CHARACTERISTICS OF FINITE ELEMENTS
FOR ANALYSIS OF BEAMS SUB-
JECTED TO IMPACT LOAD

By

BRIJ RAJ KISHORE

Bachelor of Architecture
University of Roorkee
Roorkee, U. P., India
1961

Master of Architectural Engineering
Oklahoma State University
Stillwater, Oklahoma
1967

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1973

CHARACTERISTICS OF FINITE ELEMENTS
FOR ANALYSIS OF BEAMS SUB-
JECTED TO IMPACT LOAD

Thesis Approved:

William P. Dawkins
Thesis Adviser

R. Lumsdaine

A. E. Kelly

Donald E. Boyd

D. D. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

My first and special feelings of thanks should go to Mr. and Mrs. J. E. Robinson, for without their help, guidance, and encouragement, I could not have been able to complete this work.

I sincerely express my indebtedness to Dr. J. V. Parcher for providing the teaching assistantship which helped me to complete my graduate study.

I wish to express my thanks to Mr. Eldon Hardy for his help in the preparation of the drawings and to Miss Charlene Fries and Mrs. Carol Patterson for their hard work and careful and professional typing.

Now, I turn to the pleasant task of acknowledging the debts of gratitude to the members of my advisory committee: to Dr. W. P. Dawkins, major adviser and chairman of the committee, for his sound instruction, interest, and personal guidance; and to Drs., D. E. Boyd, A. E. Kelly, and R. K. Munshi for their advice and encouragement.

At last, with the vivid lingering memories of my mother and father, I feel most grateful to my parents, who strove to give me an education but missed seeing me fulfill their hopes.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure		Page

NOMENCLATURE

A	length of a rectangular element along x^e-direction in the element axis system
$[A]$	transformation matrix relating displacement parameters and nodal displacements
a	length of the base of an element along x^e-direction in the element axis system
B	width of a rectangular element along y^e-direction in the element coordinate system
$[B(x, y)]$	rectangular matrix: displacement parameter transformation matrix to internal strains of an element
b	height of an element along y^e-direction in the element coordinate system
C_{ij}	elements of a matrix
$[C(x, y)]$	rectangular transformation matrix relating internal displacements of an element with its nodal displacements
$[C]$ or $[C_{ij}]$	damping matrix of the system
c	x^e coordinate of the vertex of the triangular element coordinate system
$[D]$	matrix of elastic constants
E	modulus of elasticity of the material
$\overline{F}$	dissipation function of the system
G	shear modulus of the material
$[K]$ or $[K_{ij}]$	structure stiffness matrix
$[k]$	element stiffness matrix
$[\overline{k}]$	element stiffness matrix in global coordinate system

L	span of the beam
LSR	linear strain rectangular element
$\left[L(x, y) \right]$	rectangular transformation matrix relating displacement parameters and internal displacements of an element
$\left[M \right]$ or $\left[M_{ij} \right]$	structure mass matrix
$m(x, y)$	element mass per unit area at a point (x, y)
N	number of nodal points in the structure
NY	number of elements along the y-direction in the mesh
P	applied static load at a point
$\overline{P}$	applied dynamic load at a point
$\left[Q \right]$	rectangular transformation matrix relating internal stresses of an element to its nodal displacements
$\left\{ \overline{Q} \right\}$	column vector of the nonconservative generalized forces
$\left\{ Q(\tau) \right\}$	column vector of time dependent applied forces at time $t = \tau$
$\left\{ \overline{Q}_c \right\}$	column vector of applied forces invariant with time
$\left\{ q \right\}$	column vector of displacement functions $u(x, y)$ and $v(x, y)$ for an element
$\left\{ q(x, y, t) \right\}$	column vector of displacement functions $u(x, y, t)$ and $v(x, y, t)$
$\left[R \right]$	coordinate transformation matrix
S_i^e	nodal force of an element in the ith direction in an element coordinate system
S_i^g	nodal force of an element in the ith direction in a global coordinate system
$\left\{ S \right\}$	column vector of nodal forces
$\left\{ S' \right\}$	column vector of forces at the central node of the 4-CSTR element

$\{S^e\}$	column vector of nodal forces in the element coordinate system
$\{S^g\}$	column vector of nodal forces in the global coordinate system
T	central processing unit (CPU) computation time in milliseconds
$\overline{T}$	kinetic energy of the system
$\overline{T}_e$	kinetic energy of the element
t	independent variable (time)
Δt	a time interval
t_e	thickness of the element
u_i	displacement of the ith node along the x-direction in the element axis system
$u(x, y)$	displacement function for the displacement along the x^e-direction in the element coordinate system.
$\overline{u}(x, y, t)$	displacement function for the displacement along the x-direction
V	volume of an element
$\overline{v}$	potential energy of the system
$\overline{v}_e$	potential energy of the element
v_i	displacement of the ith node along the y^e-direction in the element axis system
$v(x, y)$	displacement function for the displacement along the y^e-direction in the element coordinate system
$\overline{v}(x, y, t)$	displacement function for the displacement along the y-direction
w	uniformly distributed load per unit length of span
X	an eigenvector
$\overline{x}$	x-coordinate of a point at which dynamic load $\overline{P}$ acts
x and y	coordinates in a general (structure) coordinate system
x^e and y^e	coordinates in the element coordinate system

x^g and y^g	coordinates in the global coordinate system
$\{\beta\}$	column vector of displacement parameters for an element
Δ_{DYN}	deflection due to the dynamic load
Δ_{ST}	deflection due to static loads
$\{\delta\}$	column vector of nodal displacements of an element
$\{\delta'\}$	column vector of displacements at the central node of 4-CSTR element
$\{\delta^e\}$	column vector of nodal displacements in the element coordinate system
$\{\delta^g\}$	column vector of nodal displacements in the global coordinate system
$\{\bar{\delta}\}$	column vector of virtual nodal displacements
$\{\bar{\delta}_{(t)}\}$ or $\{\bar{\delta}_i\}$	column vector of generalized coordinates representing the displacements at the nodal points
$\{\epsilon\}$	column vector of strains within an element
$\{\bar{\epsilon}\}$	column vector of virtual strains within an element
$\bar{\lambda}$	an eigenvalue
μ	Poisson's ratio of the material
ξ_1	$\frac{1}{2}(1 - \mu)$
ξ_2	$\frac{1}{2}(1 + \mu)$
$\{\sigma\}$	column vector of stresses within an element
$\{\Phi_i\}$	normal modes of vibration of the system
φ	angle between global and element coordinate system measured counterclockwise from the global system

$\psi_i(x, y)$	set of displacement functions independent of time for an element
ω	natural circular frequencies of the system
$\{\ \}$	column vector
$\{\ \}^T$	transpose of a column vector
$[\ ]^T$	transpose of a matrix
2-CSTR	two constant strain triangle rectangular elements
4-CSTR	four constant strain triangle rectangular elements

CHAPTER I

INTRODUCTION

1.1 Historical Background of Finite Element Method

The concept of the finite element method dates back to 1941, when Hrennikoff (28) introduced the concept of substituting a framework of bars for two-dimensional elasticity problems, such as bending of plates and bending of cylindrical shells, and proposed similar substitution of a three-dimensional framework of bars for a solid continuum.

In 1943, Courant (12) presented an approximate solution of St. Venant's torsion problem. The problem was formulated by the principle of minimum potential energy assuming a linear distribution of the warping function in each of the assemblages of triangular elements.

In 1947, Prager and Synge (61)(70) provided further insight into approximate solutions of boundary value problems by geometric representation in function space. The procedure often is called the Hypercircle method and was applied to the finite element idealization of solid continua.

In 1953, Levy (40) introduced the idea of replacing the continuous structure by pieces, generating a stiffness matrix for each element, and summing the stiffnesses. Thus, the use of discrete techniques was based mainly on intuition and common sense rather than on a systematic theoretical development. A more rigorous basis for the discrete analysis was first published in 1954 by Argyris (3) and in 1956 by Turner, et al. (74).

Obviously, due to the lack of high speed and large storage computers at that time, problems of only limited size and complexity could be solved. The popularity of the finite element approach started to increase exponentially during the sixties due to the availability of digital computers.

The development of the matrix formulation of the transformation theory of structures, after the fundamental work of Argyris, was an important step. The clear and elegant matrix representation not only shed light on formulation of the solution methods, but provided a powerful way of organizing the automatic computation as well. Introduction of the first two-dimensional, compatible displacement finite element, the constant strain triangle (74), provided a means of analyzing arbitrary plane stress and plane strain problems. But it did not take long to recognize that the basic characteristic of a displacement-consistent finite element was the assumed displacement function. Application of the basic property to rectangular plates (45), shells of revolutions, axisymmetric bodies, and general three-dimensional continua (23) (35) (69) was also found to be successful.

The formulation of stiffness matrices and the application of displacement methods were found to be ideally suited for this type of displacement mode analysis. While a theory to establish necessary and sufficient conditions for convergence to the true solution was lacking, the derivation of the load-displacement equations was shown to be equivalent to a piecewise Rayleigh-Ritz procedure applied to the variational principle of minimum potential energy. The basic conditions for the selection of the displacement function, continuity and completeness, were outlined by Irons and Draper (32). A method of comparison and evaluation of the stiffness matrices was introduced by Khanna and Hooley (37), and a comparative review and interpretation of the basis of finite element methods was presented by Pian and Tong (60) in 1969. Now, with a well-established basis for the selection of higher order displacement functions, a systematic development of suitable finite elements is possible.

Fraeijs de Veubeke (16) has demonstrated the application of equilibrium elements formulated by means of stress assumptions on the minimum complementary energy principle to obtain upper bounds on the influence coefficients. Hybrid models, formulated by partial assumptions of displacement and stress functions, as illustrated by Pian and Tong (57) (58) are also extremely useful for many problems. Finally, it can be pointed out that the application of the finite element technique is not restricted only to the solution of structural problems; Zienkiewicz and Cheung (84) have applied this procedure to solve problems in heat conduction and seepage flow.

1.2 Dynamic Analysis by Finite Elements

For years, in all branches of engineering, many engineers and analysts were occupied with the problems of structural dynamics. Interest in the analysis of vibration and general dynamic behavior of complex structures increased greatly during recent years. For well over a century, methods of analysis applicable to simple structural elements were known, and these methods were applied to idealized models of complex structures. The literature contains numerous publications on the flexural vibration of beams, including such problems as the approximate determination of natural frequencies for non-linear beams, the vibration of beams continuous over one or more supports, the vibration of two- or three-dimensional frames built up from beams, and the vibration of beams with attached masses, springs, and dashpots. Examples are found in the analyses of multi-story building structures and high aspect-ratio airplane wings in which these structures are dealt with ideally as beams (6) (7). The results of these analyses are quite dependable and useful so long as the bases for idealizations are valid (6) (7).

During more recent years, methods were devised by which the behavior of a structure may be predicted in terms of the properties of its elements (26) (29) (30) (39). These methods of analysis involve, explicitly, every element in the structure. The effects of changes in these elements on the behavior of the entire structure can be determined by the application of these methods.

Since the present investigation is ultimately aimed towards analyzing a plane stress problem of a beam, one of the procedures demonstrated here leads to the formulation of an eigenvalue problem in terms of the generalized coordinates and generalized inertia and stiffness coefficients. Once this problem is formulated, a direct solution is obtained by one of the several possible techniques that is well known. An approximation to the true mode is obtained by superposition of a finite number of these modes.

For the problem of a large magnitude, the number of finite elements increases very rapidly and it becomes prohibitive to use the mode-superposition technique. Therefore, this study also illustrates a step-by-step numerical integration technique. With this procedure, the equation of motion is directly integrated for an assumed variation of acceleration.

1.3 Purpose and Scope of Present Investigation

The purpose of the study is to evaluate and demonstrate the feasibility of applying the finite element method to the analysis of beams subjected to impact load. A portion of the report presents the investigation of convergence characteristics by analyzing several examples for seven configurations of idealization using five different finite elements. The suitability of the type of finite element and the configuration of idealization is determined on the basis of the needed computer time and the convergence to the closed-form solution.

The remainder of the report deals with a comparative study of the application of modal analysis and step-by-step numerical integration methods to the analysis of beams subjected to impact load. This illustrates the suitability of the methods to linear plane stress problems. A unique feature of the step-by-step integration technique permits the use of accelerations which vary linearly during each time interval without the need for the iterative operations required by other procedures.

Specific examples and the included computer programs demonstrate the methods with regard to relative computer storage requirements, accuracy, and time required for computation.

CHAPTER II

STEPS TO STRUCTURAL ANALYSIS

2.1 Approximations and Errors

Prior to using an approximate method for the solution of a structural analysis problem, one must have a clear concept of approximations and their validity. Three types of errors can be associated with approximate solutions of structural problems. These can be classified as idealization, discretization, and manipulation errors.

Idealization errors are those which are involved in formulating a mathematical model of the structure, for example, using (a) a flat surface for a curved surface, (b) a constant depth for a varying depth, or (c) pinned joints for partially restrained joints.

Discretization errors are those which are associated with replacing a continuous structure by one composed of finite elements. Discretization errors vanish as the size of the elements tends to zero. Bounding theorems are applicable to defining limits of the discretization error. In defining bounds, both the minimum potential energy and the minimum complementary energy approximations are needed (47) (60).

Manipulation errors occur in the process of computation. These include round-off, truncation, and arithmetic errors incurred in performing the calculations.

2.2 Idealization of Structure

The process of formulating a discrete element model of a structural continuum is called structural idealization. Thus, a system of an infinite number of degrees of freedom can be replaced by a system of a finite number of degrees of freedom. This leads to the well-known and powerful approach of the matrix method of structural analysis.

Generally, in structural analysis, two types of elements are chosen: (1) line elements, and (2) finite elements. One-dimensional members are line elements and are usually represented by the centroidal line of the members. Since these elements are attached to neighbors at single points, their element stiffnesses can be easily derived. Structural systems such as plates, slabs, and shells are examples of the continuous-type systems which cannot be idealized as line elements. For these systems, finite elements are discrete elements which are obtained by line-cuts. Such cutting removes the real, continuous edge connections between the elements. After cutting, these elements are connected to the neighboring elements at the nodes. Clearly, the inter-element forces at these artificial joints or nodes do not exist in reality; therefore, further fictitious stress and displacement patterns over the element field must be introduced which can be related to the node forces or displacements. This causes the discrete element to become an approximation to the original structure. The application of these finite elements for the discretization and analysis of a structural continuum is known as the finite element method. The last step in the structural idealization requires that the internal and external forces be transformed into a statically equivalent set of concentrated forces acting at the appropriate nodal points.

2.3 Selection of Finite Elements

In the selection of finite elements, the first important step is to determine the necessary features of the structure and then to explore the corresponding structural idealizations. The specification for the research reported herein was for a planar beam, with linear material properties, to be loaded by lateral impact load.

The force system is in the plane of the beam and includes no forces normal to that plane; consequently, the analysis becomes a two-dimensional plane stress problem. The choice of a finite element for idealization is thus immediately simplified. Whereas triangular finite elements have some advantages when applied to irregularly bounded regions, their combination into quadrilateral finite elements has significant advantages. Reduction of four constant strain triangular elements into one quadrilateral element by condensation of the central node reduces the computational effort and mesh details. The quadrilateral finite elements composed of these sub-triangles, along with triangular elements at the boundary, as needed, still maintain a capability for handling irregular boundaries, depending on the sequence used in the nodal description.

It is shown in this report that the linear strain rectangular element is better suited for the plane stress problems under consideration than is the constant strain triangle rectangular element. For problems with irregular boundaries, the constant strain triangular elements can still be used, together with the linear strain rectangular elements, to satisfy boundary shapes. The details of the mathematical formulations for deriving various element stiffness matrices are given in Appendix A.

2.4 The Displacement Method Analysis

The system of discrete elements, produced from an idealization of a structure, is highly indeterminate. The total stiffness matrix of the entire structure is constructed by evaluating and assembling the stiffness matrix of each element of the structure. This assemblage produces a system of linear equations which relate nodal forces to nodal displacements. This system of equations has a size determined by the total number of degrees of freedom of the idealized structure. Until the advent of the electronic digital computer, the solution of a very large system of equations was difficult, and this approach to structural analysis was impractical. In present methods, the displacement modes are used to evaluate element stiffnesses, which are then combined in a direct stiffness procedure to give compacted arrays of equations that are finally solved by taking advantage of their banded nature. Thus, it is a versatile and powerful method of analysis of complex structures. This procedure can be outlined for the static and elastic analysis of an idealized structure according to the following basic steps:

1. Assume displacement functions for the element.

2. Derive the element stiffness matrix $\left[k \right]$ in the element coordinate system.

3. Assemble the stiffness matrix for the entire structure in the global coordinate system.

4. Solve the equilibrium equations for nodal displacements.

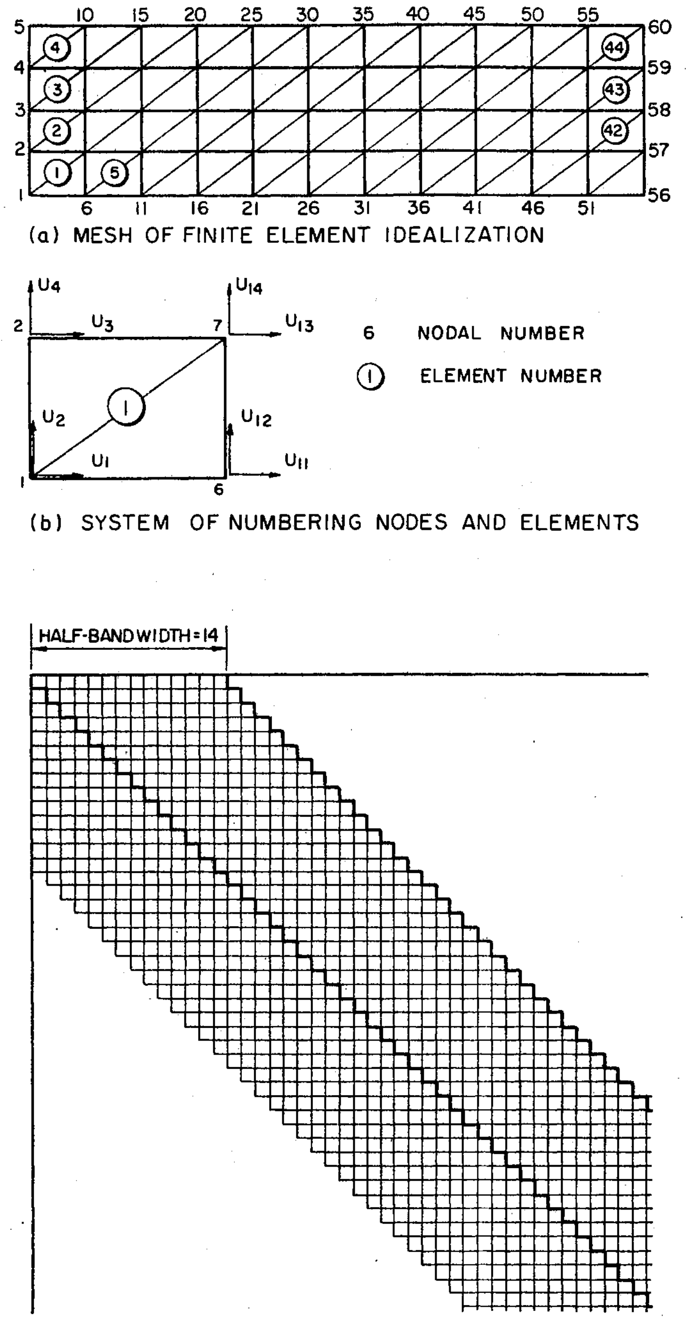
5. Compute the element stresses or the nodal forces in the desired coordinate system using the element stiffness matrix.

2.5 System for Numbering Nodes

The following properties of the structure stiffness matrix have some special advantages in the process of solving the equilibrium equations: (a) banded; (b) symmetric about the diagonal; (c) diagonally dominant. Figures 1 and 2 show that orderly numbering systems for nodes and elements produce a concentration of non-zero coefficients in a diagonal band. If "D" is the greatest difference between node numbers for any element and "n" is the number of degrees of freedom per node, the half-band width is (D + 1)n. Since the matrix is symmetrical and banded, this system of numbering nodes gives another advantage in storage processes and in writing computing algorithms..

2.6 Element Stiffness Matrices

The stiffness coefficient is defined as the force per unit displacement. The coefficients must be calculated for each element in order to construct the stiffness matrix of the entire structure. Several methods are used for determining these element stiffnesses. Most of these methods generally fall under energy methods of analysis. Approaches such as variational methods, or virtual work theorems, are widely used to derive relations between nodal forces and nodal displacements. The stiffness matrix of an element of any shape, form, or material properties may be evaluated by assuming a displacement function for the element and using a standard derivation procedure. The basic aim in the selection of these functions is to achieve compatibility of displacements between the boundaries of elements. However, in addition, the number of displacement patterns chosen must agree with the number of degrees of freedom of displacements of the element. A function of a large



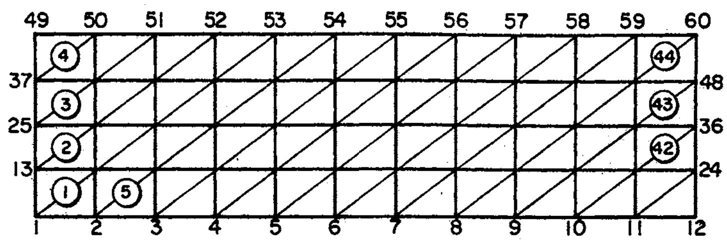
(a) MESH OF FINITE ELEMENT IDEALIZATION

(b) SYSTEM OF NUMBERING NODES AND ELEMENTS

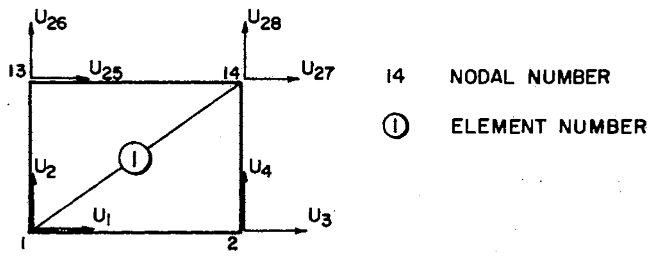
LARGEST DIFFERENCE BETWEEN TWO NODAL NUMBERS = 6
HALF-BAND WIDTH = 2 X (6+1) = 14

(c) STIFFNESS MATRIX

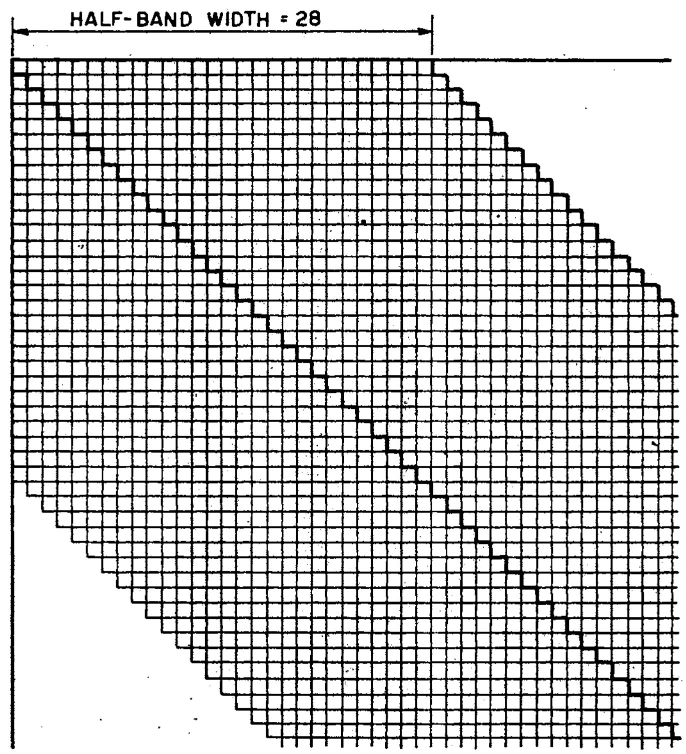
Figure 1. Band Width of Stiffness Matrix for System 1



(a) MESH OF FINITE ELEMENT IDEALIZATION



(b) SYSTEM OF NUMBERING NODES AND ELEMENTS



LARGEST DIFFERENCE BETWEEN TWO NODAL NUMBERS=13

HALF-BAND WIDTH = 2 X (13+1) = 28

(c) STIFFNESS MATRIX

Figure 2. Band Width of Stiffness Matrix
for System 2

number of displacement patterns may be assumed and reduced to the number of degrees of freedom by a Rayleigh-Ritz type of process (34) (56) (60), but this does not necessarily lead to improvement in the element stiffness properties.

2.7 Procedure to Derive Finite Element Stiffness Matrix

A systematic application of this procedure is illustrated in Appendix A.

1. To define displacements at any point within a plane stress element, the displacement functions u(x, y) and v(x, y) can be assumed for a set of two orthogonal directions. These displacement functions may be expressed using the arbitrary constant coefficients:

$$\left\{\beta\right\} = \left\{\beta_1, \beta_2, \beta_3, \ldots\right\} . \qquad (2.1a)$$

The column vector of constants, $\left\{\beta\right\}$, is the vector of generalized displacements at the boundary nodes where the displacement compatibility with neighboring nodes is required. The number of elements in the vector $\left\{\beta\right\}$ is equal to the number of the generalized displacements at the boundary nodes (refer to Equations A.2a and A.2b).

2. The displacement functions for the element can be written in matrix form as

$$\left\{q\right\} = \begin{Bmatrix} u(x, y) \\ v(x, y) \end{Bmatrix} = \left[L(x, y)\right] \left\{\beta\right\} \qquad (2.1b)$$

where $\left\{q\right\}$ is the column vector of the displacement functions u(x, y) and v(x, y), and $\left[L(x, y)\right]$ is the rectangular matrix whose elements are functions of coordinates x and y (refer to Equation A.2c).

3. Nodal displacements $\left\{\delta\right\}$ can be evaluated in terms of arbitrary constants $\left\{\beta\right\}$ by substituting the coordinates of the nodes in the

displacement functions u(x, y) and v(x, y) in Equation (2. 1b). Thus, the nodal displacements can be written in matrix form as

$$\left\{\delta\right\} = \left\{\begin{matrix} u_i \\ v_i \end{matrix}\right\}_{i = 1, 2, \ldots n} = \left[A\right] \left\{\beta\right\} \qquad (2, 1c)$$

where n is the number of nodes in the plane stress element, and the elements of the square matrix $\left[A\right]$ are constants.

4. The column vector of arbitrary constants $\left\{\beta\right\}$ can be evaluated in terms of nodal displacements $\left\{\delta\right\}$ from Equation (2, 1c), as

$$\left\{\beta\right\} = \left[A\right]^{-1} \left\{\delta\right\} . \qquad (2, 1d)$$

5. The column vector of unknown arbitrary constants $\left\{\beta\right\}$ can be eliminated by substituting Equation (2, 1d) into Equation (2. 1b) to evaluate displacement functions $\left\{q\right\}$ in terms of nodal displacements $\left\{\delta\right\}$:

$$\left\{q\right\} = \left[L(x, y)\right] \left[A\right]^{-1} \left\{\delta\right\} = \left[C(x, y)\right] \left\{\delta\right\} \qquad (2, 1e)$$

where the elements of the rectangular matrix $\left[C(x, y)\right]$ are functions of the coordinates x and y.

6. Internal strains $\left\{\epsilon\right\}$ in the element can be evaluated in terms of nodal displacements $\left\{\delta\right\}$ by using displacement functions from Equation (2. 1b) and the column vector of arbitrary constants $\left\{\beta\right\}$ from Equation (2, 1d). Thus,

$$\left\{\epsilon\right\} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} = \left[B(x, y)\right] \left\{\beta\right\}$$

$$= \left[B(x, y)\right] \left[A\right]^{-1} \left\{\delta\right\} \qquad (2, 1f)$$

7. Internal stresses $\{\sigma\}$ in the element can be obtained in terms of nodal displacements $\{\delta\}$ by using stress-strain relations and substituting strains $\{\epsilon\}$ from Equation (2.1f). Thus,

$$\{\sigma\} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = [D]\{\epsilon\} = [D][B(x,y)][A]^{-1}\{\delta\} \qquad (2.1g)$$

where $[D]$ is the matrix of elastic constants.

8. If $\{\bar{\epsilon}\}$ are the virtual strains in the element caused by the nodal virtual displacements $\{\bar{\delta}\}$, internal virtual work can be written by using Equations (2.1f) and (2.1g), as:

$$\int_V \{\bar{\epsilon}\}^T \{\sigma\} \, dV = \int_V \{\bar{\delta}\}^T [A^{-1}]^T [B(x,y)]^T [D] [B(x,y)] [A]^{-1} \{\delta\} \, dV$$

$$= \{\bar{\delta}\}^T [A^{-1}]^T \int_V [B(x,y)]^T [D] [B(x,y)] \, dV [A]^{-1} \{\delta\}. \qquad (2.1h)$$

9. If $\{S\}$ are the nodal forces and $[k]$ is the element stiffness matrix, the relation of the nodal forces to nodal displacements can be given by

$$\{S\} = [k]\{\delta\}. \qquad (2.1i)$$

10. For virtual nodal displacements $\{\bar{\delta}\}$, the external virtual work by nodal forces can be written by using Equation (2.1i), as

$$\{\bar{\delta}\}^T \{S\} = \{\bar{\delta}\}^T [k] \{\delta\}. \qquad (2.1j)$$

11. Equating internal work and external work of Equations (2.1h) and (2.1j), the element stiffness matrix can be obtained as

$$\left[k\right] = \left[A^{-1}\right]^T \int_V \left[B(x,y)\right]^T \left[D\right] \left[B(x,y)\right] \, dV \left[A\right]^{-1} . \qquad (2.1k)$$

2.8 Selection of Displacement Functions

A key to the derivation of a deformation-consistent stiffness matrix is the selection of a displacement function satisfying the following requirements (32) (47):

1. Must be continuous over the element. They need not have continuous derivatives.

2. Must maintain the continuity with displacements of adjacent elements. This can be accomplished in two ways:

   a. when nodal displacements are selected as generalized displacements, and

   b. when displacements along any side of the element are selected so that they depend only on the displacements at the nodes bounding the side.

3. Must be a linear function of the generalized displacements. This is necessary so that the force-displacement equations will be linear, i.e., independent of the position of the external reference system.

4. Must include rigid body displacement states. It is necessary to include the conditions of global static equilibrium; otherwise, self-straining would result from rigid body motions.

A check on whether or not these states are included can be made by showing that the forces in each column of the stiffness matrix satisfy the macroscopic equations of equilibrium for the element (32) (47).

5. All of these stipulations are independent of element geometry, material characteristics, smallness of strains, and displacements,

2.9 Stiffness Matrices of Special Elements

The stiffness matrices given in this section have been used for studying convergence behavior of the elements. These matrices have been obtained from the combination of stiffness matrices derived in Appendix A and as described below.

2.9.1 Rectangular Element No. 1

Using Equation (A. 13) and combining stiffness matrices of two triangular elements of Figures 3 and 4, the stiffness matrix for a rectangular element shown in Figure 5 can be obtained as indicated in Equation (2. 2), where

μ = Poisson's ratio of the material

A = length of the element along the x^e-direction

B = width of the element along the y^e-direction

$\xi_1 = \frac{1}{2}(1 - \mu)$

$\xi_2 = \frac{1}{2}(1 + \mu)$

E = modulus of elasticity

t_e = thickness of the element

S = nodal forces of the element.

The stresses within the element can be given by

$$\{\sigma\} = [Q]\{\delta\} \tag{2.3}$$

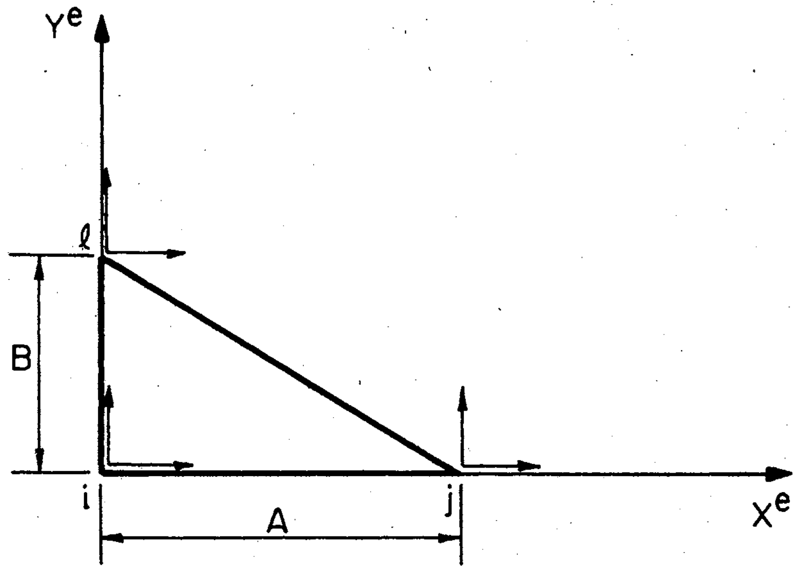


Figure 3. Constant Strain Triangular Element No. 1

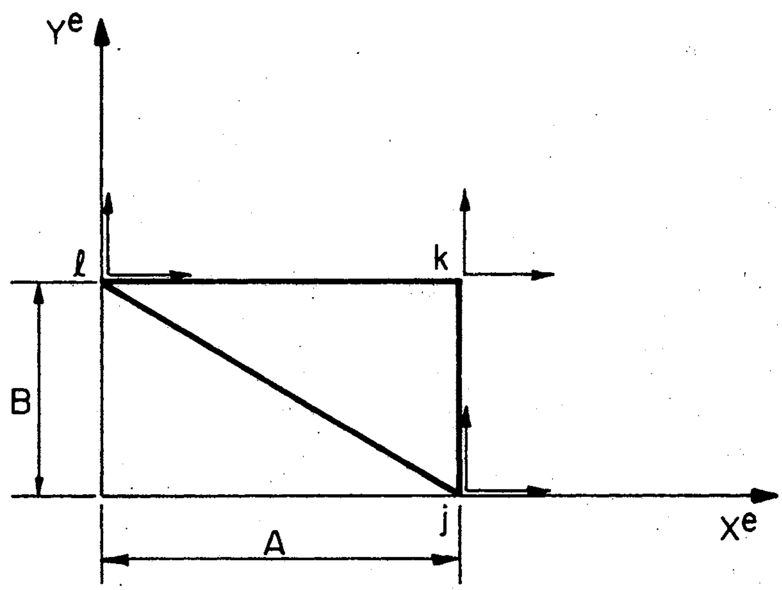


Figure 4. Constant Strain Triangular Element No. 2

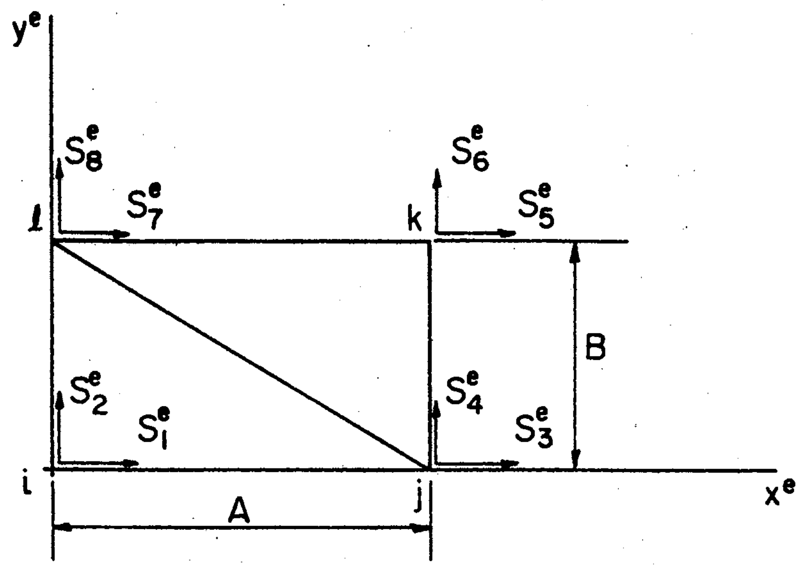


Figure 5. Rectangular Element No. 1

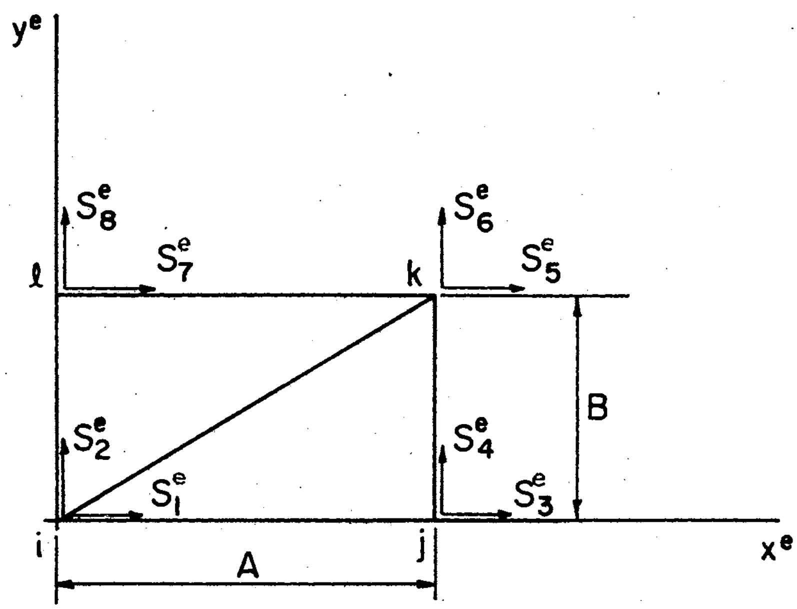


Figure 6. Rectangular Element No. 2

$$[k] = \frac{Et_e}{2(1-\mu^2)AB} \begin{bmatrix} \xi_1 A^2 + B^2 & & & & & & & \\ \xi_2 AB & A^2 + \xi_1 B^2 & & & & \text{Symmetric} & & \\ -B^2 & -\mu AB & \xi_1 A^2 + B^2 & & & & & \\ -\xi_1 AB & -\xi_1 B^2 & 0 & A^2 + \xi_1 B^2 & & & & \\ 0 & 0 & -\xi_1 A^2 & -\mu AB & \xi_1 A^2 + B^2 & & & \\ 0 & 0 & -\xi_1 AB & -A^2 & \xi_2 AB & A^2 + \xi_1 B^2 & & \\ -\xi_1 A^2 & -\xi_1 AB & 0 & \xi_2 AB & -B^2 & -\mu AB & \xi_1 A^2 + B^2 & \\ -\mu AB & -A^2 & \xi_2 AB & 0 & -\xi_1 AB & -\xi_1 B^2 & 0 & A^2 + \xi_1 B^2 \end{bmatrix} \tag{2.2}$$

where matrix $[Q]$ can be written as

$$\frac{E}{2(1-\mu^2)AB}\left[\begin{array}{c|c|c|c|c|c|c|c} -B & -\mu A & B & -\mu A & B & \mu A & -B & \mu A \\ \hline -\mu B & -A & \mu B & -A & \mu B & A & -\mu B & A \\ \hline -A\xi_1 & -B\xi_1 & -A\xi_1 & B\xi_1 & A\xi_1 & B\xi_1 & A\xi_1 & -B\xi_1 \end{array}\right] \qquad (2.4)$$

2.9.2 Rectangular Element No. 2

Using Equation (A. 13) and combining stiffness matrices of two triangular elements, the stiffness matrix for a rectangular element shown in Figure 6 can be obtained as indicated in Equation (2. 5).

The stresses within the element can be given by

$$\{\sigma\} = [Q]\{\delta\} \qquad (2.6)$$

where matrix $[Q]$ can be written as

$$\frac{E}{2(1-\mu^2)AB}\left[\begin{array}{c|c|c|c|c|c|c|c} -B & -\mu A & B & -\mu A & B & \mu A & -B & \mu A \\ \hline -\mu B & -A & \mu B & -A & \mu B & A & -\mu B & A \\ \hline -A\xi_1 & -B\xi_1 & -A\xi_1 & B\xi_1 & A\xi_1 & B\xi_1 & A\xi_1 & -B\xi_1 \end{array}\right] \qquad (2.7)$$

2.9.3 Rectangular Element No. 3

This element (Figure 7) was obtained by combining the two elements which are shown in Figures 5 and 6. The stiffness matrix, obtained by averaging the stiffnesses of the two elements, has improved stiffness characteristics. This combination will yield results of greater accuracy than the worst combination of the two triangular elements.

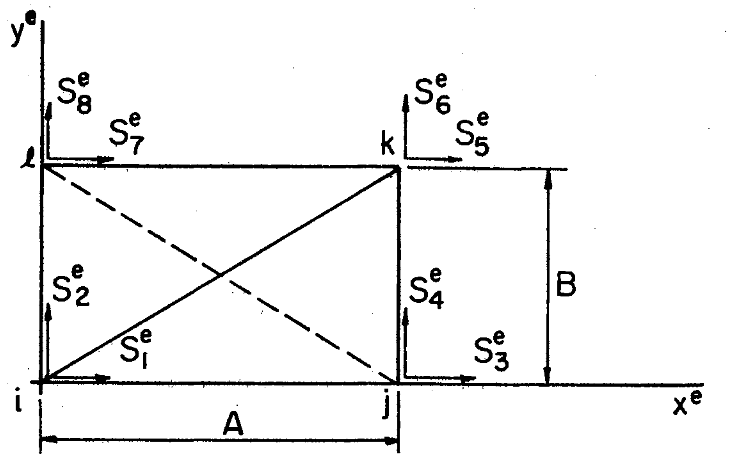


Figure 7. Rectangular Element No. 3

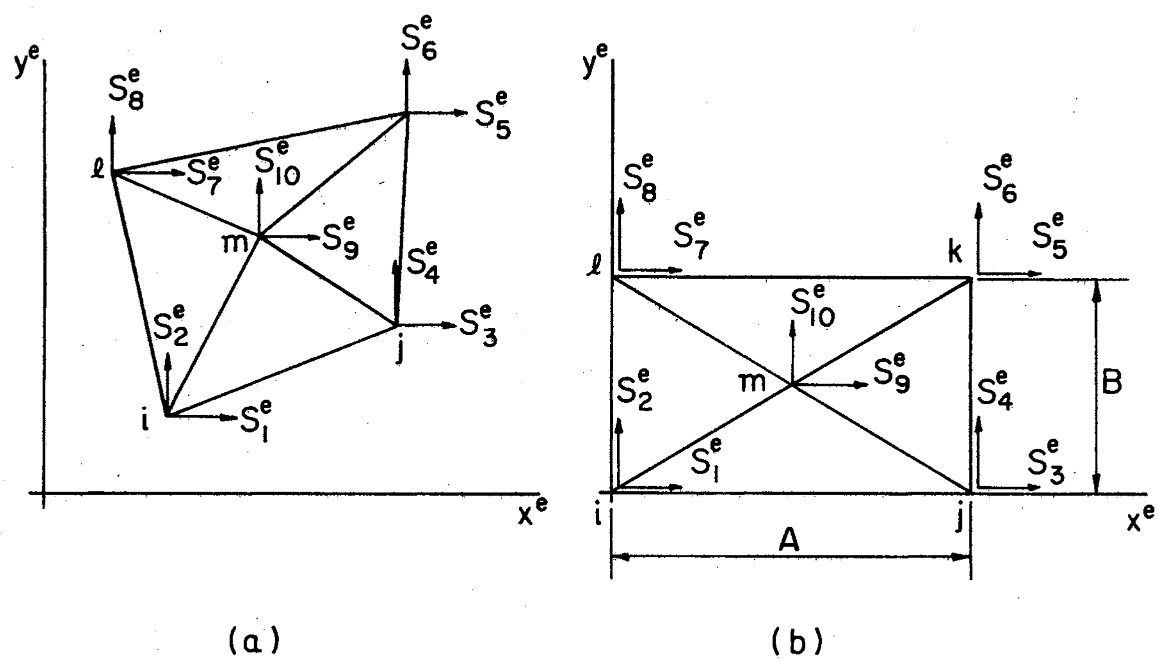


Figure 8. Rectangular Element No. 4

$$[k] = \frac{Et_e}{2(1-\mu^2)AB}\begin{bmatrix} \xi_1A^2+B^2 & & & & & & & \\ 0 & A^2+\xi_1B^2 & & & & \text{Symmetric} & & \\ -B^2 & \xi_1AB & \xi_1A^2+B^2 & & & & & \\ -\mu AB & -\xi_1B^2 & -\xi_2AB & A^2+\xi_1B^2 & & & & \\ 0 & -\xi_2AB & -\xi_1A^2 & \xi_1AB & \xi_1A^2+B^2 & & & \\ -\xi_2AB & 0 & \mu AB & -A^2 & 0 & A^2+\xi_1B^2 & & \\ -\xi_1A^2 & \mu AB & 0 & 0 & -B^2 & \xi_1AB & \xi_1A^2+B^2 & \\ \xi_1AB & -A^2 & 0 & 0 & \mu AB & -\xi_1B^2 & -\xi_2AB & A^2+\xi_1B^2 \end{bmatrix} \tag{2.5}$$

Thus, summing and averaging of the stiffness matrices of Equations (2. 2) and (2. 5) will result in Equation (2. 8).

2.9.4 Rectangular Element No. 4

To obtain the stiffness matrix for the quadrilateral element built of four triangular elements, as shown in Figure 8, the stiffness matrix $[k]$ of the triangular element, Equation (A. 13), can be used. By assembling the stiffness matrices of four triangles, a stiffness matrix of size 10 x 10 is obtained to represent the quadrilateral having five nodal points. If no load is applied at the central node, the matrix can be condensed to obtain an 8 x 8 stiffness matrix for the quadrilateral element with four nodes as follows:

$$\begin{bmatrix} S_{8x1} \\ S'_{2x1} \end{bmatrix}_{10x1} = \left[\begin{array}{c|c} k_{11_{8x8}} & k_{12_{8x2}} \\ \hline k_{21_{2x8}} & k_{22_{2x2}} \end{array}\right]_{10x10} \begin{bmatrix} \delta_{8x1} \\ \delta'_{2x1} \end{bmatrix}_{10x1} \tag{2.9}$$

$$\{S\}_{8x1} = [k_{11}]_{8x8} \{\delta\}_{8x1} + [k_{12}]_{8x2} \{\delta'\}_{2x1} \tag{2.10}$$

where

S_{8x1} = column vector of the nodal forces at the boundary nodes

S'_{2x1} = column vector of the nodal forces at the internal node

$k_{11_{8x8}}$ = square matrix of stiffness coefficient forces induced at the boundary nodes due to unit nodal displacements at the boundary nodes

$k_{12_{8x2}}$ = rectangular matrix of stiffness coefficient forces induced at the boundary nodes due to unit nodal displacements at the internal node

$$
[k] = \frac{Et_e}{2(1-\mu^2)AB}
\begin{bmatrix}
\xi_1 A^2 + B^2 & & & & & & & \\
\frac{1}{2}\xi_2 AB & A^2 + \xi_1 B^2 & & & & \text{Symmetric} & & \\
-B^2 & \frac{1}{2}(\xi_1 - \mu)AB & \xi_1 A^2 + B^2 & & & & & \\
\frac{1}{2}(-\xi_1+\mu)AB & -\xi_1 B^2 & -\frac{1}{2}\xi_2 AB & A^2 + \xi_1 B^2 & & & & \\
0 & -\frac{1}{2}\xi_2 AB & -\xi_1 A^2 & \frac{1}{2}(\xi_1-\mu)AB & \xi_1 A^2 + B^2 & & & \\
-\frac{1}{2}\xi_2 AB & 0 & \frac{1}{2}(-\xi_1+\mu)AB & -A^2 & \frac{1}{2}\xi_2 AB & A^2 + \xi_1 B^2 & & \\
-\xi_1 A^2 & \frac{1}{2}(-\xi_1+\mu)AB & 0 & \frac{1}{2}\xi_2 AB & -B^2 & \frac{1}{2}(\xi_1-\mu)AB & \xi_1 A^2 + B^2 & \\
\frac{1}{2}(\xi_1-\mu)AB & -A^2 & \frac{1}{2}\xi_2 AB & 0 & \frac{1}{2}(-\xi_1+\mu)AB & -\xi_1 B^2 & -\frac{1}{2}\xi_2 AB & A^2 + \xi_1 B^2
\end{bmatrix}
\tag{2.8}
$$

$k_{21_{2x8}}$ = rectangular matrix of stiffness coefficient forces induced at the internal node due to unit nodal displacements at the boundary nodes

$k_{22_{2x2}}$ = square matrix of stiffness coefficient forces induced at the internal node due to unit nodal displacements at the internal node

δ_{8x1} = column vector of nodal displacements at the boundary nodes

δ'_{2x1} = column vector of nodal displacements at the internal node.

Since no load is applied at node 'm', vector $\{S'\}$ is zero, and

$$\{\delta'\}_{2x1} = -\left[k_{22}\right]_{2x2}^{-1}\left[k_{21}\right]_{2x8}\{\delta\}_{8x1} \,. \qquad (2.11)$$

Substituting $\{\delta'\}$ from Equation (2. 11) into Equation (2. 10) gives

$$\{S\} = \left[k_{11}\right]\{\delta\} - \left[k_{12}\right]\left[k_{22}\right]^{-1}\left[k_{21}\right]\{\delta\}$$

or

$$\{S\} = \left[k_{11}\right] - \left[k_{12}\right]\left[k_{22}\right]^{-1}\left[k_{21}\right]\{\delta\} \,. \qquad (2.12)$$

Therefore,

$$\left[k\right] = \left[k_{11}\right] - \left[k_{12}\right]\left[k_{22}\right]^{-1}\left[k_{21}\right] . \qquad (2.13)$$

To eliminate the need for a matrix inversion subroutine in the computer program, the following procedure illustrated for Equation (2. 14) was used to obtain the stiffness matrix of Equation (2. 13) by taking the value of n equal to 9:

If $S_{n+1} = 0$, the $(n+1)^{th}$ row can be written as

$$0 = \sum_{i=1}^{n} C_{n+1,\,2}\,\delta_i + C_{n+1,\,n+1}\,\delta_{n+1} \qquad (2.15)$$

$$
\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_n \\ S_{n+1} \end{bmatrix}
=
\begin{bmatrix}
C_{11} & C_{21} & \cdots C_{1,n-3} \cdots & C_{1,n-2} & C_{1,n-1} & C_{1,n} & C_{1,n+1} \\
C_{21} & C_{22} & \cdots & C_{2,n-2} & C_{2,n-1} & C_{2,n} & C_{2,n+1} \\
C_{31} & \cdots & \cdots & \cdots & \cdots & C_{3,n} & C_{3,n+1} \\
\vdots & & & & & & \vdots \\
C_{n,1} & C_{n,2} & \cdots & \cdots & \cdots & \cdots & \\
C_{n+1,1} & C_{n+1,2} & C_{n+1,3} & \cdots & \cdots & \cdots & C_{n+1,n+1}
\end{bmatrix}
\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \vdots \\ \\ \delta_{n+1} \end{bmatrix}
\qquad (2.14)
$$

from which

$$\delta_{n+1} = - \frac{1}{C_{n+1,\, n+1}} \sum_{i=1}^{n} C_{n+1,\, 2}\, \delta_i \, . \qquad (2.16)$$

The expression δ_{n+1} can be eliminated from Equation (2.14) by substituting the value of δ_{n+1}; thus,

$$S_j = \sum_{i=1}^{n} \left(C_{j,\, 2} - \frac{C_{j,\, n+1} \cdot C_{n+1,\, i}}{C_{n+1,\, n+1}} \right) \delta_i \qquad (2.17)$$

or

$$S_j = \sum_{i=1}^{n} C'_{j,\, i}\, \delta_i \qquad (j = 1, 2, 3, \ldots n) \qquad (2.18)$$

where $C'_{j,\, i}$ are new elements of matrix (n x n).

Similarly, by repeating the procedure for the n^{th} row, the matrix of size (n x n) can be further reduced.

2.9.5 Rectangular Element No. 5

This is a linear strain rectangular element shown in Figure 9, and its stiffness matrix is given in Equation (A.20). The stresses within the element can be found by using Equations (A.21) and (A.22). For a point at the center of the element, that is, for $x = \frac{a}{2}$ and $y = \frac{b}{2}$ in Equation (A.22),

$$\left[Q \right] = \frac{E}{2(1-\mu^2)\, ab} \begin{bmatrix} -b & -\mu a & b & -\mu a & b & \mu a & -b & \mu a \\ -\mu b & -a & \mu b & -a & \mu b & b & -\mu b & a \\ -\xi_1 a & -\xi_1 b & -\xi_1 a & \xi_1 b & \xi_1 a & \xi_1 b & \xi_1 a & -\xi_1 b \end{bmatrix} \qquad (2.19)$$

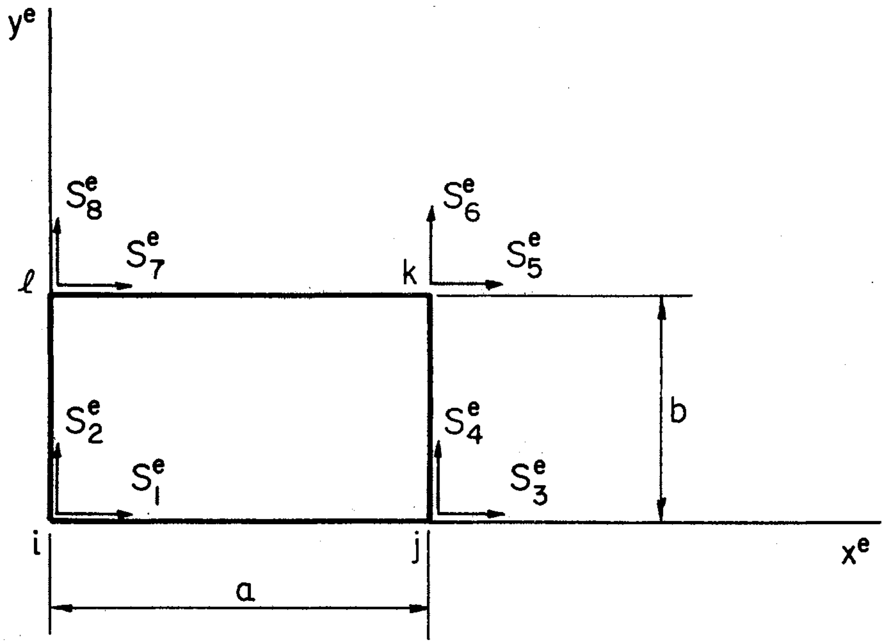


Figure 9. Rectangular Element No. 5

From the experience of the analysts, it was found that the elements with more complete displacement functions give accurate solutions much more efficiently, in terms of computation time, than do the simpler elements. However, such elements are not always easy to devise because of the geometric continuity requirement of the displacements at the element junctions (79).

CHAPTER III

EXAMPLES OF LINEAR ELASTIC PROBLEMS WITH STATIC LOADS AND THEIR RESULTS

3.1 Procedure

Several examples were selected to study the convergence characteristics, computation time requirements, and the suitability of the finite element for application to some plane stress problems of linear elasticity. To simplify the comparison with analytic solutions, only isotropic material was considered.

Five elements and seven configurations (as shown in Figures 10, 11, 17, and 18) were used in the examples. All meshes were processed by a digital computer program using four nodal point rectangles, which consist of two or four constant strain triangles, or a linear strain rectangle as basic elements. The computer programs are given in Appendix B. These programs generate the meshes, compute the stiffness matrices, assemble the total stiffness matrix, calculate the displacements at all nodal points and the stresses at the central point of the rectangular elements, and print out the computation time for each example.

For all seven configurations of idealization, an equal number of rectangular elements was taken for each set of points on the deflection convergence curve. This enables the curves to be directly comparable.

The elasticity and beam theory solutions are shown in Figures 12, 13, 14, 15, 16, 19, 20, 21, 22, and 23 to illustrate the convergence behavior.

3.2 Examples

3.2.1 Simply Supported Beam

This beam was analyzed for two kinds of loading: (a) a uniformly distributed load, and (b) a point load at midspan, as shown in Figures 10 and 11. Central nodes along the vertical edge at the ends were restrained against vertical displacement, and the central node at the midspan was restrained against horizontal displacement to maintain the symmetry of the problem.

A comparison of the midspan deflections and some internal stresses with elasticity and beam theory solutions is presented in Figures 12, 13, 14, 15, and 16. The values used as theoretical midspan deflections are:

(a) For a uniformly distributed load:

$$\Delta_{ST} = \frac{5wL^4}{384EI}\left[1 + \frac{24E}{25G}\left(\frac{2h}{L}\right)^2\right] = 1.664 \times 10^{-5} \text{ inches}$$

(Beam Theory) (3.1)

$$\Delta_{ST} = \frac{5wL^4}{384EI}\left[1 + \frac{6(8+5\mu)}{25}\left(\frac{2h}{L}\right)^2\right] = 1.658 \times 10^{-5} \text{ inches}$$

(Elasticity) (3.2)

(b) For a point load at midspan:

$$\Delta_{ST} = \frac{PL^3}{48EI}\left[1 + \frac{6E}{5G}\left(\frac{2h}{L}\right)^2\right] = 1.344 \times 10^{-5} \text{ inches}$$

(Beam Theory) (3.3)

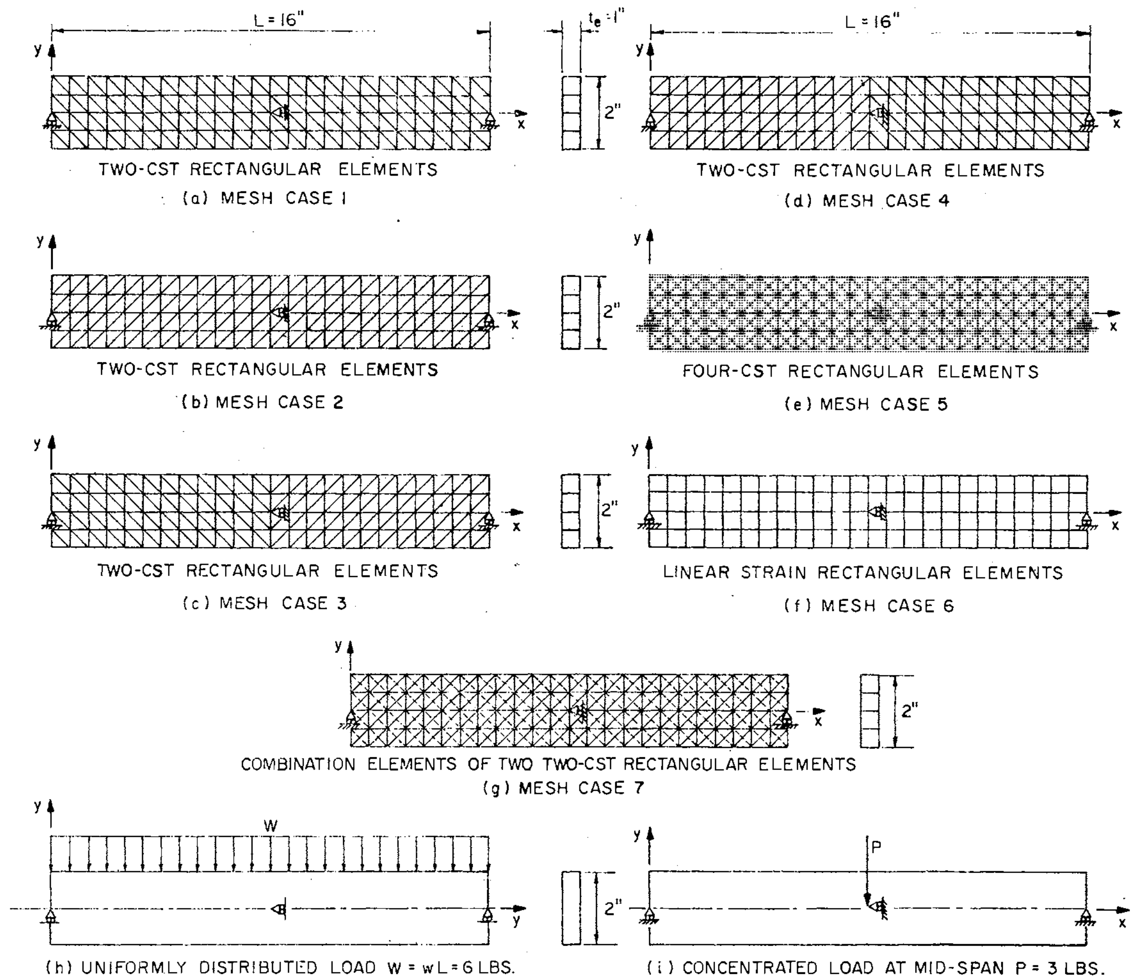


Figure 10. Simply Supported Beams and Their Finite Element Idealizations (NY = 4)

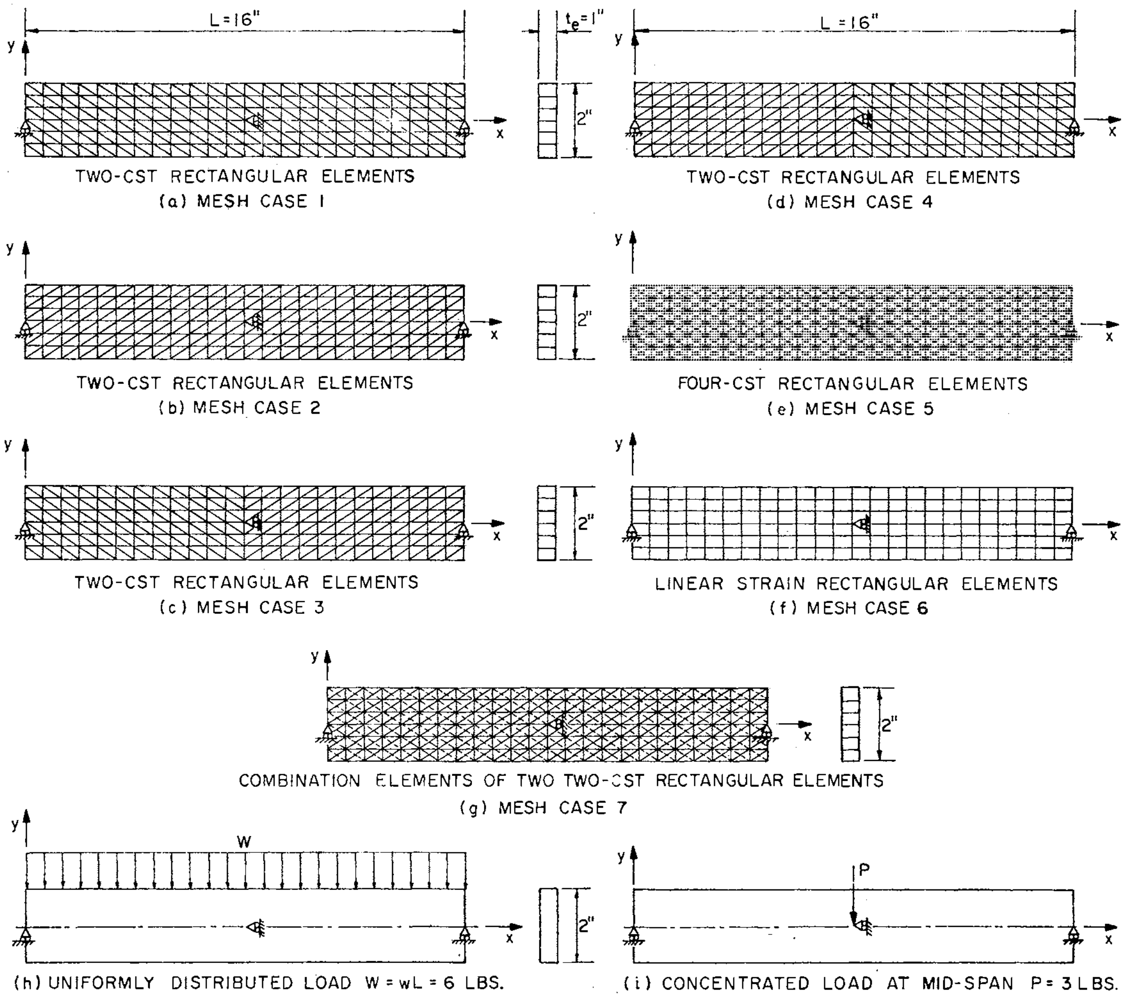


Figure 11. Simply Supported Beams and Their Finite Element Idealizations (NY = 6)

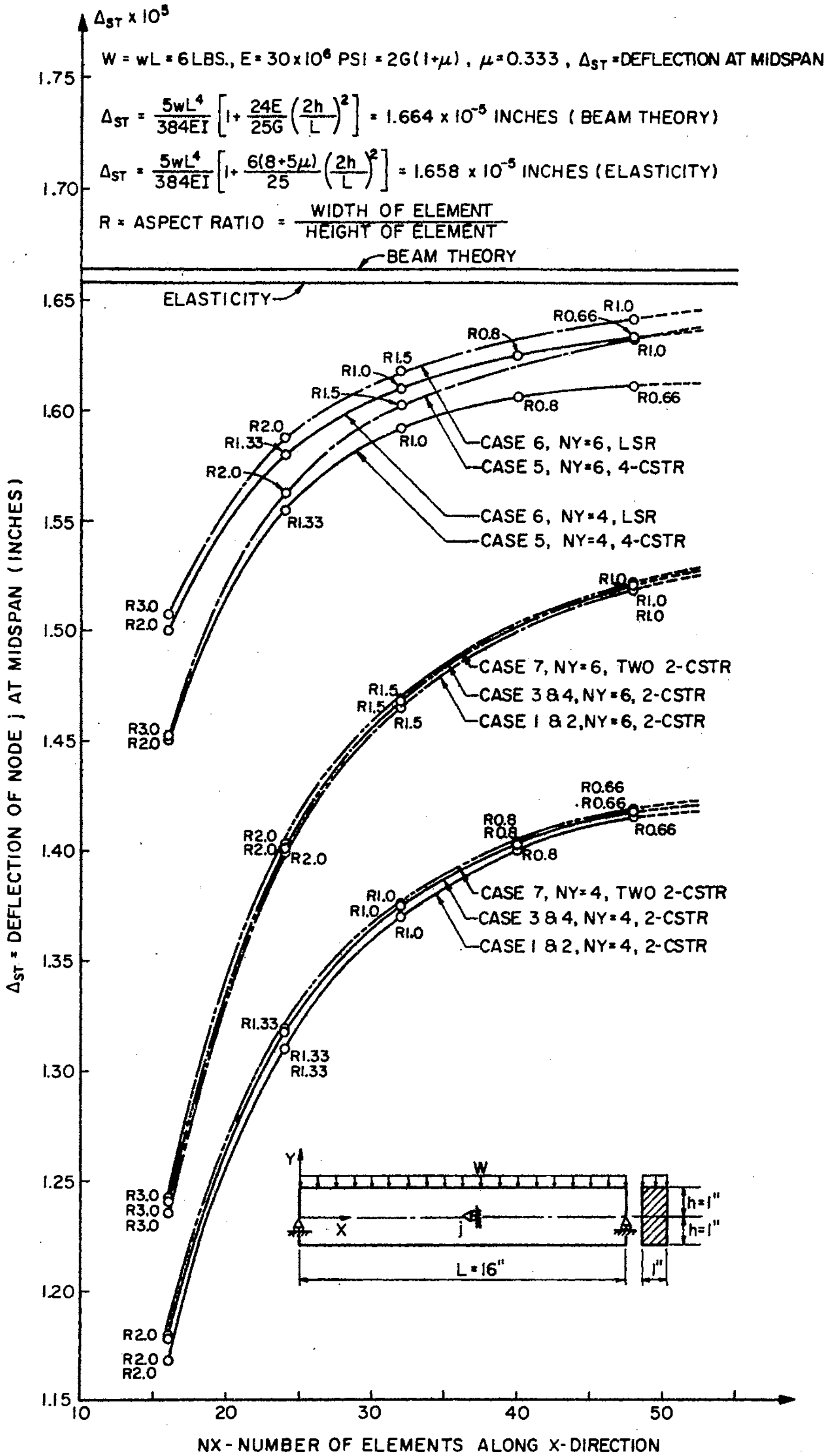


Figure 12. Displacement Convergence with Respect to Number of Elements Along X-Direction for a Simply Supported Beam with Distributed Load

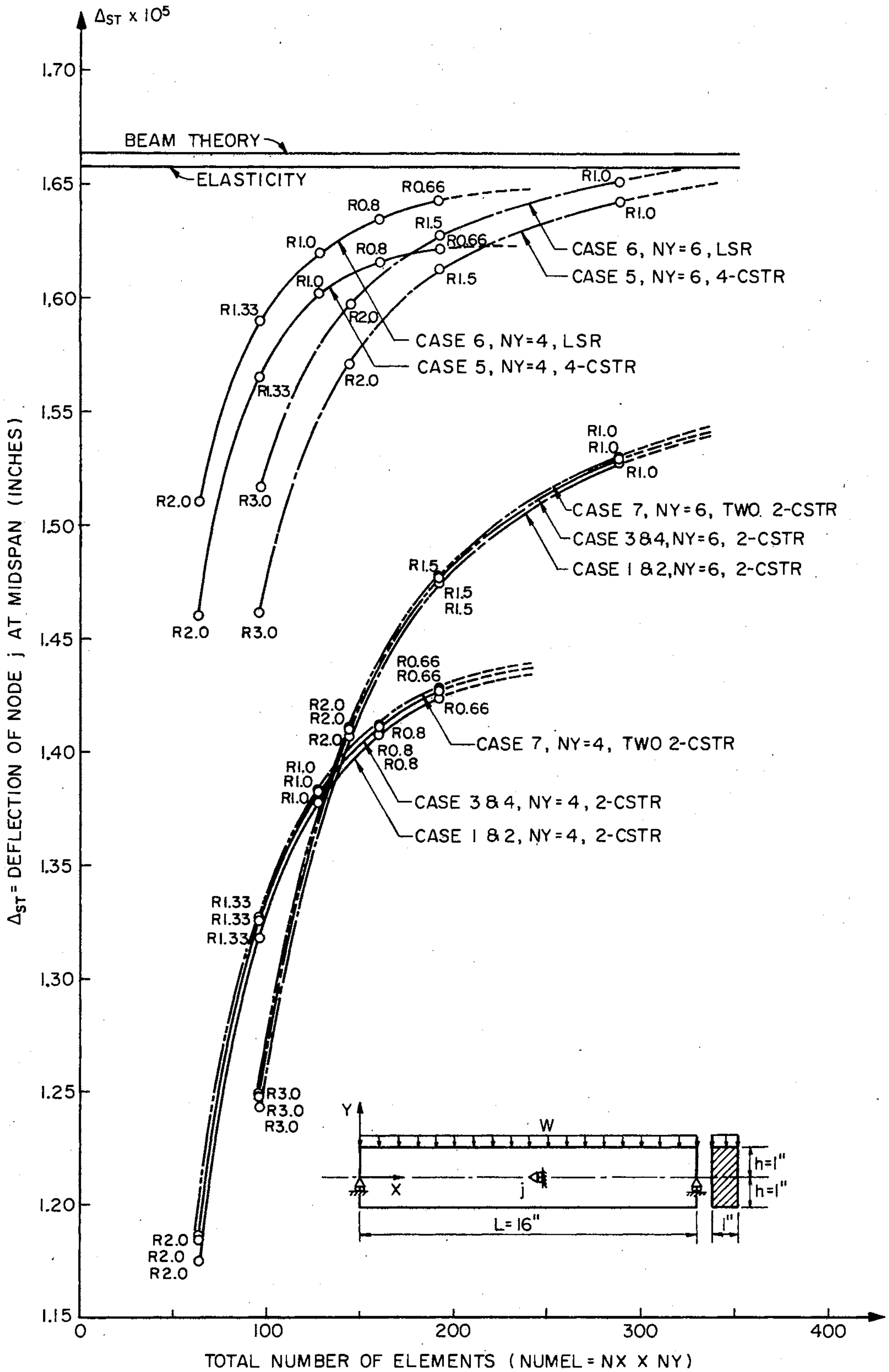


Figure 13. Displacement Convergence with Respect to Total Number of Elements for a Simply Supported Beam with Distributed Load

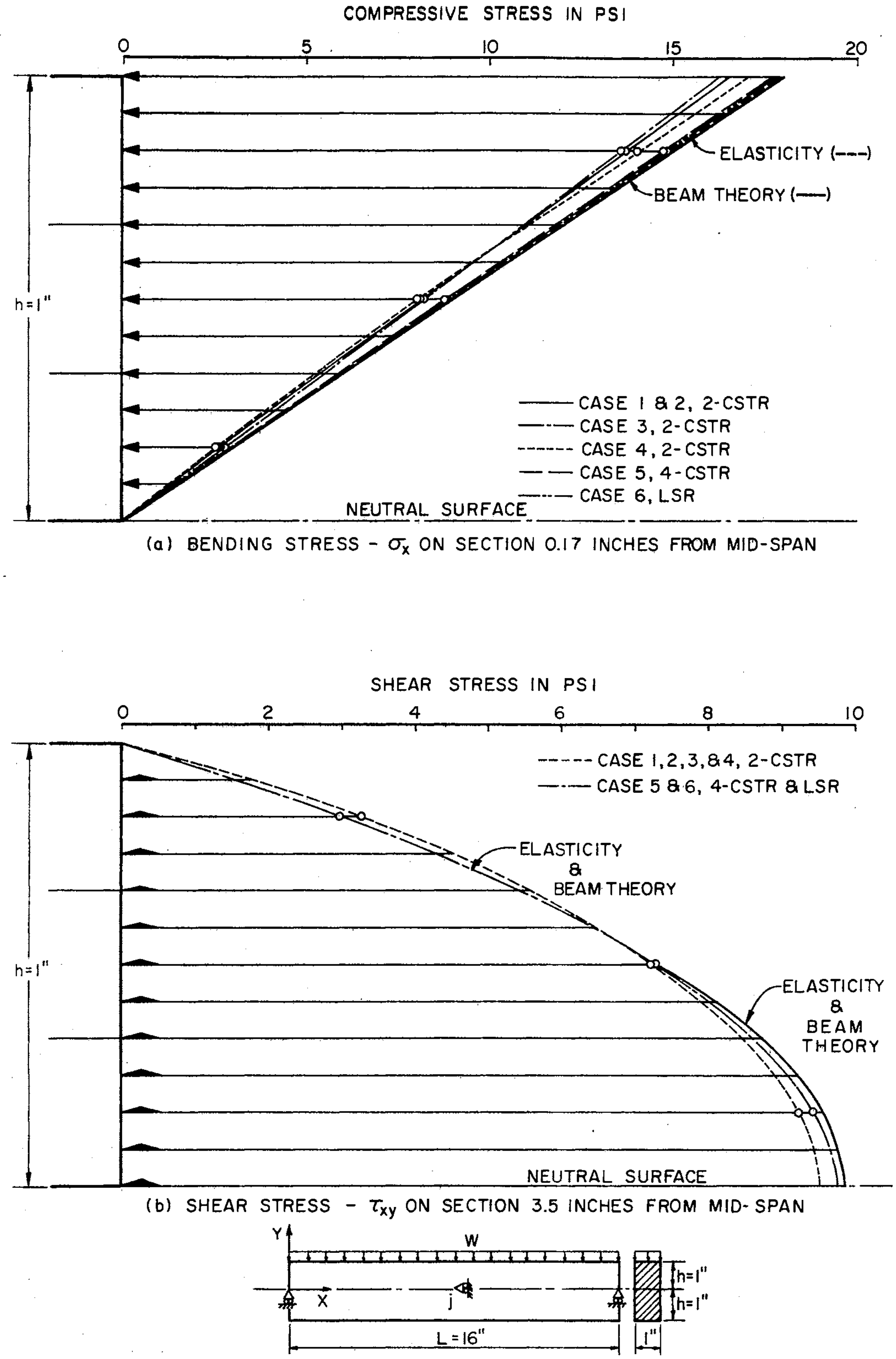


Figure 14. Bending and Shearing Stresses for a Simply Supported Beam with Distributed Load

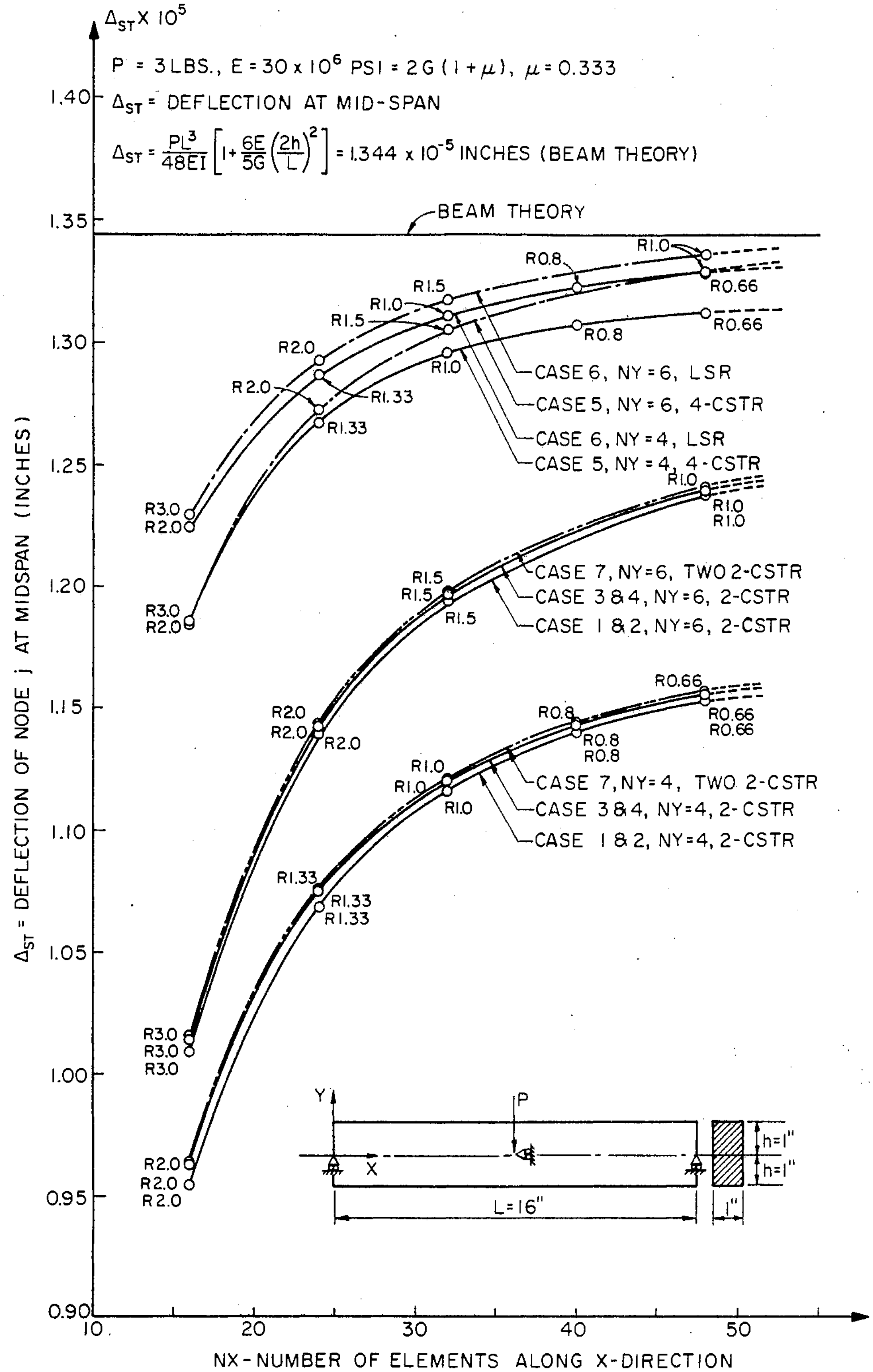


Figure 15. Displacement Convergence with Respect to Number of Elements Along X-Direction for a Simply Supported Beam with Point Load at Mid-Span

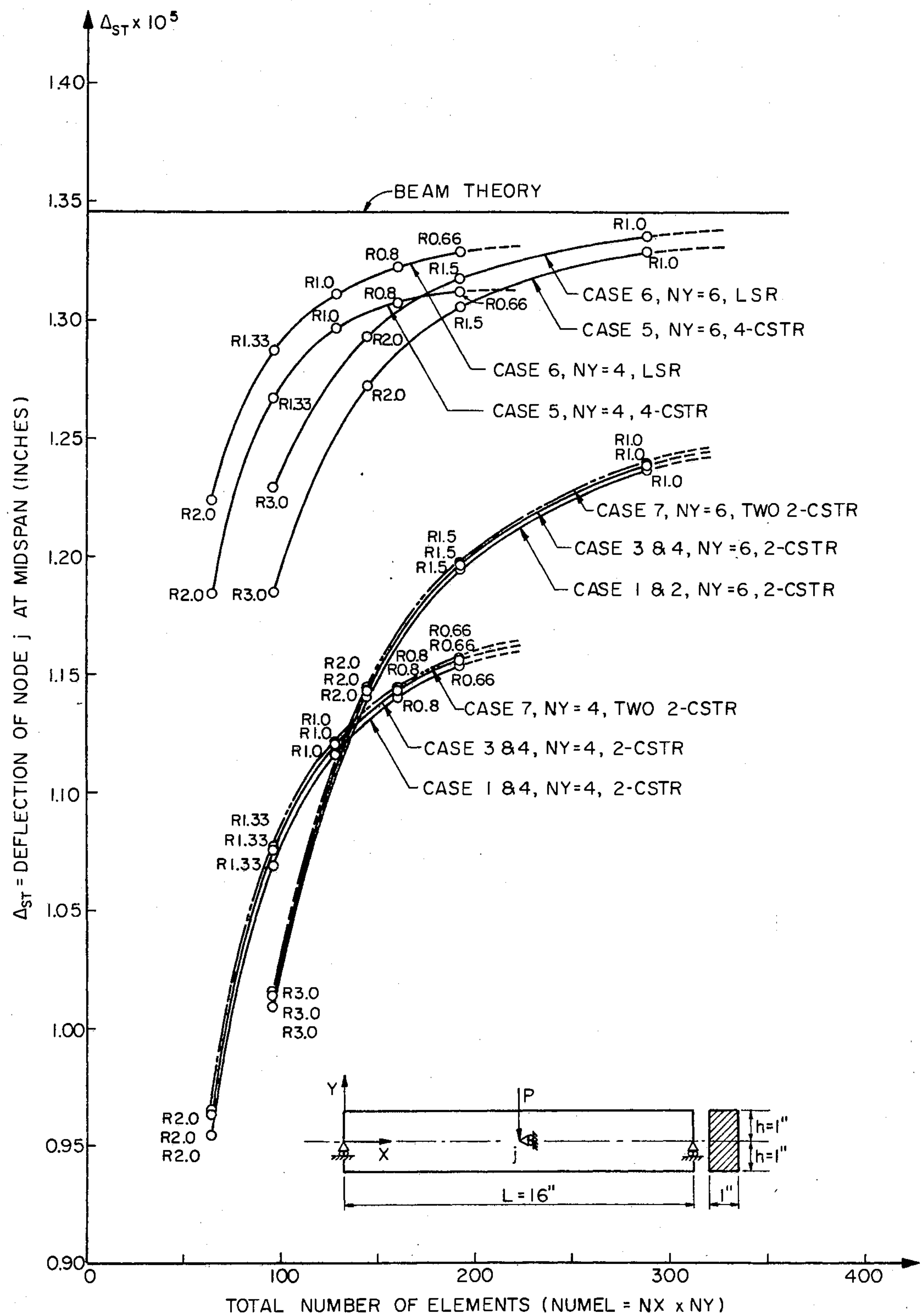


Figure 16. Displacement Convergence with Respect to Total Number of Elements for a Simply Supported Beam with Point Load at Mid-Span

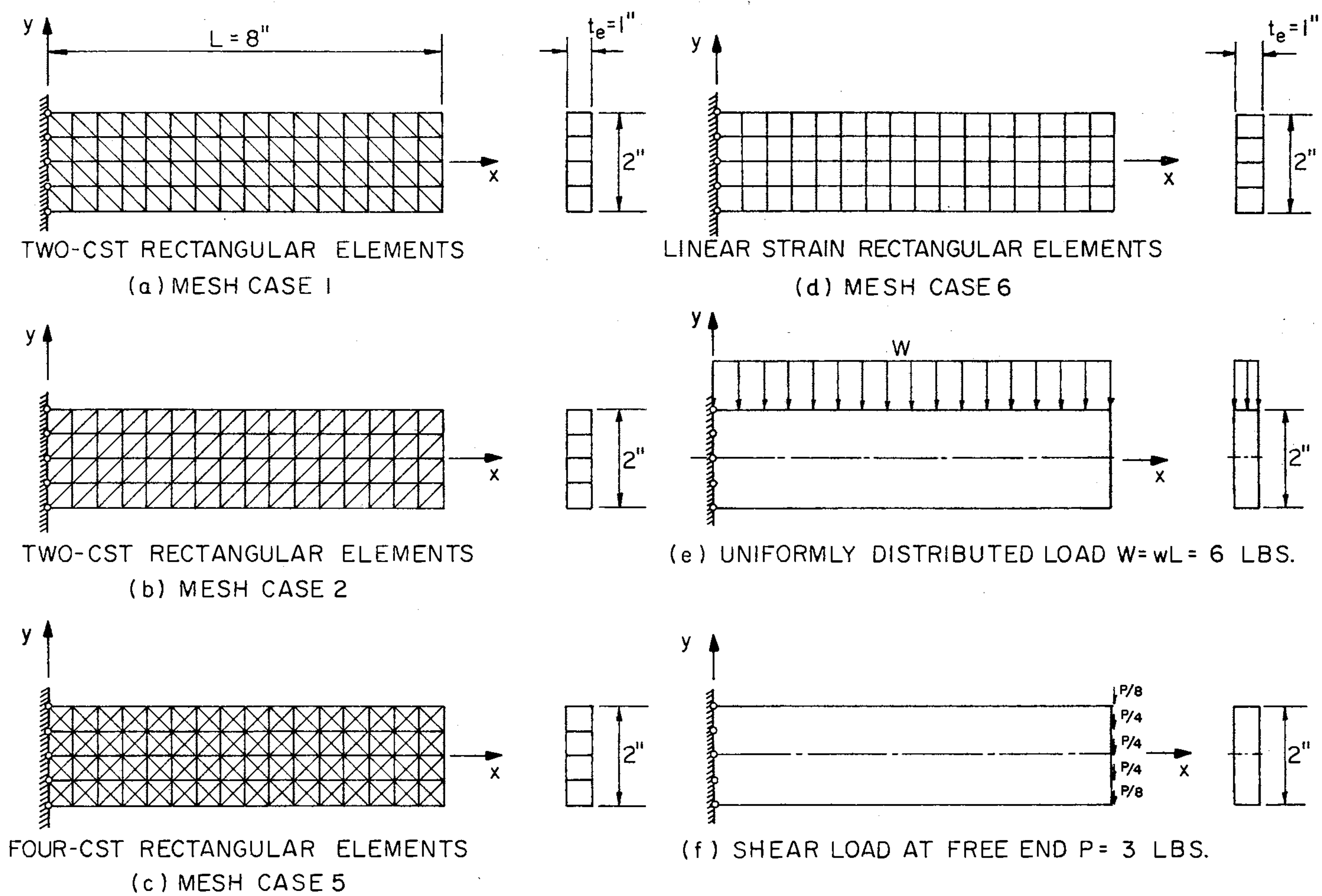


Figure 17. Cantilever Beams and Their Finite Element Idealizations (NY = 4)

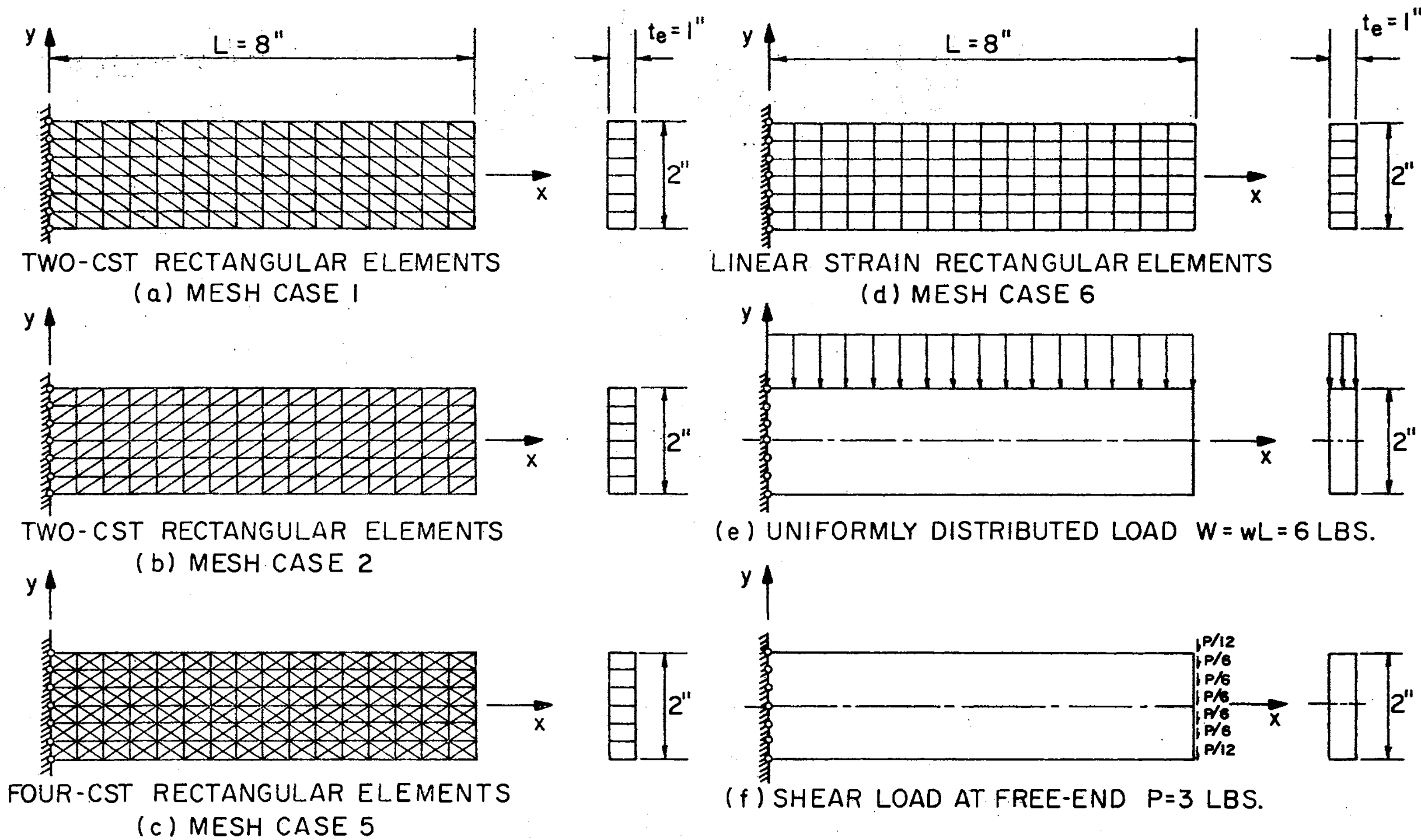


Figure 18. Cantilever Beams and Their Finite Element Idealizations (NY = 6)

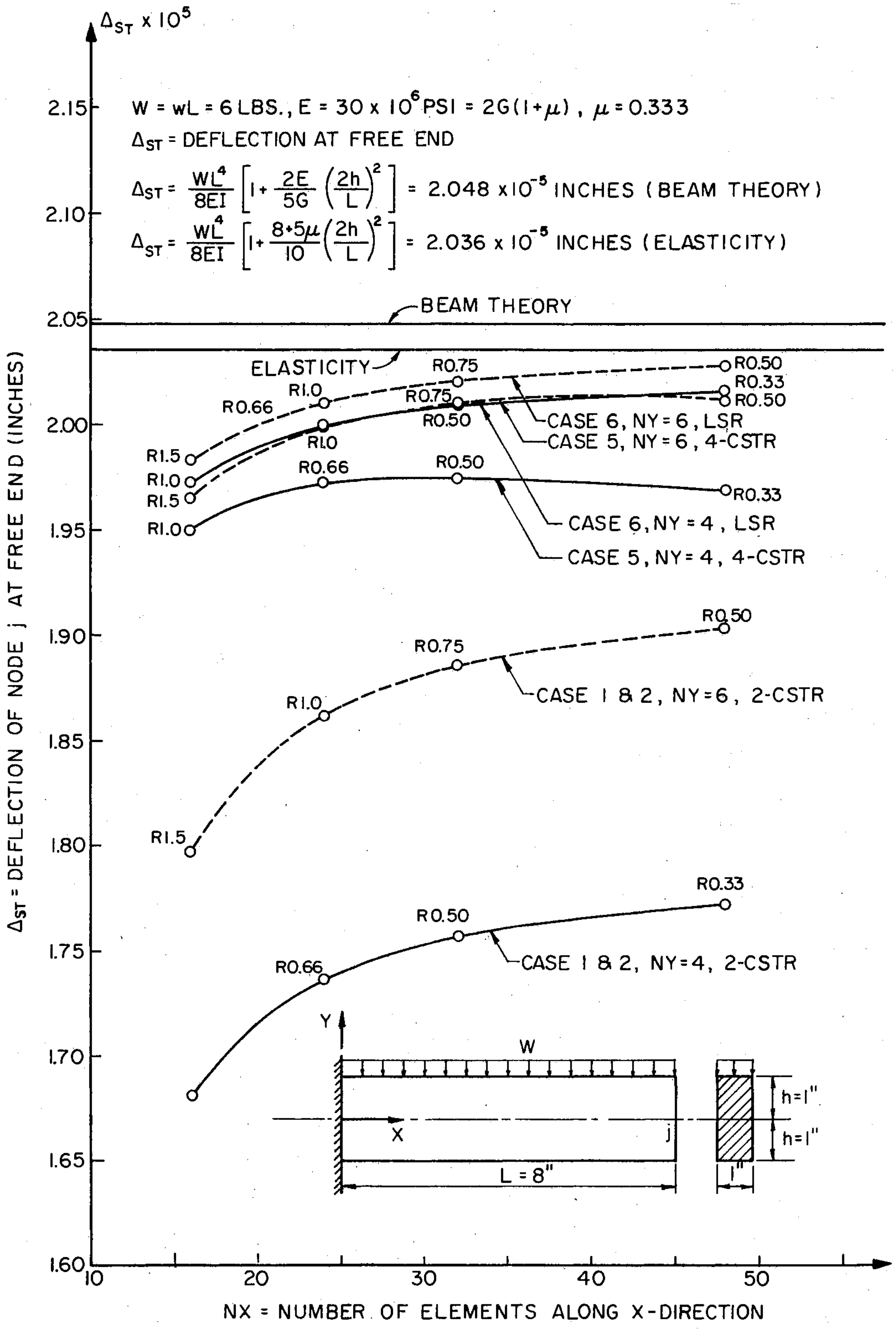


Figure 19. Displacement Convergence with Respect to Number of Elements Along X-Direction for a Cantilever Beam with Distributed Load

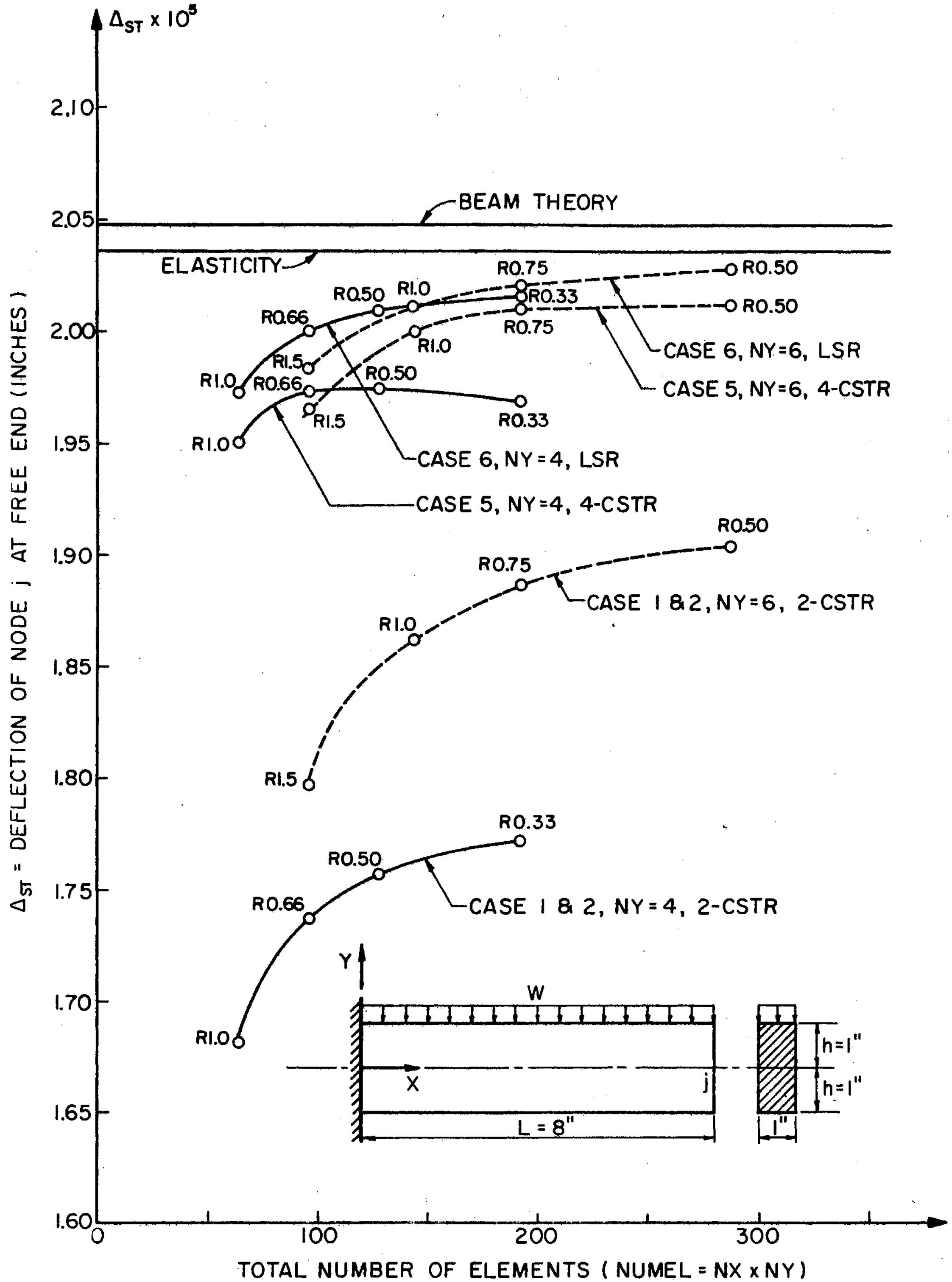


Figure 20. Displacement Convergence with Respect to Total Number of Elements for a Cantilever Beam with Distributed Load

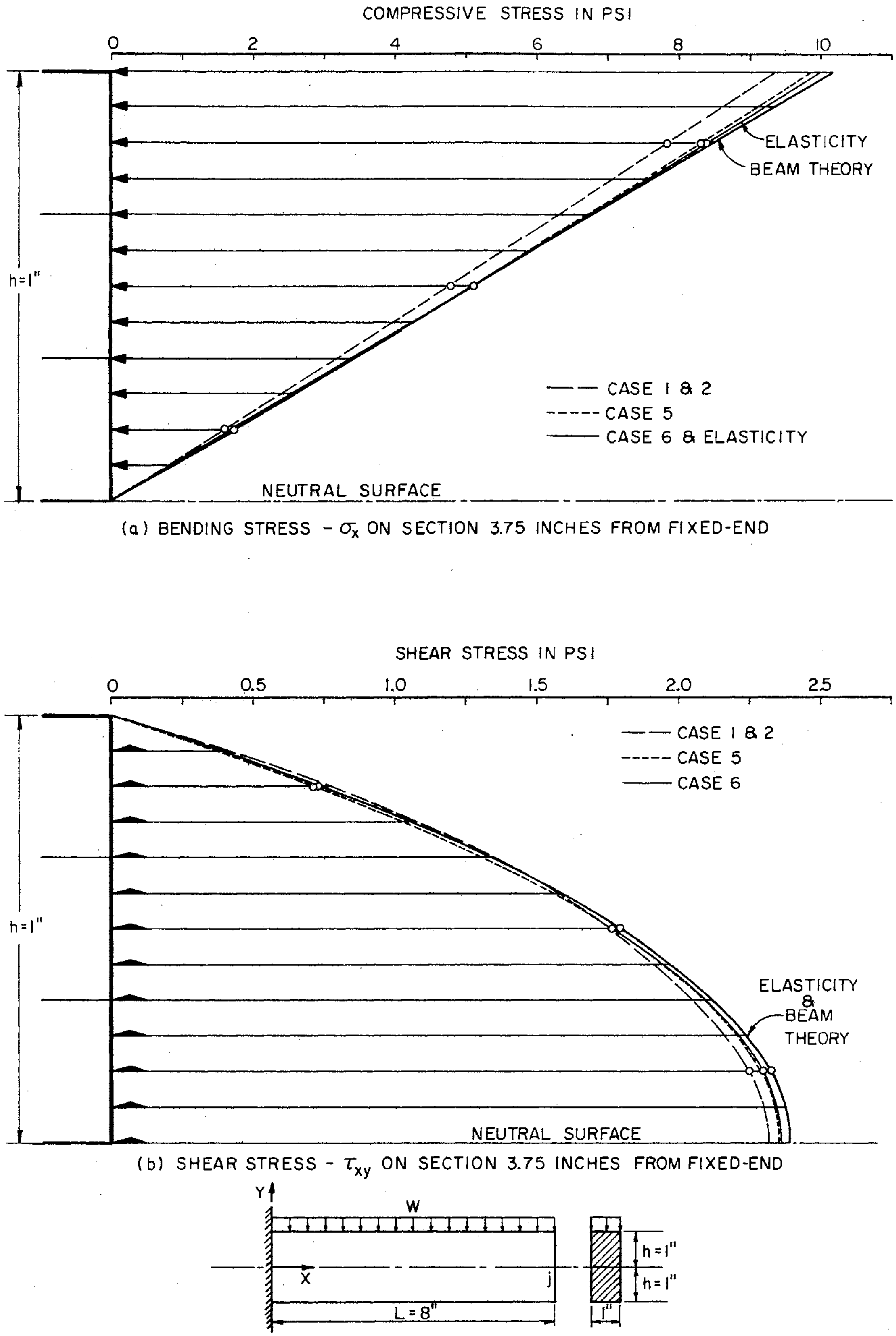


Figure 21. Bending and Shearing Stresses for a Cantilever Beam with Distributed Load

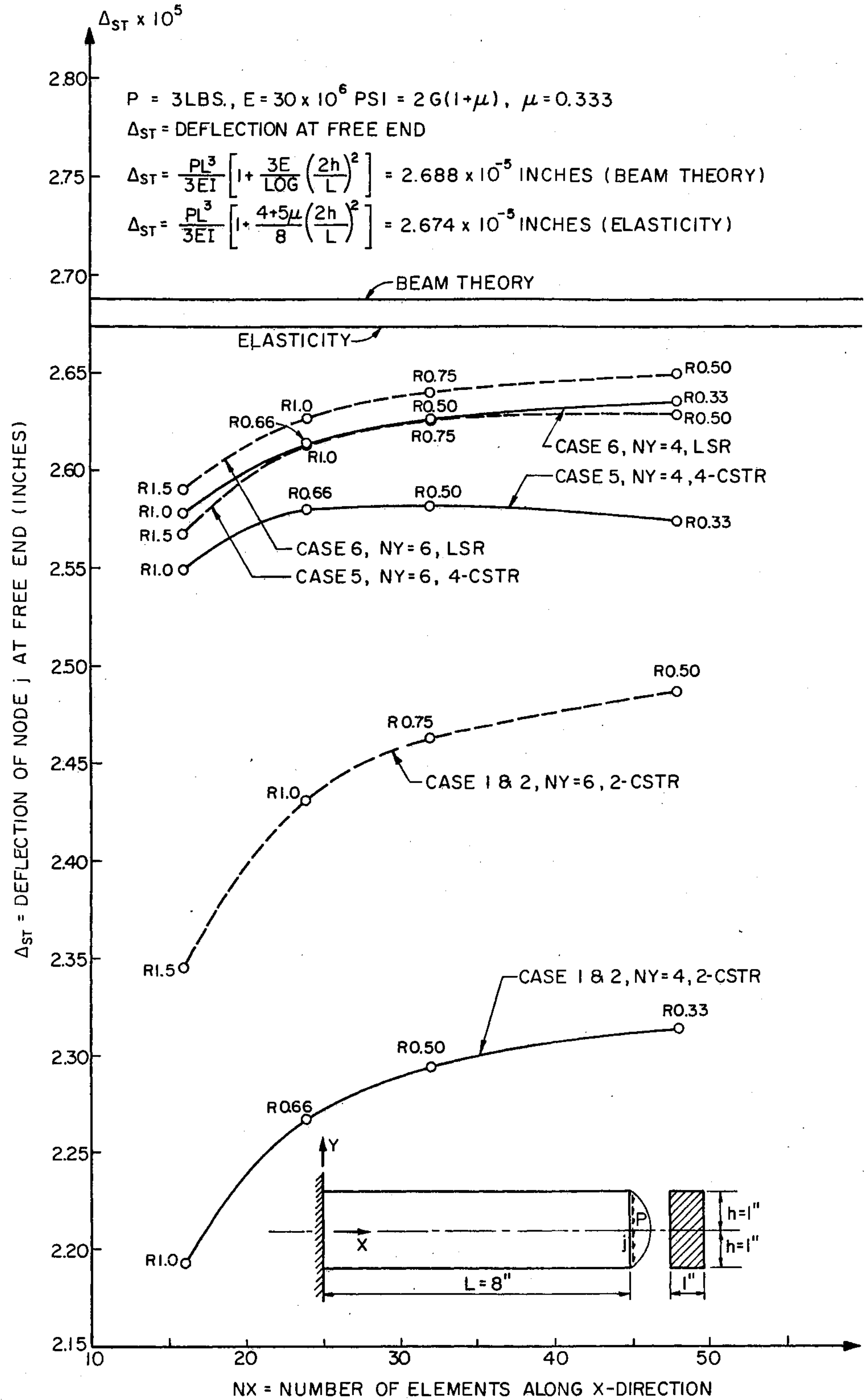


Figure 22. Displacement Convergence with Respect to Number of Elements Along X-Direction for a Cantilever Beam with Load at Free End

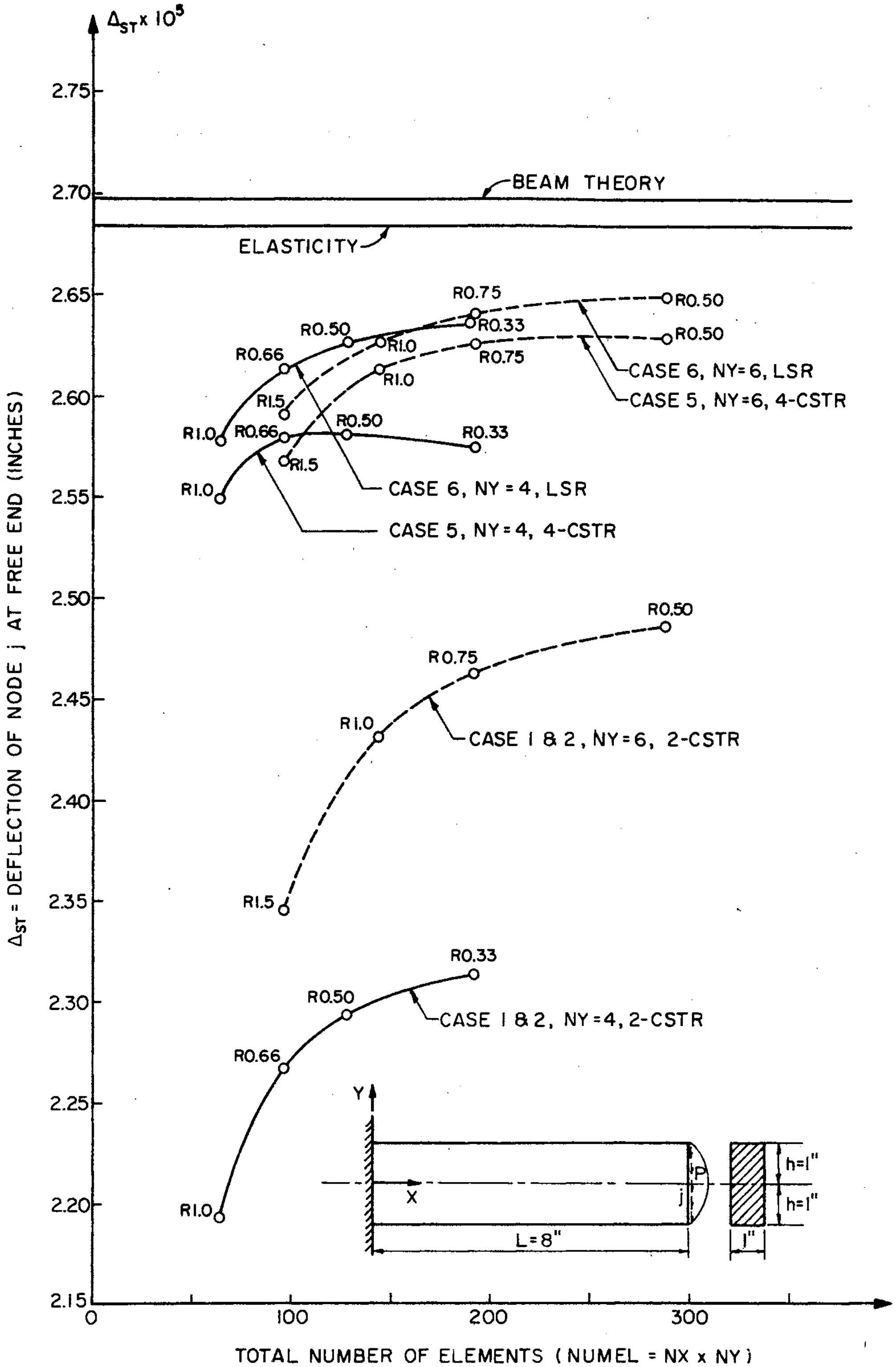


Figure 23. Displacement Convergence with Respect to Total Number of Elements for a Cantilever Beam with Load at Free End

where

$$wL = 6 \text{ lbs}$$

$$P = 3 \text{ lbs}$$

$$\mu = 0.333$$

$$E = 30 \times 10^6 \text{ psi} = 2G(1 + \mu).$$

3.2.2 Cantilever Beam

As shown in Figures 17 and 18, a cantilever beam was also analyzed for two kinds of loading: (a) a uniformly distributed load, and (b) a parabolically varying end shear. The fixed-end condition was introduced by restraining all the nodes along one of the end-edges of the beam against all displacements.

A comparison of free-end deflections and some internal stresses with elasticity and beam theory solutions is presented in Figures 19, 20, 21, 22, and 23. The values used as theoretical free-end deflection are:

(a) For a uniformly distributed load

$$\Delta_{ST} = \frac{wL^4}{8EI}\left[1 + \frac{2E}{5G}\left(\frac{2h}{L}\right)^2\right] = 2.048 \times 10^{-5} \text{ inches}$$

(Beam Theory) (3.4)

$$\Delta_{ST} = \frac{wL^4}{8EI}\left[1 + \frac{8 + 5\mu}{10}\left(\frac{2h}{L}\right)^2\right] = 2.036 \times 10^{-5} \text{ inches}$$

(Elasticity) (3.5)

(b) For a parabolically varying end shear

$$\Delta_{ST} = \frac{PL^3}{3EI}\left[1 + \frac{3E}{10G}\left(\frac{2h}{L}\right)^2\right] = 2.688 \times 10^{-5} \text{ inches}$$

(Beam Theory) (3.6)

$$\Delta_{ST} = \frac{PL^3}{3EI}\left[1 + \frac{4+5\mu}{8}\left(\frac{2h}{L}\right)^2\right] = 2.674 \times 10^{-5} \text{ inches}$$

(Elasticity) (3.7)

where

$wL = 6$ lbs

$P = 3$ lbs

$\mu = 0.333$

$E = 30 \times 10^6$ psi $= 2G(1 + \mu)$.

3.3 Discussion of Results

In the problems considered here for illustration, the elasticity solutions are approximately equal to the beam theory solutions, with some exceptions, such as in the proximity of the built-in end where the full clamping condition constitutes a mixed problem of elasticity for which there is no closed form solution. The elasticity solution assumes that the section at the built-in end is free to warp, while the finite element method of solution considers all nodal points fixed at this section. Therefore, the elasticity solution (shown in Figures 12, 13, 15, 16, 19, 20, 22, and 23) is an upper bound for the exact solution. On the other hand, the deflections computed by a compatible finite element analysis are lower bounds for the exact solution.

The normal stresses and shear stresses obtained by the finite element analysis are plotted in Figures 14 and 21 and compared with the beam theory and theory of elasticity for a cross section at a certain distance far from the cross section where the displacement-boundary conditions are specified.

It can be observed from Figures 12, 15, 19, and 22 that a considerable improvement in the results occurs as the number of elements along the height of the beam increases from NY = 4 to NY = 6 for the 2-CSTR idealization. This is obviously due to the fact that the improved convergence curve represents a better approximation to the actual deformation pattern than the other convergence curve. Case 5 represents the idealization containing 4-CSTR, which gives not only a finer division of the rectangular element than the 2-CSTR for the same rectangular element to achieve a monotonic convergence, but it also provides a better freedom of deformations and stress distribution. Thus, it gives smoother curve approximation to the actual stress and deformation curves. In general (and as shown in Figures 13, 14, 16, 20, 21, and 23), the linear strain rectangular elements (where applicable) yield slightly better results for stresses and deformations for a given nodal pattern than 4-CSTR elements because they employ a more refined displacement function and deformation approximation.

3.4 Computing Time

The computing time required for each problem is examined considering many factors which are involved in the total process of computation. The computation time depends on the number of elements and the nodal displacements in the idealized structure, band width of the total stiffness matrix, the time required to transmit information between core memory and the magnetic disk storage units, and the kind of computer used. The time to read or write on magnetic storage units varies directly with the band width and the number of records in each matrix.

Four polynomials were derived to give an estimate of the approximate computation time. These polynomials were obtained by evaluating the four arbitrary constants α_1, α_2, α_3, α_4 of Equation (3.8) for each of the four tables, I, II, III, and IV.

$$T = \alpha_1 + \alpha_2 \left(\frac{N}{100}\right) + \alpha_3 \left(\frac{N}{100}\right)^2 + \alpha_4 \left(\frac{N}{100}\right)^3 \qquad (3.8)$$

where

T = time in milliseconds

N = number of nodal points.

3.4.1 Simply Supported Beam (NY = 4, Table I)

Table I gives four values of the average time--3530, 4760, 6220, and 7450 milliseconds--corresponding to the four idealizations with the numbers of nodal points 85, 125, 165, and 245, respectively. Substituting these four sets of values into Equation (3.8), a set of four simultaneous equations with the unknown constants α_1, α_2, α_3, and α_4 were obtained. From the evaluation and substitution of these four constants into Equation (3.8), the following polynomial was obtained:

$$T = 4396.44 - 5449.85\left(\frac{N}{100}\right) + 6529.45\left(\frac{N}{100}\right)^2 - 1549.51\left(\frac{N}{100}\right)^3. \qquad (3.9)$$

This equation can be used to obtain an estimate of the execution time for other idealizations (NY = 4) which are not listed in Table I by substituting the value of the number of nodal points in the configuration of idealization.

3.4.2 Simply Supported Beam (NY = 6, Table II)

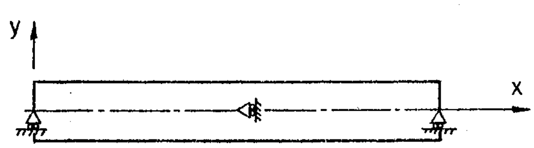
The polynomial

$$T = -3861.84 + 9924.02\left(\frac{N}{100}\right) - 2640.69\left(\frac{N}{100}\right)^2 + 351.14\left(\frac{N}{100}\right)^3 \qquad (3.10)$$

TABLE I

TIME COMPARISON FOR VARIOUS FINITE ELEMENT CONFIGURATIONS FOR A SIMPLY SUPPORTED BEAM
(NY = 4)

Number of Elements Along x-Direction	Number of Elements Along y-Direction	Type of Configuration (Case No.)	Time in Milli-seconds	Average Time in Milli-seconds T	Number of Elements	Number of Nodal Points N
16	4	1 or 2	3330	3530	64	85
		3 or 4	3690			
		5	3690			
		6	3430			
24	4	1 or 2	4640	4760	96	125
		3 or 4	4720			
		5	4910			
		6	4790			
32	4	1 or 2	6038	6220	128	165
		3 or 4	6372			
		5	6137			
		6	6338			
48	4	1 or 2	7435	7450	192	245
		3 or 4	7420			
		5	7600			
		6	7340			

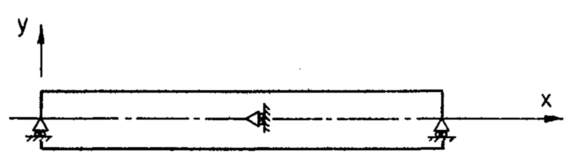


$$T = 4396.44 - 5449.85\left(\frac{N}{100}\right) + 6529.45\left(\frac{N}{100}\right)^2 - 1549.51\left(\frac{N}{100}\right)^3 \qquad (3.9)$$

TABLE II

TIME COMPARISON FOR VARIOUS FINITE ELEMENT CONFIGURATIONS FOR A SIMPLY SUPPORTED BEAM (NY = 6)

Number of Elements Along x-Direction	Number of Elements Along y-Direction	Type of Configur-ation (Case No.)	Time in Milli-seconds	Average Time in Milli-seconds T	Number of Elements	Number of Nodal Points N
16	6	1 or 2	4240	4800	96	119
		3 or 4	4700			
		5	5100			
		6	5290			
24	6	1 or 2	7170	7300	144	175
		3 or 4	7390			
		5	7320			
		6	7320			
32	6	1 or 2	9380	9300	192	231
		3 or 4	9100			
		5	9430			
		6	9380			
48	6	1 or 2	13080	13280	288	343
		3 or 4	13240			
		5	13390			
		6	13390			



$$T = -3861.84 + 9924.02\left(\frac{N}{100}\right) - 2640.69\left(\frac{N}{100}\right)^2 + 351.14\left(\frac{N}{100}\right)^3 \qquad (3.10)$$

was obtained using the same procedure as explained for Equation (3.9) in the preceding section. For Equation (3.10), values of the average time--4800, 7300, 9300, and 13280 milliseconds--corresponding to the idealizations with the numbers of nodal points 119, 175, 231, and 343, respectively, were used to evaluate the arbitrary constants α_1, α_2, α_3, and α_4 in Equation (3.8).

3.4.3 Cantilever Beam (NY = 4, Table III)

It can be observed from Tables I and III, by comparing the two idealizations corresponding to the same number of nodal points, that the solution of the cantilever beam problem required approximately 11 percent more execution time than that taken by the simply supported beam. This difference in time was due to the fact that the extra time was used in the modification of the matrix form of the equilibrium equation for the specified boundary conditions. Obviously, the specified boundary conditions for the cantilever beam are more in number than the number of boundary conditions specified for the simply supported beam.

The polynomial

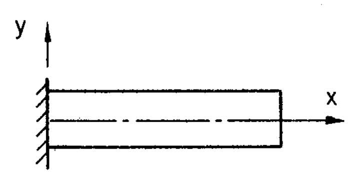
$$T = 4903.36 - 4744.11\left(\frac{N}{100}\right) + 5609.45\left(\frac{N}{100}\right)^2 - 1145.85\left(\frac{N}{100}\right)^3 \quad (3.11)$$

was derived by substituting into Equation (3.8) the values of the average time--4220, 5500, 7200, and 10100 milliseconds--corresponding to the values of the number of nodal points 85, 125, 165, and 245, given in Table III, and evaluating the arbitrary constants α_1, α_2, α_3, and α_4.

TABLE III

TIME COMPARISON FOR VARIOUS FINITE ELEMENT CONFIGURATIONS FOR A CANTILEVER BEAM (NY = 4)

Number of Elements Along x-Direction	Number of Elements Along y-Direction	Type of Configur-ation (Case No.)	Time in Milli-seconds	Average Time in Milli-seconds T	Number of Elements	Number of Nodal Points N
16	4	1 or 2	4090	4220	64	85
		5	4274			
		6	4290			
24	4	1 or 2	5322	5500	96	125
		5	5720			
		6	5455			
32	4	1 or 2	6918	7200	128	165
		5	7435			
		6	7285			
48	4	1 or 2	9598	10100	192	245
		5	10663			
		6	9981			



$$T = 4903.36 - 4744.11\left(\frac{N}{100}\right) + 5609.45\left(\frac{N}{100}\right)^2 - 1145.85\left(\frac{N}{100}\right)^3 \quad (3.11)$$

3.4.4 Cantilever Beam (NY = 6, Table IV)

By comparing Tables II and IV, a similar difference in time can be observed as discussed in the preceding section; however, this difference in time is roughly 50 percent larger than that between Tables I and III. This is due to the fact that the number of boundary conditions specified for the problem in Table IV are more than that specified for the problem in Table III.

For the derivation of the polynomial

$$T = 2652.77 + 1586.89\left(\frac{N}{100}\right) + 1280.06\left(\frac{N}{100}\right)^2 - 192.19\left(\frac{N}{100}\right)^3, \quad (3.12)$$

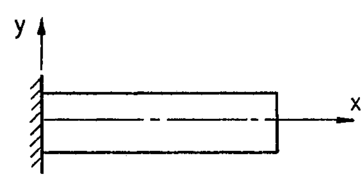
the set of values of the average time--6030, 8320, 10780, and 15400 milliseconds--corresponding to the values of the number of nodal points 119, 175, 231, and 343, given in Table IV, were substituted in Equation (3.8) to evaluate the values of the constants α_1, α_2, α_3, and α_4.

Tables I, II, III, and IV show the comparison of the execution times in milliseconds for the complete solution, including both stresses and displacements, but excluding the program compilation time. The polynomials of Equations (3.9), (3.10), (3.11), and (3.12) were derived to represent these tables and to delineate themselves as tools in estimating an approximate computation time for a configuration of idealization which is not listed in the tables. For example, if the number of elements in the y-direction of a simply supported beam is 6 (NY = 6, refer to Table II) and the number of nodal points are equal to 287, Equation (3.10) would give an approximate value of average computation time T = 11170 milliseconds upon substituting N = 287 into the equation.

TABLE IV

TIME COMPARISON FOR VARIOUS FINITE ELEMENT CONFIGURATIONS FOR A CANTILEVER BEAM (NY = 6)

Number of Elements Along x-Direction	Number of Elements Along y-Direction	Type of Configuration (Case No.)	Time in Milli-seconds	Average Time in Milli-seconds T	Number of Elements	Number of Nodal Points N
16	6	1 or 2	5987	6030	96	119
		5	6120			
		6	5971			
24	6	1 or 2	8234	8320	144	175
		5	8317			
		6	8401			
32	6	1 or 2	10531	10780	192	231
		5	11013			
		6	10796			
48	6	1 or 2	15106	15400	288	343
		5	15737			
		6	15355			



$$T = 2652.77 + 1586.89\left(\frac{N}{100}\right) + 1280.06\left(\frac{N}{100}\right)^2 - 192.19\left(\frac{N}{100}\right)^3 \qquad (3.12)$$

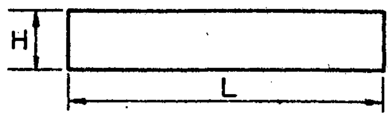
3.5 Aspect Ratio

From a study of Figures 13, 16, 20, and 23, it can be observed that reducing the size of the mesh increases the accuracy of the results. For example, the curve obtained for Case 5 (Figure 8) shows far better displacement convergence than do Cases 1, 2, 3, and 4 (Figures 5, 6, and 7), which contain triangular elements of twice the size of the triangular elements in Case 5. It is important to note that such a wide difference in the behavior of these two kinds of elements exists in spite of the fact that the number of degrees of freedom for the two kinds of elements differs only by two (Figures 5, 6, 7, and 8). Figures 13, 16, 20, and 23 show that the system of reducing the size of the mesh is controlled by another factor which is called the "aspect ratio." Here, the aspect ratio is defined as the ratio between the base and the height of the rectangular elements. It can be seen in the convergence curve for Case 5 that the result starts to diverge if the aspect ratio goes below 0.8, regardless of the size of the mesh. For Cases 1, 2, 3, 4, and 6, approximately at an aspect ratio of 0.8, the gradient of the curves has become very small and any further increase in the number of elements adds little to the convergence of displacement.

From a general observation of the displacement convergence curves, the empirical expressions of Table V can be given as a guide for the first trial. Further modifications in the size of the mesh can be made for the desired accuracy of the results. It is important to keep the value of the aspect ratio within certain limits to avoid ill-conditioning of the matrix. According to the illustration in Reference (41), the best performance results for the aspect ratio 1.0, when the number of degrees of freedom is maintained at the same value and only

TABLE V

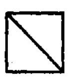
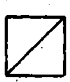
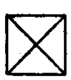
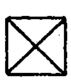
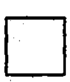
ESTIMATION OF A FINITE ELEMENT MESH



H = Height of Beam

L = Beam Span

R = L/H

Types of Element	Divisions in 'H'	Aspect Ratio Range	Recommended Number of Elements for First Trial	Expected Deviation from Beam Theory Solution
	8	0.80 to 0.90	36R to 40R	3%
	4	0.80 to 0.90	18R to 20R	4%
	6	0.80 to 0.90	38R to 42R	2%
	4	0.80 to 0.90	18R to 20R	3%
	6	0.80 to 0.90	38R to 42R	1%

the size of the mesh varies. It is recommended here that the aspect ratio for a mesh should be chosen within the limits of 0.75 and 1.50.

CHAPTER IV

DYNAMIC RESPONSE BY MODE SUPERPOSITION AND STEP-BY-STEP PROCEDURE

4.1 General

In this report, the finite element technique is applied to analyze the dynamic behavior of beams under free vibration and under a time-dependent forcing function.

For dynamic analysis by the finite element displacement method, the structure is discretized into a number of finite elements; and after the individual matrices are assembled, a set of simultaneous ordinary second-order differential equations of motion in terms of generalized displacements, velocities, and accelerations is obtained.

The solution of these differential equations of motion can be obtained using two different approaches. One of them is called the "classical mode superposition method." This method requires the extraction of a few normalized eigenvectors and eigenvalues, the number depending on the kind of the structure and the exciting force pattern. It is necessary to determine these eigenvectors accurately. Thus, the dynamic response analysis by the mode superposition method is to be preceded by a free vibration analysis to determine the eigenvalues and the corresponding eigenvectors. Recently, the mode superposition method was used by Clough and Chopra (11) to determine the response of earth dams under earthquakes as a plane strain problem and by

Idriss (31) to analyze the effects of the finite element mesh and earth-bank size on the accuracy of the results and suggested criterion for their selection.

The dynamic response of a problem with forced vibration can be determined directly by a numerical integration approach. In this approach, a certain variation of acceleration is assumed within the short period of integration. This method yields stable results using a reasonable amount of computer time. This method is more versatile than the normal mode superposition method and it can be extended to non-linear materials without further difficulties. This approach also retains the generality of the damping matrix, although in this report damping is neglected.

The matrix formulation of dynamic response problems uses a stiffness matrix to define the elastic characteristics and a mass matrix to define the inertial characteristics of the structure. The formulation of the stiffness matrix for various types of structures is well described in the literature (21) (62). The formulation of the mass matrix, on the other hand, is usually accomplished by the physical lumping of the structural mass at the nodes where the stiffness coefficients are defined. This leads to the formulation of the simple diagonal mass matrix, and thus to a simple technique of solution. Nevertheless, the resulting eigenvalues and eigenvectors may be quite different from the solution of the exact problem.

In spite of the fact that this report does not attempt to illustrate the application of the consistent mass matrix (1) (2) (25) and its effect on the accuracy of the results due to prohibitive computer time requirements, it does cover its derivation to some extent for a future opportunity for its application. To improve the accuracy of the dynamic analysis as it is affected by the mass matrix, the use of a consistent mass matrix is desirable. The evaluation of the consistent mass matrix is given in section 4. 2.

In brief, the studies show that the accuracy of a dynamic response analysis depends on the accuracy of the derivation of the mass and stiffness matrices and the degree to which the assumed displacement functions can represent the actual displacements. Thus, the utmost importance should be given to the choice of the type of finite element to be used.

4. 2 Differential Equation of Motion and Consistent Mass Matrix

To obtain the dynamic response (1) (2) (25) (44) of a structural system with small displacements, the applicable form of Lagrange's equations can be written as

$$\frac{d}{dt}\left\{\frac{\partial \overline{T}}{\partial \dot{\overline{\delta}}_i}\right\} + \left\{\frac{\partial \overline{F}}{\partial \dot{\overline{\delta}}_i}\right\} + \left\{\frac{\partial \overline{V}}{\partial \overline{\delta}_i}\right\} = \left\{\overline{Q}_i\right\} \quad (1 = 1, 2, 3, \ldots n) \qquad (4.1)$$

where $\left\{\overline{\delta}_i\right\}$ are the finite number of generalized coordinates representing the deformation of the structure at the nodal points and $\left\{\overline{Q}_i\right\}$ are the non-conservative generalized forces.

In the Rayleigh-Ritz technique, the deformation of a system can be expressed by a set of n independent displacement functions $\psi_i(x, y)$, so that the total displacement of the element area dA can be written as

$$\left\{\overline{q}(x, y, t)\right\}_{2x1} = \left\{\begin{matrix}\overline{u}_x(x, y, t)\\ \overline{v}_y(x, y, t)\end{matrix}\right\} = \left[\psi(x, y)\right]_{2x4}\left\{\overline{\delta}(t)\right\}_{4x1} \qquad (4.2)$$

where the function $\left\{\overline{q}(x, y, t)\right\}$ is the total displacement vector obtained by the superposition of the component vectors $\left[\psi(x, y)\right]$ and time-variant amplitudes $\left\{\bar{\delta}(t)\right\}$, which are the generalized nodal displacements of the element. If the structural system behaves linearly, the kinetic energy $\overline{T}_e$ of an element may be written as

$$\overline{T}_e = \frac{1}{2}\int_A m(x,y) \left\{\dot{\bar{q}}(x,y,t)\right\}^T_{1x4} \left\{\dot{\bar{q}}(x,y,t)\right\}_{4x1} dA \qquad (4.3)$$

where superscript T denotes the transpose of the matrix. Substituting Equation (4, 2) into Equation (4. 3),

$$\overline{T}_e = \frac{1}{2}\int_A m(x,y) \left\{\dot{\bar{\delta}}(t)\right\}^T_{1x4} \left[\psi(x,y)\right]^T_{2x4} \left[\psi(x,y)\right]_{2x4} \left\{\dot{\bar{\delta}}(t)\right\}_{4x1} dA$$

$$= \frac{1}{2}\left\{\dot{\bar{\delta}}(t)\right\}^T \left(\int_A m(x,y) \left[\psi(x,y)\right]^T \left[\psi(x,y)\right] dA\right) \left\{\dot{\bar{\delta}}(t)\right\}$$

$$= \frac{1}{2}\left\{\dot{\bar{\delta}}(t)\right\}^T_{1x4} \left[m_{ij}\right]_{4x4} \left\{\dot{\bar{\delta}}(t)\right\}_{4x1} \qquad (4.4)$$

where the coefficient m_{ij} is the symmetric mass inertia force acting at coordinate i concurrent with a unit acceleration of coordinate j and determined from

$$\left[m_{ij}\right] = \int_A m(x,y) \left[\psi(x,y)\right]^T \left[\psi(x,y)\right] dA \,. \qquad (4.5a)$$

The matrix $\left[m_{ij}\right]$ obtained from Equation (4. 5a) is called the consistent mass matrix of the element. If the mass per unit area of the element m(x, y) is a constant value m, then

$$\left[m_{ij}\right] = m\int_A \left[\psi(x,y)\right]^T \left[\psi(x,y)\right] dA \,. \qquad (4.5b)$$

The total mass matrix of the entire structure can be obtained by assembling the element mass matrices. The kinetic energy of the total structure can then be written as

$$\overline{T} = \frac{1}{2} \left\{\dot{\overline{\delta}}(t)\right\}^T_{1xn} \left[\overline{M}\right]_{nxn} \left\{\dot{\overline{\delta}}(t)\right\}_{nx1} . \tag{4.6}$$

Similarly, the potential energy $\overline{V}_e$ of the element can be written as

$$\overline{V}_e = \frac{1}{2} \left\{\overline{\delta}\right\}^T_{1x4} \left[k_{ij}\right]_{4x4} \left\{\overline{\delta}\right\}_{4x1} \tag{4.7a}$$

where the coefficient k_{ij} is the symmetric elastic stiffness coefficient and represents the restraining force acting at coordinate i concurrent with a unit displacement of coordinate j. The derivation of these coefficients has already been dealt with in section 2.7 and Appendix A. The total potential energy $\overline{V}$ of the linear structural system can be written as

$$\overline{V} = \frac{1}{2} \left\{\overline{\delta}\right\}^T_{1xn} \left[\overline{K}\right]_{nxn} \left\{\overline{\delta}\right\}_{nx1} . \tag{4.7b}$$

Assuming the damping force proportional to the velocity, it will be convenient to introduce a function

$$\overline{F} = \frac{1}{2} \left\{\dot{\overline{\delta}}\right\}^T \left[\overline{C}\right] \left\{\dot{\overline{\delta}}\right\} \tag{4.8}$$

which was called the "dissipation function" by Lord Rayleigh (64).

Introducing Equations (4.6), (4.7b), and (4.8) into Equation (4.1), a set of n coupled ordinary differential equations describing the motion of a viscously damped, linear system is obtained in the form

$$\left[\overline{M}_{ij}\right] \left\{\ddot{\overline{\delta}}_j\right\} + \left[\overline{C}_{ij}\right] \left\{\dot{\overline{\delta}}_j\right\} + \left[\overline{K}_{ij}\right] \left\{\overline{\delta}_j\right\} = \left[\overline{Q}_i\right] \binom{i = 1,2,3,\ldots n}{j = 1,2,3,\ldots n} \tag{4.9}$$

These equations are, in general, complex; but in the special case in which the damping matrix $\left[\overline{C}_{ij}\right]$ is a linear combination of the matrices $\left[\overline{M}_{ij}\right]$ and $\left[\overline{K}_{ij}\right]$, the uncoupled equations are real. This fact was pointed out by Lord Rayleigh (64), who stated that the uncoupling is achieved when $\overline{F}$ is a linear function of $\overline{T}$ and $\overline{V}$.

4.3 Mass Matrix, $[\overline{M}]$

In order to simplify the technique of solution, save computer storage, facilitate computations in generating the mass matrix, and economize computer time, it was assumed that the mass of the element is equally distributed among the nodes of the element. Thus, the resulting matrix is diagonal.

4.4 Stiffness Matrix, $[\overline{K}]$

The discussion given in section 2.4 also applies to this case.

4.5 Methods of Solution

The differential equation of vibration (Equation 4.9) for the dynamic response of a structural system can be solved by either of the following methods.

4.5.1 The Normal Mode Superposition Method

When the system is linear, that is, the elements of the matrices $[\overline{M}_{ij}]$ and $[\overline{K}_{ij}]$ are constant and the elements of the matrix $[\overline{C}_{ij}] \rightarrow 0$, the normal mode superposition technique can be used by extracting the eigenvalues and the eigenvectors of the structural system. The procedure of finding the eigenvalues and eigenvectors is discussed in section 4.6.

For a general multidegree of freedom system, the differential equation of the dynamic response is

$$[\overline{M}]\{\ddot{\overline{\delta}}\} + [\overline{K}]\{\overline{\delta}\} = \{\overline{Q}\} \qquad (4.10)$$

and in the case of free vibration, its solution can be written as

$$\left\{\bar{\delta}\right\} = \sum_{i=1}^{n} \left(d_i \left\{\Phi_i\right\} \cos \omega_i t + c_i \left\{\Phi_i\right\} \sin \omega_i t\right) \tag{4.11}$$

where $\left\{\Phi_i\right\}$ and ω_i are the normal modes and the natural frequencies, respectively, if the initial conditions at time $t = 0$ are

$$\left\{\bar{\delta}\right\} = \left\{\bar{\delta}_o\right\} \quad \text{and} \quad \left\{\dot{\bar{\delta}}\right\} = \left\{\dot{\bar{\delta}}_o\right\}. \tag{4.12}$$

The values of the coefficients d_i and c_i can be expressed as

$$d_i = \frac{\left\{\Phi_i\right\}^T \left[\overline{M}\right] \left\{\bar{\delta}_o\right\}}{\left\{\Phi_i\right\}^T \left[\overline{M}\right] \left\{\Phi_i\right\}} \quad \text{and} \quad c_i = \frac{1}{\omega_i} \frac{\left\{\Phi_i\right\}^T \left[\overline{M}\right] \left\{\dot{\bar{\delta}}_o\right\}}{\left\{\Phi_i\right\}^T \left[\overline{M}\right] \left\{\Phi_i\right\}}. \tag{4.13}$$

For the forced vibration, when the time-wise variation $f(\tau)$ is the same for every force, and the system is initially at rest,

$$\left\{\bar{\delta}\right\} = \sum_{i=1}^{n} \frac{1}{\omega_i^2} \frac{\left\{\Phi_i\right\}^T \left\{\overline{Q}_c\right\}}{\left\{\Phi_i\right\}^T \left[\overline{M}\right] \left\{\Phi_i\right\}} \left\{\Phi_i\right\} \int_{\tau=0}^{\tau=t} \omega_i f(\tau) \sin \omega_i (t-\tau)\, d\tau \tag{4.14}$$

where $\left\{\overline{Q}(\tau)\right\} = \left\{\overline{Q}_c\right\} f(\tau)$.

When the initial displacement and velocity of the system are zero (i. e., the system is at rest), only Equation (4. 14) will solve the problem. Also, if the load applied on the system remains constant for all time

$$\left\{\overline{Q}(\tau)\right\} = \left\{\overline{Q}_c\right\} \tag{4.15}$$

then Equation (4. 14) can be written as

$$\left\{\bar{\delta}\right\} = \sum_{i=1}^{n} \frac{1}{\omega_i^2} \frac{\left\{\Phi_i\right\}^T \left\{\overline{Q}_c\right\}}{\left\{\Phi_i\right\}^T \left[\overline{M}\right] \left\{\Phi_i\right\}} \left\{\Phi_i\right\} \int_{\tau=0}^{\tau=t} \omega_i \sin \omega_i (t-\tau)\, d\tau \tag{4.16a}$$

or

$$\left\{\bar{\delta}\right\} = \sum_{i=1}^{n} \frac{1}{\omega_i^2} \frac{\left\{\Phi_i\right\}^T \left\{\bar{Q}_c\right\}}{\left\{\Phi_i\right\}^T \left[\bar{M}\right] \left\{\Phi_i\right\}} \left\{\Phi_i\right\}(1 - \cos \omega_i t). \quad (4.16b)$$

It can be seen in Equation (4.16) that the term

$$\frac{1}{\omega_i^2} \frac{\left\{\Phi_i\right\}^T \left\{\bar{Q}_c\right\}}{\left\{\Phi_i\right\}^T \left[\bar{M}\right] \left\{\Phi_i\right\}}$$

is constant for a given value of i. By calling this term C_i, Equation (4.16) can be written as

$$\left\{\bar{\delta}\right\} = \sum_{i=1}^{n} C_i \left\{\Phi_i\right\}(1 - \cos \omega_i t) . \quad (4.17)$$

Once the natural frequencies ω_i and normalized modes $\left\{\Phi_i\right\}$ are known from the extracted eigenvalues and eigenvectors (section 4.6), the solution of differential Equation (4.10) can be found from Equation (4.17).

4.5.2 The Step-by-Step Integration Procedure

A direct integration of Equation (4.9) can be performed on the assumption that the accelerations at the discrete points vary in an arbitrary fashion during a small interval of time. The dynamic response is then determined by a step-by-step integration technique (83) following a sequence of matrix operations. The advantage of this procedure is that the pre-extraction of the eigenvalues and eigenvectors is not necessary.

The equation of motion, with a viscous form of damping, at time t can be written as (Equation (4.9)):

$$\left[\overline{M}\right]\left\{\ddot{\overline{\delta}}\right\}_t + \left[\overline{C}\right]\left\{\dot{\overline{\delta}}\right\}_t + \left[\overline{K}\right]\left\{\overline{\delta}\right\}_t = \left\{\overline{Q}\right\}_t \tag{4.18}$$

where

$\left\{\ddot{\overline{\delta}}\right\}_t$ = acceleration vector of the system at time t;

$\left\{\dot{\overline{\delta}}\right\}_t$ = velocity vector of the system at time t;

$\left\{\overline{\delta}\right\}_t$ = displacement vector of the system at time t;

$\left\{\overline{Q}\right\}$ = force vector on the system at time t;

$\left[\overline{M}\right]$ = total mass matrix;

$\left[\overline{C}\right]$ = damping matrix;

$\left[\overline{K}\right]$ = total stiffness matrix.

It is assumed here that within a small increment of time, the acceleration for each node of the system varies linearly, as shown in Figure 24. This permits the reduction of the second-order differential equation into a sequence of recurring matrix equations.

A direct integration over a time interval for all nodes yields the following matrix equations for the velocity and displacement at the end of the time interval Δt:

$$\left\{\dot{\overline{\delta}}\right\}_t = \left\{\dot{\overline{\delta}}\right\}_{t-\Delta t} + \frac{\Delta t}{2}\left\{\ddot{\overline{\delta}}\right\}_{t-\Delta t} + \frac{\Delta t}{2}\left\{\ddot{\overline{\delta}}\right\}_t = \left\{a\right\}_{t-\Delta t} + \frac{\Delta t}{2}\left\{\ddot{\overline{\delta}}\right\}_t \tag{4.19}$$

and

$$\left\{\overline{\delta}\right\}_t = \left\{\overline{\delta}\right\}_{t-\Delta t} + \Delta t\left\{\dot{\overline{\delta}}\right\}_{t-\Delta t} + \frac{(\Delta t)^2}{3}\left\{\ddot{\overline{\delta}}\right\}_{t-\Delta t} + \frac{(\Delta t)^2}{6}\left\{\ddot{\overline{\delta}}\right\}_t$$

$$= \left\{b\right\}_{t-\Delta t} + \frac{(\Delta t)^2}{6}\left\{\ddot{\overline{\delta}}\right\}_t \tag{4.20}$$

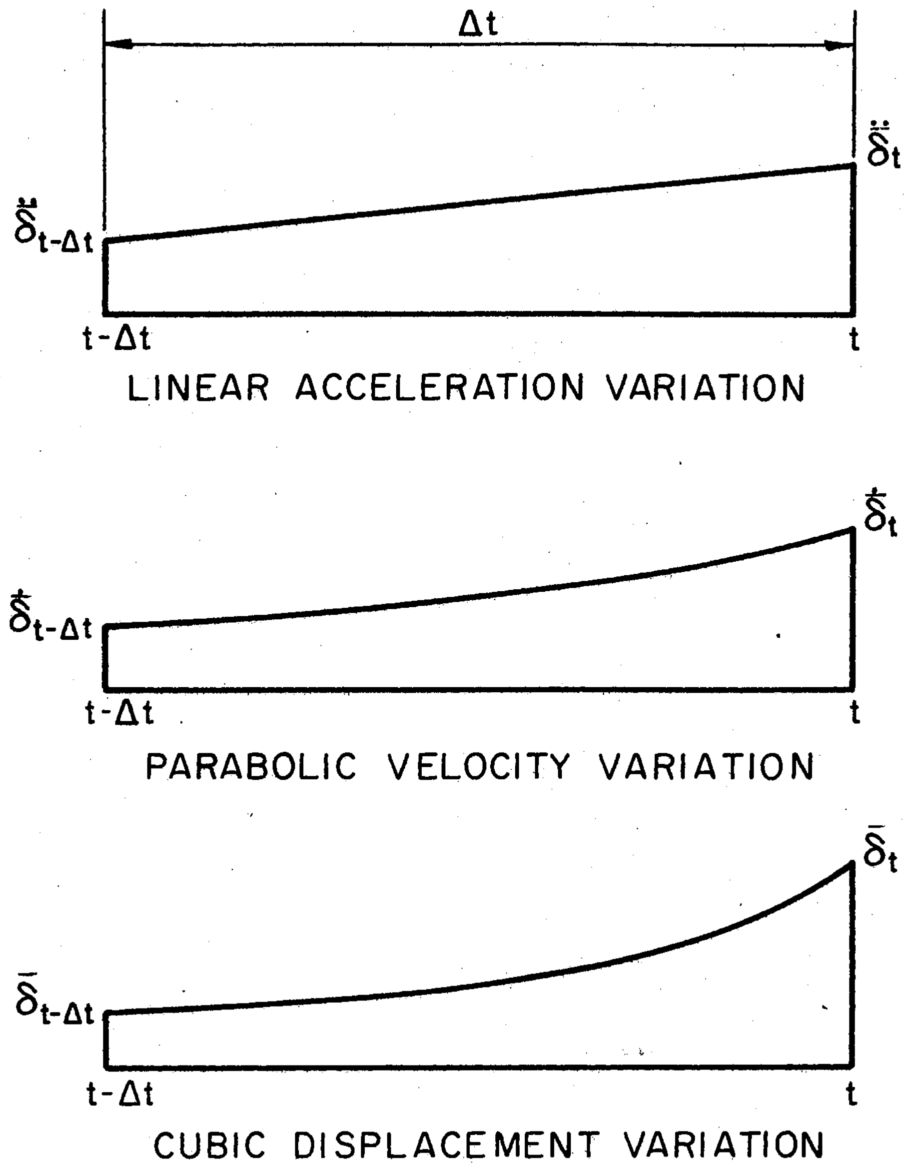


Figure 24. Displacement Functions

where

$$\{a\}_{t-\Delta t} = \{\dot{\bar{\delta}}\}_{t-\Delta t} + \frac{\Delta t}{2}\{\ddot{\bar{\delta}}\}_{t-\Delta t} \qquad (4.21)$$

and

$$\{b\}_{t-\Delta t} = \{\bar{\delta}\}_{t-\Delta t} + \Delta t\{\dot{\bar{\delta}}\}_{t-\Delta t} + \frac{(\Delta t)^2}{3}\{\ddot{\bar{\delta}}\}_{t-\Delta t} . \qquad (4.22)$$

Introducing Equations (4.19) and (4.20) into Equation (4.18) gives

$$[\overline{M}]\{\ddot{\bar{\delta}}\}_t + [\overline{C}]\left(\{a\} + \frac{\Delta t}{2}\{\ddot{\bar{\delta}}\}_t\right) + [\overline{K}]\left(\{b\} + \frac{(\Delta t)^2}{6}\{\ddot{\bar{\delta}}\}_t\right) = \{\overline{Q}\}_t$$

or

$$\left[[\overline{M}] + \frac{\Delta t}{2}[\overline{C}] + \frac{(\Delta t)^2}{6}[\overline{K}]\right]\{\ddot{\bar{\delta}}\}_t = \{\overline{Q}\}_t - [\overline{C}]\{a\} - [\overline{K}]\{b\}$$

or

$$\{\ddot{\bar{\delta}}\}_t = \left[[\overline{M}] + \frac{\Delta t}{2}[\overline{C}] + \frac{(\Delta t)^2}{6}[\overline{K}]\right]^{-1}\Big\{\{\overline{Q}\}_t - [\overline{C}]\{a\}_{t-\Delta t} - [\overline{K}]\{b\}_{t-\Delta t}\Big\} . \qquad (4.23)$$

Since damping is being neglected here, ($[\overline{C}] \to 0$), Equation (4.23) becomes

$$\{\ddot{\bar{\delta}}\}_t = \left[[\overline{M}] + \frac{(\Delta t)^2}{6}[\overline{K}]\right]^{-1}\Big\{\{\overline{Q}\}_t - [\overline{K}]\{b\}_{t-\Delta t}\Big\} . \qquad (4.24)$$

Since the initial displacement and velocity are zero, the procedure can be outlined as follows:

Step 1: from Equation (4.18) at time t = 0

$$\{\ddot{\bar{\delta}}\}_0 = [\overline{M}]^{-1}\{\overline{Q}\}_0 \qquad (4.25a)$$

Step 2: from Equations (4.21) and (4.22) at time t = 0

$$\{a\}_0 = \frac{\Delta t}{2}\{\ddot{\bar{\delta}}\}_0 \tag{4.25b}$$

$$\{b\}_0 = \frac{(\Delta t)^2}{3}\{\ddot{\bar{\delta}}\}_0 \tag{4.25c}$$

Step 3: from Equation (4.24) at time t = Δt

$$\{\ddot{\bar{\delta}}\}_{\Delta t} = \left[[\overline{M}] + \frac{(\Delta t)^2}{6}[\overline{K}]\right]^{-1}\left\{\{\overline{Q}\}_{\Delta t} - [\overline{K}]\{b\}_0\right\} \tag{4.25d}$$

Step 4: from Equations (4.19) and (4.20) at time t = Δt

$$\{\dot{\bar{\delta}}\}_{\Delta t} = \{a\}_0 + \frac{\Delta t}{2}\{\ddot{\bar{\delta}}\}_{\Delta t} \tag{4.25e}$$

$$\{\bar{\delta}\}_{\Delta t} = \{b\}_0 + \frac{(\Delta t)^2}{6}\{\ddot{\bar{\delta}}\}_{\Delta t}\ . \tag{4.25f}$$

After the completion of the above four steps of Equation (4.25), the following steps can be repeated in sequence for each increment of time Δt:

Step 1:

$$\{a\}_{t-\Delta t} = \{\dot{\bar{\delta}}\}_{t-\Delta t} + \frac{\Delta t}{2}\{\ddot{\bar{\delta}}\}_{t-\Delta t} \tag{4.26a}$$

Step 2:

$$\{b\}_{t-\Delta t} = \{\bar{\delta}\}_{t-\Delta t} + \Delta t\{\dot{\bar{\delta}}\}_{t-\Delta t} + \frac{(\Delta t)^2}{3}\{\ddot{\bar{\delta}}\}_{t-\Delta t} \tag{4.26b}$$

Step 3:

$$\{\ddot{\bar{\delta}}\}_t = \left[[\overline{M}] + \frac{(\Delta t)^2}{6}[\overline{K}]\right]^{-1}\left\{\{\overline{Q}\}_t - [\overline{K}]\{b\}_{t-\Delta t}\right\} \tag{4.26c}$$

Step 4:

$$\{\dot{\bar{\delta}}\}_t = \{a\}_{t-\Delta t} + \frac{\Delta t}{6}\{\ddot{\bar{\delta}}\}_t \tag{4.26d}$$

Step 5:

$$\left\{\bar{\delta}\right\}_t = \left\{b\right\}_{t-\Delta t} + \frac{(\Delta t)^2}{6}\left\{\ddot{\bar{\delta}}\right\}_t \qquad (4.26e)$$

Thus, the step-by-step response of the system can be obtained by the repeated application of the above five steps of Equation (4.26).

4.6 Normal Modes and Frequencies of a Free Vibration Problem

The modes and frequencies can be obtained by letting $\left\{\bar{Q}\right\} = 0$ and assuming the displacements $\left\{\delta\right\}$ to be sinusoidal functions of time with frequency ω. Thus,

$$\left\{\bar{\delta}\right\} = \left\{X\right\} \sin \omega t \qquad (4.27a)$$

$$\left\{\ddot{\bar{\delta}}\right\} = -\omega^2 \left\{X\right\} \sin \omega t \qquad (4.27b)$$

and Equation (4, 10) becomes

$$-\omega^2 \left[\bar{M}\right] \left\{X\right\} + \left[\bar{K}\right] \left\{X\right\} = 0 \qquad (4.28a)$$

or

$$\left[\bar{M}\right]^{-1} \left[\bar{K}\right] \left\{X\right\} = \omega^2 \left\{X\right\} \qquad (4.28b)$$

or

$$\left[\bar{A}\right] \left\{X\right\} = \bar{\lambda} \left\{X\right\} . \qquad (4.28c)$$

The solution of Equation (4.28c) can be obtained by using a classical method of finding the eigenvalues and eigenvectors using the computer subroutines given in Appendix C. This subroutine (22) (38) is written to find all the real eigenvalues and eigenvectors of a real general matrix. The eigenvalues are computer by the "QR double-step" method and the eigenvectors by the inverse iteration technique (17) (80).

To improve the accuracy of the results, the following modifications are carried out:

1. The matrix is scaled by a sequence of similarity transformations so that the absolute sums of corresponding rows and columns are roughly equal.

2. The scaled matrix is normalized so that the value of the Euclidean norm is equal to one.

The main part of the process commences with the reduction of an (n x n) real matrix $\left[\overline{A}\right]$ by a similarity transformation (Householder's method) to an upper, almost triangular Hessenberg form (80). Then the QR double-step iterative process is performed on the Hessenberg matrix until all elements of the subdiagonal that converge to zero are in modulus less than $2^{-t}\ \|H\|_E$, where t is the number of significant digits in the mantissa of a binary floating-point number and $\|H\|_E$ is the Euclidean (Frobenius) norm of the triangular Hessenberg matrix. For the IBM 360/65 computer, the value of t is equal to 53. Since the Hessenberg form is preserved under the QR iteration, a reduction of the initial matrix $\left[\overline{A}\right]$ to the Hessenberg form provides a significant saving of computation in each iteration for the QR decomposition.

After the eigenvectors $\{X\}$ and eigenvalues $\overline{\lambda}$ are determined, normalized eigenvectors $\{\Phi\}$ and natural frequencies ω are determined as follows:

$$\{\Phi_i\} = \frac{1}{\sqrt{\{X_i\}^T\{X_i\}}}\{X_i\} \tag{4.29}$$

and

$$\omega_i = \sqrt{\lambda_i} \tag{4.30}$$

where i = 1, 2, 3, ... n.

4.7 Closed Form Solution

In the case of a beam with simply supported ends (as shown in Figure 25), the displacements are represented by the summation of a series of the sinusoidal displacements.

$$\Delta_{DYN} = \sum_{i=1,2,3,\ldots} \bar{\Phi}_i(t) \sin \frac{i\pi x}{L} \quad . \tag{4.31}$$

The functions of time, $\bar{\Phi}_i(t)$, are determined from the differential equation of dynamic response which can be derived using d'Alembert's principle combined with the principle of virtual work. The equation for the vibration, produced by some disturbing force $\bar{P}$ applied at a distance $\bar{x}$ from the end support, is obtained as

$$\ddot{\bar{\Phi}}_i + \frac{i^4\pi^4 EI}{L^4 m} \bar{\Phi}_i = \frac{2\bar{P}}{mL} \sin \frac{i\pi\bar{x}}{L} \quad . \tag{4.32}$$

The general solution of Equation (4.32) is

$$\bar{\Phi}_i = \alpha_1 \cos \frac{i^2\pi^2}{L^2} \sqrt{\frac{EI}{m}}\, t + \alpha_2 \sin \frac{i^2\pi^2}{L^2} \sqrt{\frac{EI}{m}}\, t + \frac{L^2}{i^2\pi^2} \sqrt{\frac{m}{EI}} \left(\frac{2}{mL}\right) \int_{\tau=0}^{\tau=t} \bar{P} \sin \frac{i\pi\bar{x}}{L} \sin\left\{\frac{i^2\pi^2}{L^2} \sqrt{\frac{EI}{m}}\, (t-\tau)\right\} d\tau \quad , \tag{4.33}$$

Integrating the third term, which represents the forced vibration, and ignoring the first two terms, which represent the free vibration, the displacement equation is obtained from Equation (4.33) as

$$\Delta_{DYN} = \frac{2\bar{P}L^3}{\pi^4 EI} \sum_{i=1,2,3,\ldots} \frac{1}{i^4} \sin \frac{i\pi\bar{x}}{L} (1 - \text{cox}\ \omega_i t) \sin \frac{i\pi x}{L} . \tag{4.34}$$

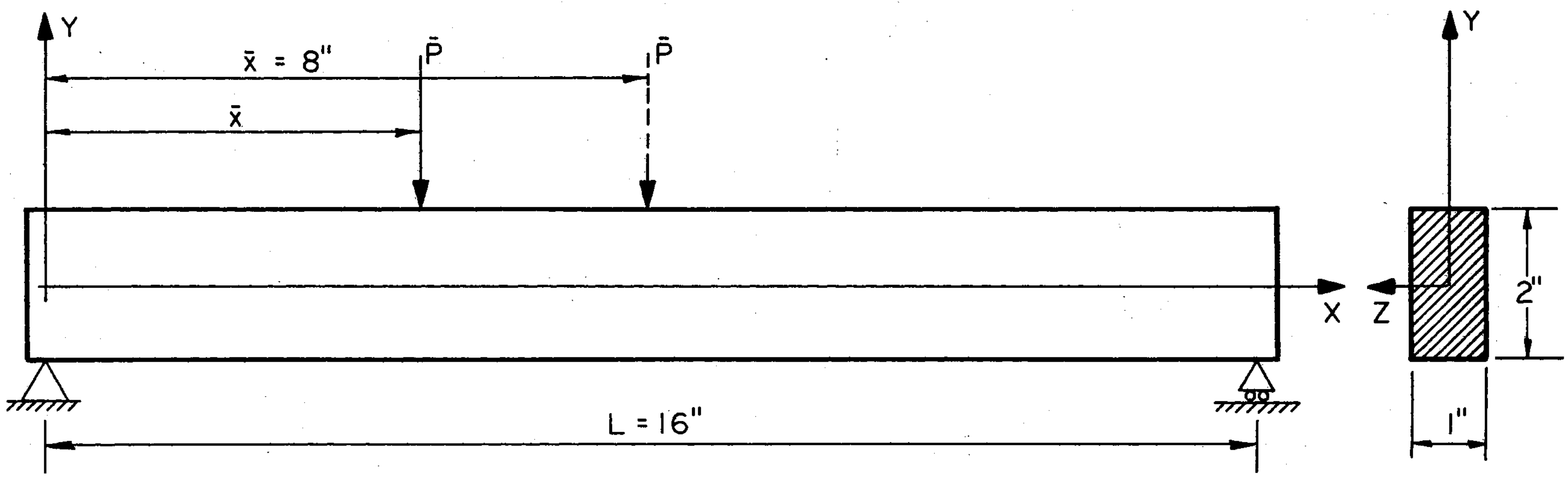


Figure 25. Simply Supported Beam with an Applied Disturbing Load $\bar{P}$

When the load P is applied at midspan (i. e., $\bar{x} = \frac{L}{2}$),

$$\Delta_{DYN} = \frac{2\bar{P}L^3}{\pi^4 EI} \sum_{i=1,3,5,\ldots} \frac{(-1)^{(i-1)/2}}{i^4} (1 - \cos \omega_i t) \sin \frac{i\pi x}{L} . \qquad (4.35)$$

Thus, the displacement for the point at midspan can be written as

$$\Delta_{DYN} = \frac{2\bar{P}L^3}{\pi^4 EI} \sum_{i=1,3,5,\ldots} \frac{1}{i^4} (1 - \cos \omega_i t) . \qquad (4.36)$$

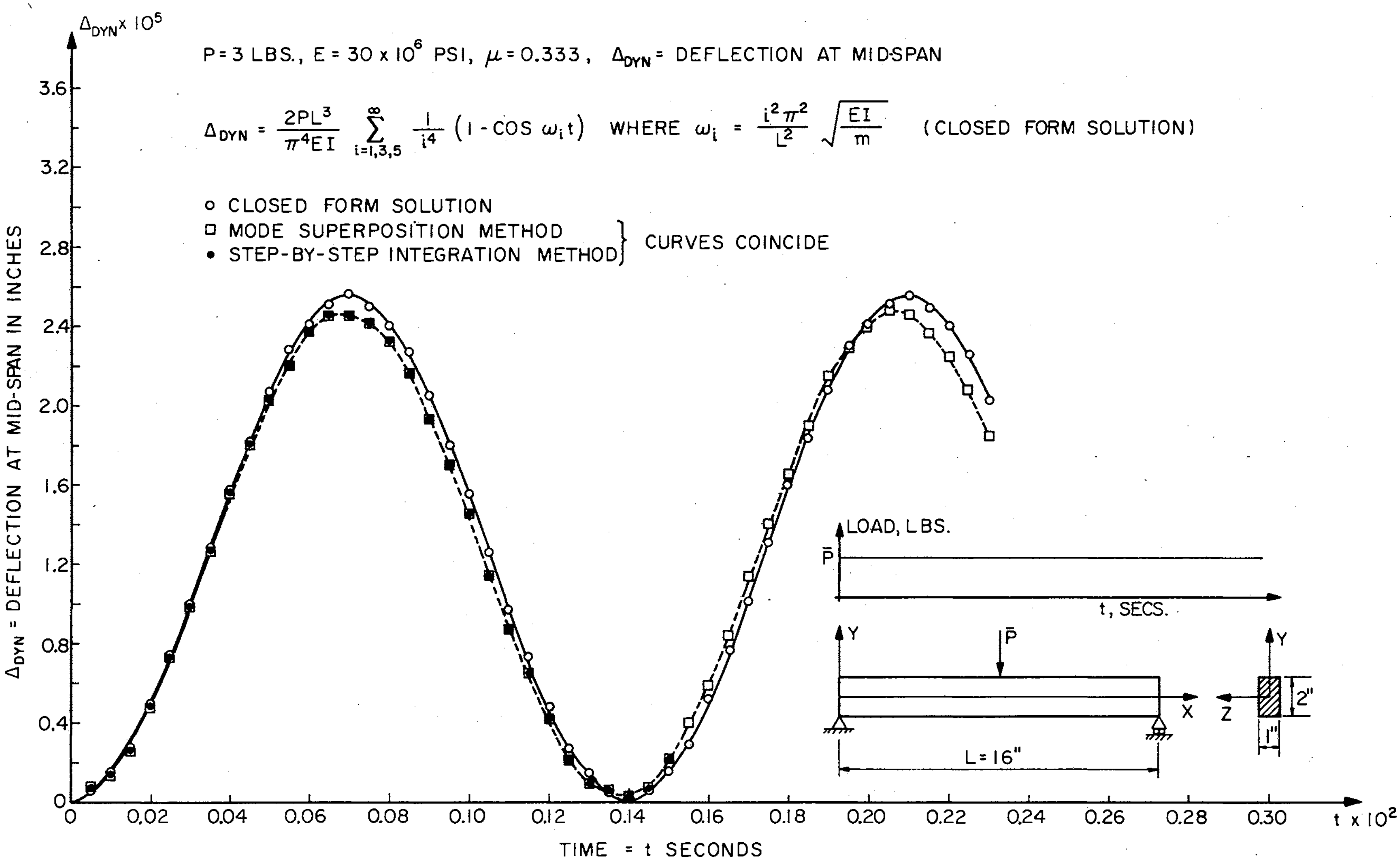


Figure 26. Comparison of Dynamic Solutions

CHAPTER V

AN EXAMPLE OF A LINEAR ELASTIC PROBLEM WITH A DYNAMIC LOAD AND ITS RESULTS

5.1 Procedure

The numerical procedures presented in Chapter IV are illustrated here by using an example of a beam with both ends pinned and a suddenly applied constant impact load $\overline{P}$ of infinite duration, as shown in Figure 26. To maintain the symmetry of the problem, both ends of the beam were restrained against vertical displacement and the node at the mid-span was restrained against horizontal displacement, as shown in Figures 10 and 11. For ease in comparing the two numerical procedures, an equal number of finite elements was chosen for the finite element configuration of the example. The comparison was made for the computation time, storage requirement, and the convergence of the deflection solution at midspan to the closed form solution.

The finite element mesh was processed by a digital computer program using a linear strain rectangle as the basic element. The computer programs are given in Appendices B and C. These programs generate the mesh, compute the stiffness matrix, assemble the total stiffness matrix, formulate mass matrix, modify the matrices for the given boundary conditions, calculate the eigenvalues and eigenvectors, select the flexural modes, perform the modal superposition and step-by-step integration of the differential equation of vibration. The computed displacements at the midspan of the beam by the normal mode superposition and step-by-step integration methods are presented in Table II and plotted in a graphical form in Figure 26 for a comparison of the results.

5.2 Selection of the Finite Element Mesh

The linear strain rectangular element was selected for this example in view of the fact that it represents a better deformation approximation than the constant strain triangle rectangular element as discussed in section 3.3.

According to section 3.5, the selection of a mesh should be determined on the basis of an aspect ratio ranging within the limits of 0.75 and 1.50 as a first trial, and then the succeeding trials could be estimated to meet the degree of accuracy desired for the solution of a problem. For the problem considered here for illustration, the selection of the mesh size was dictated by two major factors: (a) the amount of funds available for the computer use and the number of K-bytes of available main core memory in the computer, as discussed in section 5.3, and (b) the desired degree of accuracy of the results. Since the first factor was the dominating one, a mesh of (4 x 18) with an aspect ratio of 1.77 was selected to save computer time. A mesh within the recommended limits of aspect ratio would have required a larger number of elements and thus more computation time. It can be observed from Figure 16 that by violating the recommended limits of the aspect ratio, a considerable degree of accuracy of the solution was sacrificed. This illustrates that the aspect ratio of 1.77 would not have been a good trial value if one intends to achieve a good degree of accuracy of the solution. It can be noticed also in Figure 16 that the maximum deflection for the static load at midspan of the beam for the (4 x 18) mesh differs by 7 percent from the closed form solution, while if one chooses the aspect ratio of 1.50, this maximum deflection will differ only by 4.8 percent from the closed form solution.

5.3 Example

5.3.1 Normal Mode Superposition Method

1. Computer program: The computer program (22) to determine the eigenvalues and eigenvectors was rewritten to make full use of the available main core memory and to keep the reading and writing on the magnetic disk as small as possible. This approach saves computer time and is thus economical.

2. Selection of modes: The flexural modes of vibration were selected by comparing them with the flexural modes and the corresponding natural frequencies obtained from beam theory solution. A computer program FLXMOD (Appendix C) was written on the basis of Equation (4.35) to select all symmetrical modes and rearrange them in order of their increasing natural frequencies. From these orderly arranged 47 modes, the modes whose period was greater than 3 percent was chosen on the basis of a judgment that the superposition of the modes whose period is larger than 3 percent would satisfactorily represent the required solution. On the basis of the above criterion of judgment, only five flexural modes were selected for the superposition to obtain a satisfactory solution. These first five normal modes of the lowest natural frequencies are presented graphically in Figure 27.

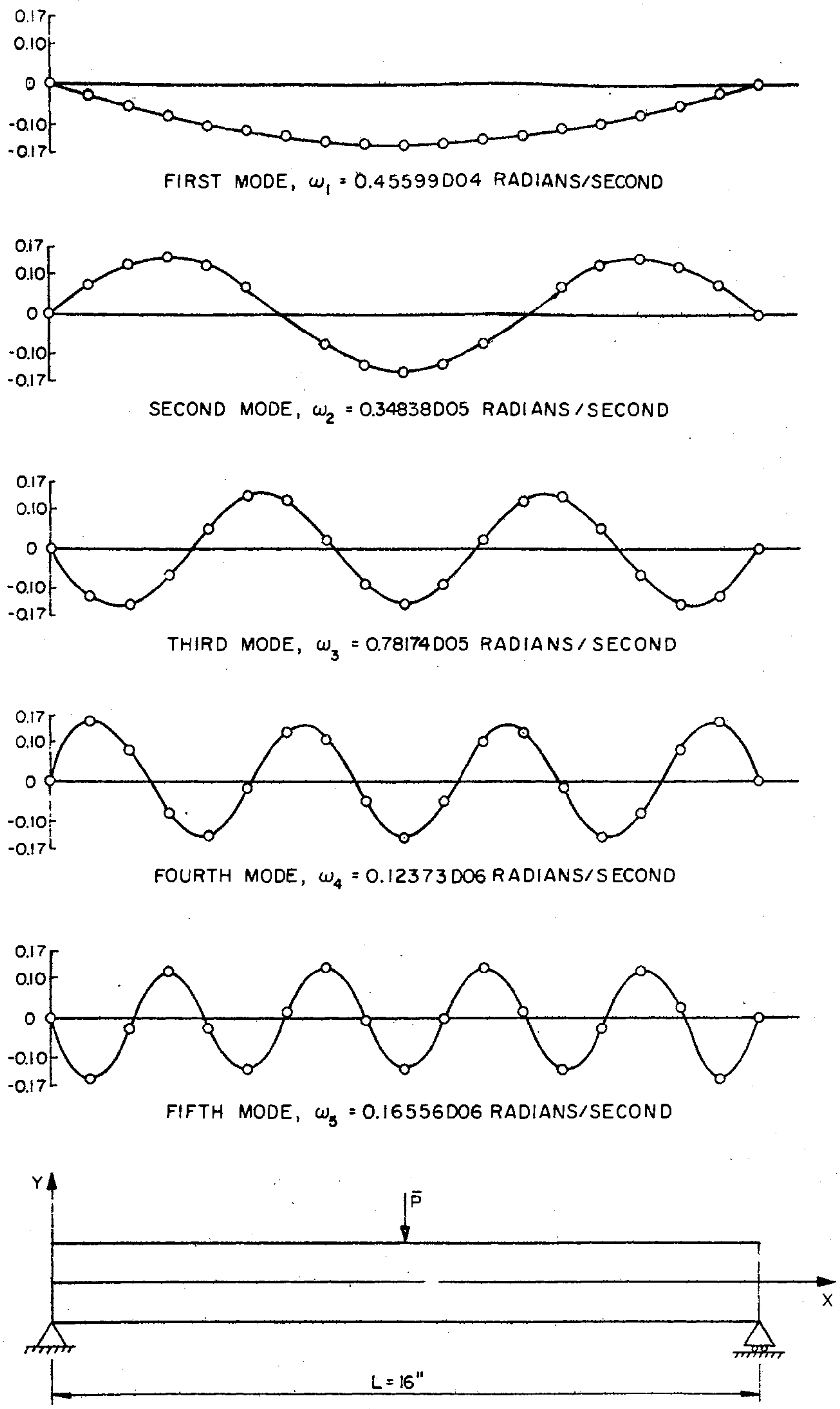


Figure 27. Mode Shapes for the First Five Modes

3. Superposition of normal flexural modes: The superposition of all the 47 normal modes and the five normal modes corresponding to the lowest frequencies was accomplished in accordance with section 4.5 by the computer program FLXMOD. The displacements obtained from both superpositions of the normal modes are given in Table VI. The displacements obtained from the superposition of 47 flexural modes are also illustrated graphically in Figure 26.

5.3.2 Step-By-Step Integration Method

1. Computer program: This computer program is the modification of the computer program (82) written for the static analysis of structures. The computer program was modified to suit the requirements of the problems considered in this report. Since a large number of subscripted variables are involved in this program, it was preferred to take advantage of the banded nature of the matrix. This procedure required a large amount of reading and writing on the magnetic disk; and, especially, the time increment in the process of integration is usually very small, which increases the use of the magnetic disc considerably and thus the total computer time. Since this program is designed to keep only a small portion of the matrices in the main core memory at a time, the program has a capability of solving problems of a large magnitude.

2. Selection of time increment Δt: Newmark (51) has suggested that, for the linear acceleration method, Δt should be less than one tenth of the smallest period of the structure. Since the smallest period of the structure is not usually known, several different time increments

TABLE VI

COMPARISON OF DISPLACEMENTS AT MID-SPAN OBTAINED FROM CLOSED FORM SOLUTION, MODE SUPERPOSITION AND STEP-BY-STEP INTEGRATION METHODS

Number of Time Steps	Time (Seconds)	Closed Form Solution (inches)	Superposition of first 5 Flexural Modes (inches)	Superposition of all 47 Flexural Modes (inches)	Step-By-Step Integration (inches)
1	0.0	0.0	0.0	0.0	0.0
2	0.005 D-02	0.0553 D-05	0.0626 D-05	0.0664 D-05	0.0665 D-05
3	0.010 D-02	0.1536 D-05	0.1681 D-05	0.1725 D-05	0.1725 D-05
4	0.015 D-02	0.2807 D-05	0.2852 D-05	0.2896 D-05	0.2893 D-05
5	0 020 D-02	0.5011 D-05	0.4852 D-05	0.4900 D-05	0.4901 D-05
6	0.025 D-02	0.7503 D-05	0.7462 D-05	0.7505 D-05	0.7510 D-05
7	0.030 D-02	0.9917 D-05	1.0067 D-05	1.0108 D-05	1.0113 D-05
8	0.035 D-02	1.2866 D-05	1.2561 D-05	1.2612 D-05	1.2618 D-05
9	0.040 D-02	1.5790 D-05	1.5396 D-05	1.5435 D-05	1.5434 D-05
10	0.045 D-02	1.8188 D-05	1.8278 D-05	1.8310 D-05	1.8314 D-05
11	0.050 D-02	2.0685 D-05	2.0307 D-05	2.0338 D-05	2.0345 D-05
12	0.055 D-02	2.2854 D-05	2.2025 D-05	2.2054 D-05	2.2054 D-05
13	0.060 D-02	2.4095 D-05	2.3733 D-05	2.3788 D-05	2.3780 D-05
14	0.065 D-02	2.5075 D-05	2.4522 D-05	2.4568 D-05	2.4567 D-05
15	0.070 D-02	2.5595 D-05	2.4450 D-05	2.4492 D-05	2.4496 D-05
16	0.075 D-02	2.5013 D-05	2.4038 D-05	2.4086 D-05	2.4089 D-05
17	0.080 D-02	2.4030 D-05	2.3211 D-05	2.3264 D-05	2.3255 D-05
18	0.085 D-02	2.2727 D-05	2.1550 D-05	2.1590 D-05	2.1589 D-05
19	0.090 D-02	2.0492 D-05	1.9178 D-05	1.9230 D-05	1.9235 D-05
20	0.095 D-02	1.8009 D-05	1.6951 D-05	1.6986 D-05	1.6989 D-05
21	0.100 D-02	1.5577 D-05	1.4516 D-05	1.4548 D-05	1.4551 D-05
22	0.105 D-02	1.2604 D-05	1.1399 D-05	1.1443 D-05	1.1437 D-05
23	0.110 D-02	0.9704 D-05	0.8647 D-05	0.8685 D-05	0.8693 D-05
24	0.115 D-02	0.7319 D-05	0.6430 D-05	0.6475 D-05	0.6485 D-05
25	0.120 D-02	0.4819 D-05	0.4145 D-05	0.4192 D-05	0.4200 D-05
26	0.125 D-02	0.2685 D-05	0.2133 D-05	0.2179 D-05	0.2177 D-05
27	0.130 D-02	0.1471 D-05	0.0964 D-05	0.1013 D-05	0.1010 D-05
28	0.135 D-02	0.0497 D-05	0.0567 D-05	0.0618 D-05	0.0610 D-05
29	0.140 D-02	0.0012 D-05	0.0348 D-05	0.0379 D-05	0.0389 D-05
30	0.145 D-02	0.0619 D-05	0.0694 D-05	0.0735 D-05	0.0736 D-05
31	0.150 D-02	0.1602 D-05	0.2222 D-05	0.2254 D-05	0.2247 D-05

need to be tried until the proper time increment is found to give the desired degree of accuracy.

For the example presented here, a trial time increment was found using a procedure of three steps (Figure 28): (a) A group of four elements adjoining a node "i" was selected; (b) The boundary conditions were assumed, as shown in Figure 28; (c) The mass of node "i" was obtained by lumping one-fourth of the mass of each of the four elements then using the expression

$$\Delta t = \frac{T}{10} = \frac{1}{10}\left(\frac{2L^2}{\pi}\sqrt{\frac{m}{EI}}\right)$$

or

$$\Delta t = \frac{L^2}{5\pi}\sqrt{\frac{m}{EI}} \qquad (5.1)$$

where

$$m = \frac{\rho(A_1 + A_2)(B_1 + B_2)}{4g}$$

A_1 and A_2 = length of the elements in inches

B_1 and B_2 = height of the elements in inches

ρ = weight of the material in pounds per cubic inch (for steel, $\rho = 0.283565$ lbs/in^3)

g = 386.4 in/sec^2

L = $A_1 + A_2$, in inches

E = modulus of elasticity (for steel, $E = 30 \times 10^6$ psi)

I = moment of inertia, i.e., $(B_1 + B_2)^3/12$ in^4.

The dimensions of the elements in the example illustrated here are shown in Figure 29. Since the material used for the example is steel, a trial time increment Δt was found equal to 0.115×10^{-5} seconds from Equation (5.1).

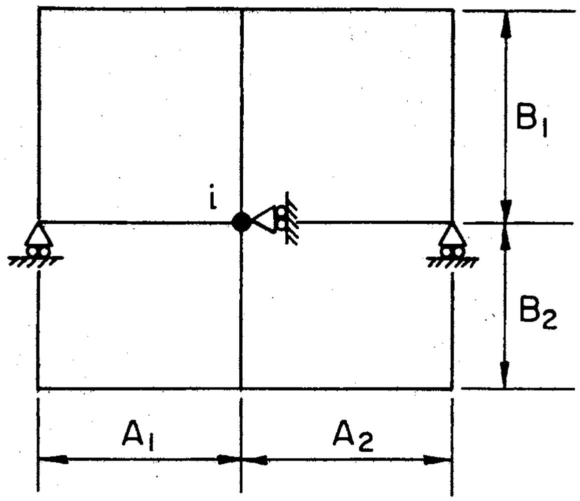


Figure 28. A General Four-Element Model for Δt

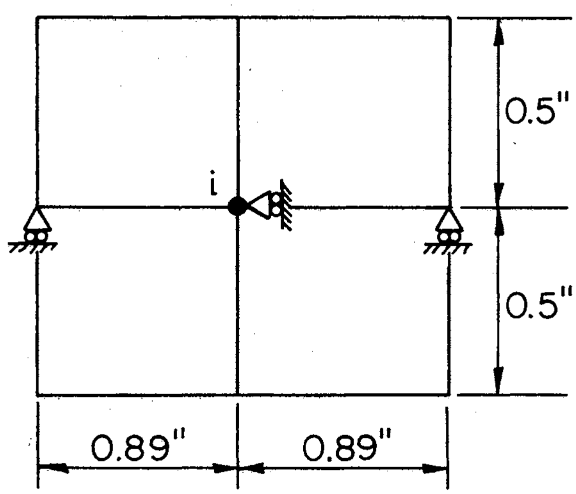


Figure 29. An Illustrated Four-Element Model for Δt

Two trial values of time increment Δt ($\Delta t = 0.1 \times 10^{-5}$ seconds and $\Delta t = 0.2 \times 10^{-5}$ seconds) were tested for the first few points of the deflection curve. The deflections obtained by using $\Delta t = 0.1 \times 10^{-5}$ seconds were almost the same as those obtained by the modal superposition method. The values of the deflections obtained by using $\Delta t = 0.2 \times 10^{-5}$ seconds were also in accord with the deflections obtained by the modal superposition method, but the acceleration and velocity values at each time increment were very much different from those obtained by using $\Delta t = 0.1 \times 10^{-5}$ seconds. This deviation in acceleration and velocity values causes one to suspect a possibility of instability of the solution at some later step of integration. Since, from modal analysis, the minimum period was found to be 0.733×10^{-5} seconds, the value of the time increment equal to 0.1×10^{-5} seconds was judged to be an adequate trial value to test the solution for a larger number of time increments. Tests for determining a larger number of points on the deflection curve for the other time increments were not attempted due to a prohibitive requirement of the computer time by the step-by-step integration method.

5.3.3 Computing Time

The computing time for the normal mode superposition method depends on the number of degrees of freedom of the system. For the step-by-step integration method, the computing time not only depends on the number of degrees of freedom of the system but also on the time increment Δt and on the number of steps of increments desired for the displacement curve.

The distribution of computer time for the two methods was as follows:

1. Normal mode superposition method:

Formulation of $\left[K\right]$ and $\left[M\right]$ matrices	17 seconds
Formulation of $\left[M\right]^{-1}\left[K\right]$ matrix	20 seconds
Computation of all eigenvalues and eigenvectors	1869 seconds
Selection and superposition of 47 flexural modes	37 seconds
Total	1943 seconds

Main core memory of 374 K-bytes was used.

2. Step-by step integration method:

Formulation of $\left[K\right]$ and $\left[M\right]$ matrices	17 seconds
Step-by-step integration	4986 seconds
Total	5003 seconds

Main core memory of 188 K-bytes was used.

5.4 Discussion of Results

For the problem considered here for illustration, although the continuous system is idealized by a medium-size mesh, the results shown in Table VI and in Figure 26 are in excellent agreement with the closed form solution. From the standpoint of the computational time, it is apparent from section 5.3 that the mode superposition of 47 modes required much less computer time than did the step-by-step integration method. It should be noted that the distribution of computer time given in section 5.3 depends on the specific computer used. The step-by-step integration program performs computations on blocks of the banded matrix, and the number and size of the records written on the magnetic disk are proportional to the number of elements in the band width of the matrices. For example, if two problems of an equal number of degrees of freedom are to be solved, the one having the larger band width would need more computer time.

Although, for the problem illustrated here, the modal analysis was found to be considerably more economical than the step-by-step integration method, the direct integration of the differential equation of vibration by the step-by-step procedure would be economical for large systems with loads varying with time, because the numerical integration of the convolution integral in the modal superposition would involve similar difficulties with the time step, Δt, as in the step-by-step integration method. Modal superposition is particularly recommended for undamped systems; but in special cases where the damping matrix is a linear combination of the mass and stiffness matrices, it can be used for damped systems. Furthermore, the modal superposition method is based on the assumption of linear structure, whereas the step-by-step method may be applied to nonlinear systems simply by modifying the assumed linear properties approximately at each successive step of integration.

One of the principal advantages of the mode superposition method lies in the fact that the response of the structure is largely expressed by the first few flexural modes of vibration of the system due to the presence of a large number of dilatory modes with varied frequencies. The program used here can easily be modified to extract only a certain number of modes, but it would be necessary to select the corresponding eigenvalues from all the eigenvalues of the system. Since determination of all the eigenvalues of the system is required, the modification may save only up to 25 percent of the computer time used in computation of all eigenvalues and eigenvectors (e. g., from section 5.3.3, a savings of approximately 470 seconds out of a total 1869 seconds).

CHAPTER VI

SUMMARY AND CONCLUSIONS

6.1 Summary

The primary aim of this investigation was to determine a suitable simple finite element and to evaluate its application to plane stress problems for the idealization of a structure under dynamic loads. The major items of this study included the convergence of solution, the requirement of computer time, and the storage in K-bytes of the core memory. To simplify the investigation, only the simple examples of beams with pinned supports and cantilever beams of isotropic material were considered.

For the static analysis, each problem was idealized as a structural model using plane stress rectangular elements. Seven configurations were analyzed using five types of rectangular elements. Four of these rectangular elements were composed of constant strain triangular elements, and the fifth element was the linear strain rectangular element. For all seven configurations, the analysis was performed using the IBM 360/65 digital computer, and the results are presented graphically to illustrate the convergence behavior.

To study the problem with dynamic loads, only a simply supported beam with an impact load of infinite duration at midspan was considered. The beam was idealized as a structural model using linear strain rectangular elements. The problem was solved by the mode superposition method and the step-by-step integration method to investigate the overall practicality of each method. All computations were performed using the IBM 360/65 digital computer.

6.2 Conclusions

From the study of the convergence, it was found that the behavior of the linear strain rectangular element is the most satisfactory of the five elements studied for the purpose. The results obtained for the problem with static loads were in excellent agreement with beam theory and elasticity solutions; convergence to the solution required fewer elements and less computation time than that required by the other elements.

On the basis of the above results, the linear strain rectangular element was used for the analysis of the problem with an impact load using the mode superposition method and the step-by-step integration method. Both methods of solution gave values of deflections which were almost the same and were found to be in excellent agreement with the closed form solution. The mode superposition method required considerably less computing time than did the step-by-step integration procedure.

Thus, from this investigation it was concluded that the linear strain rectangular element is a satisfactory element for the analysis of a plane stress problem with in-plane dynamic loads. Also, if the undamped structural system does not have a large number of degrees of freedom and is not subjected to time dependent varying loads, the mode superposition method would be an economical procedure. For a general case of a viscously damped system where damping is not a linear combination of the mass and stiffness matrices, the systematic integration of the step-by-step procedure may be comparatively more economical and practical than the mode superposition technique, which may require considerably more time and computer storage space in determining modes and frequencies. Determination of the modes and frequencies for a damped system of n degrees of freedom is equivalent to solving an eigenvalue problem of an undamped system of 2n degrees of freedom (44, page 206).

A SELECTED BIBLIOGRAPHY

(1) Archer, J. S. "Consistent Mass Matrix for Distributed Mass Systems." Proceedings, American Society of Civil Engineers, Vol. 89 (August, 1963), 161-178.

(2) Archer, J. S. "Consistent Matrix Formulations for Structural Analysis Using Finite Element Techniques." AIAA Journal, Vol. 3, No. 10 (October, 1965), 1910-1918.

(3) Argyris, J. H. "Energy Theorems and Structural Analysis." Aircraft Engineering, Vol. 26 (October-November, 1954), 345-356; Vol. 27 (February-May, 1955), 383-387.

(4) Argyris, J. H. "Continua and Discontinua." Opening paper presented at the Conference on Matrix Methods in Structural Mechanics, AFIT, Ohio, 1965.

(5) Argyris, J. H., S. Kelsey, and H. Kamel. "Matrix Methods of Structural Analysis: A Precis of Recent Developments." Published in "Matrix Methods of Structural Analysis." AGARDograph '72. Ed. F. De Veubeke. Oxford: Pergamon Press, 1964.

(6) Bishop, R. E. D. "The Analysis of Vibrating Systems Which Embody Beams in Flexure." Proceedings, Institution of Mechanical Engineers, Vol. 169, No. 51 (1955), 1031-1045.

(7) Bishop, R. E. D. "The Vibration of Frames." Proceedings, Institution of Mechanical Engineers, Vol. 170, No. 29 (1956) 955-967

(8) Clough, R. W. "Structural Analysis by Means of a Matrix Algebra Program." Proceedings, First Conference Electronic Computation, ASCE, Kansas City, Missouri. (November 20-21, 1958), 109-132.

(9) Clough R. W. "The Finite Element Method in Plane Stress Analysis." Proceedings, Second Conference Electronic Computation, ASCE, Pittsburgh, Pennsylvania (September 8-9, 1960), 345-377.

(10) Clough, R. W. "The Finite Element Method in Structural Mechanics." Stress Analysis. Ed. O. C. Zienkiewicz and G. S. Holister. New York: John Wiley and Sons, 1965.

(11) Clough, R. W. and A. K. Chopra. "Earthquake Stress Analysis in Earth Dams." Proceedings, ASCE, Journal of Engineering Mechanics Division (April, 1966), 197-211.

(12) Courant, R. "Variational Methods for the Solution of Problems of Equilibrium and Vibrations." Bulletin, American Mathematical Society, No. 49 (January, 1943), 1-23.

(13) Elias, Z. M. "Duality in Finite Element Methods." Proceedings, ASCE, Journal of Engineering Mechanics Division, Vol. 94, EM 4 (August, 1968), 931-946.

(14) Fellippa, C. A. "Refined Finite Element Analysis of Linear and Nonlinear Two-Dimensional Structures." SESM Report No, 66-22, Department of Civil Engineering, University of California at Berkeley, 1966.

(15) Fraeijs De Veubeke, B. M. "Upper and Lower Bounds in Matrix Structural Analysis." Published in AGARDograph '72. Ed. Fraeijs De Veubeke, B. M. Oxford: Pergamon Press, 1964.

(16) Fraeijs De Veubeke, B. M. "Displacement and Equilibrium Models in Finite Element Method." Stress Analysis. Eds. O. C. Zienkiewicz and G. S. Holister. New York: John Wiley and Sons, 1965.

(17) Francis, J. G. F. "The QR Transformation--A Unitary Analogue to the LR Transformation." Computer Journal, No. 4-3 (October, 1961), 265-271; No. 4-4 (January, 1962), 332-345.

(18) Franklin, H. A. "Nonlinear Analysis of Reinforced Concrete Frames and Panels." SESM Report No. 70-5, Department of Civil Engineering, University of California at Berkeley, March, 1970.

(19) Fulton, R., R. Eppink, and J. Walz. "The Accuracy of Finite Element Methods in Continuum Problems." Proceedings, Fifth U. S. National Congress of Applied Mechanics, ASME (June 17-19, 1966), 272.

(20) Gear, C. W. "A Sample Set of Test Matrices for Eigenvalue Programs." Mathematics of Computation, Vol. 23, No. 1 (January, 1969), 119-125.

(21) Gere, J. M. and W. Weaver. Analysis of Framed Structures. New Jersey: D. Van Nostrand Company, Inc., 1965.

(22) Grad, J. and M. A. Brebner. "Algorithm 343, Eigenvalues, and Eigenvectors of a Real General Matrix." Communications of the ACM 11 (December, 1968), 820-826.

(23) Grafton, P. E. and D. R. Strome. "Analysis of Axisymmetric Shells by the Direct Stiffness Method." AIAA Journal, Vol. 1, No. 10 (October, 1963), 2342-2347.

(24) Gregory, R. T. and D. L. Karney. A Collection of Matrices for Testing Computational Algorithms. New York: John Wiley and Sons, 1969.

(25) Guyan, R. J. "Distribution Mass Matrix for Plate Element Bending." AIAA Journal (March, 1965), 567-568.

(26) Henshell, R. D and G. B. Warburton. "Transmission of Vibration in Beam Systems." International Journal for Numerical Methods in Engineering, Vol. 1, No. 1 (January, 1969), 47-66.

(27) Hooley, R. F. and P. D. Hibbert. "Bounding Plane Stress Solution by Finite Elements." Proceedings, Journal of Structural Division, ASCE, ST 1, Vol. 65, No. 9 (February, 1966), 39-48.

(28) Hrennikoff, A. "Solutions of Problems of Elasticity by the Framework Method." Journal of Applied Mechanics (December, 1941), A.169-A.175.

(29) Hurty, W. C. "Vibrations of Structural Systems by Component Mode Synthesis." Proceedings, Journal of the Engineering Mechanics Division, ASCE, EM4, Vol. 86 (August, 1960), 51-69.

(30) Hurty, W. C. "Dynamic Analysis of Structural Systems Using Component Modes." AIAA Journal, Vol. 3, No. 4 (April, 1965), 678-685.

(31) Idriss, I. M. "Finite Element Analysis for the Seismic Response of Earth Banks." Proceedings, Journal of Soil Mechanics and Foundation Division, ASCE (May, 1968), 99-137.

(32) Irons, B. H. R. and K. J. Draper. "Inadequacy of Nodal Connections in a Stiffness Solution for Plate Bending." AIAA Journal, Vol. 3 (May, 1965), 961.

(33) Johnson, M. and R. McLay. "Convergence of the Finite Element Method in the Theory of Elasticity." Transactions, ASME Journal of Applied Mechanics, Vol. 90 (June, 1968), 274-278.

(34) Jones, R. E. "A Generalization of the Direct Stiffness Method of Structural Analysis." AIAA Journal, Vol, 2, No. 5 (May, 1964), 821-826.

(35) Jones, R. E. and D. R. Strome. "Direct Stiffness Method of Analysis Utilizing Curved Elements." AIAA Journal, Vol. 4, No. 9 (September, 1966), 1519-1525.

(36) Khanna, J. "Criterion for Selecting Stiffness Matrices." AIAA Journal. Vol. 3, No. 10 (October, 1965), 1976.

(37) Khanna, J. and R. Hooley. "Comparison and Evaluation of Stiffness Matrices." AIAA Journal, Vol. 4, No. 12 (December, 1966), 2105-2111.

(38) Knoble, H. D. "Certification of Algorithm 343, Eigenvalues, and Eigenvectors of a Real General Matrix." Communications of the ACM 13 (February, 1970), 122-124.

(39) Laursen, H. I., R. P. Shubinshi, and R. W. Clough. "Dynamic Matrix Analysis of Framed Structures." Proceedings, Fourth U. S. National Congress of Applied Mechanics, University of California at Berkeley (June 18-21, 1962), 99-105.

(40) Levy, Samuel. "Structural Analysis and Influence Coefficients for Delta Wings." Journal of the Aeronautical Sciences, Vol. 20 (July, 1953), 449-454.

(41) MacLeod, I. A. "New Rectangular Finite Element for Shear Wall Analysis." Journal of the Structural Division, ASCE, Vol. 95 ST 3 (March, 1969), 399-409.

(42) Marcel, P. V. and I. P. King. "Elastic-Plastic Analysis of Two-Dimensional Stress Systems by the Finite Element Method." International Journal of Mechanical Sciences, Vol. 9 (September, 1967), 143-155.

(43) McCormick, C. W. "Plane Stress Analysis." Journal of Structural Division, ASCE, ST 4 (August, 1963), 37-54.

(44) Meirovitch, Leonard. Analytical Methods in Vibrations. New York: The MacMillan Company, 1967.

(45) Melosh, R. J. "A Stiffness Matrix for the Analysis of Thin Plates in Bending." Journal of Aeronautical Sciences, Vol. 28 (1961) 34.

(46) Melosh, R. J. "Development of the Stiffness Method to Define Bounds on Elastic Behavior of Structures." (Ph.D. dissertation, University of Washington, 1962).

(47) Melosh, R. J. "Basis of Derivation of Matrices for the Direct Stiffness Method." AIAA Journal, Vol. 1, No. 7 (July, 1963), 1631-1637.

(48) Melosh, R. J. "Structural Analysis of Solids." Proceedings, Journal of Structural Division, ASCE, St 4 (August, 1963), 205-223.

(49) NASA. "Finite Element Approximation Accuracy and Extent of Errors in Structural Problems." Journal of Scientific and Technical Aerospace Reports, Vol. 8 (1970), 1741.

(50) Neubert, V. H. "Computer Methods for Dynamic Structural Response." Second Conference on Electronic Computation, ASCE, (September, 1960).

(51) Newmark, N. M. "A Method of Computation for Structural Dynamics." Proceedings, Journal of Engineering Mechanics Division, ASCE (July, 1959), 67-94.

(52) Ngo, D. and A. C. Scordelis. "Finite Element Analysis of Reinforced Concrete Beams." ACI Journal, No. 64-14 (March, 1967), 152-163.

(53) Nilson, A. H. "Nonlinear Analysis of Reinforced Concrete by Finite Element Method." ACI Journal (September, 1968), 757-766.

(54) Oliveira, E. R. A. "Completeness and Convergence in the Finite Element Method." Proceedings, Second Air Force Conference on Matrix Methods in Structural Mechanics, Wright-Patterson AFB, Ohio (1968).

(55) Oliveira, E. R. A. "Theoretical Foundations of the Finite Element Method." International Journal of Solids and Structures. Vol. 4, No. 10 (October, 1968), 929-951.

(56) Pian, T. H. H. "Derivation of Element Stiffness Matrices." AIAA Journal, Vol. 2, No. 3 (March, 1964), 576-577.

(57) Pian, T. H. H. "Element Stiffness Matrices for Boundary Compatibility and for Prescribed Boundary Stresses." Proceedings, First Conference on Matrix Methods in Structural Mechanics, Wright-Patterson AFB, Ohio (1965).

(58) Pian, T. H. H. "Rationalization in Deriving Element Stiffness Matrix by Assumed Stress Approach." Proceedings, Second Conference on Matrix Methods in Structural Mechanics, Wright-Patterson AFB, Ohio (1968).

(59) Pian, T. H. H. and P. Tong. "The Convergence of Finite Element Method in Solving Linear Elastic Problems." International Journal of Solids and Structures, Vol. 3 (March, 1967), 865-879.

(60) Pian, T. H. H. and P. Tong. "Basis of Finite Element Methods for Solid Continua." International Journal for Numerical Methods in Engineering, Vol. 1, No. 1 (January, 1969), 2-28.

(61) Prager, W. and J. L. Synge. "Approximations in Elasticity Based on the Concept of Function Space." Brown University-Quarterly Applied Mathematics, Vol. 5, No. 3 (October, 1947), 241-269.

(62) Przemieniecki, J. S. Theory of Matrix Structural Analysis. New York: McGraw-Hill Publishing Company, 1968.

(63) Rashid, Y. R. "Analysis of Axisymmetric Composite Structures by the Finite Element Method." Nuclear Engineering and Design, Vol. 3 (March, 1966), 163-182.

(64) Rayleigh, Lord. The Theory of Sound. Vol. 1. New York: Dover Publication, 1945.

(65) Reissner, E. "On a Variational Theorem in Elasticity." Journal of Mathematics and Physics, Vol. 29, No. 2 (July, 1950), 90-95.

(66) Rutishauser, Heinz. Handbook for Automatic Computation; Description of Algol 60. New York: Springer-Verlag, Inc., 1967.

(67) Salonen, E. M. "A Rectangular Plate Bending Element, the Use of Which is Equivalent to the Use of the Finite Difference Method." International Journal for Numerical Methods in Engineering, Vol. 1 (1969), 261-274.

(68) Stewart, G. W. "Incorporating Origin Shifts into the Symmetric QR Algorithm for Symmetric Tridiagonal Matrices." Communications of the ACM 13 (June, 1970), 365-367.

(69) Strickland, J. A., D. R. Navaratna, and T. H. H. Pian. "Improvements on the Analysis of Shells of Revolution by the Matrix Displacement Method." AIAA Journal, Vol. 4, No. 11 (November, 1966), 2069-2071.

(70) Synge, J. L. The Hypercircle in Mathematical Physics. Cambridge, England: Cambridge University Press, 1957.

(71) Temple, G. and W. G. Bickley. Rayleigh's Principle and Its Application to Engineering. New York: Dover Publication, 1956.

(72) Tocher, J. L. and W. Hartz. "A Higher Order Plane Stress Element." Proceedings, Journal of Engineering Mechanics Division, ASCE (August, 1967), 149-172.

(73) Turner, M. J. "The Direct Stiffness Method of Structural Analysis." AGARDograph '72, London: Pergamon Press, 1964.

(74) Turner, M. J., et al. "Stiffness and Deflection Analysis of Complex Structures." Journal of Aerospace Sciences, Vol. 23, No. 9 (September, 1956), 805-823, 854.

(75) Venkatraman, B. and S. A. Patel. Structural Mechanics with Introduction to Elasticity and Plasticity. New York: McGraw-Hill Book Company, 1970.

(76) Walz, J., R. E. Fulton, and N. J. Cyrus. "Accuracy and Convergence of Finite Element Approximations." Proceedings, Second Air Force Conference on Matrix Methods in Structural Mechanics, Wright-Patterson AFB, Ohio (1968).

(77) Walz, J., et al. "Accuracy of Finite Element Approximations to Structural Problems." Journal of Scientific and Technical Aerospace Reports, Vol. 8, No. 70-21708 (1970), 1741.

(78) Washizu, K. On Some Variational Principles of Elasticity and Plasticity. Oxford: Pergamon Press, 1968.

(79) Webster, J. J. "Free Vibration Analysis of Structures Using Rayleigh-Ritz and Finite Element Methods." Symposium on Structural Dynamics, Loughborough University of Technology, England, (March 23-25, 1970), B.1.1-B.1.18.

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

(81) Wilkinson, J. H. and C. Reinsch. Handbook for Automatic Computation; Linear Algebra. New York: Springer-Verlag, Inc., 1971.

(82) Wilson, E. L. "Finite Element Analysis of Two-Dimensional Structures." SESM Report No. 63-2, University of California at Berkeley, 1963.

(83) Wilson, E. L. and R. W. Clough. "Dynamic Response by Step-by-Step Matrix Analysis." Symposium on the Use of Computers in Civil Engineering, Paper No. 45, Lisbon, 1962.

(84) Zienkiewicz, O. C. The Finite Element Method in Engineering Science. New York: McGraw-Hill Book Company, 1971.

APPENDIX A

STIFFNESS MATRICES FOR ELEMENTS IN PLANE STRESS

A. 1 Constant Strain Triangular Element

The displacement function for any point within the triangular element can be uniquely given by the displacements, u and v. Thus, the nodal displacements shown in Figure 30 may be represented by

$$\left\{\delta\right\} = \left\{u_i\ u_j\ u_k\ v_i\ v_j\ v_k\right\}. \qquad (A.1)$$

The element displacement functions, u and v, for a constant-strain, triangular element are taken to be linear functions of x and y, as

$$u = \beta_1 + \beta_2 x + \beta_3 y \qquad (A.2a)$$

$$v = \beta_4 + \beta_5 x + \beta_6 y \qquad (A.2b)$$

which can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \left[\begin{array}{c|c|c|c|c|c} 1 & x & y & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & x & y \end{array}\right] \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} \qquad (A.2c)$$

Following the scheme summarized in Section 2.7, the nodal displacements $\left\{\delta\right\}$ in terms of the displacement parameters $\left\{\beta\right\}$ may be expressed as

$$\begin{bmatrix} u_i \\ u_j \\ u_k \\ v_i \\ v_j \\ v_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & a & 0 & 0 & 0 & 0 \\ 1 & c & b & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & a & 0 \\ 0 & 0 & 0 & 1 & c & b \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} \qquad (A.3a)$$

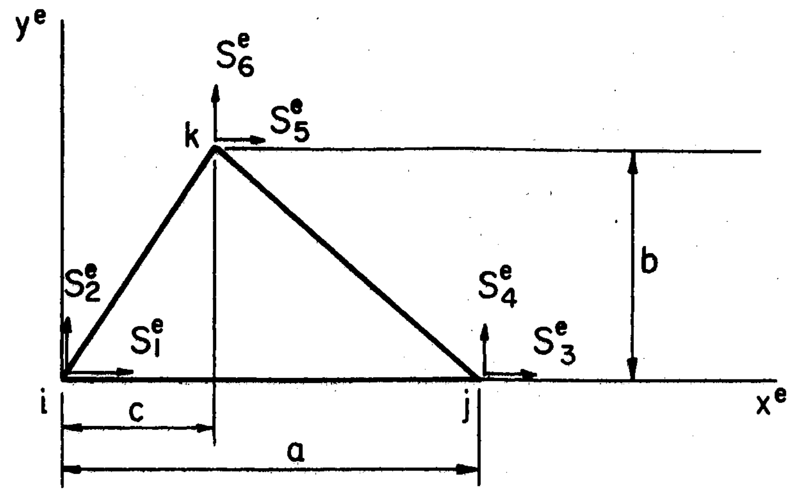


Figure 30. Triangular Plane Stress Element with Element Coordinate System

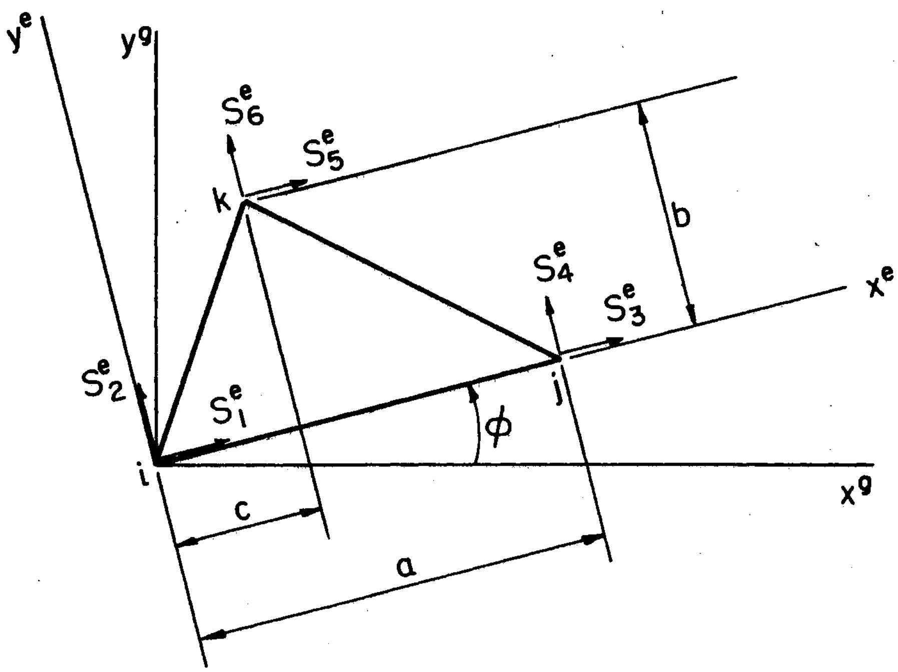


Figure 31. Triangular Plane Stress Element with Global Coordinate System

which can be written as

$$\{\delta\} = [A] \{\beta\}. \tag{A.3b}$$

The internal strains in the element are obtained by taking the partial derivative of displacements

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} \tag{A.4a}$$

which can be written as

$$\{\epsilon\} = [B] \{\beta\}. \tag{A.4b}$$

Since the elements of matrix $[B]$ are all constant, it follows that the strains within the element are constant. The stress-strain relationships for an isotropic material, for a plane stress element, with Poisson's ratio μ, are given by

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E}{1-\mu^2} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\mu) \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} \tag{A.5a}$$

which can be written as

$$\{\sigma\} = [D] \{\epsilon\}. \tag{A.5b}$$

The stiffness matrix of the element may be obtained by using step 11 in section 2.7:

$$[k] = [A^{-1}]^T \int_V [B]^T [D] [B] \, dV \, [A^{-1}] \tag{A.6}$$

Since the matrices $[B]$ and $[D]$ contain only constant terms, the expression for the stiffness matrix becomes

$$[k] = [A^{-1}]^T [B]^T [D] [B] \frac{t_e ab}{2} [A^{-1}] \qquad (A.7)$$

where t_e is the thickness and $\frac{t_e ab}{2}$ is the volume of the element.

Thus, from Equation (A.7), by substituting $\xi_1 = \frac{1}{2}(1 - \mu)$ and $\xi_2 = \frac{1}{2}(1 + \mu)$, the element stiffness matrix can be obtained as given on the following page in Equation (A.8a), and which can be written as

$$\{S^e\} = [k] \{\delta\} . \qquad (A.8b)$$

The stresses within the element are given by

$$\begin{aligned} \{\sigma\} &= [D] [B] [A]^{-1} \{\delta\} \\ &= [Q] \{\delta\} \end{aligned} \qquad (A.9)$$

where matrix $[Q]$ can be written as

$$\frac{E}{(1-\mu^2)ab} \left[\begin{array}{c|c|c|c|c|c} -b & \mu(c-a) & b & -\mu c & 0 & ba \\ \hline -\mu b & (c-a) & \mu b & -c & 0 & a \\ \hline \xi_1(c-a) & -\xi_1 b & -\xi_1 c & \xi_1 b & \xi_1 a & 0 \end{array} \right] \qquad (A.10)$$

A.2 Transformation to Global Coordinates

The relationship between the nodal forces in the global coordinate system and the element coordinate system may be obtained by coordinate transformation. The transformation of the stiffness matrix from $x^e - y^e$ element coordinate system to $x^g - y^g$ global coordinate system is carried out as shown in Equation (A.11a) and Figure 31.

$$
\begin{bmatrix} S_1^e \\ S_2^e \\ S_3^e \\ S_4^e \\ S_5^e \\ S_6^e \end{bmatrix}
= \frac{Et_e}{\frac{1}{2}(1-\mu^2)\,ab}
\begin{bmatrix}
b^2 + \xi_1(c-a)^2 & & & & \text{Symmetric} & \\
-\xi_2 b(c-a) & \xi_1 b^2 + (c-a)^2 & & & & \\
-b^2 - \xi_1 c(c-a) & \xi_2 bc - \mu ab & b^2 + \xi_1 c^2 & & & \\
\xi_1 b(c-a) + \mu bc & -\xi_1 b^2 - c(c-a) & -\xi_2 bc & \xi_1 b^2 + c^2 & & \\
\xi_1 a(c-a) & -\xi_1 ab & -\xi_1 ac & \xi_1 ab & \xi_1 a^2 & \\
-\mu ab & a(c-a) & \mu ab & -ac & 0 & a^2
\end{bmatrix}
\begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix}
\qquad \text{(A.8a)}
$$

$$\begin{bmatrix} S_1^g \\ S_2^g \\ S_3^g \\ S_4^g \\ S_5^g \\ S_6^g \end{bmatrix} = \begin{bmatrix} \text{Cos}\varphi & \text{Sin}\varphi & 0 & 0 & 0 & 0 \\ -\text{Sin}\varphi & \text{Cos}\varphi & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{Cos}\varphi & \text{Sin}\varphi & 0 & 0 \\ 0 & 0 & -\text{Sin}\varphi & \text{Cos}\varphi & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{Cos}\varphi & \text{Sin}\varphi \\ 0 & 0 & 0 & 0 & -\text{Sin}\varphi & \text{Cos}\varphi \end{bmatrix} \begin{bmatrix} S_1^e \\ S_2^e \\ S_3^e \\ S_4^e \\ S_5^e \\ S_6^e \end{bmatrix} \quad \text{(A.11a)}$$

This can be written as

$$\{S^g\} = [R]\{S^e\} . \quad \text{(A.11b)}$$

Similarly,

$$\{\delta^g\} = [R]\{\delta^e\} , \quad \text{(A.12)}$$

and therefore,

$$[\bar{k}] = [R][k][R]^T . \quad \text{(A.13)}$$

A. 3 <u>Linear Strain Rectangular Element</u>

The procedure used to develop the stiffness matrix for this element (Figure 7) follows closely to the outline given in section 2.7. The element displacement functions u and v are taken to be

$$u = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 xy \quad \text{(A.14)}$$

$$v = \beta_5 + \beta_6 x + \beta_7 y + \beta_8 xy . \quad \text{(A.15)}$$

Thus, the nodal displacements $\{\delta\}$ can be expressed in terms of the displacement parameters $\{\beta\}$ as

$$\begin{bmatrix} u_i \\ u_j \\ u_\ell \\ u_k \\ v_i \\ v_j \\ v_\ell \\ v_k \end{bmatrix} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & & & & \\ 1 & a & 0 & 0 & & 0 & & \\ 1 & 0 & b & 0 & & & & \\ 1 & a & b & ab & & & & \\ \hline & & & & 1 & 0 & 0 & 0 \\ & 0 & & & 1 & a & 0 & 0 \\ & & & & 1 & 0 & b & 0 \\ & & & & 1 & a & b & ab \end{array}\right] \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \end{bmatrix} \qquad \text{(A.16a)}$$

which can be written as

$$\{\delta\} = [A]\{\beta\}. \qquad \text{(A.16b)}$$

The internal strains in the element are

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} = \left[\begin{array}{c|c|c|c|c|c|c} 0 & 1 & 0 & y & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & x \\ \hline 0 & 0 & 1 & x & 1 & 0 & y \end{array}\right] \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \end{bmatrix} \qquad \text{(A.17a)}$$

which can be written as

$$\{\epsilon\} = [B]\{\beta\}. \qquad \text{(A.17b)}$$

The stress-strain relationships in an isotropic material for the plane stress element are given by

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1-\mu^2)} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\mu) \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \qquad (A.18a)$$

which can be written as

$$\{\sigma\} = [D]\{\epsilon\} . \qquad (A.18b)$$

The stiffness matrix of the element may now be obtained by using step 11 in section 2.7, as

$$[k] = [A^{-1}]^T \int_V [B]^T [D] [B] \, dV \, [A^{-1}] . \qquad (A.19)$$

Thus, from Equation (A.19) by substituting $\xi_1 = \frac{1}{2}(1-\mu)$ and $\xi_2 = \frac{1}{2}(1+\mu)$, the element stiffness matrix can be obtained as given on the following page in Equation (A.20).

The stresses within the element are given by

$$\{\sigma\} = [D][B][A]^{-1}\{\delta\} \qquad (A.21)$$

$$= [Q]\{\delta\}$$

where matrix $[Q]$ may be written as

$$\frac{E}{(1-\mu^2)ab} \begin{bmatrix} (y-b) & \mu(x-a) & (b-y) & -\mu x & y & \mu x & -y & \mu(a-x) \\ \mu(y-b) & (x-a) & \mu(b-y) & -x & \mu y & x & -\mu y & (a-x) \\ \xi_1(x-a) & \xi_1(y-b) & -\xi_1 x & \xi_1(b-y) & \xi_1 x & \xi_1 y & \xi_1(a-x) & -\xi_1 y \end{bmatrix}$$

(A.22)

$$
[k] = \frac{Et_e}{12(1-\mu^2)AB}
\begin{bmatrix}
4(b^2+\xi_1 a^2) & & & & & & & \\
3ab\xi_2 & 4(a^2+\xi_1 b^2) & & & & \text{Symmetric} & & \\
2(\xi_1 a^2 - 2b^2) & 3ab(\xi_1-\mu) & 4(b^2+\xi_1 a^2) & & & & & \\
3ab(\mu-\xi_1) & 2(a^2-2\xi_1 b^2) & -3ab\xi_2 & 4(a^2+\xi_1 b^2) & & & & \\
-2(b^2+\xi_1 a^2) & -3ab\xi_2 & 2(b^2-2\xi_1 a^2) & 3ab(\xi_1-\mu) & 4(b^2+\xi_1 a^2) & & & \\
-3ab\xi_2 & -2(a^2+\xi_1 b^2) & 3ab(\mu-\xi_1) & 2(\xi_1 b^2-2a^2) & 3ab\xi_2 & 4(a^2+\xi_1 a^2) & & \\
2(b^2-2\xi_1 a^2) & 3ab(\mu-\xi_1) & -2(b^2+\xi_1 a^2) & 3ab\xi_2 & 2(\xi_1 a^2-2b^2) & 3ab(\xi_1-\mu) & 4(b^2+\xi_1 a^2) & \\
3ab(\xi_1-\mu) & 2(\xi_1 b^2-2a^2) & 3ab\xi_2 & -2(a^2+\xi_1 b^2) & 3ab(\mu-\xi_1) & 2(a^2-2\xi_1 b^2) & -3ab\xi_2 & 4(a^2+\xi_1 b^2)
\end{bmatrix}
\quad (A.20)
$$

APPENDIX B

FLOW CHART AND LISTING OF COMPUTER PROGRAMS FOR STATIC ANALYSIS

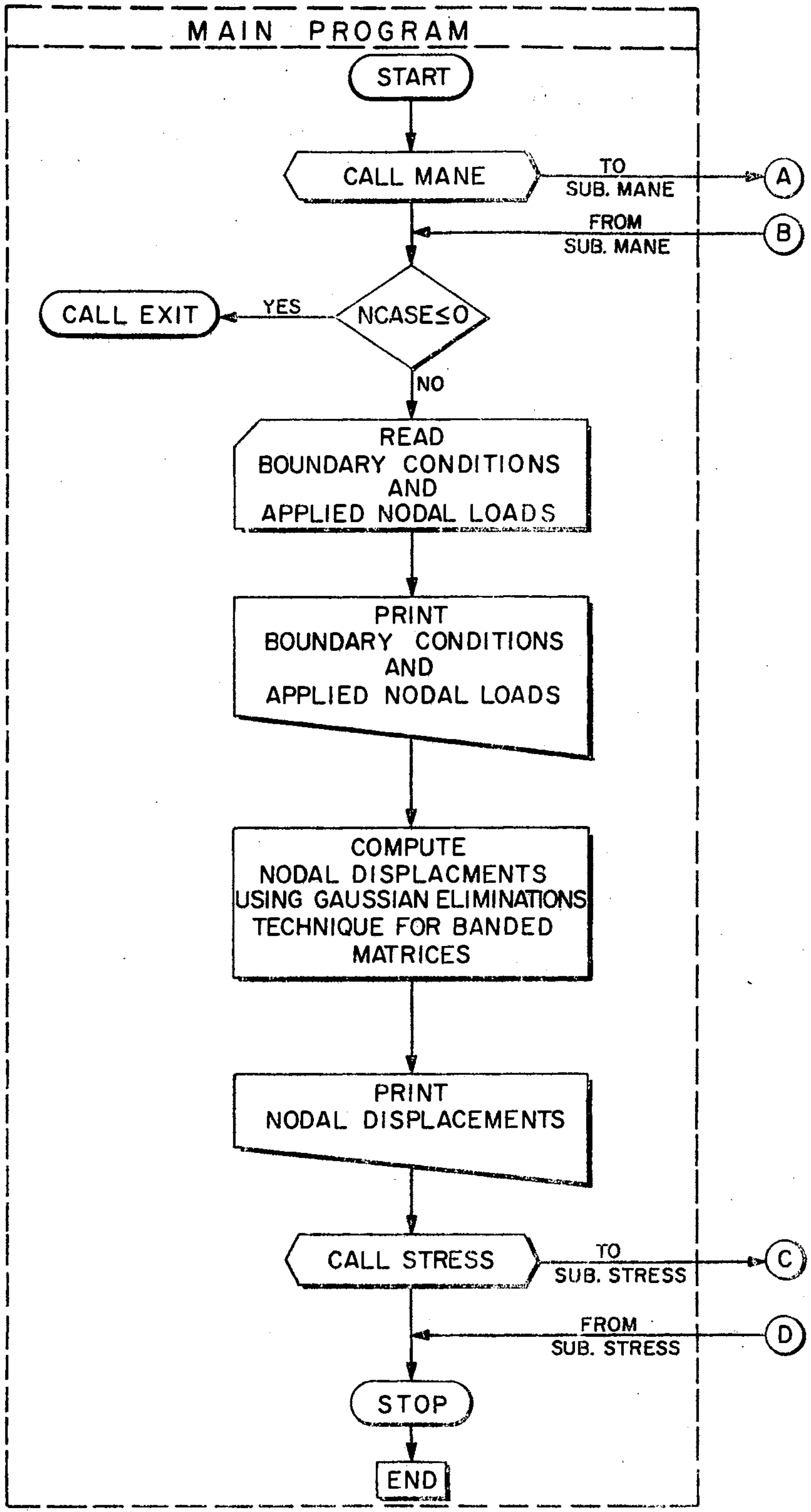


Figure 32. Flow Chart for Static Analysis

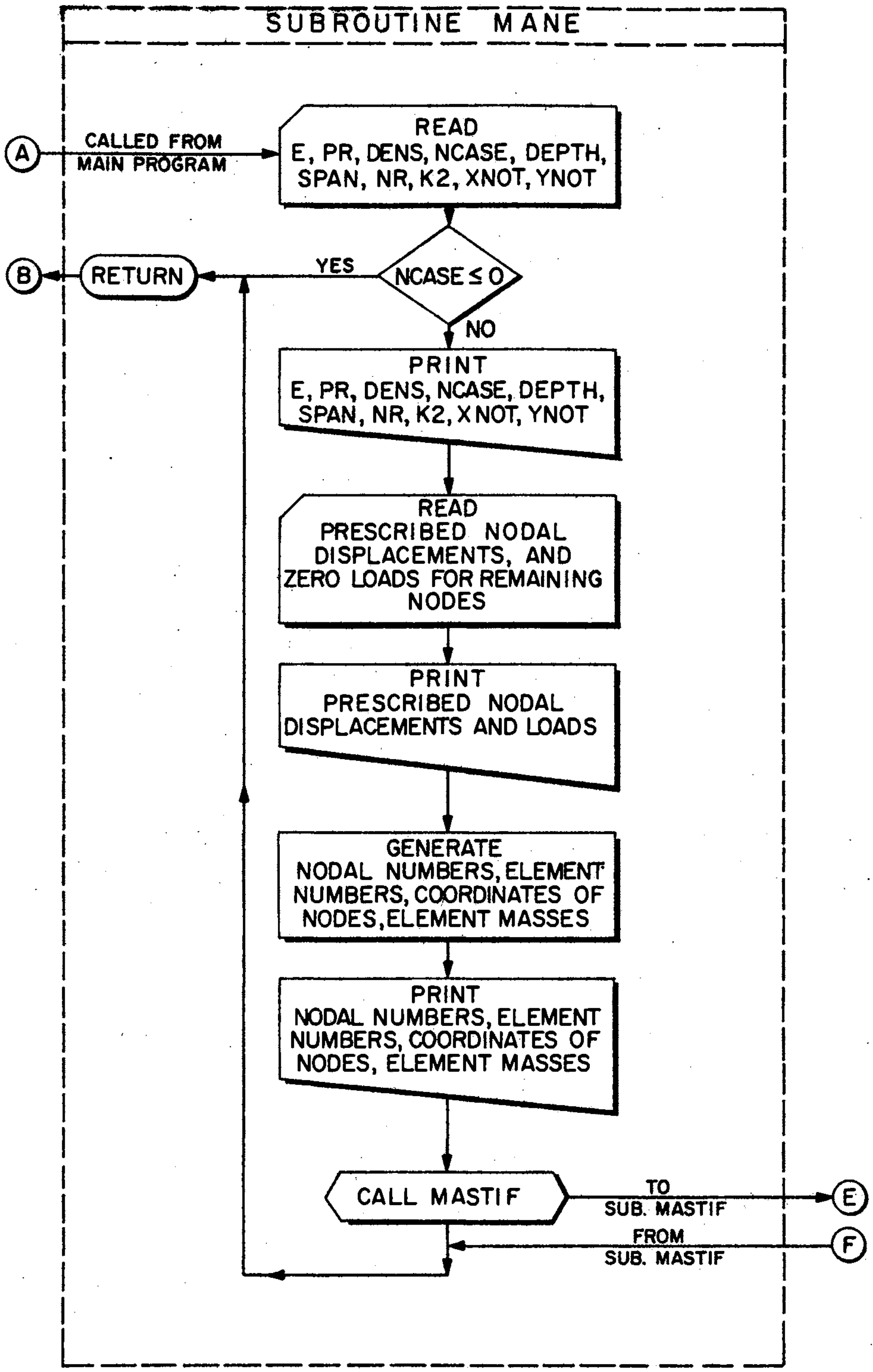


Figure 32. (Continued)

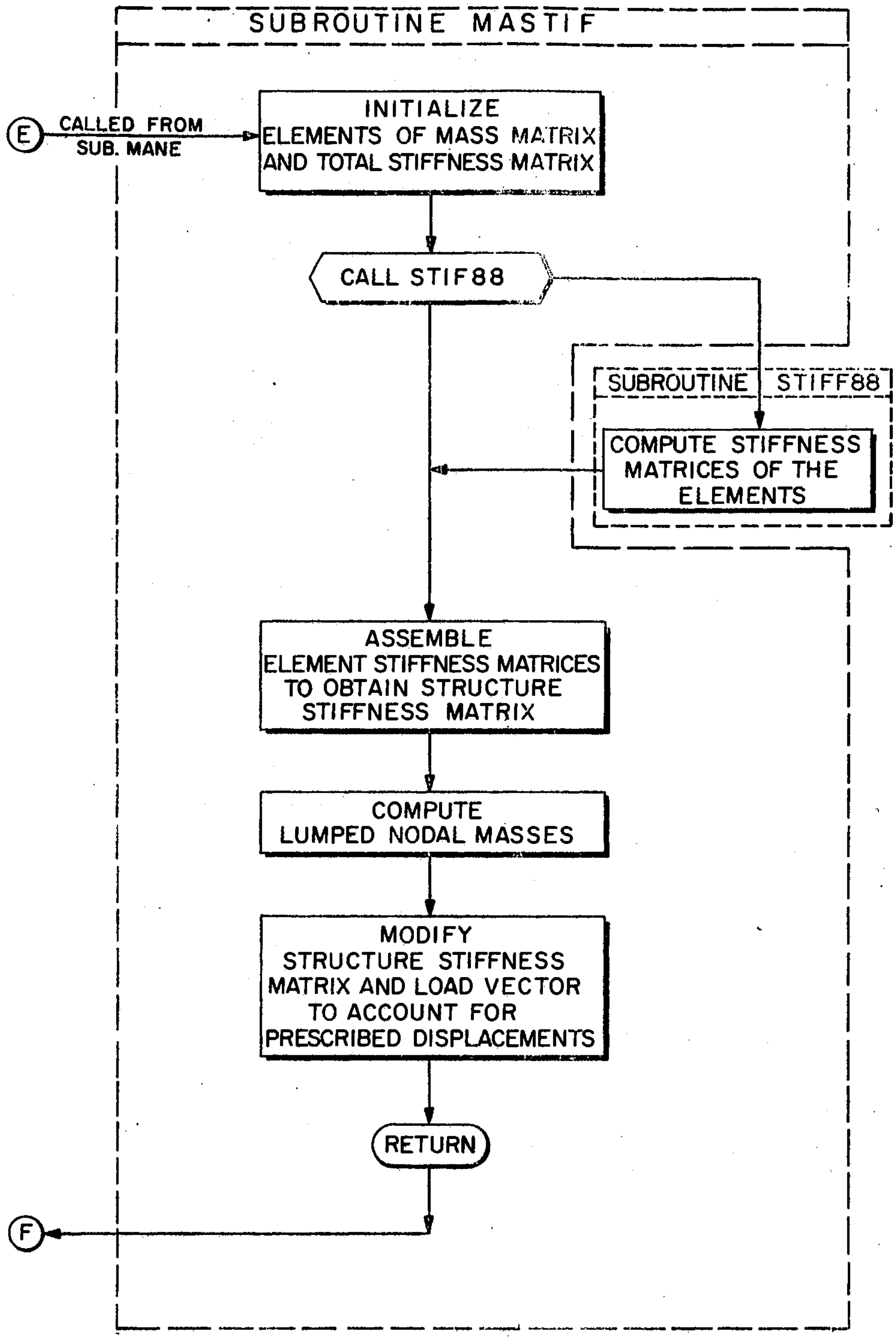


Figure 32. (Continued)

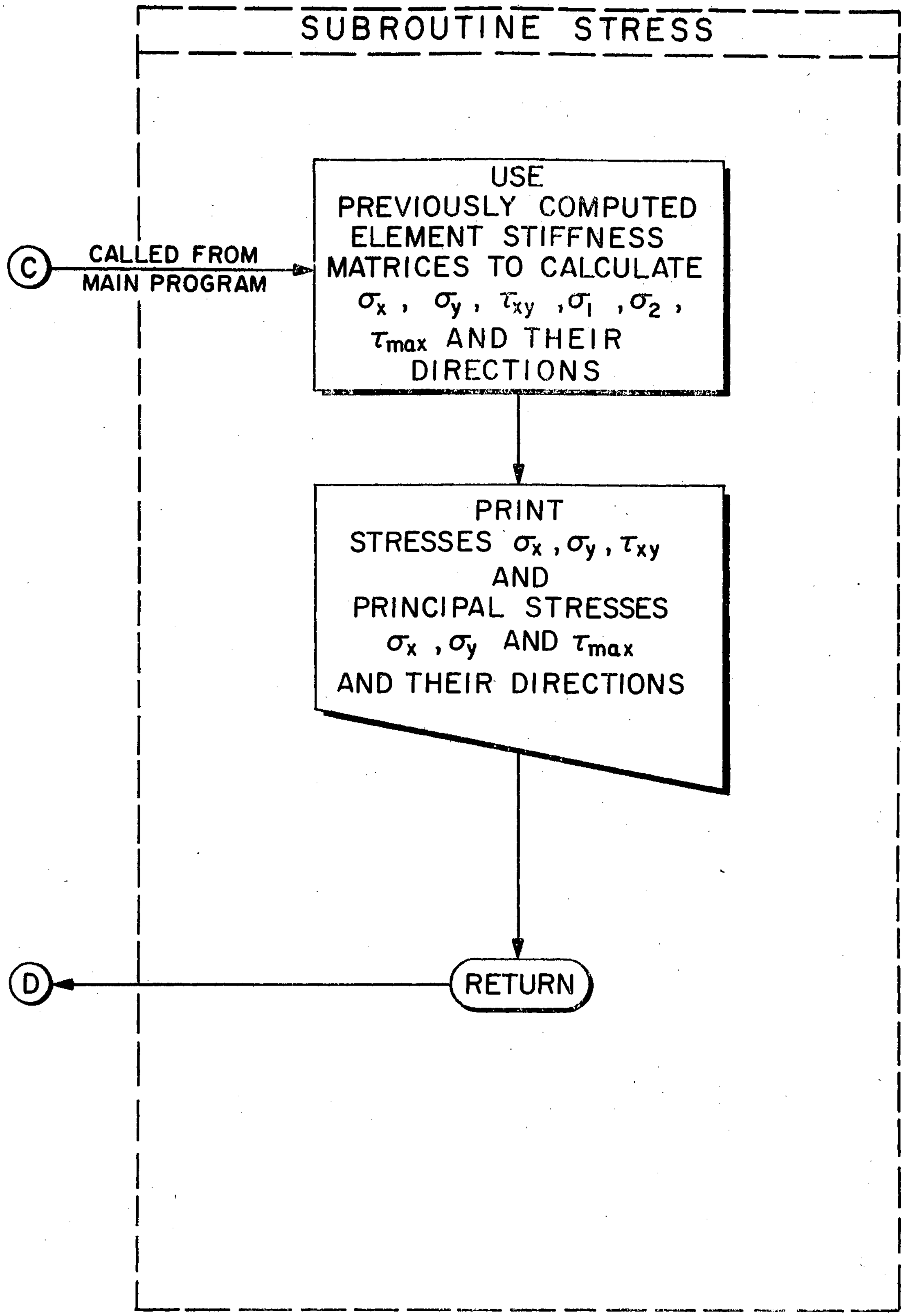


Figure 32. (Continued)

```
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    +                                                                      +
C    + PROGRAM                                                              +
C    +                                                                      +
C    + ''STATIC ANALYSIS USING FIVE TYPES OF FINITE ELEMENTS IN             +
C    + SEVEN CONFIGURATIONS''                                               +
C    +                                                                      +
C    + LANGUAGE              :  FORTRAN IV                                  +
C    + DIGITAL COMPUTER      :  IBM 360/65                                  +
C    + PROGRAMMER            :  BRIJ R. KISHORE                             +
C    +                          STRUCTURAL ENGINEER                         +
C    +                          U. S. ARMY, CORPS OF ENGINEERS              +
C    +                          CHICAGO, ILLINOIS                           +
C    +                                                                      +
C    + PURPOSE                                                              +
C    +                                                                      +
C    + THIS PROGRAM PERFORMS THE STATIC ANALYSIS OF THE PLANE               +
C    + STRESS PROBLEMS. DETAILED INFORMATION CAN BE FOUND IN:               +
C    + ''CHARACTERISTICS OF FINITE ELEMENTS FOR ANALYSIS OF                 +
C    + BEAMS SUBJECTED TO IMPACT LOAD'', PH.D. DISSERTATION                 +
C    + BY BRIJ R. KISHORE, SCHOOL OF CIVIL ENGINEERING,                     +
C    + OKLAHOMA STATE UNIVERSITY, JULY 1972.                                +
C    +                                                                      +
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    +                                                                      +
C    + DESCRIPTION OF PARAMETERS:                                           +
C    +                                                                      +
C    + MM            = BLOCK WIDTH                                          +
C    + NN            = BLOCK LENGTH                                         +
C    + IY            = NUMBER OF NODAL LOADS IN Y-DIRECTION                 +
C    + LOADY (I)     = NODAL NUMBERS WHICH HAVE LOAD IN Y-DIRECTION         +
C    + YLOAD (I)     = MAGNITUDE OF LOAD AT NODE LOADY (I)                  +
C    + JX            = NUMBER OF NODAL LOADS IN X-DIRECTION                 +
C    + LOADX (I)     = NODAL NUMBERS WHICH HAVE LOAD IN X-DIRECTION         +
C    + XLOAD (I)     = MAGNITUDE OF LOAD AT NODE LOADX (I)                  +
C    + ISUNIF        = LOGICAL VARIABLE;  .TRUE. IF BEAM HAS ANY            +
C    +                 UNIFORMLY DISTRIBUTED LOAD, OTHERWISE .FALSE.        +
C    + UNIFLD        = MAGNITUDE OF TOTAL UNIFORMLY DISTRIBUTED LOAD        +
C    + NR            = NUMBER OF VERTICAL LINES IN THE FINITE               +
C    +                 ELEMENT MESH                                         +
C    + B (N)         = LOAD VECTOR                                          +
C    + A(N,M)        = BLOCKS OF STIFFNESS MATRIX                           +
C    + NBLK          = NUMBER OF BLOCKS                                     +
C    + NUMNP         = NUMBER OF NODAL POINTS                               +
C    + FFM(I1,I2)    = MASS MATRIX                                          +
C    + HED           = HEADING                                              +
C    + E             = MODULUS OF ELASTICITY                                +
C    + PR            = POISSON'S RATIO                                      +
C    + DEPTH         = DEPTH OF BEAM                                        +
C    + WIDTH         = WIDTH OF BEAM                                        +
C    + SPAN          = SPAN OF BEAM                                         +
C    + NUMEL         = NUMBER OF ELEMENTS                                   +
C    + X             = X-COORDINATE OF NODAL POINTS                         +
C    + Y             = Y-COORDINATE OF NODAL POINTS                         +
C    + UX            = INITIAL LOAD OR DISPLACEMENT VALUE AT NODES          +
C    +                 IN X-DIRECTION                                       +
C    + UY            = INITIAL LOAD OR DISPLACEMENT VALUE AT NODES          +
C    +                 IN Y-DIRECTION                                       +
C    + NCASE         = TYPE OF FINITE ELEMENT CONFIGURATION                 +
```

```
C    + MBAND         = BAND WIDTH                                           +
C    +                                                                      +
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /AAA/ X(343),Y(343),UX(343),UY(343),UXTYPE(343),UYTYPE(343)
     1,ELMAS(288),HED(18),TYPE(8),E,DENS,PR,VOL,MTYPE(288),NUMNP,
     2NUMEL,NUMAT,KN,NCASE,KOUNT
      COMMON /ARG/ XXX(5),YYY(5),S(10,10),DD(3,3),HH(6,10),P(10),XX(4),
     1YY(4),C(4,4),H(6,10),D(6,6),F(6,10),
     2TYPE1,TYPE2,TEST1,TEST2,IX(288,4),LM(4),NR,LIMIT,ISTART
      COMMON /BANARG/ A(36,18),FM(36),B(36),MBAND,NUMBLK
      COMMON/AIJFM/AA(40,18,18),FFM(40,18)
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      COMMON /AZZA/ AZZ(18,40)
      COMMON/SQUAD/SQ(10,10)
      DIMENSION YLOAD(9),XLOAD(9),LOADY(9),LOADX(9),NBCY(9),NBCX(9)
     1,KDISP(18,40)
      LOGICAL *1 ISUNIF
C
      KOUNT=0
    1 CONTINUE
      CALL ELAPSE(I)
      PRINT 3224, I
      KOUNT=KOUNT+1
      CALL MANE
      CALL ELAPSE(I)
      PRINT 3224, I
C
C******NCASE BEING READ IN SUBROUTINE MANE
      IF(NCASE.EQ.0) GOTO 9999
C
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
      MBAND=2*LIMIT
      MM=MBAND
      NN=MBAND
      NL=NN+1
      NH=2*NN
      NUMBLK=0
C
C
      READ(5,3224)IY
      IF(IY.EQ.0) GO TO 10
      READ(5,13)(LOADY(I),YLOAD(I),I=1,IY)
      PRINT 3, IY,(LOADY(I),YLOAD(I),I=1,IY)
      DO 6 I=1,IY
    6 UY(LOADY(I))=YLOAD(I)
   10 CONTINUE
      READ(5,3224)JX
      IF(JX.EQ.0) GO TO 11
      READ(5,13)(LOADX(I),XLOAD(I),I=1,JX)
      PRINT 3, JX,(LOADX(I),XLOAD(I),I=1,JX)
      DO 7 I=1,JX
    7 UX(LOADX(I))=XLOAD(I)
   11 CONTINUE
      READ 20, ISUNIF,UNIFLD
```

```
      IF(ISUN(F) GOTO 25
      GOTC 35
   25 YI=UNIFLD/DFLOAT(NR-1)
      I=LIMIT-2
C
C*****IF DISTRIBUTED LOAD IS ON NEUTRAL SURFACE LJ=1+I/2
C     LJ=I-I/2
C*****IF DISTRIBUTED LOAD IS CN TOP SURFACE LJ=I
      LJ=I
C
      IF(UYTYPE(LJ).EQ.TYPE1)UY(LJ)=UY(LJ)+YI/TWO
      JI=(NR-1)*I+LJ
      IF(UYTYPE(JI).EQ.TYPE1)UY(JI)=UY(JI)+YI/TWO
      DO 30 J=3,NR
      JI=(J-2)*I+LJ
      IF(UYTYPE(JI).EQ.TYPE1)UY(JI)=UY(JI)+YI
   30 CONTINUE
      PRINT 3244,(J,UY(J),J=LJ,NUMNP,I)
   35 CONTINUE
C
C*****INITIALIZE MATRICES B(N) AND A(N,M)
C
      DO 5C N=1,NH
      B(N)=ZERO
      DC 50 M=1,NN
   5C A(N,M)=ZERO
C
C     COMPUTE 'NBLK' ............
C
      NBLK=NUMNP/LIMIT
      I=MOD(NUMNP,LIMIT)
      IF(I.GT.0) NBLK=NBLK+1
      PRINT 3224,NBLK
C
C     INITIALIZE & READ IN TOTAL STIFFNESS MATRIX
C
      DO 691 I1=1,NBLK
      DO 691 I2=1,NN
      DO 691 I3=1,NN
  691 AA(I1,I2,I3)=ZERC
C
C
      DO 693 I1=1,NBLK
      READ(3)(FFM(I1,I2),I2=1,NN)
      READ(2) N
      PRINT3224,N
      READ(4)(I2,I3,AA(I1,I2,I3),I4=1,N)
      GC TC 124
C
C*****SHIFT BLOCK OF EQUATIONS & MODIFY EQUATIONS BY BLOCKS
C
  122 NUMBLK=NUMBLK+1
      NS=LIMIT*(NUMBLK+1)
      NK=NS-LIMIT
      NP=NK-LIMIT+1
      KSHIFT=2*NP-2
      IF(NK.GT.NUMNP)NK=NUMNP
C
      DO 123 N=1,NN
      NM=NN+N
      B(N)=B(NM)
      B(NM)=ZERO
      DO 123 M=1,MM
      A(N,M)=A(NM,M)
  123 A(NM,M)=ZERO
      IF(NUMBLK .EQ. NBLK) GO TO 126
  124 CONTINUE
      N=NL-1
      DO 711 I2=1,NN
      N=N+1
      DO 711 I3=1,MM
      A(N,I3)=AA(NUMBLK+1,I2,I3)
  711 CONTINUE
      IF(NUMBLK .EQ. 0) GO TO 122
C
C*****ADD CONCENTRATED FORCES
C
  126 CONTINUE
      DO 250 N=NP,NK
      K=2*N-KSHIFT
      IF(UYTYPE(N) .NE. TYPE1) GO TO 240
      B(K)=B(K)+UY(N)
  240 IF(UXTYPE(N) .NE. TYPE1) GO TO 250
      B(K-1)=B(K-1)+UX(N)
  250 CONTINUE
C
C*****BOUNDARY CONDITIONS
C
      DO 410 M=NP,NK
      IF(M-NUMNP)315,315,410
  315 N=2*M-KSHIFT-1
      IF(UXTYPE(M) .NE. TYPE2) GO TO 320
      U=UX(M)
C
      CALL MODIFY(NH,N,U)
C
  320 N=N+1
      IF(UYTYPE(M) .NE. TYPE2) GO TO 410
      U=UY(M)
      CALL MODIFY(NH,N,U)
  410 CONTINUE
C
C*****REDUCE EQUATIONS BY BLOCKS
C
  200 DO 300 N=1,NN
      IF(A(N,1)) 225,300,225
  225 B(N)=B(N)/A(N,1)
      DO 275 L=2,MM
      IF(A(N,L)) 230,275,230
  230 Q=A(N,L)/A(N,1)
      I=N+L-1
      J=0
      DO 255 K=L,MM
      J=J+1
  255 A(I,J)=A(I,J)-Q*A(N,K)
      B(I)=B(I)-A(N,L)*B(N)
      A(N,L)=Q
  275 CONTINUE
```

```
  300 CONTINUE
      IF(NUMBLK-NBLK)375,400,375
  375 DO 720 N=1,NN
      WRITE(1) B(N),(A(N,M),M=2,MM)
  720 CONTINUE
      GO TO 122
C
C*****BACK SUBSTITUTION
C
  400 DO 450 M=1,NN
      N=NN+1-M
      DO 425 K=2,MM
      L=N+K-1
  425 B(N)=B(N)-A(N,K)*B(L)
      NM=N+NN
      B(NM)=B(N)
  450 AZZ(N,NUMBLK)=B(N)
      NUMBLK=NUMBLK-1
      IF(NUMBLK)475,500,475
  475 CONTINUE
      DO 729 N=1,NN
      BACKSPACE 1
  729 CONTINUE
      DO 730 N=1,NN
      READ(1) B(N),(A(N,M),M=2,MM)
  730 CONTINUE
      DO 731 N=1,NN
      BACKSPACE 1
  731 CONTINUE
      GO TO 400
  500 CONTINUE
C
C*****PRINT DISPLACEMENTS
C
      PRINT 2009
      K=0
      DO 352 NB=1,NBLK
      DO 350 N=1,MBAND,2
      K=K+1
      KDISP(N,NB)=K
      IF(DABS(AZZ(N   ,NB)) .LT. 1.0D-06)AZZ(N   ,NB)=ZERO
      IF(DABS(AZZ(N+1,NB)) .LT. 1.0D-06)AZZ(N+1,NB)=ZERO
  350 CONTINUE
      IF(K-NUMNP) 352,360,360
  352 CONTINUE
  360 CONTINUE
      PRINT 2010,((KDISP(N,NB),AZZ(N,NB),AZZ(N+1,NB),N=1,MBAND,2),
     1NB=1,NBLK)
      CALL ELAPSE(I)
      PRINT 3224, I
C
      CALL  STRESS
C
    2 FORMAT (I10,/,16I5)
    3 FORMAT (I10,/,5(I5,D10.2))
   12 FORMAT(16I5)
   13 FORMAT(5(I5,D10.2))
   20 FORMAT(L5,D15.2)
 2004 FORMAT ( 10X, I5, 3X, 1P2D12.3, 4X, A4, 1PD12.4, 4X, A4, 1PD12.4 )
 2009 FORMAT('1',2(7X,'NODE NO.' ,13X,'X-DISP.',13X,'Y-DISP.'))
 2010 FORMAT (2(7X,I8,5X,1PD15.6,5X,1PD15.6))
 3220 FORMAT(1X,3D26.16,1X)
 3223 FORMAT (2(2X,I3,1X,I3,1X,D23.16,7X))
 3224 FORMAT (I10)
 3244 FORMAT (/////,3X,'COMPUTED NODAL-LOAD EQUIVALENT TO UNIFORMLY DIST
     1RIBUTED LOAD',//,3X,5('NODE NO.',3X,'Y-NODAL-LOAD',3X),/,
     2   5(I10,3X,D13.2))
      GO TO 1
 9999 STOP
      END
```

```
      SUBROUTINE MANE
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /AAA/ X(343),Y(343),UX(343),UY(343),UXTYPE(343),UYTYPE(343)
     1,ELMAS(288),HED(18),TYPE(8),E,DENS,PR,VOL,MTYPE(288),NUMNP,
     2NUMEL,NUMAT,KN,NCASE,KOUNT
      COMMON /ARG/ XXX(5),YYY(5),S(10,10),DD(3,3),HH(6,10),P(10),XX(4),
     1YY(4),C(4,4),H(6,10),D(6,6),F(6,10),
     2TYPE1,TYPE2,TEST1,TEST2,IX(288,4),LM(4),NR,LIMIT,ISTART
      COMMON /BANARG/ A(36,18),FM(36),B(36),MBAND,NUMBLK
      COMMON/AIJFM/AA(40,18,18),FFM(40,18)
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      COMMON /AZZA/ AZZ(18,40)
      COMMON/SQUAD/SQ(10,10)
C
C*****READ AND PRINT OF CONTROL INFORMATION AND MATERIAL PROPERTIES
C
      IF(KOUNT.GT.1) GOTO 5
      READ 1000,HED,TYPE
      READ  1001,    E    ,PR    ,DENSTY
    5 CONTINUE
      READ(5,3224)NCASE
      IF(NCASE.EQ.0) RETURN
      PRINT2000,HED,TYPE
      PRINT 2001
      DENS=DENSTY
      PRINT 2002,    E    ,PR    ,DENS
C
C*****READ AND PRINT OF NODAL POINT DATA
C
      READ 402,DEPTH,WIDTH       ,SPAN,LIMIT
      DENS     = (ONE/(386.4000*1728.0000))*DENS
      READ 430, NR,K2,XNOT,YNOT
      PRINT 435
      PRINT 432,DEPTH,WIDTH,NR,K2,SPAN,XNOT,YNOT
      DR2=DEPTH/DFLOAT(K2)
      K3=NR-1
      DR3= SPAN/DFLOAT(K3)
      PRINT 2010,     DR2,DR3,LIMIT
      PRINT 3226, NCASE
      K=K2+1
      KN=K
      NUMNP=K*NR
      NUQUAD = (K-1)*(NR-1)
```

```
      NUMEL=NUQUAD
      PRINT 440
      DO 315 J =1,K
      X(J)=XNOT
  315 Y(J)=YNOT + DFLOAT(J-1)*DR2
      Y(K2/2 + 1) = 0.0D00
C
      DO 320 I=2,NR
      XZ=DR3*(I-1)+XNOT
      N1=K*(I-1)+1
      DO 320 KK=1,K
      J=N1+KK-1
      X(J)=XZ
  320 Y(J)=Y(KK)
      DO 321 I=1,NUMNP
      UXTYPE(I)=TYPE1
      UYTYPE(I)=TYPE1
      UX(I)= ZERO
      UY(I)= ZERO
  321 CONTINUE
C
      READ 1002, IUMNP
      READ 1002,(I,UXTYPE(I),UX(I),UYTYPE(I),UY(I),N=1,IUMNP)
      PRINT 2004,(I,X(I),Y(I),UXTYPE(I),UX(I),UYTYPE(I),UY(I),I=1,NUMNP)
      CALL GRAPH (X,Y,NUMNP,1)
C
C*****READ AND PRINT OF ELEMENT PROPERTIES
C
      PRINT  2006
      NM=0
      NRM1=NR-1
      K=KN
      DO 610 I=1,NRM1
      L1=(K-1)*(I-1)+1
      L2=(K-1)*I
      LI=(L2-L1+1)*(I-1)
      DO 610 N=L1,L2
      MTYPE(N)=1
      IX(N,1)=N+I-1
      IX(N,2)=IX(N,1)+K
      IX(N,3)=IX(N,2)+1
      IX(N,4)=IX(N,1)+1
      ELMAS(N)=(Y(N+I)-Y(N+I-1))*DR3*DENS
      DO 633 M=1,4
      DO 633 MM=1,4
      KK=IABS(IX(N,M)-IX(N,MM))
      IF(KK-NM) 633,633,631
  631 NM=KK
      IF (NM - LIMIT) 633,634,634
  634 PRINT 2008
      CALL EXIT
  633 CONTINUE
  610 CONTINUE
      XZ = ONE/FOUR
      DO 613 N=1,NUMEL
  613 ELMAS(N)=ELMAS(N) * XZ
      PRINT 2007,(I,(IX(I,J),J=1,4),MTYPE(I),ELMAS (I),I=1,NUMEL)
      MBAND=2* LIMIT
C
      PRINT 3000
      CALL ELAPSE(I)
      PRINT 3224,I
      CALL MASTIF
      CALL ELAPSE(K)
      PRINT 3224,K
      PRINT 3010
C
  402 FORMAT (3F10.0, I10)
  430 FORMAT (2I5,2F10.0)
  432 FORMAT (1P2D14.3,2I7,1P3D14.3)
  435 FORMAT (//,1X,'DEPTH OF BEAM',' WIDTH OF BEAM', 5X,'NR',5X,'K2',
     110X,'SPAN',10X,'XNOT',10X,'YNOT')
  440 FORMAT ('1', 10X,'NODE',5X,'X-ORDINATE  Y-ORDINATE  X-LOAD',3X,
     1'OR DISPL.  Y-LOAD   OR DISPL.')
 1000 FORMAT (18A4,/,8A3 )
 1001 FORMAT (3(3X,D11.4) )
 1002 FORMAT (I5,6X,A4,F10.0,6X,A4,F10.0)
 2000 FORMAT ('1',4X,18A4,//5X,8A3,//)
 2001 FORMAT (6X,'MODULUS OF  POISSON S  DENSITY OF',/,6X,'ELASTICITY',
     15X,'RATIO',5X,'MATERIAL',/)
 2002 FORMAT (5X,   1P3D12.3)
 2004 FORMAT ( 10X, I5, 3X, 1P2D12.3, 4X, A4, 1PD12.4, 4X, A4, 1PD12.4 )
 2006 FORMAT  ('1', 13X,'EL. NO.',9X,'I',9X,'J',9X,'K',9X,'L',2X,
     1'MATERIAL',  7X,'EL. MASS'  )
 2007 FORMAT (10X, 6I10,D23.16)
 2008 FORMAT (30H0 BAND WIDTH EXCEEDS ALLOWABLE)
 2010 FORMAT (/,5X,'DR2=',D11.4,5X,'DR3=',D11.4,5X,'LIMIT=',I4)
 3000 FORMAT(//, 1X,'$$$$-STARTS-CALL STIFF PRINT OUT',//)
 3010 FORMAT(//, 1X,'$$$$- ENDS -CALL STIFF PRINT OUT',//)
 3224 FORMAT(I10)
 3226 FORMAT (/,5X,'CASE NO. =', I3)
      RETURN
      END
```

```
      SUBROUTINE MASTIF
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /AAA/ X(343),Y(343),UX(343),UY(343),UXTYPE(343),UYTYPE(343)
     1,ELMAS(288),HED(18),TYPE(8),E,DENS,PR,VOL,MTYPE(288),NUMNP,
     2NUMEL,NUMAT,KN,NCASE,KOUNT
      COMMON /ARG/ XXX(5),YYY(5),S(10,10),DD(3,3),HH(6,10),P(10),XX(4),
     1YY(4),C(4,4),H(6,10),D(6,6),F(6,10),
     2TYPE1,TYPE2,TEST1,TEST2,IX(288,4),LM(4),NR,LIMIT,ISTART
      COMMON /BANARG/ A(36,18),FM(36),B(36),MBAND,NUMBLK
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      COMMON/SQUAD/SW(10,10)
      DIMENSION DUMMY(1000),IZ1(1000),JZ1(1000)
      DIMENSION S1(8,8)
C
C*****INITIALIZATION
C
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
```

```
      NB = LIMIT
      ND=2*NB
      ND2=2*ND
      NUMBLK=0
      PRINT 3009 ,NB,ND,ND2,NUMBLK
C
      DO 50 N=1,ND2
      B(N)=ZERO
      FM(N)= ZERO
      DO 50 M=1,ND
   50 A(N,M)=ZERO
C
C*****FORM MASS AND STIFFNESS MATRICES IN BLOCKS
C
   60 NUMBLK=NUMBLK+1
      NH=NB*(NUMBLK+1)
      NM=NH-NB
      NL=NM-NB+1
      KSHIFT=2*NL-2
      BLK=DFLOAT(NUMBLK)
C
      NUMEL1=1+NUMEL/2
      DO 210 N=1,NUMEL
C
      IF (MTYPE(N)) 210,210,65
   65 DO 80 I=1,4
      IF(IX(N,I) .GE. NL .AND. IX(N,I) .LE. NM) GO TO  90
   80 CONTINUE
      GO TO 210
   90 CONTINUE
      IF(N.GT.NUMEL1)GOTO 94
      IF(NCASE.NE.3.AND.NCASE.NE.4)GOTO 93
      IF(N.EQ.NUMEL1)GOTO 96
   93 IF (N .GT. 1) GO TO 94
   96 CONTINUE
C
      CALL ELAPSE(I)
      PRINT 3224, I
      CALL STIF88(N)
      CALL ELAPSE(J)
      PRINT 3224, J
C
      MTYPE(N)=-MTYPE(N)
      IF(NCASE.NE.5)GOTO98
      DO 147 I=1,10
      DO 147 J=1,10
  147 SQ(I,J)=S(I,J)
      DO 150 II=1,9
      CC=S(II,10)/S(10,10)
      P(II)=P(II)-CC*P(10)
      DO 150 JJ=1,9
  150 S(II,JJ)=S(II,JJ)-CC*S(10,JJ)
C
      DO 160 II=1,8
      CC=S(II,9)/S(9,9)
      P(II)=P(II)-CC*P(9)
      DO 160 JJ=1,8
  160 S(II,JJ)=S(II,JJ)-CC*S(9,JJ)
   98 CONTINUE
      DO 99 I=1,8
      DO 99 J=1,8
   99 S1(I,J)=S(I,J)
      PRINT 3003,((S1(I,J),J=1,8),I=1,8)
 9999 CONTINUE
      GO TO 165
   94 MTYPE(N)=-MTYPE(N)
      DO 97 I=1,8
      DO 97 J=1,8
   97 S(I,J)=S1(I,J)
      DO 151 I=1,8
  151 P(I)=ZERO
C
C*****ADD ELEMENT STIFFNESS TO STRUCTURE STIFFNESS
C
  165 DO 166 I=1,4
  166 LM(I)=2*IX(N,I)-2
C
      ELM = ELMAS(N)
      DO 199 I=1,4
      DO 199 K=1,2
      II=LM(I)+K-KSHIFT
      FM(II)= FM(II) + ELM
C
      KK=2*I-2+K
      B(II) =B(II)+P(KK)
      DO 200 J=1,4
      DO 200 L=1,2
      JJ=LM(J)+L-II+1-KSHIFT
      LL=2*J-2+L
      IF(JJ) 200,200,175
  175 IF(ND-JJ)180,195,195
  180 PRINT 2001, N
      CALL EXIT
  195 CONTINUE
      A(II,JJ)=A(II,JJ)+S(KK,LL)
  200 CONTINUE
  199 CONTINUE
  210 CONTINUE
C
C*****ADD CONCENTRATED FORCES WITHIN BLOCK
C
      DO 250 N=NL,NM
      K=2*N-KSHIFT
      IF ( UYTYPE(N) .NE. TYPE1 ) GO TO 240
      B(K) = B(K) + UY(N)
  240 IF ( UXTYPE(N) .NE. TYPE1 ) GO TO 250
      B(K-1) = B(K-1) + UX(N)
  250 CONTINUE
C
C*****WRITE BLOCK OF EQUATIONS ON TAPE AND SHIFT UP LOWER BLOCK
C
      WRITE(3)(FM(N),N=1,ND)
      N=0
      DO 860 I=1,ND
      DO 860 J=1,ND
      IF (A(I,J) .GT. 1.0 .OR.  A(I,J) .LT. -1.0) GO TO 161
      GO TO 860
  161 N=N+1
```

```
      IZ1(N)=I
      JZ1(N)=J
      DUMMY(N)=A(I,J)
  86C CONTINUE
      PRINT 3224,N
      PRINT 3222, (IZ1(I),JZ1(I),DUMMY(I),I=1,N)
      WRITE(2) N
      WRITE(4)(IZ1(I),JZ1(I),DUMMY(I),I=1,N)
C
  455 DO 420 N=1,ND
      K=N+ND
      B(N)=B(K)
      FM(N)=FM(K)
      B(K)=ZERO
      FM(K)= ZERO
      DO 420 M=1,ND
      A(N,M)=A(K,M)
  420 A(K,M)=ZERO
C
C*****CHECK FOR LAST BLOCK
C
      IF (NM-NUMNP) 60,480,480
  480 CONTINUE
C
  500 RETURN
 2000 FORMAT (26HONEGATIVE AREA ELEMENT NO. I4)
 20C1 FORMAT (29HOBAND WIDTH EXCEEDS ALLOWABLE I4)
 3003 FORMAT (10X,'MATRIX- S1'/ (3X,8D16.6))
 3009 FORMAT (//,5X,'NB,ND,ND2,NUMBLK',4I10,//)
 301C FORMAT (2X,'B',7D17.5 )
 3011 FORMAT(/,' NUMBLK,NH,NM,NL,KSHIFT',5I10,/)
 3025 FORMAT (5(1X,D23.16,2X))
 3220 FORMAT (5X,'FM(N)'/,(1X,3D26.16,1X))
 3222 FORMAT (4(1X,'(',I3,',',I3,')',D23.16))
 3223 FORMAT (2(2X,I3,1X,I3,1X,D23.16,7X))
 3224 FORMAT (I10)
      ENC



      SUBROUTINE STIF88(N)
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON /AAA/ X(343),Y(343),UX(343),UY(343),UXTYPE(343),UYTYPE(343)
     1,ELMAS(288),HED(18),TYPE(8),E,DENS,PR,VOL,MTYPE(288),NUMNP,
     2NUMEL,NUMAT,KN,NCASE,KOUNT
      COMMON /ARG/ XXX(5),YYY(5),S(10,10),DD(3,3),HH(6,10),P(10),XX(4),
     1YY(4),C(4,4),H(6,10),D(6,6),F(6,10),
     2TYPE1,TYPE2,TEST1,TEST2,IX(288,4),LM(4),NR,LIMIT,ISTART
      COMMON /BANARG/ A(36,18),FM(36),B(36),MBAND,NUMBLK
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      DIMENSION SS(8,8)
C
      XLAM1=HALF*(ONE-PR)
      XLAM2=HALF*(ONE+PR)
      N12=0
      CC 1210 II=1,8
      DO 1210 JJ=1,8
 121C SS(II,JJ)=ZERO
    5 I=IX(N,1)
      J=IX(N,2)
      K=IX(N,3)
      L=IX(N,4)
      XA=DABS(X(J)-X(I))
      YB=DABS(Y(K)-Y(J))
      NUMEL1=1+NUMEL/2
      IF(N.NE.NUMEL1)GCTO 200
      IF(NCASE.EQ.3)GOTO 2
      IF(NCASE.EQ.4)GOTO 1
  2CC CONTINUE
      GO TO ( 1,2,1,2,3,4),NCASE
C     PRINT 2002,NCASE
C
      IF(NCASE .NE. 12) CALL EXIT
    1 CONTINUE
C
C  STIFFNESS MATRIX 0135
C     SUBROUTINE TRI135(N)
C
      COEF=E/( 2.0D00*XA*YB*(ONE-PR*PR))
      S(1,1)=      (YB*YB+XLAM1*XA*XA)
      S(2,2)=      ( XA*XA+XLAM1*YB*YB)
      S(2,1)=        XA*YB*XLAM2
      S(6,1)= ZERO
      S(5,2)= ZERO
      S(4,3)= ZERO
      S(8,7)= ZERO
      S(8,3)= XA*YB*(PR+XLAM1)
      S(7,4)= XA*YB*(PR+XLAM1)
      S(6,5)= S(2,1)
      S(3,1)=-YB*YB
      S(7,5)=S(3,1)
      S(4,1)=-XLAM1*XA*YB
      S(8,1)=-PR*XA*YB
      S(3,2)= S(8,1)
      S(7,2)= S(4,1)
      S(6,3)= S(4,1)
      S(5,4)= S(8,1)
      S(7,6)= S(8,1)
      S(8,5)= S(4,1)
      S(7,1)=-XLAM1*XA*XA
      S(5,3)= S(7,1)
      S(5,1)= ZERO
      S(7,3)= S(5,1)
      S(4,2)= -XLAM1*YB*YB
      S(8,6)= S(4,2)
      S(8,2)= -XA*XA
      S(6,4)= S(8,2)
      S(6,2)= ZERO
      S(8,4)= S(6,2)
      GOTC 3545
    2 CONTINUE
C
C  STIFFNESS MATRIX 0045
C     SUBROUTINE TRI045(N)
C
      COEF=E/( 2.0D00*XA*YB*(ONE-PR*PR))
```

```
      S(1,1)=      (YB*YB+XLAM1*XA*XA)
      S(2,2)=      (XA*XA+XLAM1*YB*YB)
      S(2,1)=ZERO
      S(6,1)= -XA*YB*(XLAM1+PR)
      S(5,2)=S(6,1)
      S(4,3)=-XLAM2*XA*YB
      S(8,7)=S(4,3)
      S(8,3)=ZERO
      S(7,4)= ZERO
      S(6,5)= S(2,1)
      S(3,1)=-YB*YB
      S(7,5)=S(3,1)
      S(4,1)=     PR*XA*YB
      S(8,1)= XLAM1*XA*YB
      S(3,2)= S(8,1)
      S(7,2)= S(4,1)
      S(6,3)= S(4,1)
      S(5,4)= S(8,1)
      S(7,6)= S(8,1)
      S(8,5)= S(4,1)
      S(7,1)=-XLAM1*XA*XA
      S(5,3)= S(7,1)
      S(5,1)= ZERO
      S(7,3)= S(5,1)
      S(4,2)= -XLAM1*YB*YB
      S(8,6)= S(4,2)
      S(8,2)= -XA*XA
      S(6,4)= S(8,2)
      S(6,2)= ZERO
      S(8,4)= S(6,2)
      GOTO 3545
    4 CONTINUE
C
C**STIFFNESS MATRIX RECTNG
C     SUBROUTINE RECTNG(N)
C
      COEF=E/(12.0D00*XA*YB*(ONE-PR*PR))
      S(1,1)=FOUR*(YB*YB+XLAM1*XA*XA)
      S(2,2)=FOUR*(XA*XA+XLAM1*YB*YB)
      S(2,1)=THREE*XA*YB*XLAM2
      S(6,1)=-S(2,1)
      S(5,2)= S(6,1)
      S(4,3)= S(6,1)
      S(8,7)= S(6,1)
      S(8,3)= S(2,1)
      S(7,4)= S(2,1)
      S(6,5)= S(2,1)
      S(3,1)=TWO*(XLAM1*XA*XA-TWO*YB*YB)
      S(7,5)=S(3,1)
      S(4,1)=THREE*XA*YB*(PR-XLAM1)
      S(8,1)=-S(4,1)
      S(3,2)= S(8,1)
      S(7,2)= S(4,1)
      S(6,3)= S(4,1)
      S(5,4)= S(8,1)
      S(7,6)= S(8,1)
      S(8,5)= S(4,1)
      S(7,1)=TWO*(YB*YB-TWO*XLAM1*XA*XA)
      S(5,3)= S(7,1)
      S(5,1)=-TWO*(YB*YB+XLAM1*XA*XA)
      S(7,3)= S(5,1)
      S(4,2)= TWO*(XA*XA-TWO*XLAM1*YB*YB)
      S(8,6)= S(4,2)
      S(8,2)= TWO*(XLAM1*YB*YB-TWO*XA*XA)
      S(6,4)= S(8,2)
      S(6,2)=-TWO*(XA*XA+XLAM1*YB*YB)
      S(8,4)= S(6,2)
      GOTC 3545
    3 CONTINUE
C
C**STIFFNESS MATRIX QUAD
C     SUBROUTINE QUAD (N)
C
C*****FORM STRESS-STRAIN RELATIONSHIP
C
      IF (TYPE(4) - TEST1) 10,30,10
   1C IF (TYPE(4) - TEST2) 20,40,20
   20 PRINT 2000
      CALL EXIT
   30 COMM = E       /(ONE -PR       * PR      )
      C(1,1) = COMM
      C(1,2) = COMM * PR
C
      C(1,3) = ZERO
      C(1,4) = ZERO
      C(2,1) = C(1,2)
      C(2,2) = COMM
      C(2,3) = ZERO
      C(2,4) = ZERO
      C(3,1) = ZERO
      C(3,2) = ZERO
      C(3,3) = ZERO
      C(3,4) = ZERC
      C(4,1) = ZERO
      C(4,2) = ZERO
      C(4,3) = ZERO
      C(4,4) = COMM *HALF * (ONE- PR      )
      GO TO 50
   4C COMM = E       / ((ONE+ PR      ) * (ONE-HALF* PR      ))
      C(1,1) = COMM * (ONE- PR      )
      C(1,2) = COMM * PR
      C(1,3) = ZERO
      C(1,4) = ZERO
      C(2,1) = C(1,2)
      C(2,2) = C(1,1)
      C(2,3) = ZERO
      C(2,4) = ZERO
      C(3,1) = COMM * PR
      C(3,2) = C(3,1)
      C(3,3) = ZERO
      C(3,4) = ZERO
      C(4,1) = ZERC
      C(4,2) = ZERO
      C(4,3) = ZERO
      C(4,4) = COMM *HALF * (ONE-TWO * PR      )
C
C*****FORM QUADRILATERAL STIFFNESS MATRIX
C
```

```
   50 XXX(5) = (X(I) + X(J) + X(K) + X(L)) / FOUR
      YYY(5) = (Y(I) + Y(J) + Y(K) + Y(L)) / FOUR
      DO 94 M = 1,4
      MM = IX(N,M)
      XXX(M) = X(MM)
   94 YYY(M) = Y(MM)
C
      DO 100 II=1,10
      P(II)=ZERO
      DO 95 JJ=1,6
   95 HH(JJ,II)=ZERO
      DO 100 JJ=1,10
  100 S(II,JJ)=ZERO
C
      IF (K-L) 125,120,125
  120 CALL TRISTF(1,2,3)
      XXX(5) = (XXX(1) + XXX(2) + XXX(3)) / THREE
      YYY(5) = (YYY(1) + YYY(2) + YYY(3)) / THREE
      GO TO 130
C
  125 VOL=ZERO
      CALL TRISTF(4,1,5)
      CALL TRISTF(1,2,5)
      CALL TRISTF(2,3,5)
      CALL TRISTF(3,4,5)
C
      DO 140 II=1,6
      DO 140 JJ=1,10
  140 HH(II,JJ)=HH(II,JJ)/FOUR
C
  130 RETURN
 2000 FORMAT(41H0 PLANE STRESS OR STRAIN TYPE ERROR        )
C
 3545 CONTINUE
      DO210 II=3,8,2
      JJ=II+1
      S(II,II)=S(1,1)
  210 S(JJ,JJ)=S(2,2)
      DO220 II=1,8
      DO230 JJ=II,8
  230 S(II,JJ)=S(JJ,II)
      DO220 KK=1,8
  220 S(II,KK)=COEF*S(II,KK)
      IF(NCASE.NE.12) RETURN
      N12=N12+1
      DO 1220 II=1,8
      DO 1220 JJ=1,8
      SS(II,JJ)=SS(II,JJ)+S(II,JJ)
      S(II,JJ)=ZERO
 1220 CONTINUE
      IF(N12 .EQ.1) GOTO 2
      DO 1230 II=1,8
      DO 1230 JJ=1,8
 1230 S(II,JJ)=SS(II,JJ)*HALF
      RETURN
 2002 FORMAT(5X,'NCASE =',I5,'***********',//)
      END
```

```
      SUBROUTINE STRESS
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /AAA/ X(343),Y(343),UX(343),UY(343),UXTYPE(343),UYTYPE(343)
     1,ELMAS(288),HED(18),TYPE(8),E,DENS,PR,VOL,MTYPE(288),NUMNP,
     2NUMEL,NUMAT,KN,NCASE,KOUNT
      COMMON /ARG/ XXX(5),YYY(5),S(10,10),DD(3,3),HH(6,10),P(10),XX(4),
     1YY(4),C(4,4),H(6,10),D(6,6),F(6,10),
     2TYPE1,TYPE2,TEST1,TEST2,IX(288,4),LM(4),NR,LIMIT,ISTART
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      COMMON /AZZA/ AZZ(18,40)
      COMMON/SQUAD/SQ(10,10)
      DIMENSION Q(3,8),SIG(4),TP(6),QQ(3,8)
C
      NN=2*LIMIT
      XLAM1=HALF*(ONE-PR)
      MPRINT=0
      NUMEL1=1+NUMEL/2
C
      DO 90 N=1,NUMEL
      N12=0
      DO 1210 II=1,3
      DO 1210 JJ=1,8
 1210 QQ(II,JJ)=ZERO
      I=IX(N,1)
      J=IX(N,2)
      K=IX(N,3)
      L=IX(N,4)
      XA=DABS(X(J)-X(I))
      YB=DABS(Y(K)-Y(J))
      COE =E/(XA*YB*(ONE-PR*PR))
      XZX=(X(I)+X(J)+X(K)+X(L))/FOUR
      YZY=(Y(I)+Y(J)+Y(K)+Y(L))/FOUR
      XXA=XA/TWO
      YYB=YB/TWO
      DO 10 I1=1,4
      II=2*I1
      KK=2*IX(N,I1)
      NB=KK/NN
      NJ=MOD(KK,NN)
      IF(NJ) 5,6,5
    5 NB=NB+1
      JJ=NJ
      GO TO 8
    6 JJ=NN
    8 P(II-1)=AZZ(JJ-1,NB)
      P(II)=AZZ(JJ,NB)
   10 CONTINUE
      IF(NCASE.EQ.3.AND.N.GE.NUMEL1)GOTO 2
      IF(NCASE.EQ.4.AND.N.GE.NUMEL1)GOTO 1
      GOTO(1,2,1,2,3,4),NCASE
C
      IF(NCASE .NE. 12) CALL EXIT
    1 CONTINUE
C
C   STRESS Q135
C
      COEF=HALF*COE
      Q(1,1)=-YB
      Q(1,3)= YB
```

```
      Q(1,5)= YB
      Q(1,7)=-YB
      Q(1,2)=-PR*XA
      Q(1,4)=-PR*XA
      Q(1,6)= PR*XA
      Q(1,8)= PR*XA
      Q(2,1)=-PR*YB
      Q(2,3)= PR*YB
      Q(2,5)= PR*YB
      Q(2,7)=-PR*YB
      Q(2,2)=-XA
      Q(2,4)=-XA
      Q(2,6)= XA
      Q(2,8)= XA
      Q(3,1)=-XA*XLAM1
      Q(3,3)=-XA*XLAM1
      Q(3,5)= XA*XLAM1
      Q(3,7)= XA*XLAM1
      Q(3,2)=-YB*XLAM1
      Q(3,4)= YB*XLAM1
      Q(3,6)= YB*XLAM1
      Q(3,8)=-YB*XLAM1
      GOTO29
    2 CONTINUE
C
C   STRESS 0045
C
      COEF=HALF*COE
      Q(1,1)=-YB
      Q(1,3)= YB
      Q(1,5)= YB
      Q(1,7)=-YB
      Q(1,2)=-PR*XA
      Q(1,4)=-PR*XA
      Q(1,6)= PR*XA
      Q(1,8)= PR*XA
      Q(2,1)=-PR*YB
      Q(2,3)= PR*YB
      Q(2,5)= PR*YB
      Q(2,7)=-PR*YB
      Q(2,2)=-XA
      Q(2,4)=-XA
      Q(2,6)= XA
      Q(2,8)= XA
      Q(3,1)=-XA*XLAM1
      Q(3,3)=-XA*XLAM1
      Q(3,5)= XA*XLAM1
      Q(3,7)= XA*XLAM1
      Q(3,2)=-YB*XLAM1
      Q(3,4)= YB*XLAM1
      Q(3,6)= YB*XLAM1
      Q(3,8)=-YB*XLAM1
      GOTO29
C
    4 CONTINUE
      COEF=COE
      Q(1,1)=YYB-YB
      Q(2,1)=PR*Q(1,1)
      Q(2,2)=XXA-XA
      Q(3,1)=XLAM1*Q(2,2)
      Q(1,2)=PR*Q(2,2)
      Q(3,2)=XLAM1*Q(1,1)
      Q(1,3)=-Q(1,1)
      Q(2,3)=PR*Q(1,3)
      Q(3,3)=-XLAM1*XXA
      Q(1,4)=-PR*XXA
      Q(2,4)=-XXA
      Q(3,4)=-XLAM1*Q(1,1)
      Q(1,5)=YYB
      Q(2,5)=PR*YYB
      Q(3,5)=XLAM1*XXA
      Q(1,6)=PR*XXA
      Q(2,6)=XXA
      Q(3,6)=XLAM1*YYB
      Q(1,7)=-YYB
      Q(2,7)=-PR*YYB
      Q(3,7)=-XLAM1*Q(2,2)
      Q(1,8)=-PR*Q(2,2)
      Q(2,8)=-Q(2,2)
      Q(3,8)=-XLAM1*YYB
C
   29 CONTINUE
      IF(NCASE.NE.12) GOTO 1240
C
      N12=N12+1
      DO 1220 II=1,3
      DO 1220 JJ=1,8
 1220 QQ(II,JJ)=QQ(II,JJ)+Q(II,JJ)
      IF(N12 .EQ.1) GOTO 2
      DO 1230 II=1,3
      DO 1230 JJ=1,8
 1230 Q(II,JJ)=QQ(II,JJ)*HALF
 1240 DO 30 II=1,3
      SIG(II)=ZERO
      DO 30 JJ=1,8
      Q(II,JJ)=COEF*Q(II,JJ)
   30 SIG(II)=SIG(II)+Q(II,JJ)*P(JJ)
      GOTO 17
    3 CONTINUE
      DO 15 I=1,10
      DO 15 J=1,10
   15 S(I,J)=SQ(I,J)
      DO 20 I = 1,2
      XX(I) = P(I+8)
      DO 20 K = 1,8
   20 XX(I) = XX(I) -S (I+8,K) * P(K)
C
      COMM =S (9,9) *S (10,10) -S (9,10) *S (10,9)
      IF(COMM)32,40,32
   32 P(9) =(S (10,10) * XX(1) -S (9,10) * XX(2)) / COMM
      P(10) =(-S (10,9) * XX(1) +S (9,9) * XX(2)) / COMM
C
   40 DO 50 I = 1,6
      TP(I) = 0.0
      DO 50 K = 1,10
   50 TP(I) = TP(I) + HH(I,K) * P(K)
C
   52 XX(1) = TP(2)
```

```
      XX(2) = TP(6)
      XX(3) = 0.0
      XX(4) = TP(3) + TP(5)
   56 DO 60 I = 1,4
      SIG(I) = 0.
      DO 60 K = 1,4
   60 SIG(I) = SIG(I) + C(I,K) * XX(K)
      SIG(3)=SIG(4)
   17 CONTINUE
      RAD=((SIG(1)-SIG(2))/TWO) **2+(SIG(3))**2
      TMAX=DSQRT(RAD)
      SAVR=(SIG(1)+SIG(2))/TWO
      SIG1=SAVR+TMAX
      SIG2=SAVR-TMAX
      TAN2A=TWO*SIG(3)/(SIG(1)-SIG(2))
      ANG=DATAN(TAN2A)*90.0D00/PI
   62 IF(MPRINT .GT. 0) GO TO 80
      PRINT 2000
      MPRINT= 55
   80 MPRINT=MPRINT-1
      PRINT 2001,N,XZX,YZY,(SIG(I1),I1=1,3),SIG1,SIG2,ANG
   90 CONTINUE
      RETURN
C
 2000 FORMAT ('1',5X,/,1X,'STRESS AT MID-POINT OF ELEMENTS',//,1X,
     *'EL. NO.',2X,'X-ORDINATE',4X,'Y-ORDINATE',7X,'SIGMA-X',7X,
     *'SIGMA-Y',8X,'TAU-XY',7X,'SIGMA-1',7X,'SIGMA-2',4X,
     *'ANGLE IN RADIANS',/)
 2001 FORMAT(1X,I5,8D14.4)
 2002 FORMAT(5X,'NCASE =',I5,'***********',//)
      END


      SUBROUTINE GRAPH(X,Y,NP,NS)
C*****PROGRAMMER: ROBERT WOODS, SCHOOL OF MECH. ENGINEERING, OSU.
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PLOT(121,51), X(1000), Y(1000)
      INTEGER PLOT, ROUND, DISTX, DISTY, XAXIS, YAXIS
      INTEGER DOT, BLANK, MINUS, UNITS, O(13)
      DATA DOT, BLANK, MINUS, UNITS/1H.,1H ,1H-,1HI/
      DATA O/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H*,1H1,1H2/
      ROUND(A,B) = A/B + 0.5
C
    1 FORMAT(1H1,28X,'MAJOR X HASH MARKS INDICATE ',D9.2,/
     *            29X,'MAJOR Y HASH MARKS INDICATE ',D9.2)
    3 FORMAT(///51(14X, 71A1/))
    4 FORMAT(  /51(10X,121A1/))
C
      IXM = 121
      IYM = 51
      IF(NS .GT. 0) GO TO 11
      IXM = 71
      IYM = 41
C
C
   11 XMIN = X(1)
      XMAX = X(1)
      YMIN = Y(1)
      YMAX = Y(1)
C
      DO 20 I=2,NP
      IF(X(I) .LT. XMIN) XMIN = X(I)
      IF(X(I) .GT. XMAX) XMAX = X(I)
      IF(Y(I) .LT. YMIN) YMIN = Y(I)
      IF(Y(I) .GT. YMAX) YMAX = Y(I)
   20 CONTINUE
      IF(XMAX .NE. XMIN) GO TO 21
      XMIN = 2*XMIN - 1.E-7
      XMAX = 2*XMAX + 1.E-7
   21 IF(YMAX .NE. YMIN) GO TO 22
      YMIN = 2*YMIN - 1.E-7
      YMAX = 2*YMAX + 1.E-7
C
   22 XSTEP = AMAX(XMAX,XMAX-XMIN,-XMIN)
      YSTEP = AMAX(YMAX,YMAX-YMIN,-YMIN)
      STEPX = STEP(XSTEP/(IXM-0.51))
      STEPY = STEP(YSTEP/(IYM-0.51))
      HASHX = STEPX*10.
      HASHY = STEPY*10.
C
      DISTX = ROUND(XSTEP,STEPX)
      DISTY = ROUND(YSTEP,STEPY)
      XAXIS = IYM/2 + 1 - DISTY/2
      IF(YMAX .GE. 0.0) XAXIS = XAXIS + ROUND(YMAX,STEPY)
      YAXIS = IXM/2 + 1 + DISTX/2
      IF(XMAX .GE. 0.0) YAXIS = YAXIS - ROUND(XMAX,STEPX)
      LOCNX = YAXIS - (YAXIS-1)/10*10
      LOCNY = XAXIS - (XAXIS-1)/10*10
C
      DO 40 I=1,121
      DO 40 J=1,51
   40 PLOT(I,J) = BLANK
C
      DO 41 L=1,IYM
      PLOT(1,L) = UNITS
      PLOT(IXM,L) = UNITS
   41 PLOT(YAXIS,L) = DOT
      DO 42 L=1,IXM
      PLOT(L,1) = MINUS
      PLOT(L,IYM) = MINUS
   42 PLOT(L,XAXIS) = DOT
C
      IC = 0
      DO 50 I=LOCNX,IXM,10
      NUMX = -(YAXIS-1)/10 + IC
      NX=IABS(NUMX)+1
      IC = IC+1
   50 PLOT(I,XAXIS) = O(NX)
C
      IC = 0
      DO 60 J=LOCNY,IYM,10
      NUMY =  (XAXIS-1)/10 - IC
```

```
      NY= IABS(NUMY)+1
      IC = IC+1
   60 PLOT(YAXIS,J) = O(NY)
      PLOT(YAXIS,XAXIS) = O(1)
C
      DO 70 K=1,NP
      IX = YAXIS + ROUND(X(K),STEPX)
      JY = XAXIS - ROUND(Y(K),STEPY)
   70 PLOT(IX,JY) = O(11)
C
      WRITE(6,1) HASHX, HASHY
      IF(IXM .LE. 71) WRITE(6,3) ((PLOT(I,J),I=1,71),J=1,IYM)
      IF(IXM .GT. 71) WRITE(6,4) PLOT
C
      RETURN
      END
      FUNCTION STEP(W)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      N = DLOG10(W)                                                     GF
      IF(W .LE. 1.0) N = N-1                                            GF
      K = W/10.**N + 1.0                                                GI
      STEP = K*10.**N                                                   GF
      IF((K-1)*10.**N .GE. W) STEP = W                                  GI
      RETURN                                                            GI
      END                                                               GF
      FUNCTION AMAX(A,B,C)
      IMPLICIT REAL*8 (A-H,O-Z)
      AMAX = A
      IF(B.GE.C .AND. B.GE.A) AMAX = B
      IF(C.GE.A .AND. C.GE.B) AMAX = C
      RETURN
      END
```

APPENDIX C

LISTING OF COMPUTER PROGRAMS FOR DYNAMIC ANALYSIS

```
C
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    +                                                                    +
C    +                                                                    +
C    + PROGRAM                                                            +
C    +                                                                    +
C    + ''DYNAMIC ANALYSIS USING MODAL SUPERPOSITION''                     +
C    +                                                                    +
C    + LANGUAGE              :  FORTRAN IV                                +
C    + DIGITAL COMPUTER      :  IBM 360/65                                +
C    + PROGRAMMER            :  BRIJ R. KISHORE                           +
C    +                          STRUCTURAL ENGINEER                       +
C    +                          U. S. ARMY, CORPS OF ENGINEERS            +
C    +                          CHICAGO, ILLINOIS                         +
C    +                                                                    +
C    + PURPOSE                                                            +
C    +                                                                    +
C    + THIS PROGRAM FORMULATES [M] [K] MATRIX, FINDS ALL                  +
C    + EIGENVALUES AND EIGENVECTORS, AND PERFORMS MODAL                   +
C    + SUPERPOSITION. DETAILED INFORMATION CAN BE FOUND IN:               +
C    + ''CHARACTERISTICS OF FINITE ELEMENTS FOR ANALYSIS OF               +
C    + BEAMS SUBJECTED TO IMPACT LOAD'', PH.D. DISSERTATION               +
C    + BY BRIJ R. KISHORE, SCHOOL OF CIVIL ENGINEERING,                   +
C    + OKLAHOMA STATE UNIVERSITY, JULY 1972.                              +
C    +                                                                    +
C    +                                                                    +
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    +                                                                    +
C    + DESCRIPTION OF PARAMETERS:                                         +
C    +                                                                    +
C    + SEE ALSO DESCRIPTION OF PARAMETERS GIVEN IN PROGRAM  FOR           +
C    + STATIC ANALYSIS IN APPENDIX B.                                     +
C    +                                                                    +
C    +                                                                    +
C    + A            = MATRIX FOR WHICH EIGENVALUES & EIGENVECTORS         +
C    +                TO BE FOUND.                                        +
C    + N            = ORDER OF MATRIX A                                   +
C    + AWORK & BWORK= WORK ARRAYS FOR TEMPORARY STORAGE OF                +
C    +                EIGENVECTORS, EIGENVALUES, AND OTHER WORK           +
C    +                ARRAYS.                                             +
C    + IWORK        : WORK ARRAY FOR INTEGER VARIABLES                    +
C    + T            = NUMBER OF BINARY DIGITS IN THE MANTISSA OF          +
C    +                A DOUBLE PRECISION FLOATING POINT                   +
C    +                                                                    +
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
// EXEC PGM=GOGO
//D1 DD DSN=OSU.ACT10188.NONZERON,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D2 DD DSN=OSU.ACT10188.MASSMATX,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D3 DD DSN=OSU.ACT10188.TOTLSTIF,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D4 DD DSN=OSU.ACT10188.MASSINVK,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D5 DD DSN=OSU.ACT10188.EIGVALUE,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D6 DD DSN=OSU.ACT10188.EIGVECTR,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
```

```
// EXEC FORTHCLG,REGION.GO=241K
//FORT.SYSIN DD *
C
C
C*****PROGRAM FORMULATES MASS MATRIX, STRUCTURE STIFFNESS MATRIX
C     AND STORES ON DISK ONLY NONZERO TERMS OF STIFFNESS MATRIX
C     OF EACH BLOCK. NONZERON IS NUMBER OF ELEMENTS STORED ON DISK
C     WHICH ARE NONZERO.
C
C
      IMPLICIT REAL *8 (A-H,O-Z)
C
      COMMON /AAA/ X(343),Y(343),UX(343),UY(343),UXTYPE(343),UYTYPE(343)
     1,ELMAS(288),HED(18),TYPE(8),E,DENS,PR,VOL,MTYPE(288),NUMNP,
     2NUMEL,NUMAT,KN,NCASE,KOUNT
      COMMON /ARG/ XXX(5),YYY(5),S(10,10),DD(3,3),HH(6,10),P(10),XX(4),
     1YY(4),C(4,4),H(6,10),D(6,6),F(6,10),
     2TYPE1,TYPE2,TEST1,TEST2,IX(288,4),LM(4),NR,LIMIT,ISTART
      COMMON /BANARG/ A(36,18),FM(36),B(36),MBAND,NUMBLK
      COMMON/AIJFM/AA(40,18,18),FFM(40,18)
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      COMMON /AZZA/ AZZ(18,40)
      COMMON/SQUAD/SQ(10,10)
C
      KOUNT=0
      CALL ELAPSE(I)
      PRINT 3224, I
      KOUNT=KOUNT+1
      CALL MANE
      CALL ELAPSE(I)
      PRINT 3224, I
C
 9999 STOP
 3224 FORMAT (I10)
      END




      SUBROUTINE MANE
        .
        .
        .
        .
      (THIS SUBROUTINE IS LISTED IN APPENDIX B)
        .
        .
        .
        .
      RETURN
      END
```

```
      SUBROUTINE MASTIF
        .
        .
        .
        .
      (THIS SUBROUTINE IS LISTED IN APPENDIX B)
        .
        .
        .
        .
      RETURN
      END




      SUBROUTINE STIF88 (N)
        .
        .
        .
        .
      (THIS SUBROUTINE IS LISTED IN APPENDIX B)
        .
        .
        .
        .
      RETURN
      END




C
C
C*****THE FOLLOWING PROGRAM FOLLOWS THE ABOVE PROGRAM TO READ
C     MASS MATRIX, AND STIFFNESS MATRIX AND IT FORMULATES [M]-[K]
C     MATRIX.
C
C
// EXEC PGM=GOGO
//D1 DD DSN=OSU.ACT10188.MASSINVK,
//  UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
// EXEC FORTGCLG,REGION.GO=379K
//FORT.SYSIN DD *
C
C
C     SUBROUTINE  MASINA
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      DIMENSION II2(120),II3(120),AA(14,14)
      DIMENSION A(190,190), DFMUX(190), OFFDUY(190)
C
      READ 10,LIMIT,NUMNP
      NBIG= 2*NUMNP
      MBAND=2*LIMIT
      PRINT 20,LIMIT,MBAND,NUMNP,NBIG
C
```

```
C     COMPUTE VALUE OF NO. OF BLOCKS 'NBLK'
C
      NBLK=NUMNP/LIMIT
      I=MOD(NUMNP,LIMIT)
      IF(I.GT.0) NBLK=NBLK+1
      PRINT 3224, NBLK
C
C
C     INITIALIZE MASS & STIFFNESS MATRIX (DFMUX & A)
C
      DO 691 I=1,NBIG
      DFMUX(I)=ZERO
      OFFDUY(I)=ZERO
      DO 691 I1=1,NBIG
  691 A(I,I1)=ZERO
C
C     READ BLOCKS OF MASS MATRIX & STIFFNESS MATRIX
C
      REWIND 2
      REWIND 3
      REWIND 4
      DO 693 I1=1, NBLK
      KSHIFT=MBAND*(I1-1)
      J1=KSHIFT+1
      J2=KSHIFT+MBAND
      IF(J2.GT.NBIG) GOTO 600
      READ(3)(DFMUX(I2),I2=J1,J2)
      GOTO 610
  600 READ(3)(OFFDUY(I2),I2=1,MBAND)
      I4=NBIG-J1+1
      DO 620 I2=1,I4
  620 DFMUX(J1+I2-1)=OFFDUY(I2)
  610 READ(2) N
      PRINT 3224,N
      READ(4)(II2(I4),II3(I4),AA(II2(I4),II3(I4)),I4=1,N)
      DO 693 I4=1,N
      I2=II2(I4)+KSHIFT
      A(I2,I2-1+II3(I4)) =AA(II2(I4),II3(I4))
  693 CONTINUE
C
      PRINT 3224,NBIG
      PRINT 2070, (DFMUX(I),I=1,NBIG)
C
      DO 30 I=1,NBIG
      MBIG=0
      DO 32 J=1,NBIG
      IF(DABS(A(I,J)) .GT. 1.0) GOTO 161
      GOTO 32
  161 MBIG=MBIG+1
      OFFDUY(MBIG)=A(I,J)
      II2(MBIG)=J
   32 CONTINUE
      PRINT 3224, MBIG
      PRINT 3222 ,(I,II2(J),OFFDUY(J),J=1,MBIG)
   30 CONTINUE
C
C
      DO 695 I1=1,NBIG
      DO 694 I2=I1,NBIG
```

```
  694 A(I2,I1)=A(I1,I2)
      DO 695 I2=1,NBIG
  695 A(I2,I1)=A(I2,I1)/DFMUX(I2)
C
      DO330 I=1,NBIG
      MBIG=0
      DO332 J=1,NBIG
      IF(DABS(A(I,J)) .GT. 1.0) GOTO 162
      A(I,J)=ZERO
      GOTO 332
  162 MBIG=MBIG+1
      OFFDUY(MBIG)=A(I,J)
      II2(MBIG)=J
  332 CONTINUE
      PRINT 3224, MBIG
      PRINT 3222 ,(I,II2(J),OFFDUY(J),J=1,MBIG)
  330 CONTINUE
C
      REWIND 11
      DO 697 I=1,NBIG
      WRITE(11)(A(I,J),J=1,NBIG)
  697 CONTINUE
      STOP
C     RETURN
C
   10 FORMAT(2I5)
   20 FORMAT (10X,'LIMIT=',I3,5X,'MBAND=',I3,/,
     110X,'NUMNP=',I4,'NBIG=',I4,////)
 2070 FORMAT (10D13.5)
 3222 FORMAT (6(1X,'(',I3,',',I3,')',D11.3))
 3224 FORMAT (I10)
      END



      BLOCK DATA
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI,SMALL
      DATA ZERO,HALF,ONE,TWO,THREE,FOUR /0.0D00,0.5D00,1.0D00,2.0D00,
     13.0D00,4.0D00/,SIX/6.0D00/,PI /3.141592653589793/,SMALL/.1D-5/
      END



//GO.SYSIN DD *
   7   95
//GO.FT02F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,1),
// DCB=(LRECL=8,BLKSIZE=1092,RECFM=VBS),
//     DSN=OSU.ACT10188.NONZERON
//GO.FT03F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,1),
// DCB=(LRECL=112,BLKSIZE=1684,RECFM=VBS),
//     DSN=OSU.ACT10188.MASSMATX
//GO.FT04F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,5),
// DCB=(LRECL=1800,BLKSIZE=2298,RECFM=VBS),
//     DSN=OSU.ACT10188.TOTLSTIF
//GO.FT11F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(NEW,KEEP),SPACE=(TRK,55),
// DCB=(LRECL=1680,BLKSIZE=2298,RECFM=VBS),
//     DSN=OSU.ACT10188.MASSINVK
//



C
C
C*****THE FOLLOWING PROGRAM FOLLOWS THE ABOVE PROGRAM. IT READS
C      [M]-1[K] MATRIX AND FINDS ALL EIGENVALUES AND EIGENVECTORS
C     OF [M]-1[K] MATRIX.
C
C
// EXEC PGM=GOGO
//D1 DD DSN=OSU.ACT10188.EIGVALUE,
//  UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D2 DD DSN=OSU.ACT10188.EIGVECTR,
//  UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
// EXEC FORTHCLG,REGION.GO=379K
//FORT.SYSIN DD *
C.....MAIN PROGRAM
      IMPLICIT REAL *8(A-H,O-Z)
C
C   +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C--ON FILE 1---'INDIC'                                                  +
C--ON FILE 2---'H' THEN 'A'                                             +
C--ON FILE 3---'LOCAL'                                                  +
C--ON FILE14---'VECR'                                                   +
C--ON FILE 8---'PRFACT'                                                 +
C--ON FILE 9 & 13---'EVR'                                               +
C--ON FILE 12--'SUBDIA'                                                 +
C--ON FILE 13 & 9---'EVR'                                               +
C   +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      COMMON/BLOCK1/AWORK(190),BWORK(190),A(190,190)
      COMMON/BLOCK2/IWORK(190),N,IVEC,M
      COMMON/BLOCK3/ENORM,EPS,EX,T
      COMMON/AFILE/ID1,ID2,ID3,ID4,ID8,ID9
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI,SMALL
      DEFINE FILE 1(190,1,U,ID1),2(190, 380,U,ID2),3(190,1,U,ID3),
     114(190,380,U,ID4),8(190,2,U,ID8),9(190,2,U,ID9)
      DIMENSION C(190)
C
      KKKK=11
      CALL ELAPSE (I1)
      PRINT 3989, KKKK,I1
C
      T=53.0
      TT=T
      PRINT 3
C
```

```
      READ 10,LIMIT,NUMNP
   10 FORMAT(2I5)
      N   = 2*NUMNP
      MBAND=2*LIMIT
      PRINT 15,LIMIT,MBAND,NUMNP,N
C
C*****COMPUTE VALUE OF NO. OF BLOCKS 'NBLK'
C
      NBLK=NUMNP/LIMIT
      I=MOD(NUMNP,LIMIT)
      IF(I.GT.0) NBLK=NBLK+1
      PRINT 3224, NBLK
C
      REWIND 11
      DO 30 I=1,N
      READ (11) (A(I,J),J=1,N)
   30 CONTINUE
C     PRINT 2070,((A(I,J),J=1,N ),I=1,N )
C
C*****BOUNDARY CONDITIONS :-
C     INITIALIZE NODAL DISPLACEMENTS DFMUX & OFFDUY BY GIVING THEM
C     ANY ARBITRARY VALUE GREATER THAN ONE ( SAY 'SIX' )
C
      DO 689 I=1,NUMNP
      AWORK(I)=SIX
  689 BWORK (I)=SIX
      READ 3224,JX
      IF(JX.EQ.0) GOTO 9
C
C*****'IWORK' = 'NODE NUMBERS'
C
      READ12,(IWORK(I),I=1,JX)
      PRINT2,JX,(IWORK(I),I=1,JX   )
C
C*****SET X-DISP VALUES EQUAL TO ZERO FOR THE NODES WHICH HAVE
C     X-DISP = ZERO .
C     'AWORK' = 'X-DISP'
C
      DO 5 I=1,JX
    5 AWORK(IWORK(I))=ZERO
C
    9 CONTINUE
      READ 3224,JY
      IF(JY.EQ.0) GOTO 8
C
C*****'IWORK' = 'NODE NUMBERS'
C
      READ12,(IWORK(I),I=1,JY)
      PRINT2,JY,(IWORK(I),I=1,JY)
C
C*****SET Y-DISP VALUES EQUAL TO ZERO FOR THE NODES WHICH HAVE
C     Y-DISP = ZERO .
C     'BWORK' = 'Y-DISP'
C
      DO 4 I=1,JY
    4 BWORK (IWORK(I))=ZERO
    8 CONTINUE
C
C*****MODIFY THE MATRIX A FOR THE KNOWN ZERO DISPLACEMENTS.
C     FOR EXAMPLE, IF KTH.-DISP. = ZERO, SET A(K,K)=ONE & REMAINING EL-
C     EMENTS OF KTH. COLUMN & KTH. ROW = ZERO.
C
      DO 410 M=1,NUMNP
      K=2*M-1
      IF(AWORK(M).GT.ONE) GOTO 320
      ASSIGN 320 TO MTURN
      GOTO500
  320 K=K+1
      IF(BWORK (M).GT.ONE)GOTO 410
      ASSIGN 410 TO MTURN
  500 CONTINUE
      DO 420 I=1,N
      A(I,K)=ZERO
  420 A(K,I)=ZERO
      A(K,K)=ONE
      GOTO MTURN, (320,410)
  410 CONTINUE
C
      REWIND 4
      DO 510 I=1,N
      WRITE (4) (A(I,J),J=1,N)
  510 CONTINUE
C
      KKKK=22
      CALL ELAPSE (I1)
      PRINT 3989, KKKK,I1
      CALL EIGENP
      KKKK=33
      CALL ELAPSE (I1)
      PRINT 3989, KKKK,I1
C
      REWIND 4
      DO 520 I=1,N
       READ (4) (A(I,J),J=1,N)
  520 CONTINUE
      T=TT
      PRINT 2074, T
      REWIND 13
      READ (13)(BWORK(I1),I1=1,N)
      PRINT 2080,(I1,BWORK(I1),I1=1,N)
      ID1=1
      ID4=1
      PRINT 66
      DO 20 J=1,N
      READ(1'ID1) IWORK(J)
      READ(14'ID4) (AWORK(I1),I1=1,N )
      PRINT 2081,J,(I1,AWORK(I1),I1=1,N)
C
C
C*****THE FOLLOWING PROGRAM IS ONLY TO CHECK IF ALL EIGENVALUES,
C     AND EIGENVECTORS FOUND ARE CORRECT.
C
C
C    +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      IF(J .LT. 151) GO TO 20
C    +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      DO 70 IN=1,N
      C(IN)=0.D00
```

```
      DO 72 IM=1,N
      C(IN)=C(IN)+A(IN,IM) * AWORK(IM)
   72 CONTINUE
      C(IN)=C(IN)/BWORK(J)
   70 CONTINUE
      PRINT 2081,J,(I1,    C(I1),I1=1,N)
C
      PRINT 66
   20 CONTINUE
      PRINT 66
      PRINT 66
      PRINT 2082,(I1,IWORK(I1),I1=1,N)
C
      KKKK=44
      CALL ELAPSE (I1)
      PRINT 3989, KKKK,I1
C
 9999 STOP
    2 FORMAT(I10,/,16I5)
    3 FORMAT ('1')
   12 FORMAT (16I5)
   15 FORMAT (10X,'LIMIT=',I3,5X,'MBAND=',I3,/,
     110X,'NUMNP=',I4,'N  = ',I4,////)
   66 FORMAT (//)
 2070 FORMAT (1X,10D13.5)
 2072 FORMAT (/,5X,20I6)
 2074 FORMAT (5X,'T=',D14.7)
 2080 FORMAT(5X,'EIGENVALUES ',//,(3X,7(1X,I4,'-',D12.5)))
 2081 FORMAT (2X,'EIGENVECTOR-',I4,/ ,(3X,7(1X,I4,'-',D12.5)))
 2082 FORMAT (5X,'INDICATOR',/,5X,'2-EIGENVALUE & EIGENVECTORS BOTH FOUN
     1D',/,5X,'1-ONLY EIGENVALUE FOUND',/,5X,'0-NONE FOUND',//,
     2(2X,16(I4,1X,'-',I2)))
 3220 FORMAT(1X,3D26.16,1X)
 3223 FORMAT (2(2X,I3,1X,I3,1X,D23.16,7X))
 3224 FORMAT (I10)
 3989 FORMAT ( 5X,I4,'-',I10)
      END


      BLOCK DATA
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI,SMALL
      DATA ZERO,HALF,ONE,TWO,THREE,FOUR /0.0D00,0.5D00,1.0D00,2.0D00,
     13.0D00,4.0D00/,SIX/6.0D00/,PI /3.141592653589793/,SMALL/.1D-6/
      END


      SUBROUTINE EIGENP
      IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON/BLOCK1/AWORK(190),BWORK(190),A(190,190)
      COMMON/BLOCK2/IWORK(190),N,IVEC,M
      COMMON/BLOCK3/ENORM,EPS,EX,T
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI,SMALL
      COMMON/AFILE/ID1,ID2,ID3,ID4,ID8,ID9
      DEFINE FILE 1(190,1,U,ID1),2(190, 380,U,ID2),3(190,1,U,ID3),
     14(190,380,U,ID4),8(190,2,U,ID8),9(190,2,U,ID9)
C
      KKKK= 111
      CALL ELAPSE (I1)
      PRINT 3989, KKKK ,I1
    1 CALL SCALE
      CALL ELAPSE (I1)
      PRINT 3989, KKKK ,I1
C
C.....'H' ON FILE 2
C.....'AWORK'='PRFACT' ON FILE 8
C..................................................................
C THE COMPUTATION OF THE EIGENVALUES OF THE NORMALISED
C MATRIX.
C
      EX = DEXP(-T*DLOG(2.D0))
      KKKK= 222
      CALL ELAPSE (I1)
      PRINT 3989, KKKK ,I1
      CALL HESQR
      CALL ELAPSE (I1)
      PRINT 3989, KKKK ,I1
C
C.....'BWORK'='H' ON FILE 2
C.....'AWORK'='SUBDIA' ON FILE 12
C.....'AWORK'='EVR' ON FILE 9 & 13
C.....'IWORK'= 'INDIC' ON FILE 1
C
      REWIND 13
      REWIND 12
      READ(12) (AWORK(I1) ,I1=1,N)
 2072 FORMAT (/,5X,20I6)
 2070 FORMAT(6D20.7)
C
C THE POSSIBLE DECOMPOSITION OF THE UPPER-HESSENBERG MATRIX
C INTO THE SUBMATRICES OF LOWER ORDER IS INDICATED IN THE
C ARRAY LOCAL. THE DECOMPOSITION OCCURS WHEN SOME
C SUBDIAGONAL ELEMENTS ARE IN MODULUS LESS THAN A SMALL
C POSITIVE NUMBER EPS DEFINED IN THE SUBROUTINE HESQR . THE
C AMOUNT OF WORK IN THE EIGENVECTOR PROBLEM MAY BE
C DIMINISHED IN THIS WAY.
C.....'AWORK'='SUBDIA' ON FILE 12
C
      J = N
      I = 1
C.....'LOCAL'='IWORK' ON FILE 3
      IWORK(1)=1
      IF(J.EQ.1)GO TO 4
C.....'SUBDIA'='AWORK' ON FILE 12
    2 IF(DABS(AWORK(J-1)).GT.EPS) GOTO 3
      I = I+1
      IWORK(I)=0
    3 J = J-1
      IWORK(I)=IWORK(I)+1
      IF(J.NE.1)GO TO 2
```

```
C
C THE EIGENVECTOR PROBLEM.
    4 K = 1
      L=IWORK(1)
C.....'IWORK'='LOCAL' ON FILE 3
      ID3=1
      DO 945 I1=1,N
      WRITE(3'ID3) IWORK(I1)
  945 CONTINUE
      M = N
      LOCALK=IWORK(K)
      KKKK= 333
      CALL ELAPSE (I1)
      PRINT 3989, KKKK ,I1
      DO 10 I=1,N
        IVEC = N-I+1
      PRINT 2010, L, IVEC
 2010 FORMAT (5X,'L=',I5,5X,'IVEC=', I5)
      ID1=IVEC
      READ(1'ID1) INDIVC
 2071 FORMAT (5X,'INDIC(IVEC)=',I5)
        IF(I.LE.L)GO TO 5
        K = K+1
        M = N-L
      ID3=K
      READ(3'ID3) LOCALK
      L=L+LOCALK
    5 CONTINUE
      IF(INDIVC .EQ. 0) GO TO 10
 2076 FORMAT (5X,'IVEC=',I5,'EVI=',D20.7)
C
C.....NOTE:-
C.....'COMPVE' SUBROUTINE IS CALLED ONLY WHEN 'EVI' .NE. 'ZERO'. IF
C.....'EVI' IS 'ZERO',SUBROUTINE 'COMPVE' IS SKIPPED.
C
C TRANSFER OF AN UPPER-HESSENBERG MATRIX OF THE ORDER M FROM
C THE ARRAYS 'H'= 'VECI' AND SUBDIA INTO THE ARRAY A .
C
      REWIND 12
      READ(12) (AWORK(I1) ,I1=1,N)
      ID2=1
        DO 7 K1=1,M
      READ(2'ID2) (BWORK(I1), I1=1,N)
          DO 6 L1= 1,K1
    6 A(L1,K1)= BWORK(L1)
      IF(K1.EQ.1)GOTO 7
      A(K1,K1-1)=AWORK(K1-1)
    7     CONTINUE
 2090 FORMAT (5X,'K=',I5,'LOCALK=',I5)
C
C.....................................................................
C THE COMPUTATION OF THE REAL EIGENVECTOR IVEC OF THE UPPER-
C HESSENBERG MATRIX CORRESPONDING TO THE REAL EIGENVALUE
C EVR(IVEC).
C
      CALL REALVE
C
C.....'VECR'='AWORK' ON FILE 14
   10   CONTINUE
```

```
C
      CALL ELAPSE (I1)
      PRINT 3989, KKKK ,I1
C
C THE RECONSTRUCTION OF THE MATRIX USED IN THE REDUCTION OF
C MATRIX A TO AN UPPER-HESSENBERG FORM BY HOUSEHOLDER METHOD
C
      DO 12 I=1,N
        DO 11 J=I,N
          A(I,J) = 0.D0
   11     A(J,I) = 0.D0
   12   A(I,I) = 1.D0
      IF(N.LE.2) GOTO 15
      M=N-2
      DO 14 K=1,M
      ID2=K
      READ (2'ID2) (AWORK(I2),I2=1,N)
      L=K+1
      DO 14 J=2,N
      D1=0.D00
      DO 13 I=L,N
      D2=AWORK(I)
   13 D1 =D1+D2*A(J,I)
      DO 14 I=L,N
      A(J,I)=A(J,I)-AWORK(I)*D1
   14 CONTINUE
C
C THE COMPUTATION OF THE EIGENVECTORS OF THE ORIGINAL NON-
C SCALED MATRIX.
   15 CONTINUE
      DO 24 I=1,N
C.....'AWORK'='VECR'
      ID4=I
      READ(14'ID4)(AWORK(I1),I1=1,N)
C.....'BWORK'='WORK',NOTE THIS 'WORK' HAS NO RELATION WITH
C.....'WORK' OF SUBROUTINE'REALVE'
        DO 18 J=1,N
      ID8=J
      READ(8'ID8) PRFACJ
          D1 = 0.D0
          DO 17 K=1,N
            D3 = A(J,K)
      D1=D1+D3*AWORK(K)
   17       CONTINUE
      BWORK(J)=D1/PRFACJ
   18     CONTINUE
C
C THE NORMALIZATION OF THE EIGENVECTORS AND THE COMPUTATION
C OF THE EIGENVALUES OF THE ORIGINAL NON-NORMALISED MATRIX.
        D1 = 0.D0
        DO 19 M=1,N
      W1=BWORK(M)
   19     D1 = D1+W1*W1
        D1 = DSQRT(D1)
      ID4=I
      FIND (14'ID4)
        DO 20 M=1,N
      AWORK(M)=BWORK(M)/D1
      IF(DABS(AWORK(M)).LT. 1.0D-08) AWORK(M)=0.0D00
```

```
   20 CONTINUE
      WRITE(14'ID4)(AWORK(I1),I1=1,N)
      ID9=I
      READ(9'ID9)EVRI
      ID9=I
      FIND(9'ID9   )
      EVRI=EVRI*ENORM
      WRITE(9'ID9   )EVRI
   24 CONTINUE
   25 CONTINUE
      ID9=1
      DO 40 I1=1,N
      READ (9'ID9)AWORK(I1)
   40 CONTINUE
      REWIND 13
      WRITE (13) (AWORK(I1),I1=1,N)
      RETURN
   81 FORMAT (5X, D13.6)
 2081 FORMAT (2X,'EIGENVECTOR-',I4,/ ,(3X,7(1X,I4,'-',D12.5)))
 3989 FORMAT ( 5X,I4,'-', I10 )
      END
```

```
      SUBROUTINE SCALE
      IMPLICIT REAL *8(A-H,O-Z)
C
      LOGICAL *1 SCLRQD
      SCLRQD = .FALSE.
      COMMON/BLOCK1/AWORK(190),BWORK(190),A(190,190)
      COMMON/BLOCK2/IWORK(190),N,IVEC,M
      COMMON/BLOCK3/ENORM,EPS,EX,T
      COMMON/AFILE/ID1,ID2,ID3,ID4,ID8,ID9
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI,SMALL
      DEFINE FILE 1(190,1,U,ID1),2(190, 380,U,ID2),3(190,1,U,ID3),
     114(190,380,U,ID4),8(190,2,U,ID8),9(190,2,U,ID9)
C
C.....'AWORK'='PRFACT' ON FILE  8
C.....'H'='A' ON FILE 2
      ID2=1
      DO 1 J=1,N
      WRITE(2'ID2)(A(I,J),I=1,N)
    1 AWORK(J)=ONE
      BOUND1 = 0.75D0
      BOUND2 = 1.33D0
      ITER = 0
    3 NCOUNT = 0
      DO 8 I=1,N
        COLUMN = 0.D0
        ROW = 0.D0
        DO 4 J=1,N
          IF(I.EQ.J)GO TO 4
          COLUMN = COLUMN + DABS(A(J,I))
          ROW = ROW + DABS(A(I,J))
    4     CONTINUE
C.....NOTE 'COLUMN' & 'ROW' ALWAYS POSITIVE NUMBERS.
      IF(COLUMN .LT. SMALL)GOTO 5
      IF(ROW .LT. SMALL) GOTO 5
C       IF(COLUMN.EQ.0.D0)GO TO 5
C         IF(ROW.EQ.0.D0)GO TO 5
          Q = COLUMN/ROW
          IF(Q.LT.BOUND1)GO TO 6
          IF(Q.GT.BOUND2)GO TO 6
    5   NCOUNT = NCOUNT + 1
        GO TO 8
    6   FACTOR = DSQRT(Q)
      SCLRQD = .TRUE.
        DO 7 J=1,N
          IF(I.EQ.J)GO TO 7
          A(I,J) = A(I,J)*FACTOR
          A(J,I) = A(J,I)/FACTOR
    7     CONTINUE
C.....'AWORK'='PRFACT'
      AWORK(I)=AWORK(I)*FACTOR
    8   CONTINUE
      ITER = ITER+1
      IF(SCLRQD) PRINT 91
      IF(ITER.GT.30)GO TO 11
      IF(NCOUNT.LT.N)GO TO 3
C
      FNORM = 0.D0
      DO 9 I=1,N
        DO 9 J=1,N
          Q=A(I,J)
    9     FNORM = FNORM+Q*Q
      FNORM = DSQRT(FNORM)
      DO 10 I=1,N
       DO 10 J=1,N
   10     A(I,J)=A(I,J)/FNORM
      ENORM = FNORM
C     PRINT 92
      GO TO 13
C
   11 CONTINUE
      ID2=1
      DO 12 J=1,N
      AWORK(J)=ONE
      READ(2'ID2) (A(I,J),I=1,N)
   12 CONTINUE
      PRINT 90
      ENORM=ONE
C
   13 CONTINUE
C.....'H' ON FILE 2
C.....'AWORK'='PRFACT' ON FILE 8
      ID8=1
      DO 195 I1=1,N
      WRITE (8'ID8)AWORK(I1)
  195 CONTINUE
      RETURN
   90 FORMAT (5X,'***SCALING FAILED ***')
   91 FORMAT (5X,'***SCALING WAS REQUIRED***')
   92 FORMAT (5X,'***MATRIX HAS BEEN SCALED***')
      END
```

```
      SUBROUTINE HESQR
      IMPLICIT REAL *8(A-H,O-Z)
C
      LOGICAL *1 ISHIFT
      ISHIFT=.FALSE.
      SMAL2=1.0D-20
      COMMON/BLOCK1/AWORK(190),BWORK(190),A(190,190)
      COMMON/BLOCK2/IWORK(190),N,IVEC,M
      COMMON/BLOCK3/ENORM,EPS,EX,T
      COMMON/AFILE/ID1,ID2,ID3,ID4,ID8,ID9
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI,SMALL
      DEFINE FILE 1(190,1,U,ID1),2(190, 380,U,ID2),3(190,1,U,ID3),
     114(190,380,U,ID4),8(190,2,U,ID8),9(190,2,U,ID9)
C
      M = N-2
       AWORK(N)=0.D0
      AWORK(N-1)=ZERO
      DO 12 K=1,M
      ID2=K
      READ(2'ID2)(BWORK(I1),I1=1,N)
      ID2=K
      FIND(2'ID2  )
C.....'BWORK'='H' ON FILE 2
        L = K+1
        S = 0.D0
        DO 3 I=L,N
      BWORK(I)=A(I,K)
    3     S = S+DABS(A(I,K))
C       IF(S.NE.DABS(A(K+1,K)))GO TO 4
C.....NOTE 'S' IS ALWAYS A POSITIVE NUMBER
      SDABSA=S-DABS(A(K+1,K))
      IF(DABS(SDABSA) .GT. SMAL2) GOTO 4
      PRINT 921, K,S,A(K+1,K)
  921 FORMAT (1X,'K,S,A(K+1,K)',I5,2D25.16)
C.....'AWORK'='SUBDIA' ON FILE 12
      AWORK(K)=A(K+1,K)
C       SUBDIA(K) = A(K+1,K)
C       H(K+1,K) = 0.D0
      BWORK(K+1)=ZERO
C       GO TO 12
      GOTO 112
    4   SR2 = 0.D0
        DO 5 I=L,N
          SR = A(I,K)
          SR = SR/S
          A(I,K) = SR
    5     SR2 = SR2+SR*SR
        SR = DSQRT(SR2)
        IF(A(L,K).LT.0.D0)GO TO 6
        SR = -SR
    6   SR2 = SR2-SR*A(L,K)
        A(L,K) = A(L,K)-SR
      BWORK(L)=BWORK(L)-SR*S
      AWORK(K)=SR*S
C       SUBDIA(K) = SR*S
C       H(L,K) = H(L,K)-SR*S
        X = S*DSQRT(SR2)
        DO 7 I=L,N
      BWORK(I)=BWORK(I)/X
      AWORK(I)=A(I,K)/SR2
    7 CONTINUE
C         H(I,K) = H(I,K)/X
C   7     SUBDIA(I) = A(I,K)/SR2
C PREMULTIPLICATION BY THE MATRIX PR.
        DO 9 J=L,N
          SR = 0.D0
          DO 8 I=L,N
    8       SR = SR+A(I,K)*A(I,J)
          DO 9 I=L,N
C   9       A(I,J) = A(I,J)-SUBDIA(I)*SR
    9 A(I,J)=A(I,J)-AWORK(I) *SR
C POSTMULTIPLICATION BY THE MATRIX PR.
        DO 11 J=1,N
          SR=0.D0
          DO 10 I=L,N
   10       SR = SR+A(J,I)*A(I,K)
          DO 11 I=L,N
C  11       A(J,I) = A(J,I)-SUBDIA(I)*SR
   11 A(J,I)=A(J,I)-AWORK(I)*SR
  112 CONTINUE
      WRITE(2'ID2  ) (BWORK(I1),I1=1,N)
   12   CONTINUE
      DO 13 K=1,M
   13 A(K+1,K)=AWORK(K)
C  13   A(K+1,K) = SUBDIA(K)
C TRANSFER OF THE UPPER HALF OF THE MATRIX A INTO THE
C ARRAY H AND THE CALCULATION OF THE SMALL POSITIVE NUMBER
C EPS.
C     SUBDIA(N-1) = A(N,N-1)
      AWORK(N-1)=A(N,N-1)
   14 EPS = 0.D0
C.....'IWORK'='INDIC'
      DO 15 K=1,N
      ID2=K
      IWORK(K)=0
      IF(K.NE.N)EPS=EPS+AWORK(K)*AWORK(K)
      READ(2'ID2) (BWORK(I1),I1=1,N)
      ID2=K
      FIND(2'ID2  )
      DO 155 I=1,K
      BWORK(I)=A(I,K)
      W2=A(I,K)
      EPS=EPS+W2*W2
  155 CONTINUE
      WRITE(2'ID2  ) (BWORK(I1),I1=1,N)
   15 CONTINUE
      EPS=EX*DSQRT(EPS)
C.....'AWORK'='SUBDIA' ON FILE 12
      REWIND 12
      WRITE(12)(AWORK(I1),I1=1,N)
C
C THE QR ITERATIVE PROCESS. THE UPPER-HESSENBERG MATRIX H IS
C REDUCED TO THE UPPER-MODIFIED TRIANGULAR FORM.
C
C DETERMINATION OF THE SHIFT OF ORIGIN FOR THE FIRST STEP OF
```

```
C THE QR ITERATIVE PROCESS.
      SHIFT = A(N,N-1)
      IF(N.LE.2)SHIFT = 0.D0
      IF (DABS(A(N,N)).GT. SMAL2) SHIFT=ZERO
      IF(DABS(A(N-1,N)).GT.SMAL2)  SHIFT=ZERO
      IF (DABS(A(N-1,N-1)) .GT. SMAL2) SHIFT=ZERO
C     IF(A(N,N).NE.0.D0)SHIFT = 0.D0
C     IF(A(N-1,N).NE.0.D0)SHIFT = 0.D0
C     IF(A(N-1,N-1).NE.0.D0)SHIFT = 0.D0
      M = N
      NS= 0
      MAXST = N*10
C
C TESTING IF THE UPPER HALF OF THE MATRIX IS EQUAL TO ZERO.
C IF IT IS EQUAL TO ZERO THE QR PROCESS IS NOT NECESSARY.
      DO 16 I=2,N
        DO 16 K=I,N
C           IF(A(I-1,K).NE.0.D0)GO TO 18
      IF (DABS(A(I-1,K)).GT. SMAL2) GOTO 18
      A(I-1,K) = 0.0D00
   16      CONTINUE
C.....................................................................
C.....'EVR'='AWORK' ON FILE 9 & 13
      DO 17 I=1,N
      IWORK(I)=1
      AWORK(I)=A(I,I)
   17 CONTINUE
      GOTO 37
C
C START THE MAIN LOOP OF THE QR PROCESS.
   18 K=M-1
      M1=K
      I = K
      DO 85 I1=1,N
      DO 85 I2=1,N
      IF(DABS(A(I1,I2)) .LT. 1.0D-20) A(I1,I2)=0.0D00
   85 CONTINUE
C FIND ANY DECOMPOSITIONS OF THE MATRIX.
C JUMP TO 34 IF THE LAST SUBMATRIX OF THE DECOMPOSITION IS
C OF THE ORDER ONE.
C JUMP TO 35 IF THE LAST SUBMATRIX OF THE DECOMPOSITION IS
C OF THE ORDER TWO.
      IF(K)37,34,19
   19 IF(DABS(A(M,K)).LE.EPS)GO TO 34
      IF(M-2.EQ.0)GO TO 35
   20   I = I-1
        IF(DABS(A(K,I)).LE.EPS)GO TO 21
        K = I
        IF(K.GT.1)GO TO 20
   21 IF(K.EQ.M1)GO TO 35
      IF(ISHIFT)PRINT 999,R
  999 FORMAT (5X, '***R=', D25.16)
C TRANSFORMATION OF THE MATRIX OF THE ORDER GREATER THAN TWO
      S = A(M,M)+A(M1,M1)+SHIFT
      SR = A(M,M)*A(M1,M1)-A(M,M1)*A(M1,M) + 0.25D0*SHIFT*SHIFT
      A(K+2,K) = 0.D0
C CALCULATE X1,Y1,Z1,FOR THE SUBMATRIX OBTAINED BY THE
C DECOMPOSITION.
      X = A(K,K)*(A(K,K)-S)+A(K,K+1)*A(K+1,K)+SR
      Y = A(K+1,K)*(A(K,K)+A(K+1,K+1)-S)
      R = DABS(X)+DABS(Y)
      IF(R.GT.SMAL2) GOTO 215
      IF(ISHIFT) GO TO 215
      SHIFT=A(M,M-1)
      ISHIFT=.TRUE.
      GO TO 21
  215 CONTINUE
      ISHIFT=.FALSE.
C     IF(R.EQ.0.D0)SHIFT = A(M,M-1)
C     IF(R.EQ.0.D0)GO TO 21
      Z = A(K+2,K+1)*A(K+1,K)
      SHIFT = 0.D0
      NS =NS+1
C
C THE LOOP FOR ONE STEP OF THE QR PROCESS.
      DO 33 I=K,M1
        IF(I.EQ.K)GO TO 22
C CALCULATE XR,YR,ZR.
        X = A(I,I-1)
        Y = A(I+1,I-1)
        Z = 0.D0
        IF(I+2.GT.M)GO TO 22
        Z = A(I+2,I-1)
   22   SR2 = DABS(X)+DABS(Y)+DABS(Z)
      IF(SR2.LT.SMAL2) SR2= 0.0D00
      IF(SR2.LT.SMAL2) GOTO 23
C       IF(SR2.EQ.0.D0)GO TO 23
        X = X/SR2
        Y = Y/SR2
        Z = Z/SR2
   23   S = DSQRT(X*X + Y*Y +Z*Z)
        IF(X.LT.0.D0)GO TO 24
        S = -S
   24   IF(I.EQ.K)GO TO 25
        A(I,I-1) = S*SR2
   25 IF(DABS(SR2).GT.SMAL2) GOTO 26
C  25   IF (SR2.NE.0.D0)GO TO 26
        IF(I+3.GT.M)GO TO 33
        GO TO 32
   26   SR = 1.D0-X/S
        S = X-S
        X = Y/S
        Y = Z/S
C PREMULTIPLICATION BY THE MATRIX PR.
        DO 28 J=I,M
          S = A(I,J)+A(I+1,J)*X
          IF(I+2.GT.M)GO TO 27
          S = S+A(I+2,J)*Y
   27     S = S*SR
          A(I,J) = A(I,J)-S
          A(I+1,J) = A(I+1,J)-S*X
          IF(I+2.GT.M)GO TO 28
          A(I+2,J) = A(I+2,J)-S*Y
   28     CONTINUE
C POSTMULTIPLICATION BY THE MATRIX PR.
        L = I+2
        IF(I.LT.M1)GO TO 29
        L = M
```

```
   29    DO 31 J=K,L
            S = A(J,I)+A(J,I+1)*X
            IF(I+2.GT.M)GO TO 30
            S = S + A(J,I+2)*Y
   30       S = S*SR
            A(J,I) = A(J,I)-S
            A(J,I+1)=A(J,I+1)-S*X
            IF(I+2.GT.M)GO TO 31
            A(J,I+2)=A(J,I+2)-S*Y
   31       CONTINUE
         IF(I+3.GT.M)GO TO 33
         S = -A(I+3,I+2)*Y*SR
   32    A(I+3,I) = S
         A(I+3,I+1) = S*X
         A(I+3,I+2) = S*Y + A(I+3,I+2)
   33    CONTINUE
C
      IF(NS.GT.MAXST)GO TO 37
      GO TO 18
C
C COMPUTE THE LAST EIGENVALUE.
   34 AWORK(M)=A(M,M)
      IWORK(M)=1
      M = K
      GO TO 18
C
C COMPUTE THE EIGENVALUES OF THE LAST 2X2 MATRIX OBTAINED BY
C THE DECOMPOSITION.
   35 R = 0.5D0*(A(K,K)+A(M,M))
      S = 0.5D0*(A(M,M)-A(K,K))
      S = S*S + A(K,M)*A(M,K)
      IWORK(K)=1
      IWORK(M)=1
      IF(S.LT.0.D0)GO TO 36
      T = DSQRT(S)
      AWORK(K)=R-T
      AWORK(M)=R+T
      M = M-2
      GO TO 18
   36 CONTINUE
      PRINT 3010,S
 3010 FORMAT(5X,'VALUE OF S IS NEGATIVE=',D14.6)
      T=DSQRT(-S)
      AWORK(K)=R
      AWORK(M)=R
      M=M-2
      GOTO 18
C
   37 CONTINUE
C.....'BWORK'='H' ON FILE 2
C.....'AWORK'='SUBDIA' ON FILE 12
C.....'AWORK'='EVR' ON FILE 9 & 13
C.....'IWORK'= 'INDIC' ON FILE 1
      ID1=1
      ID9=1
      DO 915 I1=1,N
      WRITE(1'ID1) IWORK(I1)
      WRITE(9'ID9 ) AWORK(I1)
  915 CONTINUE
      REWIND 13
      WRITE (13)(AWORK(I1),I1=1,N)
      PRINT 917,(I1,IWORK(I1),AWORK(I1),I1=1,N)
C     PRINT 919, (I1,IWORK(I1),I1=1,N)
      DO 925 I1=1,N
      IF(IWORK(I1) .EQ. 0) CALL EXIT
  925 CONTINUE
      RETURN
  919 FORMAT (1X,'**EVR-INDIC**',/,(1X,16(1X,I4,'-',I2)))
  917 FORMAT (1X,'***I1,INDIC(I1),EVR(I1)***', /,(1X,5(1X,I7,I4,D13.6)))
      END



      SUBROUTINE REALVE
      IMPLICIT REAL *8(A-H,O-Z)
C
      SMAL2=1.0D-20
      COMMON/BLOCK1/AWORK(190),BWORK(190),A(190,190)
      COMMON/BLOCK2/IWORK(190),N,IVEC,M
      COMMON/BLOCK3/ENORM,EPS,EX,T
      COMMON/AFILE/ID1,ID2,ID3,ID4,ID8,ID9
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI,SMALL
      DEFINE FILE 1(190,1,U,ID1),2(190, 380,U,ID2),3(190,1,U,ID3),
     114(190,380,U,ID4),8(190,2,U,ID8),9(190,2,U,ID9)
C
      VECR1I=ONE
      IF(M.EQ.1)GO TO 24
C SMALL PERTURBATION OF EQUAL EIGENVALUES TO OBTAIN A FULL
C SET OF EIGENVECTORS.
C.....'AWORK'='EVR' ON FILE 9
C.....'AWORK'='EVR' ON FILE 13
      REWIND 13
      READ (13)(AWORK(I1),I1=1,N)
      EVALUE = AWORK(IVEC)
      K = IVEC+1
      R = 0.D0
      IF(IVEC.EQ.M)GO TO 2
      DO 1 I=K,M
      IF(IDINT(EVALUE*1.D9).NE.IDINT(AWORK(I)*1.D9)) GOTO 1
C     IF(EVALUE .NE.AWORK(I)) GOTO 1
        R = R+3.D0
    1   CONTINUE
      EVALUE = EVALUE+R*EX
    2 DO 3 K=1,M
    3   A(K,K) = A(K,K)-EVALUE
C
C GAUSSIAN ELIMINATION OF THE UPPER-HESSENBERG MATRIX A. ALL
C ROW INTERCHANGES ARE INDICATED IN THE ARRAY IWORK.ALL THE
C MULTIPLIERS ARE STORED AS THE SUBDIAGONAL ELEMENTS OF A.
      K = M-1
C.....'IWORK'='IWORK' NOT REQD. TO STORE ON FILE
      DO 85 I1=1,N
      DO 85 I2=1,N
      IF(DABS(A(I1,I2)) .LT. 1.0D-20) A(I1,I2)=0.0D00
   85 CONTINUE
      DO 8 I=1,K
```

```
      L = I+1
      IWORK(I) = 0
     IF(DABS(A(I+1,I)) .GT. SMAL2) GOTO 4
     IF(DABS(A(I   ,I)) .GT. SMAL2) GOTO 8
C       IF(A(I+1,I).NE.0.D0)GO TO 4
C       IF(A(I,I).NE.0.D0)GO TO 8
        A(I,I) = EPS
        GO TO 8
    4   IF(DABS(A(I,I)).GE.DABS(A(I+1,I)))GO TO 6
        IWORK(I) = 1
        DO 5 J=I,M
          R = A(I,J)
          A(I,J) = A(I+1,J)
    5     A(I+1,J) = R
    6   R = -A(I+1,I)/A(I,I)
        A(I+1,I) = R
        DO 7 J=L,M
    7     A(I+1,J) = A(I+1,J)+R*A(I,J)
    8   CONTINUE
      IF(DABS(A(M   ,M)) .GT. SMAL2) GOTO 9
C     IF(A(M,M).NE.0.D0)GO TO 9
      A(M,M) = EPS
    9 CONTINUE
C
C THE VECTOR (1,1,....,1) IS STORED IN THE PLACE OF THE RIGHT
C HAND SIDE COLUMN VECTOR.
C.....'BWORK'='WORK' NOT REQD. TO STORE ON FILE
C.....INITIALIZE 'BWORK'
      DO 11 I=1,N
        IF(I.GT.M)GO TO 10
      BWORK(I)=ONE
        GO TO 11
   10 BWORK(I)=ZERO
   11   CONTINUE
C
C THE INVERSE ITERATION IS PERFORMED ON THE MATRIX UNTIL THE
C INFINITE NORM OF THE RIGHT-HAND SIDE VECTOR IS GREATER
C THAN THE BOUND DEFINED AS  0.01/(N*EX).
      BOUND = 0.01D0/(EX * DFLOAT(N))
      NS = 0
      ITER = 1
C
C THE BACKSUBSTITUTION.
   12 R = 0.D0
      DO 15 I=1,M
        J = M-I+1
      S=BWORK(J)
        IF(J.EQ.M)GO TO 14
        L = J+1
        DO 13 K=L,M
      SR=BWORK(K)
   13     S   = S - SR*A(J,K)
   14 BWORK(J)=S/A(J,J)
      T=DABS(BWORK(J))
        IF(R.GE.T)GO TO 15
        R = T
   15   CONTINUE
C
C THE COMPUTATION OF THE RIGHT-HAND SIDE VECTOR FOR THE NEW
C ITERATION STEP.
        DO 16 I=1,M
   16 BWORK(I)=BWORK(I)/R
C
C THE COMPUTATION OF THE RESIDUALS AND COMPARISON OF THE
C RESIDUALS OF THE TWO SUCCESSIVE STEPS OF THE INVERSE
C ITERATION.IF THE INFINITE NORM OF THE RESIDUAL VECTOR IS
C GREATER THAN THE INFINITE NORM OF THE PREVIOUS RESIDUAL
C VECTOR THE COMPUTED EIGENVECTOR OF THE PREVIOUS STEP IS
C TAKEN AS THE FINAL EIGENVECTOR.
      R1 = 0.D0
      DO 18 I=1,M
        T = 0.D0
        DO 17 J=I,M
   17 T=T+A(I,J)*BWORK(J)
        T = DABS(T)
        IF(R1.GE.T)GO TO 18
        R1= T
   18   CONTINUE
      IF(ITER.EQ.1)GO TO 19
      IF(PREVIS.LE.R1)GO TO 24
C.....'VECR(I,IVEC)'='AWORK' ON FILE 14
   19 DO 20 I=1,M
   20 AWORK(I)=BWORK(I)
      PREVIS = R1
      IF(NS.EQ.1)GO TO 24
      IF(ITER.GT.6)GO TO 25
      ITER = ITER+1
      IF(R.LT.BOUND)GO TO 21
      NS = 1
C
C GAUSSIAN ELIMINATION OF THE RIGHT-HAND SIDE VECTOR.
   21 K = M-1
      DO 23 I=1,K
      R=BWORK(I+1)
      IF(IWORK(I).EQ.0)GO TO 22
      BWORK(I+1)=BWORK(I)+BWORK(I+1)*A(I+1,I)
      BWORK(I)=R
        GO TO 23
   22 BWORK(I+1)=BWORK(I+1)+BWORK(I)*A(I+1,I)
   23   CONTINUE
      GO TO 12
C
C.....'INDIC(IVEC)'='INDIVC' ON FILE 1
   24 INDIVC=2
      IF (M.EQ.1)AWORK(1)=VECR1I
      ID1=IVEC
      WRITE(1'ID1) INDIVC
   25 IF(M.EQ.N)GO TO 27
      J = M+1
C.....'VECR(I,IVEC)'='AWORK' ON FILE 14
      DO 26 I=J,N
   26 AWORK(I)=ZERO
   27 CONTINUE
C.....'VECR'='AWORK' ON FILE 14
      ID4=IVEC
      WRITE (14'ID4) (AWORK(I1),I1=1,N)
   95 FORMAT (10X,'IVEC,INDIVC', 2I5)
C     PRINT 97, ITER, IVEC,(I1,AWORK(I1),I1=1,N)
```

```
 97 FORMAT (5X,'ITER & IVEC',2I5,/, (3X,7(1X,I4,'-',D12.5)))
    RETURN
    END




//GO.SYSIN DD *
   7   95
        1
   48
        2
   3   93
//GO.FT11F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,55),
// DCB=(LRECL=1680,BLKSIZE=2298,RECFM=VBS),
//     DSN=OSU.ACT10188.MASSINVK
//GO.FT13F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(NEW,KEEP),SPACE=(TRK,1),
// DCB=(LRECL=1680,BLKSIZE=2298,RECFM=VBS),
//     DSN=OSU.ACT10188.EIGVALUE
//GO.FT14F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(NEW,KEEP),SPACE=(TRK,55),
// DCB=(LRECL=1680,BLKSIZE=2298,RECFM=VBS),
//     DSN=OSU.ACT10188.EIGVECTR
//GO.FT01F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=1754,LRECL=1750,RECFM=VBS)
//GO.FT02F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=1754,LRECL=1750,RECFM=VBS)
//GO.FT03F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=1754,LRECL=1750,RECFM=VBS)
//GC.FT04F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=1754,LRECL=1750,RECFM=VBS)
//GO.FT08F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=1754,LRECL=1750,RECFM=VBS)
//GC.FT09F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=1754,LRECL=1750,RECFM=VBS)
//GO.FT12F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=1754,LRECL=1750,RECFM=VBS)
//




C
C
C.....THE FOLLOWING PROGRAM FOLLOWS THE ABOVE PROGRAM. IT SELECTS
C     ALL FLEXURAL MODES AND PERFORMS SUPERPOSITION.
C
C
// EXEC FORTGCLG,REGION.GO=190K
//FORT.SYSIN DD *
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION  OMGASQ(190),FORCE(190),AMAS(190),DISP(190, 65),
     1PHI(190), COSWT( 65) ,MODE(190),TIME( 65)
      DEFINE FILE 14(190,380,U,ID4)
C.....NOTE: THE VALUE OF NUMDT MUST BE EQUAL OR LESS THAN DIMENSION OF
C.....TIME (HERE 65)
      JJ=1111
      CALL ELAPSE (I1)
      PRINT 65, JJ,I1
      CALL ELAPSE (I1)
      PRINT 65, JJ,I1
      READ 10,LIMIT,NUMNP
      N=2*NUMNP
      MBAND=2*LIMIT
      PRINT 20,LIMIT,MBAND,NUMNP,N
C
C     COMPUTE VALUE OF NO. OF BLOCKS 'NBLK'
C
      NBLK=NUMNP/LIMIT
      I=MOD(NUMNP,LIMIT)
      IF(I.GT.0) NBLK=NBLK+1
      PRINT   30, NBLK
      REWIND 3
      REWIND 13
C
C    READ BLOCKS OF MASS MATRIX  (AMAS) ON UNIT 3
C
      DO 693 I1=1, NBLK
      KSHIFT=MBAND*(I1-1)
      J1=KSHIFT+1
      J2=KSHIFT+MBAND
      IF(J2.GT.N    ) GOTO 600
      READ(3)( AMAS(I2),I2=J1,J2)
      GO TO 693
  600 READ(3)(    PHI(I2),I2=1,MBAND)
      I4=N   -J1+1
      DO 620 I2=1,I4
  620  AMAS(J1+I2-1)=    PHI(I2)
  693 CONTINUE
      PRINT 60, (AMAS(I1),I1=1,N)
C
      READ 30, NUMDT,PERIOD,DT
      PRINT 30, NUMDT, PERIOD, DT
      DO 15 I=1,N
      FORCE (I)=0.0D00
      DO 15 K=1,NUMDT
   15 DISP(I,K)= 0.D00
C.....THE FOLLOWING DOLOOP COMPUTES:
C.......NMODE=NUMBER OF FLEXURAL MODES
C.......MODE(IJ) IS ARRAY CONTAINING SEQUENCE NOS. OF FLEXURAL MODES.
      READ (13) (OMGASQ(I),I=1,N)
C     PRINT 60, (OMGASQ(I),I=1,N)
      NMODE=0
      DO 80 J=1,N
      ID4=J
      READ (14'ID4) (PHI(I),I=1,N)
      IF( PHI(96).EQ. 0.0D00) GOTO 80
      IF (IDINT(PHI(86)*1.0D05) .EQ. -IDINT(PHI(106)*1.0D05)) GOTO 80
      IF (IDINT(PHI(16)*1.0D05) .EQ. -IDINT(PHI(176)*1.0D05)) GOTO 80
      IF (PHI(95) .NE. 0.0D00) GOTO 80
      NMODE=NMODE+1
      MODE(NMODE)=J
   80 CONTINUE
      PRINT 50, NMODE, (MODE(J),J=1,NMODE)
```

```
C
C.....AFTER COMPLETION OF DULOOP 80, VALUE OF NMODE IS EQUAL TO
C      TOTAL NUMBER OF FLEXURAL MODES.
       READ 35, II, (J,FORCE(J),I=1,II)
       PRINT 35,II,(J,FORCE(J), I=1,II)
       CALL ELAPSE (I)
       PRINT 65, JJ,I
C.....THE FOLLOWING DOLOOP RESTORES SEQUENCE NOS. OF MODES IN AN ORDER
C.....SUCH THAT THE LOWEST FREQUENCY MODE NOS. ARE FIRST IN ARRAY MODE(I)
C.....AND HIGHEST FREQUENCY MODE NOS. ARE LAST.
       IS=1
  300 XJ=1.0D70
       DO 400 I=IS,NMODE
       IF(OMGASQ(MODE(I)).GT.XJ) GOTO 400
       XJ=OMGASQ(MODE(I))
       JX=MODE(I)
       NI=I
  400 CONTINUE
       NTEMP=MODE(IS)
       MODE(IS)=JX
       MODE(NI)=NTEMP
       IS= IS+1
       IF(IS.LE.NMODE) GOTO 300
       PRINT 50, NMODE, (MODE(J),J=1,NMODE)
C.....THE FOLLOWING DOLOOP SELECTS THE MODES WHOSE PERIOD IS LARGER THAN
C.....3% OF THE LARGEST PERIOD FOR THE PURPOSE OF SUPERPOSITION. MODES
C.....WITH PERIOD LESS THAN 3% OF THE LARGEST PERIOD BEING NEGLECTED.
       XX=OMGASQ(MODE(1))*1.0D03
       DO 23 J=2,NMODE
       MODSML=J
       IF (OMGASQ(MODE(J)) .GT. XX) GO TO 24
   23 CONTINUE
C
C.....AFTER COMPLETION OF DULOOP 23, VALUE OF MODSML IS EQUAL
C      TO NUMBER OF FLEXURAL MODES WHOSE PERIOD IS LARGER THAN
C      3% OF THE LARGEST PERIOD.
C
   24 CONTINUE
C
C.....IF MODES, WHOSE PERIOD IS SMALLER THAN 3% OF THE LARGEST
C      PERIOD, ARE TO BE NEGLECTED, SET NMODE EQUAL TO MODSML.
C
       NMODE = MODSML
   50 CONTINUE
       PRINT 40, NMODE
C
       DO 25 JJ=1,NMODE
       J=MODE(JJ)
       ID4= J
       OMEGA = DSQRT (OMGASQ(J))
       CJN =0.0D00
       CJD =0.0D00
       READ (14'ID4) (PHI(I),I=1,N)
       IF(PHI(96) .GE. 0.0D00) GOTO 33
       DO 31 I=1,N
   31 PHI(I)= -PHI(I)
   33 CONTINUE
C      PRINT 60, (PHI(I), I=1,N)
       DO 34 I=1,N
       CJN = CJN + PHI(I) * FORCE(I)
       CJD = CJD + PHI(I) * AMAS(I)*PHI(I)
   34 CONTINUE
       CJ = CJN/ (CJD*OMGASQ(J))
       DTOMGA=DT*OMEGA
C      PRINT 60, CJ,CJN,CJD
       DO 45 IT=1,NUMDT
       OMTIME = DFLOAT (IT-1)*DTOMGA
       COSWT (IT) =1.0D0- DCOS(OMTIME)
   45 CONTINUE
       DO 55 I=1,N
C      IF (I.EQ. 7)PRINT 40,J,(DISP(6,K),K=1,NUMDT)
       PHICJ = PHI(I)*CJ
C      IF (I .EQ. 6) PRINT 40, J,PHICJ
       DO 55 K=1,NUMDT
       DISP (I,K) =DISP (I,K) + PHICJ*COSWT(K)
   55 CONTINUE
       CALL ELAPSE (I)
       PRINT 65, JJ,I
   25 CONTINUE
C      DO 64 I=6,N,10
       I=96
       PRINT 40, I,(DISP(I,K),K=1,NUMDT)
   64 CONTINUE
       JJ=2222
       CALL ELAPSE (I)
       PRINT 65, JJ,I
       DO 70 K=1,NUMDT
       COSWT(K) = DISP(96,K)
       TIME(K) = DFLOAT(K-1)* DT
   70 CONTINUE
       CALL GRAPH (TIME, COSWT,NUMDT,1)
       JJ=2222
       CALL ELAPSE (I)
       PRINT 65, JJ,I
C
       STOP
   10 FORMAT(2I5)
   20 FORMAT (10X,'LIMIT=',I3,5X,'MBAND=',I3,/,
      110X,'NUMNP=',I4,'  N =',I4,////)
   30 FORMAT (I10,2F20.6)
   35 FORMAT(I10,/,4(I5,D15.2))
   40 FORMAT(I10,/,(1X,10D13.5))
   50 FORMAT (16I5)
   60 FORMAT (  1X,10D13.5,/)
   65 FORMAT (1X,'JJ=', I5,'TIME=', I10)
       END

//GO.SYSIN DD *
    7   95
         60 0.0014                  0.00005
          1
   96       30.00 04
//GO.FT03F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,1),
```

```
// DCB=(LRECL=112,BLKSIZE=1684,RECFM=VBS),
//     DSN=OSU.ACT10188.MASSMATX
//GO.FT13F001   DD   UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,1),
// DCB=(LRECL=1680,BLKSIZE=2298,RECFM=VBS),
//     DSN=OSU.ACT10188.EIGVALUE
//GO.FT14F001   DD   UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,55),
// DCB=(LRECL=1680,BLKSIZE=2298,RECFM=VBS),
//     DSN=OSU.ACT10188.EIGVECTR
//
```

```
C    +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    +                                                                   +
C    + PROGRAM                                                           +
C    + ''DYNAMIC ANALYSIS USING STEP-BY-STEP INTEGRATION''               +
C    +                                                                   +
C    + LANGUAGE               :  FORTRAN IV                              +
C    + DIGITAL COMPUTER       :  IBM 360/65                              +
C    + PROGRAMMER             :  BRIJ R. KISHORE                         +
C    +                           STRUCTURAL ENGINEER                     +
C    +                           U. S. ARMY, CORPS OF ENGINEERS          +
C    +                           CHICAGO, ILLINOIS                       +
C    +                                                                   +
C    + PURPOSE                                                           +
C    +                                                                   +
C    + THIS PROGRAM FORMULATES MASS MATRIX, STRUCTURE STIFFNESS          +
C    + MATRIX, AND COMPUTES DISPLACEMENTS OF ALL THE NODAL POINTS        +
C    + OF THE STRUCTURE. DETAILED INFORMATION CAN BE FOUND IN:           +
C    + ''CHARACTERISTICS OF FINITE ELEMENTS FOR ANALYSIS OF              +
C    + BEAMS SUBJECTED TO IMPACT LOAD'', PH.D. DISSERTATION              +
C    + BY BRIJ R. KISHORE, SCHOOL OF CIVIL ENGINEERING,                  +
C    + OKLAHOMA STATE UNIVERSITY, JULY 1972.                             +
C    +                                                                   +
C    +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    +                                                                   +
C    +                                                                   +
C    + DESCRIPTION OF PARAMETERS:                                        +
C    +                                                                   +
C    + SEE ALSO DESCRIPTION OF PARAMETERS GIVEN IN PROGRAM  FOR          +
C    + STATIC ANALYSIS IN APPENDIX B.                                    +
C    +                                                                   +
C    + ACC (I)       = ACCELERATION IN I-TH DIRECTION                    +
C    + VEL (I)       = VELOCITY IN I-TH DIRECTION                        +
C    + DISP (I)      = DISPLACEMENT IN I-TH DIRECTION                    +
C    + T             = TIME                                              +
C    + TEND          = TIME AT WHICH FORCE BECOMES ZERO                  +
C    + FORCE (I)     = APPLIED FORCE AT ANY TIME T IN I-TH DIRECTION     +
C    + TEND1         = TEND - DT/2                                       +
C    + TEND2         = TEND + DT/2                                       +
C    + DT            = STEP INCREMENT OF TIME                            +
C    + RERUN         = LOGICAL VARIABLE; IF PROGRAM IS BEING RUN         +
C    +                 FOR 1-ST TIME: RERUN = .FALSE. . IF THE           +
C    +                 PROGRAM IS BEING RUN FOR THE CONTINUATION         +
C    +                 OF THE PREVIOUS INTEGRATION STARTING FROM         +
C    +                 THE TIME T WHERE FIRST RUN STOPPED,               +
C    +                 RERUN = .TRUE. . THIS MAKES PROGRAM TO READ       +
C    +                 DATA FOR CONTINUATION FROM THE DISK.              +
C    + RERUNT        = TIME FOR WHICH THE PROGRAM CALCULATED             +
C    +                 DISPLACEMENT IN THE PREVIOUS RUN.                 +
C    +                                                                   +
C    +                                                                   +
C    +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
// EXEC PGM=GOGO
//D1 DD DSN=OSU.ACT10188.NONZERON,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D2 DD DSN=OSU.ACT10188.MASSMATX,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D3 DD DSN=OSU.ACT10188.TOTLSTIF,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
// EXEC FORTHCLG,REGION.GO=241K
//FORT.SYSIN DD *
C
C
C*****PROGRAM FORMULATES MASS MATRIX, STRUCTURE STIFFNESS MATRIX
C     AND STORES ON DISK ONLY NONZERO TERMS OF STIFFNESS MATRIX
C     OF EACH BLOCK. NONZERON IS NUMBER OF ELEMENTS STORED ON DISK
C     WHICH ARE NONZERO.
C
C
      IMPLICIT REAL *8 (A-H,O-Z)
C
      COMMON /AAA/ X(343),Y(343),UX(343),UY(343),UXTYPE(343),UYTYPE(343)
     1,ELMAS(288),HED(18),TYPE(8),E,DENS,PR,VOL,MTYPE(288),NUMNP,
     2NUMEL,NUMAT,KN,NCASE,KOUNT
      COMMON /ARG/ XXX(5),YYY(5),S(10,10),DD(3,3),HH(6,10),P(10),XX(4),
     1YY(4),C(4,4),H(6,10),D(6,6),F(6,10),
     2TYPE1,TYPE2,TEST1,TEST2,IX(288,4),LM(4),NR,LIMIT,ISTART
      COMMON /BANARG/ A(36,18),FM(36),B(36),MBAND,NUMBLK
      COMMON/AIJFM/AA(40,18,18),FFM(40,18)
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      COMMON /AZZA/ AZZ(18,40)
      COMMON/SQUAD/SQ(10,10)
C
      KOUNT=0
      CALL ELAPSE(I)
      PRINT 3224, I
      KOUNT=KOUNT+1
      CALL MANE
      CALL ELAPSE(I)
      PRINT 3224, I
C
 9999 STOP
 3224 FORMAT (I10)
      END



      SUBROUTINE MANE
        .
        .
        .
        .
      (THIS SUBROUTINE IS LISTED IN APPENDIX B)
        .
        .
        .
        .
      RETURN
      END
```

```
      SUBROUTINE MASTIF
         .
         .
         .
         .
      (THIS SUBROUTINE IS LISTED IN APPENDIX B)
         .
         .
         .
         .
      RETURN
      END
```

```
      SUBROUTINE STIF88 (N)
         .
         .
         .
         .
      (THIS SUBROUTINE IS LISTED IN APPENDIX B)
         .
         .
         .
         .
      RETURN
      END
```

```
// EXEC PGM=GOGO
//D1 DD DSN=OSU.ACT12387.ACVELDIS,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D2 DD DSN=OSU.ACT12387.BFMIJK,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
//D3 DD DSN=OSU.ACT12387.TYMACDIS,
//   UNIT=2314,VOL=SER=DISK06,DISP=(OLD,DELETE)
// EXEC FORTHCLG,REGION.GO=190K
//FORT.SYSIN DD *
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C- ON FILE   1---'BXB(N),(AFM(N,M),M=2,MM)
C- ON FILE   2---'N' OR 'NONZERON'
C- ON FILE   3---'FFM' OR 'MASSMATX'
C- ON FILE   4---'AA(I1,I2,I3)' OR 'TOTLSTIF' IN BLOCKS
C- ON FILE   8---'FORCE(N),BX(N),AX(N),N=1,ND'
C- ON FILE   9---'ACC(N),VEL(N),DISP(N),N=1,ND'
C- ON FILE  11---'BXB(N),N=NL,NH'
C- ON FILE  12---'BFM(I1,I2,I3)' OR 'AFM(I2,I3)' IN BLOCKS
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      IMPLICIT REAL *8 (A-H,O-Z)
      LOGICAL *1 RERUN
      COMMON /AAA/ X(95),Y(95),UX(95),UY(95),UXTYPE(95),UYTYPE(95),
     1ELMAS(72),HED(18),TYPE( 8),E,DENS,PR,VOL,MTYPE(72), NUMNP,
     2NUMEL,NUMAT,KN
      COMMON/ARG/
     2 TYPE1,TYPE2,TEST1,TEST2,IX(72 ,4),LM(4),NR,LIMIT,    ISTART
      COMMON /BANARG/ A(28,14),FM(28),B(28),MBAND,NUMBLK,NBLK
      COMMON/ AIJFM/ AA(14,14,14),FFM(14,14)
      COMMON/DYNAM/AFM(28,14),BXB(28),BX(28),XB(28),AX(28),ACC(28),
     1VEL(28),DISP(28),FSTRT(28),FORCE(28),DT ,NODE(15)
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      COMMON/BLOCKZ/T,TEND,TEND1,TEND2,RERUN
CC    DIMENSION DISPX(200),DISPY(200) ,TIME(200)
      DIMENSION DISPX(50),DISPY(50),TIME(50)
C*************************************************************************
      RERUNT =0.0D00
C     RERUN=.TRUE.
      RERUN=.FALSE.
      IF(RERUN) READ 3220,RERUNT
      PRINT 3220, RERUNT
C*************************************************************************
      LIMIT = 7
      SPAN =16.0D00
      NUMNP= 95
      MBAND = 2*LIMIT
      CUT = 0.5D 01
C**** ********************************************************************
C**** ********************************************************************
C**** ********************************************************************
      DO 689 I=1,NUMNP
      UXTYPE(I) = TYPE1
      UYTYPE(I) =TYPE1
      UX(I) =ZERO
  689 UY(I) =ZERO
      REWIND 11
      REWIND 2
      REWIND 3
      REWIND 4
C---THIS 'REWIND 9' IS NEEDED FOR 'RERUN'.
      REWIND 9
      UXTYPE(48) = TYPE2
      UYTYPE (3) = TYPE2
      UYTYPE(93) = TYPE2
C.....INCDE= TOTAL NUMBER OF NODES WITH APPLIED IMPACT LOAD.
C.....NODE(I)=NODAL NUMBERS WHICH HAVE APPLIED LOAD .
      READ 690 ,INODE,(NODE(I),I=1,INODE)
      PRINT690 ,INODE,(NODE(I),I=1,INODE)
      NB=LIMIT
      ND=2*NB
      ND2= 2*ND
      NBLK=NUMNP/LIMIT
      I=MOD(NUMNP,LIMIT)
      IF(I .GT. 0)  NBLK=NBLK+1
C     PRINT 3224,NBLK
      PRINT 3224,NBLK
      DO 691 I1=1,NBLK
      DO 691 I2=1,ND
      FFM(I1,I2)=ZERO
      DO 691 I3=1,ND
  691 AA(I1,I2,I3)=ZERO
C
C
```

```
      DO 693 I1=1,NBLK
      READ (3)  (FFM(I1,I2),I2=1,ND)
      READ (2) N
C     PRINT3224,N
      PRINT3224,N
      READ (4) (I2,I3,AA(I1,I2,I3),I4=1,N)
  693 CONTINUE
      DT  = 0.000002D00
      TEND = DT * 39.0
      TEND1=TEND-DT/TWO
      TEND2=TEND+DT/TWO
      TF = TEND
      DO 709 N=1,ND
      ACC(N)=ZERO
      VEL(N)=ZERO
      DISP(N)=ZERO
      FSTRT(N)=ZERO
      BX(N) = ZERO
  709 AX(N)=ZERO
      KOUNT = 0
      JSTART = -1
C***********************************************************************
C
      ISTART =0
      T=ZERO
      IF(RERUN) T= RERUNT
      IF(RERUN) ISTART=1
C
C**** *****************************************************************
      NDT=0
      CALL ELAPSE (IDT)
      PRINT 3230 ,NDT,IDT
C***********************************************************************
      CALL ELAPSE (IDT)
      PRINT 3230 ,NDT,IDT
C
  720 T=T+DT
      NDT=NDT + 1
      ISTART=ISTART + 1
      JSTART = JSTART + 1
      REWIND 8
      NUMBLK =0
      NB = LIMIT
      ND = 2*NB
  730 NUMBLK =NUMBLK+1
      NH =NB*(NUMBLK+1)
      NM=NH - NB
      NL = NM-NB + 1
      KSHIFT = 2*NL -2
      IF( NM .GT. NUMNP ) NM=NUMNP
      DO 710 I=1,ND
  710 FORCE(I) = ZERO
      IF (T .GT. TF) GO TO 735
      DO 712 I=1,INODE
      N=NODE(I)
      IF(N .LT. NL .OR. N .GT. NM) GO TO 712
      JJ=2*N -KSHIFT
      IF(ISTART .EQ. 1) FSTRT(JJ)=-30.000004
      FORCE(JJ) = -30.000004
  712 CONTINUE
      IF (ISTART .NE. 1) GO TO 735
       DO 726 I=1,ND
      AX(I)=ZERO
  726 BX(I)=ZERO
      DO 790 I=1,INODE
      J=NODE(I)
      IF(J .LT. NL .OR. J .GT. NM) GO TO 790
      L=2*J-KSHIFT - 2
      DO 792 LK=1,2
      II= L+LK
      AX(II) = FSTRT(II) *DT/TWO
      BX(II)=FSTRT(II)*(DT**2)/THREE
      FSTRT(II)=ZERO
  792 CONTINUE
  790 CONTINUE
      GO TO 795
  735 READ (9) (ACC(N),VEL(N),DISP(N),N=1,ND)
      DO 791  J =NL,NM
      L=2*J-KSHIFT - 2
      DO 791 LK =1,2
      II= L+LK
      AX(II)=VEL(II)+(DT/TWO)*ACC(II)
  791 BX(II)=DISP(II)+DT*VEL(II)+((DT**2)/THREE)*ACC(II)
  795 CONTINUE
      WRITE(8) (FORCE(N),BX(N),AX(N), N=1,ND)
C**** *****************************************************************
C*              CHECK FOR LAST BLOCK
C**** *****************************************************************
      IF(NM-NUMNP) 730,810,810
C**** *****************************************************************
  810 CALL BANSOL
C**** *****************************************************************
C**** *****************************************************************
C*        ACCELERATION,VELOCITY AND DISPLACEMENT AT TIME T            *
C**** *****************************************************************
C
      IF(T.GT. TEND1) JSTART=0
      IF(JSTART .EQ. 5 ) JSTART=0
      IF(JSTART.NE. 0) GOTO 815
      KOUNT = KOUNT + 1
C     TYME=T
      TIME(KOUNT) =T
      NB=LIMIT
      ND=2*NB
      ND2=2*ND
      NUMBLK =0
C***********************************************************************
      J = 48
C***********************************************************************
  805 NUMBLK = NUMBLK+1
      NH=NB *(NUMBLK+1)
      NM=NH-NB
      NL=NM-NB+1
      KSHIFT =2*NL-2
      IF( NM .GT. NUMNP ) NM=NUMNP
      READ(9) (ACC(N),VEL(N),DISP(N),N=1,ND)
      IF(J .LT. NL .OR. J .GT. NM) GO TO 805
      L=2*J - KSHIFT
```

```
      DISPX(KOUNT) =ACC(L)
      DISPY(KOUNT) = DISP(L)
C     ACCLNN=ACC(L)
C     DISPLY=DISP(L)
C     CUTT=DISPLY
      CUTT = DISPY(KOUNT)
      PRINT 816,KOUNT,TIME(KOUNT),DISPX(KOUNT),DISPY(KOUNT)
C     WRITE(13) KOUNT,TYME,ACCLNN,DISPLY
      IF ( DABS(CUTT)  .GT. CUT) CALL EXIT
      REWIND 9
  815 CONTINUE
      IF(NDT.EQ. 1) JSTART=1
      IF (T .LE. TEND) GO TO  720
      CALL ELAPSE (IDT)
      PRINT 3230 ,NDT,IDT
C     PRINT 816,(I,TIME(I),DISPX(I),DISPY(I), I=1,KOUNT)
C     CALL GRAPH (TIME,DISPY,KOUNT ,1)
C
      CALL ELAPSE (IDT)
      PRINT 3230 ,NDT,IDT
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
  690 FORMAT ( I5,/, 16(I5))
  816 FORMAT (5X,I5,3D26.16)
 3220 FORMAT                 (1X,3D26.16,1X)
 3223 FORMAT (2(2X,I3,1X,I3,1X,D23.16,7X))
 3224 FORMAT (I10)
 3230 FORMAT (1X,'NDT=',I5,'TIME=',I10)
      STOP
      END



      BLOCK DATA
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ARG/
     2  TYPE1,TYPE2,TEST1,TEST2,IX(72 ,4),LM(4),NR,LIMIT,    ISTART
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      DATA ZERO,HALF,ONE,TWO,THREE,FOUR /0.0D00,0.5D00,1.0D00,2.0D00,
     13.0D00,4.0D00/,SIX/6.0D00/,PI /3.141592653589793/
      DATA TYPE1/4HLOAD/,TYPE2/4HDISP/,TEST1/3HESS/,TEST2/3HAIN/
      END



      SUBROUTINE BANSOL
      IMPLICIT REAL *8 (A-H,O-Z)
      LOGICAL *1 RERUN
      COMMON /AAA/ X(95),Y(95),UX(95),UY(95),UXTYPE(95),UYTYPE(95),
     1ELMAS(72),HED(18),TYPE( 8),E,DENS,PR,VOL,MTYPE(72), NUMNP,
     2NUMEL,NUMAT,KN
      COMMON/ARG/
     2  TYPE1,TYPE2,TEST1,TEST2,IX(72 ,4),LM(4),NR,LIMIT,    ISTART
      COMMON /BANARG/ A(28,14),FM(28),B(28),MBAND,NUMBLK,NBLK
      COMMON/AIJFM/AA(14,14,14),FFM(14,14)
      COMMON/DYNAM/AFM(28,14),BXB(28),BX(28),XB(28),AX(28),ACC(28),
     1VEL(28),DISP(28),FSTRT(28),FORCE(28),DT ,NODE(15)
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      COMMON/BLOCKZ/T,TEND,TEND1,TEND2,RERUN
C     DIMENSION  DUMMY ( 300) , IZ1( 300) , JZ1( 300)
      DIMENSION BFM(14,14,14)
C     PRINT 3224,NUMBLK
      REWIND 1
      REWIND 8
      REWIND 9
      REWIND 11
      REWIND 12
      NN=2*LIMIT
      NL=NN+1
      NH=NN+NN
      NB=0
      MM=MBAND
C
      DO 50 I=1,NH
      BXB(I)= ZERO
       XB(I)= ZERO
      B(I) = ZERO
      DO 50 J=1,NN
   50 AFM(I,J)= ZERO
      GO TO 150
C**** *****************************************************************
C*            REDUCE EQUATIONS BY BLOCKS - SHIFT BLOCK OF EQUATIONS
C**** *****************************************************************
  100 NB=NB+1
C
      DO 125 N=1,NN
      NM=NN+N
      BX(N)=BX(NM)
      BX(NM)= ZERO
      AX(N)=AX(NM)
      AX(NM)= ZERO
      FM(N)=FM(NM)
      FM(NM)= ZERO
      B(N)=B(NM) + FORCE(NM)
      B(NM)= ZERO
      XB(N)=XB(NM)
      XB(NM) = ZERO
      DO 125 M=1,MM
      AFM(N,M)=AFM(NM,M)
      AFM(NM,M)=ZERO
      A(N,M)=A(NM,M)
  125 A(NM,M)= ZERO
C**** *****************************************************************
C*                READ NEXT BLOCK OF EQUATIONS INTO CORE              *
C**** *****************************************************************
      IF(NUMBLK - NB) 150,110,150
C
  150 N=NL-1
      DO 149 I2=1,NN
      N=N+1
      FM(N)=FFM(NB+1,I2)
      DO 149 I3 =1,NN
      A(N,I3)=AA(NB+1,I2,I3)
  149 CONTINUE
```

```
      READ (8) (FORCE(N),BX(N),AX(N),N=NL,NH)
      IF (ISTART .NE. 1) GO TO 152
      DO 151 I=NL,NH
      FMI=FM(I)
      IF (FMI .EQ. ZERO) GO TO 151
      AX(I)=AX(I)/FMI
      BX(I)=BX(I)/FMI
  151 CONTINUE
  152 IF(NB) 162,100,162
C**** ************************************************************
C*              FORM AFM AND BXB MATRICES                        *
C**** ************************************************************
  162 DO 198 N=1,NH
      DO 195 M=1,MM
      IF(NB-1) 164,166,164
  164 IF(N-NN) 176,176,166
  166 IF(M-1) 170,168,170
  168 XB(N)=XB(N)+A(N,M)*BX(N)
      IF (ISTART .GT. 1 ) GO TO 195
      AFM(N,M)=FM(N)+((DT**2)/SIX)*A(N,M)
      GO TO 195
  170 K=N-M+1
      IF(K) 176,176,172
  172 XB(N)=XB(N)+A(K,M)*BX(K)
  176 K=N+M-1
      IF(NH-K) 195,178,178
  178 IF(NB-1) 180,182,180
  180 IF(K-NN) 195,195,182
  182 XB(N)=XB(N)+A(N,M)*BX(K)
      IF (ISTART .GT. 1 ) GO TO 195
      AFM(N,M)=((DT**2)/SIX)*A(N,M)
  195 CONTINUE
  198 CONTINUE
C
  110 DO 120 N=1,NN
  120 BXB(N)=B(N)-XB(N)
C**** ************************************************************
      IF (ISTART .GT. 1) GO TO 999
      DO 680 I2=1,NN
      WRITE (12) (AFM(I2,I3), I3=1,MM)
      DO 680 I3=1,MM
  680 BFM(NB,I2,I3) = AFM(I2,I3)
C     N=0
C     Z98 = 0.1D-11
C     DO 160 I=1,NN
C     DO 160 J=1,NN
C     IF(AFM(I,J) .GT. Z98 .OR. AFM(I,J) .LT. -Z98) GO TO 161
C     GO TO 160
C 161 N=N+1
C     IZ1(N)=I
C     JZ1(N)=J
C     DUMMY(N) = AFM(I,J)
C 160 CONTINUE
C     PRINT 3224, NB
C     PRINT 3224,N
C     PRINT 3222, (IZ1(I),JZ1(I),DUMMY(I), I=1,N)
C     PRINT 3224, NB
C**** ************************************************************
  999 CONTINUE
      WRITE (11)(BXB(N),N=1,NN)
C     DO 700 N=1,NN
C     WRITE (11) BXB(N)
C     WRITE (11) BXB(N),(AFM(N,M),M=I,MM)
C     PRINT 3227,(AFM(N,M),M=1,MM)
C 700 CONTINUE
      IF (NUMBLK .NE. NB) GO TO 100
      IF(RERUN) GOTO 697
      GO TO 698
  697 CONTINUE
      IF (ISTART .NE. 2) GO TO 698
      DO 699 I1=1,NBLK
      DO 699 I2=1,NN
      READ (12) (BFM(I1,I2,I3), I3=1,MM)
  699 CONTINUE
  698 CONTINUE
      REWIND 11
      NB=0
      GO TO 124
  122 NB=NB+1
      NS=LIMIT*(NB+1)
      NK=NS-LIMIT
      NP=NK-LIMIT+1
      KSHIFT = 2* NP -2
      IF( NK .GT. NUMNP ) NK=NUMNP
      DO 123 N=1,NN
      NM = NN+ N
      BXB(N)=BXB(NM)
      BXB(NM)=ZERO
      DO 123 M=1,MM
      AFM(N,M)=AFM(NM,M)
  123 AFM(NM,M)=ZERO
      IF(NUMBLK .EQ. NB) GO TO 126
  124 CONTINUE
      READ (11)(BXB(N),N=NL,NH)
C 124 DO 710 N=NL,NH
C     READ  (11) BXB(N)
C     READ (11) BXB(N),(AFM(N,M),M=1,MM)
C 710 CONTINUE
      N=NL-1
      DO 711 I2=1,NN
      N=N+1
      DO 711 I3=1,MM
      AFM(N,I3)=BFM(NB+1,I2,I3)
  711 CONTINUE
      IF(NB .EQ. 0) GO TO 122
C**** ************************************************************
C**** ************************************************************
C*                         BOUNDARY CONDITIONS                   *
C**** ************************************************************
  126 DO 410 M=NP,NK
      IF(M-NUMNP) 315,315,410
  315 N=2*M-KSHIFT-1
      IF(UXTYPE(M) .NE. TYPE2) GO TO 320
      U=UX(M)
      CALL MODIFY (NH,N,U)
  320 N=N+1
      IF(UYTYPE(M) .NE. TYPE2) GO TO 410
      U=UY(M)
```

```
      CALL MODIFY (NH,N,U)
  410 CONTINUE
C**** ******************************************************************
C*                  REDUCE BLOCK OF EQUATIONS                         *
C**** ******************************************************************
  200 DO 300 N=1,NN
      IF(AFM(N,1)) 225,300,225
  225 BXB(N)=BXB(N)/AFM(N,1)
      DO 275 L=2,MM
      IF(AFM(N,L)) 230,275,230
  230 Q=AFM(N,L)/AFM(N,1)
      I=N+L-1
      J=0
      DO 250 K=L,MM
      J=J+1
  250 AFM(I,J)=AFM(I,J)-Q*AFM(N,K)
      BXB(I)=BXB(I)-AFM(N,L)*BXB(N)
      AFM(N,L)=Q
  275 CONTINUE
  300 CONTINUE
C**** ******************************************************************
C*                 WRITE BLOCK OF REDUCED EQUATIONS ON TAPE 1         *
C**** ******************************************************************
      IF(NUMBLK-NB) 375,400,375
  375 DO 720 N=1,NN
      WRITE(1)  BXB(N),(AFM(N,M),M=2,MM)
  720 CONTINUE
      GO TO 122
C**** ******************************************************************
C*                        BACK SUBSTITUTION                           *
C**** ******************************************************************
  400 DO 450 M=1,NN
      N=NN+1-M
      DO 425 K=2,MM
      L=N+K-1
  425 BXB(N)=BXB(N)-AFM(N,K)*BXB(L)
      NM=N+NN
      BXB(NM)=BXB(N)
  450 AFM(NM,NB)=BXB(N)
      NB=NB-1
      IF(NB) 475,500,475
  475 CONTINUE
      DO 729 N=1,NN
      BACKSPACE 1
  729 CONTINUE
      DO 730 N=1,NN
      READ (1)  BXB(N),(AFM(N,M),M=2,MM)
  730 CONTINUE
      DO 731 N=1,NN
      BACKSPACE 1
  731 CONTINUE
      GO TO 400
C**** ******************************************************************
C*     ORDER UNKNOWNS IN ACC ARRAY AND CALCULATE VEL AND DISP ARRAY   *
C**** ******************************************************************
  500 REWIND 8
      MODL=2*MOD(NUMNP,LIMIT)
C     PRINT 4015, T
C
      DO 600 NB=1,NUMBLK
      K=0
      READ (8) (FORCE(N),BX(N),AX(N),N= 1,NN)
      IF (ISTART .NE. 1) GO TO 519
      DO 518 N=1,NN
      IF(NB .EQ. NUMBLK .AND. N .GT. MODL) GO TO 519
      IF(FFM(NB,N) .EQ. ZERO) GO TO 518
      FMI=FFM(NB,N)
      AX(N)=AX(N)/FMI
      BX(N)=BX(N)/FMI
  518 CONTINUE
  519 CONTINUE
      DO 520 N=1,NN
      IF(NB .EQ. NUMBLK .AND. N .GT. MODL) GO TO 520
      K=K+1
      NM=N+NN
      ACC(K)=AFM(NM,NB)
      VEL(K)=AX(K)+(DT/TWO)*ACC(K)
      DISP(K)=BX(K)+((DT**2)/SIX)*ACC(K)
  520 CONTINUE
      IF(NB .EQ. NUMBLK ) GO TO 523
      GO TO 525
  523 CONTINUE
      MODL1=MODL +1
      DO 524 K=MODL1 , NN
      ACC(K)= ZERO
      VEL (K)=ZERO
      DISP (K) = ZERO
  524 CONTINUE
  525 CONTINUE
      WRITE(9) (ACC(N),VEL(N),DISP(N),N=1,NN)
C     IF (ISTART .GT. 3) GO TO 600
C     IF(T.LT.TEND1.OR.T.GT.TEND2)GOTO600
C     K=(NB-1)*NN-1
C     DO 650 N=1,NN,2
C     K=K+2
C     K1=K+1
C     N1=N+1
C     PRINT 4004,K,ACC(N),VEL(N),DISP(N),K1,ACC(N1),VEL(N1),DISP(N1)
C     WRITE(7,4005) K,ACC(N),VEL(N),DISP(N)
C     WRITE(7,4005) K1,ACC(N1),VEL(N1),DISP(N1)
C 650 CONTINUE
  600 CONTINUE
      REWIND 9
      RETURN
 3025 FORMAT (5(1X,D23.16,2X))
 3221 FORMAT (5X,'BXB  '/,(5X,3D23.16,48X))
 3222 FORMAT (4(1X,'(',I3,',',I3,')',D23.16))
 3224 FORMAT (I10)
 3225 FORMAT (10X,'NB=',I2)
 3226 FORMAT (10X,'NP=',I4,'NK=',I4)
 3227 FORMAT (5(1X,D23.16,2X),1X)
 4004 FORMAT (2(5X,I3,3D12.4))
 4005 FORMAT (I4,3D25.16)
 4015 FORMAT (1X,'ACC(N),VEL(N),DISP(N) ON FILE (9) FOR TIME=', D16.7)
      END
```

```
      SUBROUTINE MODIFY (NEQ,N,U)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /BANARG/ A(28,14),FM(28),B(28),MBAND,NUMBLK,NBLK
      COMMON/DYNAM/AFM(28,14),BXB(28),BX(28),XB(28),AX(28),ACC(28),
     1VEL(28),DISP(28),FSTRT(28),FORCE(28),DT ,NODE(15)
      COMMON/SETNUM/ZERO,HALF,ONE,TWO,THREE,FOUR ,SIX,PI
      DO 250 M=2,MBAND
      K=N-M+1
      IF(K) 235,235,230
  230 BXB(K)=BXB(K)-AFM(K,M)*U
      AFM(K,M)= ZERO
  235 K=N+M-1
      IF(NEQ-K) 250,240,240
  240 BXB(K)= BXB(K)-AFM(N,M)*U
      AFM(N,M)= ZERO
  250 CONTINUE
      AFM(N,1)= ONE
      BXB(N)= U
      RETURN
      END


//GO.SYSIN DD *
    1
   48
//GO.FT01F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=7196,LRECL=116,RECFM=VBS)
//GO.FT02F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,1),
// DCB=(LRECL=8,BLKSIZE=1092,RECFM=VBS),
//     DSN=OSU.ACT10188.NONZERON
//GO.FT03F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,1),
// DCB=(LRECL=112,BLKSIZE=1684,RECFM=VBS),
//     DSN=OSU.ACT10188.MASSMATX
//GO.FT04F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(OLD,KEEP),SPACE=(TRK,5),
// DCB=(LRECL=1800,BLKSIZE=2298,RECFM=VBS),
//     DSN=OSU.ACT10188.TOTLSTIF
//GO.FT08F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=7144,LRECL=340,RECFM=VBS)
//GO.FT09F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(NEW,KEEP),SPACE=(TRK,2),
// DCB=(LRECL=340,BLKSIZE=7144,RECFM=VBS),
//     DSN=OSU.ACT12387.ACVELDIS
//GO.FT11F001  DD  UNIT=SYSDA,SPACE=(TRK,(10,10)),DISP=NEW,
// DCB=(BLKSIZE=7196,LRECL=116,RECFM=VBS)
//GO.FT12F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(NEW,KEEP),SPACE=(TRK,5),
// DCB=(LRECL=116,BLKSIZE=7196,RECFM=VBS),
//     DSN=OSU.ACT12387.BFMIJK
//GO.FT13F001  DD  UNIT=2314,VOL=SER=DISK06,
// DISP=(NEW,KEEP),SPACE=(TRK,11),
// DCB=(LRECL=36,BLKSIZE=1092,RECFM=VBS),
//     DSN=OSU.ACT12387.TYMACDIS
//
```

VITA

Brij Raj Kishore

Candidate for the Degree of

Doctor of Philosophy

Thesis: CHARACTERISTICS OF FINITE ELEMENTS FOR ANALYSIS OF BEAMS SUBJECTED TO IMPACT LOAD

Major Field: Civil Engineering

Biographical:

Personal Data: Born in Kheri-Lakhimpur, India, July 8, 1938, the son of Mr. and Mrs. Mritunjai Bats.

Education: Graduated from D. A. V. High School, Lucknow, India, in 1953; received the I. Sc. from Lucknow Christian College, Lucknow, India, in 1955; received the Bachelor of Science, Part I, from Lucknow University, Lucknow, India, in 1956; received the Bachelor of Architecture degree from the University of Roorkee, Roorkee, India, in 1961; received the Master of Architectural Engineering degree from Oklahoma State University, Stillwater, Oklahoma, in 1967; completed the requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1973.

Professional Experience: Graduate teaching assistant, School of Architecture, Oklahoma State University, 1962; Structural Designer for Cronheim and Weger, Architects and Engineers, Philadelphia, Pennsylvania, 1963-1967; graduate teaching assistant, School of Civil Engineering, Oklahoma State University, 1967-1972; Structural Engineer for U. S. Army, Corps of Engineers, Chicago, Illinois, since 1972.

2