAT(10X,'COMPUTED ACC., VEL., DISP. VALUES IN GENERALISED',
           1          ' COORDINATES, IN INCREASING MODE NO.')
       3020 FORMAT(5X,'STEP = ',I8,'  TIME = ',G15.9,'  BASE ACC. = ',
           1       G15.9)
       3035 FORMAT(/1X,120(1H-))
       3040 FORMAT(///10X,' NO.',2X,'      PERIOD      ',2X,'  STEP  ',2X,
           1       '     TIME     ',2X,' ACC. SPECT. ',/10X,
           2       4(1H-),2X,13(1H-),2X,8(1H-),2(2X,13(1H-)))
       3045 FORMAT(10X,I4,2X,G13.7,2X,I8,2(2X,G13.7))
      C
            END
```

--VARIABLE MAP--(LO=A)

-NAME-	-ADDRESS-	-BLOCK-	-PROPERTIES-	-TYPE-	-SIZE	-NAME-	-ADDRESS-	-BLOCK-	-PROPERTIES-	-TYPE-	-SIZE
A	5130B			REAL	10	IPRT1	5323B			INTEGER	
AA	5345B			REAL		IPRT2	5330B			INTEGER	
ACC	5342B			REAL		K	5336B			INTEGER	
A0	5212B			REAL	10	KSPEC	5274B			INTEGER	10
A1	5250B			REAL	10	K1	5337B			INTEGER	
B	5142B			REAL	10	K2	5340B			INTEGER	
BB	5346B			REAL		NFRC	5325B			INTEGER	
D	5116B			REAL	10	NSPEC	5320B			INTEGER	
D0	5166B			REAL	10	PERIOD	5072B			REAL	10
D1	5224B			REAL	10	SCACC	5327B			REAL	
E	5154B			REAL	10	SCATIM	5326B			REAL	
F	1152B			REAL	1000	SPECA	5262B			REAL	10
H	5321B			REAL		T	3122B			REAL	1000
HEAD	1126B			REAL	20	TMAX	5322B			REAL	
I	5324B			INTEGER		TSPEC	5306B			REAL	10

```
         PROGRAM SPEC      74/176  OPT=0,ROUND= A/ S/ M/-D,-DS     FTN 5.1+577        85/09/19. 17.35.52        PAGE    4
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  TT       5335B                              REAL                   V1      5236B                                REAL          10
  V0       5200B                              REAL           10      W       5104B                                REAL          10


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  N        INTEGER                         10
  NF       INTEGER                       1000


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ABS       GENERIC        1     INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF          -LABEL-ADDRESS-----PROPERTIES----DEF          -LABEL-ADDRESS-----PROPERTIES----DEF

     5  INACTIVE   DO-TERM       30               90  INACTIVE   DO-TERM       91             3004     653B      FORMAT       132
    15  INACTIVE   DO-TERM       47              100    317B                   65             3006     656B      FORMAT       133
    20  INACTIVE   DO-TERM       34              110  INACTIVE   DO-TERM      106             3010     660B      FORMAT       134
    30    244B                   50              200    525B                  110             3015     700B      FORMAT       138
    40    300B                   62              210  INACTIVE   DO-TERM      112             3020     717B      FORMAT       141
    45  INACTIVE   DO-TERM       63              220    475B                  100             3025     704B      FORMAT       139
    50    326B                   68             3000    600B     FORMAT       122             3035     726B      FORMAT       143
    70    362B                   80             3001    603B     FORMAT       123             3040     731B      FORMAT       144
    75  INACTIVE   DO-TERM       85             3002    636B     FORMAT       128             3045     750B      FORMAT       147


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  SPEC       20B     0


--I/O UNITS--(LO=A)
 -NAME---  PROPERTIES------------

  TAPE5    FMT/SEQ
  TAPE6    FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH          5352B =  2794
  SCM STORAGE USED            63700B =  26560
  COMPILE TIME                1.412 SECONDS
 17.48.02.UCLP, AA, P04    ,     0.266KLNS.
```

APPENDIX C - COMPUTER PROGRAM MODAL

DATA INPUT TO MODAL

I. HEADING CARD (20A4)

Input List : HEAD

Explanation :

HEAD : character array containing the heading for the problem

The rest of the cards are free-formatted.

II. ACCELERATION FUNCTION DEFINITION CARDS

II. A CONTROL CARD

Input List : NFRC, SCATIM, SCACC

Explanation :

NFRC : number of acceleration function definition points

SCATIM : time scale

SCACC : acceleration scale

II.B TIME v.s. ACCELERATION CARD(S)

Input List	:	(T(I), F(I), I=1, NFRC)
Explanation	:	
T	:	array containing the time values
F	:	array containing the acceleration values
Note	:	F(1) corresponds to T(1), etc. Pairs must be input in the order of ascending time value.

III. CONTROL CARD

Input List	:	NDYN, NMOD, TT, TMAX, H, IPRT1, IOPT1
Explanation	:	
NDYN	:	number of dynamic degrees of freedom
NMOD	:	number of modes to be superposed
TT	:	integration start time
TMAX	:	integration finish time
H	:	integration time step
IPRT1	:	option for printing the computed accelerations, velocities and displacements in generalized and geometric coordinates at every integration point ;

= 0 ; prints

≠ 0 ; does not print

IOPT1 : option for printing the maximum displacement value occured at evey dynamic degree-of-freedom;

= 0 ; does not print

≠ 0 , prints

IV. MODAL DAMPING RATIO CARD(S)

Input List : (D(I), I=1, NMOD)

Explanation :

D : array containing the modal damping ratios in the order of ascending mode number.

V. DISPLACEMENT OUTPUT REQUEST CARD(S)

Input List : NREQ, (KR(I), I=1, NREQ)

Explanation :

NREQ : number of integration points where the displacements will be output

KR : array containing the step numbers for which the displacements will be printed.

Note : only those displacements in the direction of dynamic degrees-of-freedom will be output.

Program MODAL must be executed after program EIG3.

DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=MODAL,B=B,L=LMODAL.

```
      PROGRAM MODAL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      PARAMETER (N =50)
      PARAMETER (NF=1000)
C
      DIMENSION HEAD(20)
      DIMENSION F(NF),T(NF)
      DIMENSION PHI(N,N),W(N),PF(N),D(N),DMAX(N)
      DIMENSION A(N),B(N),E(N),D0(N),V0(N),A0(N),D1(N),V1(N),A1(N)
      DIMENSION DM(N),KM(N),TM(N)
      DIMENSION KR(N)
C
      READ(5,'(20A4)') HEAD
      WRITE(6,3000) HEAD
C
      READ(5,*) NFRC,SCATIM,SCACC
      READ(5,*) (T(I),F(I),I=1,NFRC)
C
      READ(5,*) NDYN,NMOD,TT,TMAX,H,IPRT1,IOPT1
C
      WRITE(6,3002) NDYN,NMOD,TT,TMAX,H,IPRT1,IOPT1
C
      READ(5,*) (D(I),I=1,NMOD)
C
      READ(5,*) NREQ,(KR(I),I=1,NREQ)
C
      OPEN(UNIT=20,STATUS='OLD',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     FILE='TAPE20',RECL=2+NDYN)
C
      OPEN(UNIT=21,STATUS='SCRATCH',ACCESS='DIRECT',FORM='UNFORMATTED',
     1                 RECL=2+NDYN)
C
      WRITE(6,3010)
      DO 5 I=1,NMOD
        READ(20,REC=I) W(I),PF(I),(PHI(J,I),J=1,NDYN)
        A(I)=2.0*D(I)*W(I)
        B(I)=W(I)*W(I)
        E(I)=1.0+A(I)*H/2.0+B(I)*H*H/6.0
        WRITE(6,3015) I,W(I),D(I),PF(I)
    5 CONTINUE
C
      WRITE(6,3003) (I,I=1,8)
      DO 10 I=1,NDYN
   10   WRITE(6,3004) (PHI(I,J),J=1,NMOD)
C
      DO 20 I=1,NFRC
        T(I)=T(I)*SCATIM
   20   F(I)=F(I)*SCACC
      IF(IPRT1.NE.0) WRITE(6,3001) NFRC,SCATIM,SCACC,(I,T(I),F(I),
     1                             I=1,NFRC)
C
      DO 15 I=1,NDYN
        DM(I)=0.0
        KM(I)=0
   15   TM(I)=0.0
```

```
C
      WRITE(6,3035)
      K=0
      K1=1
      K2=2
C
      IF(TT.GT.TMAX) GO TO 200
   30 IF(K2.GT.NFRC) THEN
        ACC=0.0
        GO TO 40
      ENDIF
      IF(TT.LE.T(K2)) THEN
        ACC=F(K1)+(F(K2)-F(K1))/(T(K2)-T(K1))*(TT-T(K1))
      ELSE
        K1=K2
        K2=K2+1
        GO TO 30
      ENDIF
C
   40 DO 45 I=1,NMOD
   45   A0(I)=-PF(I)*ACC-A(I)*V0(I)-B(I)*D0(I)
C
  100 TT=TT+H
      K=K+1
      IF(TT.GT.TMAX) GO TO 200
   50 IF(K2.GT.NFRC) THEN
        ACC=0.0
        GO TO 70
      ENDIF
      IF(TT.LE.T(K2)) THEN
        ACC=F(K1)+(F(K2)-F(K1))/(T(K2)-T(K1))*(TT-T(K1))
      ELSE
        K1=K2
        K2=K2+1
        GO TO 50
      ENDIF
C
   70 DO 75 I=1,NMOD
        AA=V0(I)+A0(I)*H/2.0
        BB=D0(I)+V0(I)*H+A0(I)*H*H/3.0
        A1(I)=(-PF(I)*ACC-A(I)*AA-B(I)*BB)/E(I)
        V1(I)=AA+A1(I)*H/2.0
   75   D1(I)=BB+A1(I)*H*H/6.0
C
      DO 90 I=1,NMOD
        A0(I)=A1(I)
        V0(I)=V1(I)
        D0(I)=D1(I)
   90 CONTINUE
C
      DO 300 I=1,NDYN
        DMAX(I)=0.0
        DO 310 J=1,NMOD
  310     DMAX(I)=DMAX(I)+PHI(I,J)*D1(J)
  300 CONTINUE
C
      IF(IOPT1.NE.0) THEN
```

```
            DO 110  I=1,NDYN
              IF(ABS(DMAX(I)).GT.ABS(DM(I))) THEN
                DM(I)=DMAX(I)
                KM(I)=K
                TM(I)=TT
              ENDIF
  110       CONTINUE
            WRITE(21,REC=K) K,TT,(DMAX(I),I=1,NDYN)
            GO TO 100
          ENDIF
C
          WRITE(6,3020) K,TT,ACC
          WRITE(6,3025)
          WRITE(6,3004) (A1(I),I=1,NMOD)
          WRITE(6,3004) (V1(I),I=1,NMOD)
          WRITE(6,3004) (D1(I),I=1,NMOD)
C
          WRITE(6,3030)
          WRITE(6,3004) (DMAX(I),I=1,NDYN)
C
          GO TO 100
C
  200 IF(IOPT1.NE.0) THEN
            WRITE(6,3040)
            DO 210 I=1,NDYN
  210         WRITE(6,3045) I,KM(I),TM(I),DM(I)
            WRITE(6,3050)
            DO 220 I=1,NREQ
              READ(21,REC=KR(I)) K,TT,(DMAX(J),J=1,NDYN)
  220         WRITE(6,3055) K,TT,(DMAX(J),J=1,NDYN)
          ENDIF
C
          WRITE(6,3006)
C
          STOP
C
C ****************************************************************
C         FORMATS
C ****************************************************************
C
 3000 FORMAT(1H1,///25X,20A4,/)
 3001 FORMAT(/25X,'NUMBER OF ACCELERATION ORDINATES = ',I5,
     1       /25X,'TIME         SCALE                = ',G13.7,
     2       /25X,'ACCELERATION SCALE                = ',G13.7,
     3      //25X,'  NO  ',2X,'    TIME     ',2X,'ACCELERATION ',
     4       /25X,6(1H-),2(2X,13(1H-)),/(25X,I5,3X,2(G13.7,2X)))
 3002 FORMAT(///30X,'DYNAMIC DOF = ',I5,
     1         /30X,'NMODES      = ',I5,
     2         /30X,'START TIME  = ',G13.7,
     3         /30X,'FINISH TIME = ',G13.7,
     4         /30X,'TIME STEP   = ',G13.7,
     5         /30X,'PRINT OPT.1 = ',I5,
     6         /30X,'EXEC. OPT.1 = ',I5,/)
 3003 FORMAT(///30X,'NORMALIZED MODE SHAPES',/30X,22(1H-),
     1       /1X,8('     MODE ',I2,'    ',1X),
     2       /1X,8(15(1H-),1X))
 3004 FORMAT(1X,8(G15.9,1X))
```

```
      3006 FORMAT(1H1,6(/))
      3010 FORMAT(/30X,'MODAL   INFORMATION',/30X,19(1H-),
     1        /30X,'MODE',2X,'ANGULAR FREQ.    ',2X,'CRIT. DAM. RAT.',
     2         2X,'PARTICIP. FACT.',
     3        /30X,4(1H-),3(2X,15(1H-)))
      3015 FORMAT(30X,I3,3X,3(G15.9,2X))
      3025 FORMAT(10X,'COMPUTED ACC., VEL., DISP. VALUES IN GENERALISED',
     1             ' COORDINATES, IN INCREASING MODE NO.')
      3020 FORMAT(5X,'STEP = ',I8,'  TIME = ',G15.9,'  BASE ACC. = ',
     1        G15.9)
      3030 FORMAT(10X,'SUPERPOSED DISP. VALUES FOR THE DYN. D.O.F. S ',
     1        'IN INCREASING MODE NO. FROM LEFT TO RIGHT.')
      3035 FORMAT(/1X,120(1H-))
      3040 FORMAT(//30X,' DIR. ',2X,'  STEP  ',2X,'    TIME     ',2X,
     1             ' MAX. VALUE  ',
     2          /30X,6(1H-),2X,8(1H-),2(2X,13(1H-)))
      3045 FORMAT(30X,I5,3X,I8,2(2X,G13.7))
      3050 FORMAT(//20X,'DISPLACEMETS AT THE REQUESTED TIMES',
     1          /19X,36(1H-))
      3055 FORMAT(/5X,'STEP = ',I3,2X,'TIME = ',G13.7,(/1X,9(G13.7,1X)))
     C
            END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	--BLOCK--	--PROPERTIES--	--TYPE--	--SIZE
A	127048			REAL	50
AA	141518			REAL	
ACC	141468			REAL	
A0	132768			REAL	50
A1	135248			REAL	50
B	127668			REAL	50
BB	141528			REAL	
D	125408			REAL	50
DM	136068			REAL	50
DMAX	126228			REAL	50
D0	131328			REAL	50
D1	133608			REAL	50
E	130508			REAL	50
F	15508			REAL	1000
H	141278			REAL	
HEAD	15248			REAL	20
I	141218			INTEGER	
IOPT1	141318			INTEGER	
IPRT1	141308			INTEGER	
J	141348			INTEGER	
K	141438			INTEGER	
KM	136708			INTEGER	50
KP	140348			INTEGER	50
K1	141448			INTEGER	
K2	141458			INTEGER	
NDYN	141238			INTEGER	
NFRC	141168			INTEGER	
NMOD	141248			INTEGER	
NREQ	141328			INTEGER	
PF	124568			REAL	50
PHI	54708			REAL	2500
SCACC	141208			REAL	
SCATIM	141178			REAL	
T	35208			REAL	1000
TM	137528			REAL	50
TMAX	141268			REAL	
TT	141258			REAL	
V0	132148			REAL	50
V1	134428			REAL	50
W	123748			REAL	50

--SYMBOLIC CONSTANTS--(LO=A)

-NAME-	--TYPE--	--VALUE
N	INTEGER	50
NF	INTEGER	1000

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ABS        GENERIC        1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF      -LABEL-ADDRESS-----PROPERTIES----DEF      -LABEL-ADDRESS-----PROPERTIES----DEF

     5  INACTIVE    DO-TERM        40          110  INACTIVE    DO-TERM       119        3006   1122B       FORMAT       170
    10  INACTIVE    DO-TERM        44          200    704B                    135        3010   1124B       FORMAT       171
    15  INACTIVE    DO-TERM        55          210  INACTIVE    DO-TERM       138        3015   1144B       FORMAT       175
    20  INACTIVE    DO-TERM        48          220  INACTIVE    DO-TERM       142        3020   1163B       FORMAT       178
    30    347B                     63          300  INACTIVE    DO-TERM       110        3025   1150B       FORMAT       176
    40    403B                     75          310  INACTIVE    DO-TERM       109        3030   1172B       FORMAT       180
    45  INACTIVE    DO-TERM        76         3000   1021B      FORMAT        153        3035   1266B       FORMAT       182
    50    432B                     81         3001   1024B      FORMAT        154        3040   1211B       FORMAT       183
    70    466B                     93         3002   1057B      FORMAT        159        3045   1225B       FORMAT       186
    75  INACTIVE    DO-TERM        98         3003   1104B      FORMAT        166        3050   1231B       FORMAT       187
    90  INACTIVE    DO-TERM       104         3004   1117B      FORMAT        169        3055   1240B       FORMAT       189
   100    423B                     78


 --ENTRY POINTS--(LO=A)
  -NAME---ADDRESS--ARGS---

   MODAL      20B      0


 --I/O UNITS--(LO=A)
  -NAME--- PROPERTIES-------------

   TAPE20  AUX/BIN/DIR
   TAPE21  AUX/BIN/DIR
   TAPE5   FMT/SEQ
   TAPE6   FMT/SEQ


 --STATISTICS--

   PROGRAM-UNIT LENGTH           14161B =   6257
   SCM STORAGE USED              63700B =  26560
   COMPILE TIME                  1.899 SECONDS
17.43.54.UCLP, AA, P04,            0.317KLNS.
```

APPENDIX D - COMPUTER PROGRAM DSSI

DATA INPUT TO DSSI

I. HEADING CARD (20A4)

Input List : HEAD

Explanation :

HEAD : character array containing the problem description.

The rest of the data cards are free formatted.

II. CONTROL CARD

Input List : NDYN, DT, TMAX, IPRT1, IOPT1.

Explanation :

NDYN : number of dynamic degrees of freedom

DT : solution time step, Δt

TMAX : time up to which the integration will be performed

IPRT1 : option for printing the reduced flexibility, reduced stiffness, damping, [E] and $[E]^{-1}$ matrices [See Eqs. (II.40)]

= 0 ; do not print

≠ 0 ; print

IOPT1 : execution mode option for finding the maximum displacement occuring at every dynamic degree of freedom;

= 0 ; prints only the accelerations, velocities and displacements computed at every step

≠ 0 ; prints only the maximum displacements

III. DAMPING FACTORS CARD

Input List : ALFA, BETA

Explanation :

ALFA : damping factor α

BETA : damping factor β

Note : ALFA and BETA are factors to compute the Rayleigh damping matrix given by Eq.(III.1)

IV. ACCELERATION FUNCTION DEFINITION CARDS

IV.A CONTROL CARD

Input List : NFRC, SCATIM, SCACC, IPRT2

Explanation

NFRC	:	number of function definition points
SCATIM	:	time scale
SCACC	:	acceleration scale
IPRT2	:	option for printing the input acceleration function definition points; = 0 ; does not print ≠ 0 ; prints the values

VI.B TIME V.S. ACCELERATION CARD(S)

Input List : (T(I), F(I), I=1, NFRC)

Explanation :

T	:	array containing the time values
F	:	array containing the acceleration function values

Note : T(1) corresponds to F(1), etc.

```
      PROGRAM DSSI (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     DIRECT STEP-BY-STEP INTEGRATION BY LINEAR
C                  ACCELERATION METHOD
C
      PARAMETER (NF=1000,NM=50)

      REAL MASS(NM)
      DIMENSION HEAD(20)
      DIMENSION T(NF),F(NF)
      DIMENSION NDC(3),SB(NM,NM),Q(NM),C(NM,NM),E(NM,NM)
      DIMENSION DUM1(NM),D0(NM),V0(NM),A0(NM),A(NM),B(NM)
      DIMENSION D1(NM),V1(NM),A1(NM)
      DIMENSION DM(NM),KM(NM),TM(NM)
C
C     READ AND PRINT SYSTEM PARAMETERS
C
      READ(5,'(20A4)') HEAD
      READ(5,*) NDYN,DT,TMAX,IPRT1,IOPT1
      READ(5,*) ALFA,BETA
      READ(5,*) NFRC,SCATIM,SCACC,IPRT2
C
      WRITE(6,3005) HEAD
      WRITE(6,3006)NDYN,DT,TMAX,IPRT1,ALFA,BETA,NFRC,SCATIM,SCACC,IPRT2,
     1              IOPT1
C
C     READ THE BASE ACCELERATION DATA
C
      READ(5,*) (T(I),F(I),I=1,NFRC)
C
      OPEN(UNIT=10,STATUS='OLD',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     FILE='TAPE10',RECL=1+3)
      OPEN(UNIT=14,STATUS='OLD',ACCESS='DIRECT',FORM='UNFORMATTED',
     1     FILE='TAPE14',RECL=NDYN)
      OPEN(UNIT=21,STATUS='SCRATCH',ACCESS='DIRECT',FORM='UNFORMATTED',
     1                 RECL=NDYN)
C
C     READ THE MASS COEFFICIENTS, CONSTRUCT THE REDUCED FLEXIBILITY MATRIX,
C                          FORM THE Q VECTOR.
      WRITE(6,3012)
      DO 10 I=1,NDYN
        READ(5,*) NODE,IDIR,MASS(I)
        WRITE(6,3015) NODE,IDIR,MASS(I)
        READ(10,REC=NODE) JFLG,(NDC(J),J=1,3)
        READ(14,REC=NDC(IDIR)) (SB(I,J),J=1,NDYN)
        IF(IDIR.EQ.1) Q(I)=1.0
   10 CONTINUE
C
      DO 5 I=1,NFRC
        T(I)=T(I)*SCATIM
    5   F(I)=F(I)*SCACC
      IF(IPRT2.NE.0) WRITE(6,3010) (I,T(I),F(I),I=1,NFRC)
      IF(IPRT1.NE.0) THEN
        WRITE(6,3007)
        DO 15 I=1,NDYN
```

```
   15     WRITE(6,3110) (SB(I,J),J=1,NDYN)
        ENDIF
C
C       INVERT THE RED. FLEX. MATRIX TO OBTAIN THE REDUCED STIFFNESS MATRIX
C
        CALL INVERT(SB,NDYN,NM)
C
        IF(IPRT1.NE.0) THEN
          WRITE(6,3008)
          DO 16 I=1,NDYN
   16     WRITE(6,3110) (SB(I,J),J=1,NDYN)
        ENDIF
C
C       COMPUTE MATRIX C=ALFA*MASS+BETA*SB
C
        DO 20 I=1,NDYN
          DO 20 J=1,NDYN
   20       C(I,J)=BETA*SB(I,J)
        DO 30 I=1,NDYN
   30   C(I,I)=C(I,I)+ALFA*MASS(I)
C
        IF(IPRT1.NE.0) THEN
          WRITE(6,3009)
          DO 25 I=1,NDYN
   25     WRITE(6,3110) (C(I,J),J=1,NDYN)
        ENDIF
C
C       COMPUTE MATRIX E=MASS+C*DT/2+SB*DT**2/6
C
        DO 50 I=1,NDYN
          DO 50 J=1,NDYN
   50     E(I,J)=C(I,J)*DT*0.5+SB(I,J)*DT*DT/6.0
        DO 60 I=1,NDYN
   60   E(I,I)=E(I,I)+MASS(I)
C
        IF(IPRT1.NE.0) THEN
          WRITE(6,3011)
          DO 55 I=1,NDYN
   55     WRITE(6,3110) (E(I,J),J=1,NDYN)
        ENDIF
C
C       INVERT MATRIX E
C
        CALL INVERT(E,NDYN,NM)
C
        IF(IPRT1.NE.0) THEN
          WRITE(6,3013)
          DO 65 I=1,NDYN
   65     WRITE(6,3110) (E(I,J),J=1,NDYN)
        ENDIF
C
C       COMPUTE THE INITIAL ACCELERATION VECTOR
C
        DO 70 I=1,NDYN
   70   DUM1(I)=-F(1)*Q(I)*MASS(I)
        DO 80 I=1,NDYN
          SUM=0.0
```

```
          DO 90 J=1,NDYN
   90       SUM=SUM+C(I,J)*V0(J)
   80     DUM1(I)=DUM1(I)-SUM
        DO 100 I=1,NDYN
          SUM=0.0
          DO 110 J=1,NDYN
  110       SUM=SUM+SB(I,J)*D0(J)
  100     DUM1(I)=DUM1(I)-SUM
        DO 120 I=1,NDYN
  120     A0(I)=DUM1(I)/MASS(I)
C
        WRITE(6,3020)
        WRITE(6,3110) (A0(I),I=1,NDYN)
        WRITE(6,3110) (V0(I),I=1,NDYN)
        WRITE(6,3110) (D0(I),I=1,NDYN)
C
        TT=0.0
        K=0
        K1=1
        K2=2
C
        DO 125 I=1,NDYN
          DM(I)=0.0
          KM(I)=0
  125     TM(I)=0.0
C
C *** PERFORM THE INTEGRATION AT TIME TT ***
C
 1000   TT=TT+DT
        K=K+1
C
C       COMPUTE A=V0+A0*DT/2.0  AND  B=D0+V0*DT+A0*DT**2/3.0
C
        DO 130 I=1,NDYN
          A(I)=V0(I)+A0(I)*DT/2.0
  130     B(I)=D0(I)+V0(I)*DT+A0(I)*DT*DT/3.0
C
C       COMPUTE THE BASE ACCELERATION AT TIME TT BY LINEAR INTERPOLATION.
C
        IF(TT.GT.TMAX) GO TO 1500
  150   IF(K2.GT.NFRC) THEN
          ACC=0.0
          GO TO 160
        ENDIF
        IF(TT.LE.T(K2)) THEN
          ACC= F(K1)+(F(K2)-F(K1))/(T(K2)-T(K1))*(TT-T(K1))
        ELSE
          K1=K2
          K2=K2+1
          GO TO 150
        ENDIF
C
C       COMPUTE THE ACCELERATION AT TIME TT
C
  160   DO 170 I=1,NDYN
  170     DUM1(I)=-ACC*G(I)*MASS(I)
        DO 180 I=1,NDYN
```

```
              SUM=0.0
              DO 190 J=1,NDYN
  190           SUM=SUM+C(I,J)*A(J)
  180         DUM1(I)=DUM1(I)-SUM
            DO 200 I=1,NDYN
              SUM=0.0
              DO 210 J=1,NDYN
  210           SUM=SUM+SB(I,J)*B(J)
  200         DUM1(I)=DUM1(I)-SUM
            DO 220 I=1,NDYN
              SUM=0.0
              DO 230 J=1,NDYN
  230           SUM=SUM+E(I,J)*DUM1(J)
  220         A1(I)=SUM
C
C           COMPUTE THE VELOCITIES    AT TIME TT, V1=A+A1*DT/2.0
C           COMPUTE THE DISPLACEMENTS AT TIME TT, D1=B+A1*DT**2/6.0
C
            DO 240 I=1,NDYN
              V1(I)=A(I)+A1(I)*DT/2.0
  240         D1(I)=B(I)+A1(I)*DT*DT/6.0
C
            IF(IOPT1.NE.0) THEN
              DO 260 I=1,NDYN
                IF(ABS(D1(I)).GT.ABS(DM(I))) THEN
                  DM(I)=D1(I)
                  KM(I)=K
                  TM(I)=TT
                ENDIF
  260         CONTINUE
              GO TO 300
            ENDIF
C
C           PRINT THE COMPUTED ACC. VEL. AND DISP. VECTORS
C
            WRITE(6,3100) K,TT,ACC
            WRITE(6,3110) (A1(I),I=1,NDYN)
            WRITE(6,3110) (V1(I),I=1,NDYN)
            WRITE(6,3110) (D1(I),I=1,NDYN)
C
  300       DO 250 I=1,NDYN
              A0(I)=A1(I)
              V0(I)=V1(I)
  250         D0(I)=D1(I)
C
            WRITE(21,REC=K) (D1(I),I=1,NDYN)
C
            GO TO 1000
C
 1500       IF(IOPT1.NE.0) THEN
              WRITE(6,3040)
              DO 270 I=1,NDYN
  270           WRITE(6,3045) I,KM(I),TM(I),DM(I)
              WRITE(6,3120)
              DO 280 I=1,NDYN
                READ(21,REC=KM(I)) (D1(J),J=1,NDYN)
  280           WRITE(6,3110) (D1(J),J=1,NDYN)
```

```
           ENDIF
      C
           WRITE(6,3150)
           STOP
      C
      C ***********************************************************************
      C    FORMATS
      C ***********************************************************************
      C
 3005 FORMAT(1H1,///,1X,20A4,/)
 3006 FORMAT(/20X,'DYNAMIC DOF              = ',I5,
     1        /20X,'TIME STEP                = ',G13.7,
     2        /20X,'MAX. TIME                = ',G13.7,
     3        /20X,'PRINT OPTION 1           = ',I5,
     4        /20X,'ALFA                     = ',G13.7,
     5        /20X,'BETA                     = ',G13.7,
     6        /20X,'NO.OF BASE ACC. PTS.     = ',I5,
     7        /20X,'TIME          SCALE      = ',G13.7,
     8        /20X,'ACCELERATION SCALE       = ',G13.7,
     9        /20X,'PRINT OPTION 2           = ',I5,
     1        /20X,'EXECUTION MODE OPT.      = ',I5,/)
 3007 FORMAT(/20X,'REDUCED FLEXIBILITY MATRIX',/20X,26(1H-))
 3008 FORMAT(/20X,'REDUCED STIFFNESS   MATRIX',/20X,26(1H-))
 3009 FORMAT(/20X,'DAMPING MATRIX',/20X,14(1H-))
 3010 FORMAT(//30X,'BASE ACCELERATION DATA',/30X,22(1H-),
     1          /20X,' NO. ',2X,'    TIME      ',2X,'ACCELERATION ',
     2          /20X,5(1H-),2(2X,13(1H-)),/,(20X,I5,2(2X,G13.7)))
 3011 FORMAT(/20X,'MATRIX E=MASS+C*DT/2+SB*DT**2/6',/20X,31(1H-))
 3012 FORMAT(/30X,'MASS DATA',/30X,9(1H-),
     1        /20X,' NODE ',2X,'DIR.',2X,'MASS COEFFICIENT',
     2        /20X,6(1H-),2X,4(1H-),2X,16(1H-))
 3013 FORMAT(/20X,'INVERSE OF MATRIX E',/20X,19(1H-))
 3015 FORMAT(20X,I5,4X,I2,4X,G13.7)
 3020 FORMAT(//30X,'INITIAL ACC., VEL. AND DISP VECTORS',
     1         /30X,35(1H-))
 3040 FORMAT(//30X,' DIR. ',2X,'  STEP  ',2X,'     TIME      ',2X,
     1             ' MAX. VALUE  ',
     2         /30X,6(1H-),2X,8(1H-),2(2X,13(1H-)))
 3045 FORMAT(30X,I5,3X,I8,2(2X,G13.7))
 3100 FORMAT(/20X,'STEP = ',I5,2X,'TIME = ',G13.7,2X,'BASE ACC. = ',
     1       G13.7,/30X,'ACC., VEL. AND DISP. VALUES',/30X,27(1H-))
 3110 FORMAT(8(1X,G15.9))
 3120 FORMAT(//20X,'DISPLACEMENT VECTORS AT THE TIMES WHEN MAX. VAL.',
     1            ' ARE DETECTED, IN INCREASING ORDER.',/15X,100(1H-))
 3150 FORMAT(1H1,/////)
      C
           END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	--BLOCK--	-----PROPERTIES-----	-----TYPE-----	-----SIZE	-NAME-	--ADDRESS--	--BLOCK--	-----PROPERTIES-----	-----TYPE-----	-----SIZE
A	25424B			REAL	50	B	25506B			REAL	50
ACC	26326B			REAL		BETA	26252B			REAL	
ALFA	26251B			REAL		C	13304B			REAL	2500
A0	25342B			REAL	50	DN	26016B			REAL	50
A1	25734B			REAL	50	DT	26245B			REAL	

```
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 DUM1     25114B                             REAL            50    MASS     2265B                                 REAL          50
 D0       25176B                             REAL            50    NDC      6313B                                 INTEGER        3
 D1       25570B                             REAL            50    NDYN    26244B                                 INTEGER
 E        20210B                             REAL          2500    NFRC    26253B                                 INTEGER
 F         4343B                             REAL          1000    NODE    26262B                                 INTEGER
 HEAD      2347B                             REAL            20    Q       13222B                                 REAL          50
 I        26257B                             INTEGER               SB       6316B                                 REAL        2500
 IDIR     26263B                             INTEGER               SCACC   26255B                                 REAL
 IOPT1    26250B                             INTEGER               SCATIM  26254B                                 REAL
 IPRT1    26247B                             INTEGER               SUM     26313B                                 REAL
 IPRT2    26256B                             INTEGER               T        2373B                                 REAL        1000
 J        26265B                             INTEGER               TM      26162B                                 REAL          50
 JFLG     26264B            *S*              INTEGER               TMAX    26246B                                 REAL
 K        26321B                             INTEGER               TT      26320B                                 REAL
 KM       26100B                             INTEGER         50    V0      25260B                                 REAL          50
 K1       26322B                             INTEGER               V1      25652B                                 REAL          50
 K2       26323B                             INTEGER

--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  NF      INTEGER                      1000
  NM      INTEGER                        50

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ABS       GENERIC       1       INTRINSIC
  INVERT                  3       SUBROUTINE

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

     5  INACTIVE   DO-TERM      51            130  INACTIVE   DO-TERM       148          3005   1521B      FORMAT       236
    10  INACTIVE   DO-TERM      47            150   1052B                   153          3006   1524B      FORMAT       237
    15  INACTIVE   DO-TERM      56            160   1106B                   167          3007   1576B      FORMAT       248
    16  INACTIVE   DO-TERM      66            170  INACTIVE   DO-TERM       168          3008   1604B      FORMAT       249
    20  INACTIVE   DO-TERM      73            180  INACTIVE   DO-TERM       173          3009   1612B      FORMAT       250
    25  INACTIVE   DO-TERM      80            190  INACTIVE   DO-TERM       172          3010   1617B      FORMAT       251
    30  INACTIVE   DO-TERM      75            200  INACTIVE   DO-TERM       178          3011   1637B      FORMAT       254
    50  INACTIVE   DO-TERM      87            210  INACTIVE   DO-TERM       177          3012   1646B      FORMAT       255
    55  INACTIVE   DO-TERM      94            220  INACTIVE   DO-TERM       183          3013   1662B      FORMAT       258
    60  INACTIVE   DO-TERM      89            230  INACTIVE   DO-TERM       182          3015   1670B      FORMAT       259
    65  INACTIVE   DO-TERM     104            240  INACTIVE   DO-TERM       190          3020   1674B      FORMAT       260
    70  INACTIVE   DO-TERM     110            250  INACTIVE   DO-TERM       213          3040   1703B      FORMAT       262
    80  INACTIVE   DO-TERM     115            260  INACTIVE   DO-TERM       199          3045   1717B      FORMAT       265
    90  INACTIVE   DO-TERM     114            270  INACTIVE   DO-TERM       222          3100   1723B      FORMAT       266
   100  INACTIVE   DO-TERM     120            280  INACTIVE   DO-TERM       226          3110   1740B      FORMAT       268
   110  INACTIVE   DO-TERM     119            300   1351B                   210          3120   1743B      FORMAT       269
   120  INACTIVE   DO-TERM     122           1000   1016B                   141          3150   1760B      FORMAT       271
   125  INACTIVE   DO-TERM     137           1500   1400B                   219
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DSSI       20B     0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE10  AUX/BIN/DIR
  TAPE14  AUX/BIN/DIR
  TAPE21  AUX/BIN/DIR
  TAPE5   FMT/SEQ
  TAPE6   FMT/SEQ

--STATISTICS--

  PROGRAM-UNIT LENGTH          26343B =  11491
  SCM STORAGE USED             65700B =  27584
  COMPILE TIME                 2.765 SECONDS
```

```
SUBROUTINE INVERT      74/176   OPT=0,ROUND= A/ S/ M/-D,-DS       FTN 5.1+577            85/09/19. 17.37.29          PAGE        1
DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
FTN5,I=DSSI,B=B,L=LDSSI.

     1                     SUBROUTINE INVERT(A,N,NMAX)
     2               C
     3                     DIMENSION A(NMAX,NMAX)
     4               C
     5                     DO 10 K=1,N
     6                       DO 20 I=1,N
     7                         DO 20 J=1,N
     8                           IF(I.EQ.K.OR.J.EQ.K) GO TO 20
     9                           A(I,J)=A(I,J)-A(I,K)*A(K,J)/A(K,K)
    10                  20     CONTINUE
    11                     A(K,K)=-1.0/A(K,K)
    12                     DO 30 I=1,N
    13                       IF(I.EQ.K) GO TO 30
    14                       A(I,K)=A(I,K)*A(K,K)
    15                  30 CONTINUE
    16                     DO 40 J=1,N
    17                       IF(J.EQ.K) GO TO 40
    18                       A(K,J)=A(K,J)*A(K,K)
    19                  40 CONTINUE
    20                  10 CONTINUE
    21               C
    22                     DO 50 I=1,N
    23                       DO 50 J=1,N
    24                  50     A(I,J)=-A(I,J)
    25               C
    26                     RETURN
    27                     END

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 A            1   DUMMY-ARG                  REAL       ADJ-ARY      K          214B                               INTEGER
 I          216B                             INTEGER                 N             2   DUMMY-ARG                   INTEGER
 J          220B                             INTEGER                 NMAX          3   DUMMY-ARG                   INTEGER

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE    DO-TERM        20
    20      54B     DO-TERM        10
    30     114B     DO-TERM        15
    40     141B     DO-TERM        19
    50  INACTIVE    DO-TERM        24

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  INVERT      5B      3
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH              230B =   152
  SCM STORAGE USED               61700B = 25536
  COMPILE TIME                   0.388 SECONDS
17.53.12.UCLP, AA, P04      ,      0.479KLNS.
```

BIBLIOGRAPHY

(1) Tezcan, S.S., <u>Çubuk Sistemlerin Elektronik Hesap Makineleri ile Çözümü</u>, İstanbul Teknik Üniversitesi Kütüphanesi, Elektronik Hesap Bilimleri Enstitüsü Sayı:12, 1970.

(2) Çakıroğlu, A., Özden, E., ve Özmen, G., <u>Yapı Sistemlerinin Hesabı için Matris Metodları ve Elektronik Hesap Makinası Programları</u>, Cilt II, İstanbul Teknik Üniversitesi Kütüphanesi Sayı:1005, 1974.

(3) Ghali, A. and Neville, A.M., <u>Structural Analysis</u>, Chapman and Hall, New York, 1978.

(4) Tuma, J.J. and Munshi, R.K., <u>Advanced Structural Analysis</u>, McGraw-Hill Book Co., New York, 1971

(5) Hood, P., "Frontal Solution Program for Unsymmetric Matrices," <u>Int. J. Num. Meth. Engng.</u> Vol. 10, 1976, pp. 379-399.

(6) Clough, R.W. and Penzien, J., <u>Dynamics of Structures</u> McGraw-Hill Book Co., New York, 1975.

(7) Çakıroğlu, A., Özden, E., ve Özmen, G., Yapı Sistemlerinin Hesabı için Matris Metodları ve Elektronik Hesap Makinası Programları, Cilt I, İstanbul Teknik Üniversitesi Kütüphanesi Sayı:813,1970.

(8) Aktaş, Z., Öncül, H., ve Ural, S.,Sayısal Çözümleme, Cilt I, Orta Doğu Teknik Üniversitesi, Ankara, 1981.

(9) Wilkinson, J.H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

(10) Çakıroğlu, A. ve Özmen G., Yapıların Özel Periyotlarının Tayini ve Modların Süperpozisyonu Yöntemi, Uluğ Kitabevi Yayınları, No.11, İstanbul, 1977,.

(11) Efil, K., "Kemer Tipi Yapıların II. Mertebe Teorisi ile Hesabı," M.M.L.S Diploma Tezi, İstanbul Teknik Üniversitesi Maçka İnşaat Fakültesi, 1982.

(12) Tezcan, S.S., Yüksek Yapılarda Rüzgar ve Deprem Yükleri, Boğaziçi Üniversitesi, İnşaat Mühendisliği Şubesi, Dahili Rapor No:75-13T, İstanbul, 1975.

(13) Newmark, N.M. and Rosenblueth, E., Fundamentals of Earthquake Engineering, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

(14) Biggs, J.M., Introduction to Structural Dynamics, McGraw-Hill Book Co., New York, 1968.

(15) Englekirk, R.E. and Hart, G.C, Earhquake Design of Concrete Masonry Buildings, Vol.1, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

(16) Kuşcuoğlu, M., "Yatay Yük Taşıyıcı Düşey Elemanlar, Çerçeveler," M.M.L.S. Diploma Tezi, İstanbul Teknik Üniversitesi Maçka İnşaat Fakültesi, 1982.

(17) Karadoğan, H.F, "İkinci Mertebe Teorisi ile Deprem Hesabı İçin Bir Metod," Doktora Tezi, İstanbul Teknik Üniversitesi Mühendislik-Mimarlık Fakültesi, 1974.