ELASTIC BUCKLING OF ARCHES BY FINITE ELEMENT METHOD

By

Jose G. Lange

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Civil and Sanitary Engineering

1980

ABSTRACT

ELASTIC BUCKLING OF ARCHES
BY FINITE ELEMENT METHOD

BY

Jose G. Lange

A procedure for the computation of the elastic buckling load of arches is presented. The arch is represented by beam finite elements curved in one plane but deformable in three dimensional space. The curved axis of the element is represented by a fourth-order polynomial. The displacement functions are approximated by cubic polynomials. The expressions for the generalized strains include the linear and quadratic terms of the displacements. By using these functions the expression for the strain energy of an element is derived. This expression consists of three parts: the quadratic, cubic, and quartic terms. Proper differentiation of these expressions yields the linear stiffness matrix (K) and the incremental stiffness matrices (N1 and N2) of the element.

Assuming that the system is elastic and conservative, the equilibrium equation is obtained from the first variation of the potential energy. This represents a set of nonlinear algebraic equations. The equation

governing the linear incremental behavior is obtained from the second variation of the potential energy. A basis for obtaining the critical load of a structural system is the vanishing of the load increment vector corresponding to a change in the displacement vector. To avoid dealing with nonlinear equations, an estimate of the buckling load is obtained by assuming that the displacement increase linearly with the applied load until buckling occurs. This leads to a quadratic eigenvalue problem for the buckling loads and their associated buckling modes. Assuming that at buckling the displacements are sufficiently small the quadratic eigenproblem reduces to a linear one.

The quadratic eigenproblem is solved by the determinant search method in conjunction with the modified regula falsi iteration technique. Inverse vector iteration is used for the solution of the linear problem.

Eigenvalue problems are also formulated for the case of tilted loads (for example, due to the horizontal rigidity of the deck of an arch bridge). In addition, the buckling problem involving interactions between horizontal transverse loading and vertical in-plane loading is formulated.

A computer program was prepared for the implementation of the linear equilibrium solution and the buckling load solutions. Numerical results were obtained involving arch ribs with in-plane and out-of-plane behavior. The

influence of the number of elements on the accuracy of the results was investigated by considering both linear equilibrium problems and buckling problems. The types of buckling problems considered are: in-plane, out-of-plane, tilted loads, and the effect of out-of-plane horizontal loads on the in-plane buckling load. Good agreement was indicated by comparisons of the first three types of problems with existing analytical solutions based on the classical buckling theory. Results for the last type of problems indicated that while a small out-of-plane horizontal load may have little effect on the in-plane buckling load, the latter decreases rapidly with increases in the horizontal load.

TO MARIA MARGARITA AND ANDRES JOSE

ACKNOWLEDGMENTS

The writer wishes to express his appreciation to his major professor, Dr. Robert K. Wen, Professor of Civil Engineering, for his guidance and numerous helpful suggestions during the conducting of the research and preparation of this dissertation. Thanks also to members of the writer's doctoral committee:
Dr. William A. Bradley, Professor of Metallurgy, Mechanics and Materials Science, Dr. George E. Mase, Professor of Metallurgy, Mechanics and Materials Science, Dr. Harvey Davis, Professor of Mathematics. The writer also owes his appreciation to Ms. Ann Greenfield for her dedication in the typing of this dissertation. Special appreciation is also due his wife Maria Margarita, and son Andres Jose, who make it all worthwhile.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 GENERAL

The purpose of this thesis is to develop a procedure for the computation of the elastic buckling load of arches in space. In order to achieve this a three-dimensional beam element curved in one plane that takes into account the nonlinear effects of geometry changes was formulated. A computer program has been prepared to implement the analysis of arches under different loading conditions, and the results of certain numerical problems are presented.

This chapter describes the objective and scope of the present work, a literature review of related studies, and the general notation needed in the subsequent analysis.

1.2 OBJECTIVE AND SCOPE

Many engineering structures have components that may be considered as curved beams. Examples are the ribs of arch bridges and arch roofs, stiffening rings in aircraft and naval vessels, and horizontally curved highway bridges. When such elements are subjected to considerable compression such as in the case of arch bridges, their stability becomes a major consideration.

Although a considerable amount of work has been done (see literature review), there are a number of significant problems that have not been solved. Most of the previous works have dealt with the stability problem in the plane of the curved element. Past studies that had considered out of plane buckling have been either limited to circular or parabolic arches or made with the assumption that the curved element may be represented by a series of straight beam elements. The ouf-of-plane behavior of a truly curved element has not been studied.

The objective of the present study is to develop a three-dimensional nonlinear curved beam finite element. The numerical model is applied to the study of the buckling of arch ribs.

Figure 1-1 illustrates the general deformation behavior of a symmetrical arch under a symmetrical loading. It is seen that the buckled shape may be symmetric or antisymmetric. The curve "OC" represents the "exact" response which may be obtained by the solution of the (nonlinear) equilibrium equations of the system. It is called the "fundamental path." Depending upon the properties of the arch and loading, a point (e.g., point A) of "bifurcation" may occur before the peak point C or after it (i.e., point $\bar{A}$). Immediately beyond a bifurcation point on the fundamental path, the structure is unstable; so the behavior would follow the bifurcated path AB or $\bar{A}\bar{B}$. If the bifurcation point occurs before C, the buckling shape would be antisymmetrical (sometimes called "sidesway," generally occurring for "deep arches"). If the bifurcation point occurs at $\bar{A}$, the arch would have buckled at C in a symmetric buckling form ("snap-through," generally for "shallow arches").

The preceding represents the arch behavior described by an "exact" nonlinear analysis. The classical buckling theory would assume that up to the point when buckling takes place, the structure would maintain its original undeformed shape (point A'). At buckling, it goes into an adjacent equilibrium configuration (point B') which is unspecified in magnitude. The buckling load thus computed is called the "classical buckling load."

The approach followed in this investigation is that outlined by Mallett and Marcal (27)*. The theory is essentially different from either that which follows from the nonlinear equilibrium behavior or the classical theory of stability. The system is assumed to be elastic. The strain energy is written in terms of displacement variables. Geometrically nonlinear effects are considered by including the quadratic terms of displacements in the expressions for

---

* Number in parantheses refer to entries in the list of references.

the generalized strains. Thus, the strain energy may be written as the sum of one part containing quadratic terms, one containing cubic terms, and one quartic terms. The first variation of the potential energy (assuming that loads are conservative, i.e., their direction do not change with structural displacements) produces the equilibrium equation. The latter equilibrium equation may be transformed into an eigenvalue problem by assuming that the displacements increase linearly with the load parameter that controls the magnitude of the applied loads, and at the buckling load the linear incremental stiffness (tangent stiffness) vanishes. This formulation yields a quadratic eigenvalue problem; the lowest eigenvalue corresponds to the lowest buckling load.

For the finite element developed in this study, the curved shape of the element is represented by a fourth-order polynomial. This representation of the geometry can maintain continuity of position, slope, and curvature at two adjacent elements. The shape functions describing the displacements along each of the three coordinate axes and the twist along the longitudinal axis of the element are each expressed as cubic polynomials. This requires the introduction of eight degrees of freedom at each end. The linear and nonlinear stiffness matrices require numerical integration over the curved domain, for which the Gauss quadrature method was used.

The quadratic eigenvalue problem was established and solved by the determinant search method (5) in conjunction with the modified regula-falsi iteration technique (9). The effect of the quadratic term on the lowest eigenvalue was found to be small and it seems reasonable to drop that term. Hence, the problem reduces to that of a linear eigenvalue. It was solved by using the inverse iteration method (5).

A computer program was prepared to implement the linear equilibrium solution and the buckling load solution. Numerical results included systems with in-plane and out-of-plane behavior. In addition to some linear equilibrium problems solved to indicate the reliability of the model, the following types of buckling problems have been considered: (a) in-plane, (b) out-of-plane, (c) "tilted load," and (d) the effect of out-of-plane loads on the in-plane buckling load. For the first three types, comparisons were made with existing analytical solutions based on the classical theory (except in two cases of tilted loading for which no analytical solutions are available). The agreements are in general good. Results of type (d) indicated while a small out-of-plane horizontal load may have little effect on the in-plane buckling load, the latter decreases rapidly with increases in the horizontal load. As far as is known to the writer, this behavior has not been studied previously.

1.3 LITERATURE REVIEW

Ashwell and Sabir (2) discussed the use and limitations of several types of shape functions for finite elements for circular arches. Three types were considered: (a) polynomial expressions for the radial and circumferential displacements, (b) polynomial expressions modified to include certain trigonometric functions so as to make the circumferential strain and the change in curvature equal to zero when the displacements correspond to a rigid body motion, (c) expressions corresponding to axial strain and linear curvature. The latter two admit true rigid body displacement representations. They showed that the type (c) functions are only slightly inferior to type (b) when used for shallow arches. They also indicated that the performance of the shape functions would depend on the geometry of the arch. Shape functions good for shallow arches may not be good for deep arches and vice versa. This work was limited to actions of circular arches in a plane.

Dawe (13) pointed out the advantage of using higher-order polynomials for shape functions. His comparison included models using polynomials of orders quintic-quintic, cubic-quintic, quintic-cubic , cubic-cubic to represent the tangential and normal components of the displacements, and also a constant strain, linear curvature model for the shape functions.

Mebane and Stricklin (29) pointed out that rigid body motion could be considered to be implicitly included in the polynomial form of the shape functions as the number of elements used to represent the structure increases.

The preceding studies had been undertaken actually to investigate the optimal approach of using the finite element method for axi-symmetrical shells. The latter, like the arch deformed in its own plane, is a two-dimensional problem. For linear equilibrium problems of an element curved in one plane, the stiffness coefficients of the "in-plane" degrees of freedom (i.e., axial displacement, in-plane transverse displacement, and in-plane rotation) are uncoupled from the "out-of-plane" degrees of freedom. For circular elements the stiffness coefficients for in-plane degrees of freedom are well known (see e.g., Reference 37). For circular elements subjected to out-of-plane loading exact stiffness coefficients for sections with negligible warping have been reported by Lee (26). Stiffness matrices including warping had been reported by El-Amin and Brotton (16), and by Thornton and Master (40). The former employed the finite element method using cubic polynomials for the displacement functions and the stiffness coefficients were given explicitly. The latter presented expressions from which "exact" stiffness matrices may be computed from the inversion of certain component matrices.

Chauduri and Shore (8) developed a thin-walled curved beam element that also included warping effects and established a consistent mass matrix for the element.

The area of classical buckling analysis of curved structures has been investigated by several researchers. Austin (3) summarized the state of the knowledge of the in-plane bending and buckling of arches. His presentation was concerned with the available experimental and analytical data and their relations to design applications. Austin and Ross (4) compared the solution of the in-plane elastic buckling of arches between the classical buckling theory and the exact, nonlinear, buckling load analysis. They found that, except for buckling in the symmetric mode (snap-through), the buckling load obtained with the classical theory is very close to that obtained with the exact theory.

I. Ojalvo and Newman (31) reported a basic theoretical work on the elastic stability of a curved beam in space. The governing differential equations of a curved element in space were derived and a solution procedure akin to the "shooting method" was outlined. The shooting method is one that solves a boundary value problem as an initial value problem (30). If a curved beam is analyzed, the solution would be carried out from one end of the beam with the known boundary conditions plus certain assumed boundary conditions there as the "initial conditions." The solution proceeds toward the other end where, in general, it would not agree with the prescribed boundary conditions. Modifications would then be made of the boundary conditions assumed for the "initial end" such that the conditions at the "final end" would be met.

M. Ojalvo, Demuts, and Tokarz (32) followed the preceding works to study the out-of-plane buckling of a member curved in one plane. The theory was also applied by Shukla and M. Ojalvo (38) to calculate the buckling loads when they may be tilted such as those which would result from a horizontally rigid deck of an arch bridge. Extensive numerical data were presented covering ranges of parameters involving the ratio of the torsional stiffness to the out-of-plane bending stiffness and the ratio of the rise to the span length. Laboratory results that corroborated the theoretical data were presented by Tokarz (41). As a particular case of the equations presented in Reference (32), Tokarz and Sandhu (42) developed linear differential equations that define the lateral-torsional buckling of a parabolic arch subjected to a uniformly distributed load. They also made a comparative study with those results obtained experimentally by Tokarz (41).

Godden (20) studied the buckling load of a tied arch by use of the Rayleigh-Ritz method. The tilted hangers (on account of the assumed horizontal rigidity of the deck) were replaced by a continuous membrane along the arch. He verified the solution with experimental results. In a subsequent work Donald and Godden (14) reported a numerical procedure of the shooting method type for the analysis of curved beams (replaced by straight chords between panel points) subjected to loads normal to the plane of the structure.

1.4 NOTATION

The notation shown below has been used in this report:

A	=	area of cross-sections;
A, B	=	end nodes of an element;
b_2, b_3, b_4	=	element geometry coefficients (Eq. 2-12);
E	=	Young's modulus of elasticity;
G	=	shear modulus;
H	=	rise of the arch;
HD	=	difference in elevation between crown of arch and deck;
Ixx, Iyy	=	moment of inertia of cross-section (Fig. 2-2);
I_ξ, I_η	=	moment of inertia of cross-section (Fig. 2-2);
$[K]$	=	structural linear stiffness matrix;
$[K_m]$	=	modified structural linear stiffness matrix;
$[k]$	=	element linear stiffness matrix;
K_t	=	torsion constant of cross-section;

Kx,Ky,Kz	=	changes in curvature about x, y, z axes;
kx,ky,kz	=	current curvatures about x,y,z axes;
$\bar{k}$x,$\bar{k}$y,$\bar{k}$z	=	initial curvatures about x,y,z axes;
L	=	curved length of element;
L	=	span or arch;
[N1]	=	first order structural incremental stiffness matrix;
[N2]	=	second order structural incremental stiffness matrix;
[n1]	=	first order element incremental stiffness matrix;
[n2]	=	second order element incremental stiffness matrix;
{P}	=	vector of applied loads;
$\{P_H\}$	=	vector of applied horizontal loads;
$\bar{p}$, $\{\bar{P}\}$	=	a given load, a given load vector (Eq. 3-2);
p_{cr}	=	critical value of applied loads;
p_o, $\{P_o\}$	=	reference load, reference load vector (Eq. 3-2);
{q}	=	generalized coordinates, displacement vector;
$\{q_H\}$	=	displacement vector due to applied horizontal (out-of-plane) load;
$\{q_V\}$	=	displacement vector due to applied vertical (in-plane)load;
$\{\bar{q}\}$	=	reference displacement vector (Eq. 3-1);

$\{q_o\}$	=	reference displacement vector (Eq. 3-2);
R	=	radius of curvature
R1, R2	=	radii of curvature at ends of an element;
s	=	longitudinal axis of curved beam member;
[T]	=	diagonal transformation matrix (Eq. 3-14);
$[T_o]$	=	diagonal transformation matrix (Eq. 3-16);
u,v,w	=	displacements along x,y,z axes, respectively;
U	=	strain energy of an element;
U_ε	=	strain energy due to longitudinal strain;
U_t	=	strain energy due to torsion;
U_2, U_3, U_4	=	quadratic, cubic, and quartic parts of strain energy;
X,Y,Z	=	structure global coordinate system;
X_L, Y_L	=	relative position of end nodes of an element (Eq. 2-10);
x,y,z	=	element coordinate system;
(x_B, y_B)	=	coordinates of node B in element coordinate system;
α	=	angle of opening of circular arch;
β	=	twist of cross-section about z-axis;
$\beta x, \beta y, \beta z$	=	rotations about x,y,z axes, respectively;
γ	=	normalized variable (Eq. 2-19);
ε	=	longitudinal strain;

Θ	=	angle of tangent at node B (Fig. 2-5);
Θ_x	=	rotation about x-axis;
Θ_y	=	rotation about y-axis;
λ	=	buckling load parameter;
Δ	=	incremental operator;
$\{\ \}$	=	column vector;
$\lfloor\ \rfloor$	=	row vector;
$[\ ]$	=	rectangular matrix.

CHAPTER II

FINITE ELEMENT MODEL FOR A CURVED BEAM

2.1 GENERAL

In this chapter the displacement-strain relation and the expression of the strain energy of a curved beam are first presented. Next, the geometric representation and the displacement functions of the element are described. The strain energy expression of a typical curved element is developed. Finally, the general equations that govern the equilibrium and the linear incremental behavior of a structure are derived.

2.2 DISPLACEMENT-STRAIN RELATION

Consider a beam element curved in one plane as shown in Figure 2-1. The centroidal axis curves in the x-z plane with radius of curvature R (which may vary). The x, y, and z axes form a right-handed coordinate system with corresponding displacements u, v, and w as indicated in Figure 2-2. The cross-section of the element is taken to be constant. Assuming that plane sections remain plane after bending deformation, the expression for the longitudinal strain at a section s, measured along the curved centroidal axis, may be written as

$$\varepsilon)_{s,\xi,\eta} = \varepsilon_z)_o + \eta K_x - \xi K_y \qquad (2\text{-}1)$$

in which $\varepsilon_z)_o$ is the longitudinal strain and K_x and K_y are the changes in curvature of the centroidal axis.

For the general case of a beam curved in space, the changes in curvature have been derived by I. Ojalvo and Newman (31) as follows:

$$K_x = k_x - \bar{k}_x = \bar{k}_y \beta_z - \bar{k}_z \beta_y + \frac{d\beta_x}{d_s}$$

$$K_y = k_y - \bar{k}_y = - \bar{k}_x \beta_z + \bar{k}_z \beta_x + \frac{d\beta_y}{ds} \qquad (2\text{-}2)$$

$$K_z = k_z - \bar{k}_z = \bar{k}_x \beta_y - \bar{k}_y \beta_x + \frac{d\beta_z}{d_s}$$

in which k_x and $\bar{k}_x$ are the current and initial curvatures about the x-axis, respectively, similarly for k_y, $\bar{k}_y$ and k_z, $\bar{k}_z$; and β_x, β_y, β_z are the rotations about the x, y, z axes, respectively. The latter rotations are given by:

$$\beta_x = -u\ \bar{k}_z - \frac{dv}{ds} + W\ \bar{k}_x = - \frac{dv}{ds}$$

$$\beta_y = \frac{du}{ds} - v\ \bar{k}_z + \bar{k}_y\ W = \frac{du}{ds} + \frac{W}{R} \qquad (2\text{-}3)$$

$$\beta_z = \beta$$

in which β is the twist of the cross-section about the z-axis (note that $\bar{k}_x = \bar{k}_z = 0$). Substituting Equations

(2-3) into Equations (2-2),

$$K_x = \frac{\beta}{R} - \frac{d^2 v}{ds^2}$$

$$K_y = \frac{d^2 u}{ds^2} + \frac{dw}{ds}\frac{1}{R} + w\,\frac{d}{ds}\left(\frac{1}{R}\right) \qquad (2\text{-}4)$$

$$K_z = \frac{1}{R}\frac{dv}{ds} + \frac{d\beta}{ds}$$

The longitudinal strain at the centroidal axis may be written as:

$$\left.\varepsilon_z\right)_0 = \left(\frac{dw}{ds} - \frac{u}{R}\right) + \frac{1}{2}\left(\frac{du}{ds} + \frac{w}{R}\right)^2 + \frac{1}{2}\left(\frac{dv}{ds}\right)^2 \qquad (2\text{-}5)$$

in which the terms in the first parenthesis are the usual linear hoop strain for a curved element and the next two terms (which are nonlinear) represent the contribution to the strain by the rotations of the centroidal axis about the y- and x-axis, respectively. Substituting Equations (2-4) and (2-5) into Equation (2-1) the expression for the longitudinal strain for any point in the cross-section is obtained, i.e.,

$$\begin{aligned}\left.\varepsilon_z\right)_{s,\xi,\eta} &= \left(\frac{dw}{ds} - \frac{u}{R}\right) + \frac{1}{2}\left(\frac{du}{ds} + \frac{w}{R}\right)^2 + \frac{1}{2}\left(\frac{dv}{ds}\right)^2 \\ &\quad + \eta\left(\frac{\beta}{R} - \frac{d^2 v}{ds^2}\right) \qquad (2\text{-}6) \\ &\quad - \xi\left[\frac{d^2 u}{ds^2} + \frac{1}{R}\frac{dw}{ds} + w\frac{d}{ds}\left(\frac{1}{R}\right)\right]\end{aligned}$$

in which, it may be noted again that u, v, w, and β are displacements of the centroidal axis of the beam.

2.3 STRAIN ENERGY EXPRESSION

The expression for the total strain energy of the system may be written as:

$$U = U_\varepsilon + U_t \qquad (2\text{-}7)$$

where U_ε is the strain energy due to the longitudinal strain and U_t is that due to torsion of the cross-section along the axis of the beam. They are given by the expressions

$$U_\varepsilon = \int_{vol} \frac{E\,\varepsilon^2}{2}\, dV = \int_s \int_A \frac{E\,\varepsilon^2}{2}\, dA\, ds$$
$$U_t = \int_s \frac{G\,K_t}{2} \left(\beta_s + \frac{1}{R} v_s\right)^2 ds \qquad (2\text{-}8)$$

in which ε is the longitudinal strain as expressed by Equation (2-6), A is the cross-sectional area, E and G are the Young's modulus and shear modulus, and K_t is the torsion constant of the cross-section. In the preceding equation, the common notation of using a subscript to represent a differentiation has been used, e.g., $\beta_s \equiv d\beta/ds$. This notation will also be used subsequently. The total strain energy becomes

$$U = \int_s \int_A \frac{E\,\varepsilon^2}{2}\, dA\, ds + \int_s \frac{G\,K_t}{2} \left(\beta_s + \frac{1}{R} v_s\right)^2 ds \qquad (2\text{-}9)$$

Equation (2-9) will be used in Section 2.4.4 to obtain the strain energy of the curved element, and by proper differentiation to obtain its stiffness matrices.

2.4 FINITE ELEMENT FORMULATION

In the previous section the strain energy expression for a general three-dimensional beam curved in a plane has been presented. In this section, the geometry and the strain energy of a finite element model of a curved beam will be developed.

2.4.1 DEFINITION OF COORDINATE SYSTEMS

Figure 2-3 illustrates an arch with a typical constituent curved finite element AB. Additional coordinates for the element are described in Figure 2-4. Two coordinate systems are used in the analysis:

1) Structure Global Coordinate System.

This system consists of a single set of cartesian axes with the origin located at the crown of the arch (Figure 2-3). The system is oriented with the X-axis horizontal, the Y-axis vertical, and the Z-axis perpendicular to the plane of curvature. The positions of the nodes of the total structure are expressed by means of this system.

2) Element Coordinate System.

This system is illustrated in Figure 2-4. It consists of one set of cartesian axes with its origin located at node A, with the x-axis in the radial direction, the y-axis normal to the plane of curvature, and the z-axis tangent to the curved centroidal axis forming an angle ϕ_A with the global X-axis. Node B denotes the end node of the element.

2.4.2 ELEMENT GEOMETRY

Referring again to Figure 2-4, the coordinates of the nodes A and B with respect to the structure global system are (X_A, Y_A) and (X_B, Y_B) respectively. Furthermore, their relative position is defined by $X_L = X_B - X_A$ and $Y_L = Y_B - Y_A$.

To represent the element geometry in the x and z coordinates the element is redrawn in Figure 2-5. The coordinates (x_B, z_B) of node B in the element coordinate system are given by:

$$
\begin{aligned}
x_B &= -X_L \sin\phi_A + Y_L \cos\phi_A \\
z_B &= X_L \cos\phi_A + Y_L \sin\phi_A
\end{aligned}
\tag{2-10}
$$

in which ϕ_A, X_L and Y_L are defined in Figure 2-4. The angle ϕ in Figure 2-5 varies from zero at node A to Θ at node B, s varies along the longitudinal axis, and the radii of curvature R at nodes A and B are respectively R1 and R2.

At any point along the curve the following relations hold

$$
\begin{aligned}
dz &= ds \cos \phi \\
dx &= ds \sin \phi
\end{aligned}
\tag{2-11}
$$

The curve will be approximated by a fourth-order polynomial

$$S = b_0 + b_1 \phi + b_2 \phi^2 + b_3 \phi^3 + b_4 \phi^4 \tag{2-12}$$

the boundary conditions needed to solve for the coefficients b_i are (see Figure 2-5):

(1) at $\phi = 0$, $S = 0$

(2) at $\phi = 0$, $R = R1$ (2-13a)

(3) at $\phi = \Theta$, $S = L$

where L is the curved length of the element

(4) at $\phi = \Theta$, $R = R2$

(5) $x_B = \int_0^{\Theta} dx = \int_0^{\Theta} \sin\phi \, R \, d\phi$

(2-13b)

(6) $z_B = \int_0^{\Theta} dz = \int_0^{\Theta} \cos\phi \, R \, d\phi$

From (1) it is found that $b_0 = 0$. The curvature is obtained by differentiating Equation (2-12):

$$R = \frac{dS}{d\phi} = b_1 + 2\,b_2\phi + 3\,b_3\phi^2 + 4\,b_4\phi^3 \qquad (2\text{-}14)$$

From condition (2), $b_1 = R1$. When condition (3) is applied to Equation 2-12, the expression defining the length of the element is obtained as

$$L = R1\Theta + b_2\Theta^2 + b_3\Theta^3 + b_4\Theta^4 \qquad (2\text{-}15)$$

Condition (4) yields:

$$R2 = R1 + 2b_2\Theta + 3b_3\Theta^2 + 4b_4\Theta^3 \qquad (2\text{-}16)$$

Assuming that condition (3) is automatically satisfied if conditions (5) and (6) are, and that the equation defining the curve is known (which implies that R1 and R2 are prescribed) a system of three linear equations for three unknowns (b_2, b_3, b_4) can be established, i. e.,

$$R1 + 2b_2\,\Theta + 3b_3\Theta^2 + 4b_4\Theta^3 = R2$$

$$\begin{aligned} &R1\,(1 - \cos\Theta) + 2b_2\,(\sin\Theta - \Theta\cos\Theta) \\ &+ 3b_3\,(-\Theta^2\cos\Theta + 2\Theta\sin\Theta + 2\cos\Theta - 2) \\ &+ 4b_4(-\Theta^3\cos\Theta + 3\Theta^2\sin\Theta + 6\,\Theta\cos\Theta - 6\sin\Theta) \\ &\quad = x_B \end{aligned} \qquad (2\text{-}17)$$

$$\begin{aligned} &R1\sin\Theta + 2b_2\,(\Theta\sin\Theta + \cos\Theta - 1) \\ &+ 3b_3\,(\Theta^2\sin\Theta + 2\,\Theta\cos\Theta - 2\sin\Theta) \\ &+ 4b_4\,(\Theta^3\sin\Theta + 3\Theta^2\cos\Theta - 6\,\Theta\sin\Theta - 6\cos\Theta + 6) \\ &\quad = z_B \end{aligned}$$

Once the coefficients b_2, b_3, and b_4 are obtained from the solution of Equations (2-17) the geometry of the finite element is completely defined by Equation (2-12).

It may be pointed out that the values of b_2, b_3, and b_4 seem to be somewhat sensitive to the solution procedure used for Equations (2-17). A comparison of the solution obtained by several procedures is given in Appendix A. It is also shown that indeed condition (3), i.e., the length of the element, is satisfied by the procedure used.

2.4.3 DISPLACEMENT FUNCTIONS

As indicated in Equations (2-6) and (2-9), the strain energy of the curved element considered here depends on four independent displacement functions, i. e., u, v, w, and β, the displacements along the x, y, z axes and the rotation about the z-axis, respectively. For the finite element, these functions will be approximated by cubic polynomials in the variable ϕ (Figure 2-5)

$$
\begin{aligned}
u &= \alpha_1 + \alpha_2\phi + \alpha_3\phi^2 + \alpha_4\phi^3 \\
v &= \alpha_5 + \alpha_6\phi + \alpha_7\phi^2 + \alpha_8\phi^3 \\
w &= \alpha_9 + \alpha_{10}\phi + \alpha_{11}\phi^2 + \alpha_{12}\phi^3 \\
\beta &= \alpha_{13} + \alpha_{14}\phi + \alpha_{15}\phi^2 + \alpha_{16}\phi^3
\end{aligned} \tag{2-18}
$$

Note that ϕ is related to the arc length, s, by $ds/d\phi = R =$ radius of curvature. For simplicity, the independent variable ϕ in the preceding equations may be normalized by defining $\gamma = \phi/\Theta$, and the displacement functions become:

$$
\begin{aligned}
u &= \lambda_1 + \lambda_2\gamma + \lambda_3\gamma^2 + \lambda_4\gamma^3 \\
v &= \lambda_5 + \lambda_6\gamma + \lambda_7\gamma^2 + \lambda_8\gamma^3 \\
w &= \lambda_9 + \lambda_{10}\gamma + \lambda_{11}\gamma^2 + \lambda_{12}\gamma^3 \\
\beta &= \lambda_{13} + \lambda_{14}\gamma + \lambda_{15}\gamma^2 + \lambda_{16}\gamma^3
\end{aligned} \tag{2-19}
$$

2.4.4 ELEMENT STRAIN ENERGY

In terms of the new variable γ the longitudinal strain may be rewritten from Equation (2-6) as:

$$
\begin{aligned}
\varepsilon = {} & \left(\frac{w_\gamma}{R\Theta} - \frac{u}{R}\right) + \frac{1}{2}\left(\frac{u_\gamma}{R\Theta} + \frac{u}{R}\right)^2 + \frac{1}{2}\left(\frac{v_\gamma}{R\Theta}\right)^2 \\
& + \eta\left[\frac{\beta}{R} - \frac{v_{\gamma\gamma}}{(R\Theta)^2} - v_\gamma\,\gamma_{ss}\right] \\
& - \xi\left[\frac{u_{\gamma\gamma}}{(R\Theta)^2} + u_\gamma\,\gamma_{ss} + \frac{w_\gamma}{R^2\Theta} + \frac{w}{R\Theta}\,\gamma_{s\gamma}\right]
\end{aligned} \tag{2-20}
$$

in which

$$\gamma_s = \frac{d\gamma}{ds} = \frac{d\phi}{ds}\frac{1}{\theta} = \frac{1}{R\theta} \tag{2-21}$$

$$\gamma_{ss} = -\frac{1}{R^3\theta^2} R_s \tag{2-22}$$

$$\gamma_{s\gamma} = \left(\frac{1}{R}\right)_\gamma = R\theta\ \gamma_{ss} \tag{2-23}$$

Using the same change of variables for Equations (2-8),

$$U_\varepsilon = \int_0^1 \int_A \frac{E\ \varepsilon^2}{2}\, dA\ R\theta d\gamma$$

$$U_t = \int_0^1 \frac{G\ K_t}{2} (\beta_\gamma\ \gamma_s + \frac{1}{R} v_\gamma\ \gamma_s)^2\ R\theta d\gamma \tag{2-24}$$

The expression for the strain energy of an element is now obtained by substituting Equations (2-20) and (2-24) into Equation (2-7)

$$\begin{aligned} U = \frac{E}{2}\int_0^1 \{ & [(w_\gamma\gamma_s)^2 + (\frac{u}{R})^2 + \frac{1}{4}(u_\gamma\gamma_s + \frac{w}{R})^4 + \frac{1}{4}(v_\gamma\gamma_s)^4]A \\ & + I_\xi\ (\frac{\beta}{R} - v_{\gamma\gamma}\gamma_s^2 - v_\gamma\gamma_{ss})^2 \\ & + I_\eta\ (u_{\gamma\gamma}\gamma_s^2 + u_\gamma\gamma_{ss} + \frac{1}{R} w_\gamma\gamma_s + w\gamma_{s_\gamma}\gamma_s)^2 \\ & + [-2w_\gamma\gamma_s\frac{u}{R} + w_\gamma\gamma_s\ (u_\gamma\gamma_s + \frac{w}{R})^2 \\ & + w_\gamma\gamma_s\ (v_\gamma\gamma_s)^2 - \frac{u}{R}(u_\gamma\gamma_s + \frac{w}{R})^2 \\ & - \frac{u}{R}(v_\gamma\gamma_s)^2 + \frac{1}{2}(u_\gamma\gamma_s + \frac{w}{R})^2 (v_\gamma\gamma_s)^2]A\}R\theta d\gamma \\ & + \frac{GK_t}{2}\int_0^1 (\beta_\gamma\gamma_s + \frac{1}{R} v_\gamma\gamma_s)^2\quad R\theta d\gamma \end{aligned} \tag{2-25}$$

Equation (2-25) may be divided into three parts in the form:

$$U = U_2 + U_3 + U_4 \tag{2-26}$$

in which U_2, U_3, and U_4 contain respectively the quadratic, cubic, and quartic terms of the total strain energy. In explicit form, after some simplifications, they are

$$U_2 = \frac{E}{2}\int_0^1 [\ \frac{A}{R\Theta}(w_\gamma - \Theta u)^2 + I_\xi \frac{\Theta}{R^3}(R\beta - \frac{v_{\gamma\gamma}}{\Theta^2} - v_\gamma \gamma_{ss} R^2)^2$$

$$+ \frac{I_\eta}{R^3\Theta^3}(u_{\gamma\gamma} + u_\gamma \gamma_{ss} R^2\Theta^2 + w_\gamma \Theta + w\gamma_{s\gamma} R\Theta^3)^2\ d\gamma$$

$$+ \frac{GK_t}{2}\int_0^1 \frac{1}{R^3\Theta}(R\beta_\gamma + v_\gamma)^2\ d_\gamma \tag{2-27a}$$

$$U_3 = \frac{EA}{2}\int_0^1 \frac{1}{R^2\Theta^2}(w_\gamma - \Theta_u)\ [(u_\gamma + \Theta w)^2 + v_\gamma^2]\ d\gamma \tag{2-27b}$$

$$U_4 = \frac{EA}{8}\int_0^1 \frac{1}{R^3\Theta^3}[(u_\gamma + \Theta w)^2 + v_\gamma^2]^2\ d\gamma \tag{2-27c}$$

Upon substituting the displacement functions given by Equations (2-19) into Equations (2-27), U_2, U_3, and U_4 become functions of the coefficients λ_i in the displacement functions. These coefficients will be replaced by certain degrees of freedom related to the displacement variables at the ends of the element. These degrees of freedom are chosen to be:

u_A, u_B = the radial displacement;

v_A, v_B = the transverse displacement;

w_A, w_B = the longitudinal displacement;

β_A, β_B = the twist about the longitudinal axis;

$\Theta_{y_A}, \Theta_{y_B} = \left(\frac{du}{ds} + \frac{w}{R}\right)_{A, \text{or } B}$ = the rotation about y-axis;

$\Theta_{x_A}, \Theta_{x_B} = \left(-\frac{dv}{ds}\right)_{A, \text{ or } B}$ = the rotation about x-axis;

$\left(\frac{dw}{ds}\right)_A, \left(\frac{dw}{ds}\right)_B$ = part of axial strain;

$\left(\frac{d\beta}{ds}\right)_A, \left(\frac{d\beta}{ds}\right)_B$ = rate of twist;

For subsequent analysis the above degrees of freedom will be represented by the generalized coordinates $\{q\}$:

$$\lfloor q_1 \; q_2 \ldots q_8 ; \; q_9 \ldots q_{16} \rfloor = \lfloor u_A \; v_A \; w_A \; \beta_A \; \Theta_{y_A} \; \Theta_{x_A} \left(\frac{dw}{ds}\right)_A \left(\frac{d\beta}{ds}\right)_A ; \; u_B \; v_B \; w_B \; \beta_B \; \Theta_{y_B} \; \Theta_{x_B} \left(\frac{dw}{ds}\right)_B \left(\frac{d\beta}{ds}\right)_B \rfloor \quad (2\text{-}28)$$

The relation between $\{q\}$ and $\{\lambda\}$ may be obtained by use of Equations (2-19) and (2-28). Thus, U_2, U_3, and U_4 can be expressed in terms of the element nodal degrees of freedom $\{q\}$:

$$\begin{aligned} U_2 &= \int_0^1 f_2(\{q\})\, d\gamma \\ U_3 &= \int_0^1 f_3(\{q\})\, d\gamma \\ U_4 &= \int_0^1 f_4(\{q\})\, d\gamma \end{aligned} \quad (2\text{-}29)$$

in which f_2, f_3, and f_4 are, respectively, quadratic, cubic, and quartic functions of the q's.

The linear element stiffness matrix [k] may be obtained as

$$[k] = [k_{ij}] = [\frac{\partial^2 U_2}{\partial q_i \partial q_j}] = [\int_0^1 \frac{\partial^2 f_2}{\partial q_i \partial q_j} d\gamma] \quad (2\text{-}30)$$

The "incremental stiffness matrices" [n1] and [n2] are defined to be:

$$[n1] = [(n1)_{ij}] = [\frac{\partial^2 U_3}{\partial q_i \partial q_j}] = [\int_0^1 \frac{\partial^2 f_3}{\partial q_i \partial q_j} d\gamma] \quad (2\text{-}31)$$

$$[n2] = [(n2)_{ij}] = [\frac{\partial^2 U_4}{\partial q_i \partial q_j}] = [\int_0^1 \frac{\partial^2 f_4}{\partial q_i \partial q_j} d\gamma] \quad (2\text{-}32)$$

The expressions for the integrands in terms of the q's in Equation (2-30), (2-31), and (2-32) are too lengthy to be presented here. They are, however, explicitly given in the subroutine NUMINT of the computer program in Appendix C. It may be noted in passing that the integrals themselves were evaluated numerically by Gauss quadrature. It should also be noted that the elements of the matrices [n1] and [n2] are respectively linear and quadratic functions of the displacements.

2.5 EQUILIBRIUM EQUATIONS

The element linear stiffness matrix, [k], and the incremental stiffness matrices [n1] and [n2] have been derived in the preceding section. The structural linear stiffness matrix, [K], and the incremental stiffness matrices [N1] and [N2] may be obtained by "assembling" or adding up the corresponding element matrices, provided that they are all considered to be of the "structure size" and refer to the same global degrees of freedom.

As mentioned previously, the formulation followed in this section is that described by Mallett and Marcal (27). Assuming that the system is elastic and conservative, the potential energy of the system is:

$$\Phi_p = U + V \tag{2-33}$$

in which U is the strain energy and V is the potential of the external loads. The total potential energy may be expressed as:

$$\begin{aligned}\Phi_p &= \int_{vol} \frac{\varepsilon^2 E}{2} \, d_{vol} + V \\ &= \lfloor q \rfloor \left[\frac{1}{2} [K] + \frac{1}{6} [N1] + \frac{1}{12} [N2]\right] \{q\} \\ &\quad - \lfloor q \rfloor \{P\}\end{aligned} \tag{2-34}$$

in which [N1] and [N2] appear on account of the nonlinear terms in the expression for the longitudinal strain (Equation (2-6)).

The first variation of the potential energy produces the equilibrium equation:

$$\left[[K] + \frac{1}{2} [N1] + \frac{1}{3} [N2]\right] \{q\} = \{P\} \tag{2-35}$$

This represents a set of nonlinear algebraic equations.

The equations governing the linear incremental behavior follow from the second variation of the potential energy and are given by:

$$([K] + [N1] + [N2])_{\{\bar{q}\}} \{\Delta q\} = \{\Delta P\} \qquad (2\text{-}36)$$

where $\{\bar{q}\}$ is the reference equilibrium position. This equation will be used in the following chapter to develop the eigenproblem model for the buckling analysis.

CHAPTER III

BUCKLING LOAD ANALYSIS

3.1 GENERAL

The governing matrix equation of the linear incremental equilibrium for a nonlinear elastic structure (Equation (2-36)) was introduced in Section 2.5. In this chapter, the calculation of the buckling load as a quadratic or linear eigenvalue problem will be discussed. In addition, the formulation required for the computation of the buckling loads of arch ribs subjected to "tilted loads," and the effect of horizontal loads on the vertical buckling load will be presented. The implementation of the solution procedures used will also be given.

3.2 FORMULATION OF EIGENVALUE PROBLEMS

A basis for obtaining the critical load of a structural system is the vanishing of $\{\Delta P\}$ in Equation (2-36). This leads to:

$$([K] + [N1] + [N2])_{\{\bar{q}\}} \{\Delta q\} = \{0\} \qquad (3\text{-}1)$$

Equation (3-1) may be written for each point on the "fundamental path" (Figure 1-2). That path, of course, can be determined only by a solution of the nonlinear equilibrium equation, i.e., Equation (2-35). For a given load vector $\{\bar{P}\}$ and displacement vector $\{\bar{q}\}$ on the fundamental path, if a nontrivial solution of $\{\Delta q\}$ is obtainable from Equation (3-1), the load $\{\bar{P}\}$ would be the "exact" buckling load (the bifurcation or snap-through load).

Of course, for a given $\{\bar{P}\}$, $\{\bar{q}\}$ can be found only by the solution of the nonlinear equilibrium equations. In order to avoid dealing with the nonlinear equations, however, an estimate of the buckling load may be obtained from Equation (3-1) by assuming that the displacement of the structure increases linearly with applied load until buckling occurs. Thus, letting $\{q_0\}$ be some reference displacement, (e.g., $\{q_0\} = [K]^{-1}\{P_0\}$, $\{P_0\}$ being some reference load = $p_0 \ \{P_0\}$), the incremental stiffness matrices may be computed as:

$$[N1(\{\bar{q}\})] = [N1(\{q_0\})] \left(\frac{\bar{p}}{p_0}\right) \tag{3-2a}$$

and

$$[N2(\{\bar{q}\})] = [N2(\{q_0\})] \left(\frac{\bar{p}}{p_0}\right)^2 \tag{3-2b}$$

where $\{\bar{q}\} = [K]^{-1} \ \bar{p} \ \{P_0\}$, and the squared term in Equation (3-2b) follows from the fact that the elements of [N2] are quadratic in the displacement variables. At buckling,

$$[N1(\{q\})] = [N1]_{\{q_0\}} \left(\frac{p_{cr}}{p_0}\right) \tag{3-3a}$$

and

$$[N2(\{q\})] = [N2]_{\{q_0\}} \left(\frac{p_{cr}}{p_0}\right)^2 \tag{3-3b}$$

Thus, Equation (3-1) may be written as

$$([K] + \lambda\ [N1] + \lambda^2\ [N2])_{\{q_0\}} \{\Delta q\} = \{0\} \qquad (3\text{-}4)$$

where $\lambda = p_{cr}/p_0$. This equation represents a quadratic eigenvalue problem for estimating the buckling loads and their associated buckling modes.

If it can be assumed that at buckling the displacements are sufficiently small, then the matrix [N2] may be neglected. Thus, Equation (3-4) reduces to a linear eigenvalue problem, i.e.,

$$([K] + \lambda [N1])_{\{q_0\}} \{\Delta q\} = \{0\} \qquad (3\text{-}5)$$

In the following sections the solution procedures for the quadratic and the linear eigenvalue problems will be described.

3.3 SOLUTION OF EIGENVALUE PROBLEMS

The determinant search method in conjunction with a modified regula falsi iteration technique is used to obtain a solution of the quadratic problem. For the linear problem the inverse vector iteration is used.

3.3.1 QUADRATIC EIGENVALUE PROBLEM

A solution of the quadratic eigenvalue problem (Equation (3-4)) may be obtained by finding a value of λ for which,

$$\det \left| [K] + \lambda\ [N1] + \lambda^2 [N2] \right|_{\{q_0\}} = 0 \qquad (3\text{-}6)$$

For the purpose of this study only the smallest value of λ is of interest as it corresponds to the lowest buckling load and higher buckling loads have no practical significance. Therefore, the solution is carried out by evaluating the left-hand side of Equation (3-6) with increasing values of λ, starting from zero with small increments as shown in Figure 3-1. It may be noted that det $(\lambda = 0) > 0$. Let λ_A be such that det $(\lambda = \lambda_A) > 0$ but det $(\lambda = \lambda_B = \lambda_A + \Delta\lambda) < 0$. Then the solution $\lambda = \bar{\lambda}$ must lie in the interval $[\lambda_A, \lambda_B]$.

A modified regula falsi iteration technique (9) was used to obtain a closer estimate of the root $\bar{\lambda}$. This technique is described in Appendix B. The computer implementation of the solution of the quadratic eigenvalue problem is given in subroutine NLEIGNP of the computer program contained in Appendix C.

3.3.2 LINEAR EIGENVALUE PROBLEM

As mentioned in Section 3.2 that the linear eigenvalue problem is obtained from the quadratic problem by neglecting the matrix [N2] from Equation (3-1). In this thesis the linear eigenvalue problem is solved by using the inverse vector iteration technique as described by Bathe and Wilson (5). The technique may be regarded as a mathematical formulation of the Stodola method (23) in structural mechanics.

From Equation (3-5) the basic relation for the inverse vector iteration is

$$A\ q = \lambda\ B\ q \qquad (3\text{-}7)$$

in which for simplicity the usual symbols [ ] and { }

used for matrices and vectors have been dropped, and $A = [K]$ and $B = -[N1]$. The method assumes that A is positive definite and B may be a diagonal matrix with or without zero diagonal terms or may be a banded matrix, as is the case in this report.

A technique suitable for computer implementation is as follows:

(i) Assume a trial vector x_1 for the first eigenvector q_1 and that $x_1^T B q_1 \neq 0$.

(ii) for i = 1, 2,..., evaluate

$$A\bar{x}_{i+1} = y_i$$

$$\bar{y}_{i+1} = B\,\bar{x}_{i+1}$$

$$\rho(\bar{x}_{i+1}) = \frac{\bar{x}_{i+1}^T\, y_i}{\bar{x}_{i+1}^T\, \bar{y}_{i+1}} \tag{3-8}$$

$$y_{i+1} = \frac{\bar{y}_{i+1}}{(\bar{x}_{i+1}^T\, \bar{y}_{i+1})^{1/2}}$$

in which ρ is the Rayleigh quotient.

(iii) The preceding iterative process is considered to have converged if

$$\frac{\rho(\bar{x}_{i+1}) - \rho(\bar{x}_i)}{\rho(\bar{x}_{i+1})} \leq \text{EPSI} \tag{3-9}$$

where EPSI should be 10^{-2s} or smaller when the answer is required to 2s-digit accuracy. If "n" is the last iteration, i.e., Equation (3-9) is satisfied for i = n, then the smallest eigenvalue will be taken to be:

$$\lambda_1 = \rho(\bar{x}_{n+1}) \tag{3-10}$$

and the corresponding eigenvector is

$$q_1 = \frac{\bar{x}_{n+1}}{(\bar{x}^T_{n+1}\ \bar{y}_{n+1})^{1/2}} \tag{3-11}$$

The computer implementation of the techniques presented above is described in the subroutine EIGENVL listed in Appendix C.

In the next two sections two special formulations of the buckling problem will be described, namely, buckling due to tilted loads, and the effect of horizontal loads on the vertical buckling load.

3.4 BUCKLING OF ARCHES DUE TO TILTED LOADS

In considering the out-of-plane buckling of the ribs of arch bridges, researchers (38, 41) were concerned with the effect of the rigidity of the bridge deck. If the deck is assumed to be perfectly rigid in the horizontal direction, the vertical load transmitted through the columns (or hangers) to the rib would be tilted as the arch undergoes a buckling displacement.

Typical deck, through, and half-through arches are illustrated in Figure 3-2. Figure 3-3a illustrates a section A-A of a deck arch bridge. The horizontal component of the tilted load is $P_H = P\ v/H$. This horizontal load will enter in the equilibrium equation (Equation (2-35)) corresponding to the horizontal degrees of freedom in the out-of-plane direction. In this case, the load vector should be modified to include these horizontal loads P_H.

The incremental form of the equilibrium equation is given by Equation (2-36) as

$$([K] + [N1(q)] + [N2(q)]) \{\Delta q\} = \{\Delta P\} \qquad (3\text{-}12)$$

in which $\{\Delta P\}$ represents the change in load during buckling. The change of the vertical loads are of a higher order of magnitude and may be neglected. Therefore, the right-hand side of Equation (3-12) contains only the change in horizontal loads:

$$\{\Delta P\} = \begin{Bmatrix} \vdots \\ 0 \\ \frac{P\Delta v}{H} \\ \vdots \end{Bmatrix} \qquad (3\text{-}13)$$

Now Δv is a component of the vector $\{\Delta q\}$; or, considering such loadings from all the columns, $\{\Delta v\}$ is a subset of $\{\Delta q\}$. Then Equation (3-13) can be written as

$$\{\Delta P\} = [T] \{q\} \qquad (3\text{-}14)$$

in which the transformation matrix [T] is a diagonal matrix with diagonal terms equal to zero except for those terms corresponding to the v-components of vector $\{\Delta q\}$, in which case the diagonal term is equal to P_i/H_i, where P_i is load in and H_i the height of the column concerned. Similar analyses for the through bridge (Figure 3-3b) and half-through bridge would yield the same equation as Equation (3-14) provided H_i is computed from $H_i = Y_i - HD$ in which HD is the difference in elevation between the crown and the deck and Y_i is the Y-coordinate of the node on the rib to which the ith column or hanger is connected.

Substituting Equation (3-14) into Equation (3-12) one obtains:

$$([K] + [N1(q)] + [N2(q)] + [T])\ \{\Delta q\} = \{0\} \qquad (3\text{-}15)$$

Introducing Equations (3-3) into Equation (3-15) a modified version of the quadratic eigenvalue problem is obtained:

$$([K] + \lambda\ ([N1] + [T_o]) + \lambda^2[N2])_{\{q_o\}}\{\Delta q\} = \{0\} \qquad (3\text{-}16)$$

in which $[T_o]$ corresponds to the loading $\{P\} = \{P_o\}$.

Equation (3-16) defines the quadratic eigenvalue problem when tilting loads are considered. If matrix [N2] is neglected, the linear eigenvalue problem is:

$$([K] + \lambda\ ([N1] + [T_o]))_{\{q_o\}}\{\Delta q\} = \{0\} \qquad (3\text{-}17)$$

3.5 EFFECT OF OUT-OF-PLANE LATERAL LOAD ON IN-PLANE BUCKLING LOAD OF ARCHES

A curved beam may be subjected to a combination of out-of-plane lateral load and in-plane load. For example, a rib of an arch bridge may be subjected to a horizontal wind load normal to the plane of the rib in addition to the in-plane gravity load.

The approach used herein to an analysis of this problem is as follows. The horizontal load $\{P_H\}$ is applied first. The corresponding displacement $\{q_H\}$ is obtained from:

$$[K]\ \{q_H\} = \{P_H\} \tag{3-18}$$

Next the vertical load is applied and additional displacement $\{q_V\}$ would result. The magnitude of the vertical load is gradually increased until buckling takes place. Noting that [N1] is linear in {q}, Equation (3-1) may be written as (dropping the [N2] term):

$$([K] + [N1(\{q\})])\{\Delta q\} = ([K] + [N1(\{q_H\})] + [N1(\{q_V\})])\{\Delta q\} = \{0\} \tag{3-19}$$

Writing as previously $N1(\{q_V\}) = \lambda\ N1(\{q_o\})$ (see Equation (3-3a)), the above equation may be written as

$$([K_m] + \lambda\ [N1(\{q_o\})])\ \{\Delta q\} = \{0\} \tag{3-20}$$

in which

$$[K_m] = [K] + [N1(\{q_H\})] \tag{3-21}$$

Thus the vertical buckling load can be calculated from Equation (3-20), the influence of the horizontal load being accounted for in $[K_m]$ as indicated by Equation (3-21).

3.6 COMPUTER PROGRAM

An outline of the program developed for this study is presented in this section; the program itself is given in Appendix C. The major steps in the program are described in the same order in which they are executed:

1) The basic information concerning the physical description of the arch is input. This information includes the number of elements, the type of arch, the type of load, and the type of eigenvalue problem to be solved. The global coordinates of the nodes are also input with the parameters defining the boundary conditions of the arch.

2) The element data is input. The input of element properties is general for prismatic members of any cross-section. These properties include the modulus of elasticity, shear modulus, density of the material, area of the cross-section, moments of inertia about the two principal axes and the torsion constant of the cross-section. The program has been prepared such that separate properties for each element may be used if this is desired (other types of elements may be included in the structural system by providing additional subroutines for their stiffnesses).

3) Next, the geometry of each curved element is defined, i.e., radius of curvature at each node, coordinates of the nodes in the element system. With these coordinates (x_B, z_B) the coefficients b_i are computed.

4) The applied loads are next input. Their orientation and type of loading (i.e., concentrated, uniformly distributed) was defined with the information input in 1.

5) From the information input in 1, the semibandwidth of the structure stiffness matrix is computed. The element linear stiffness matrices are computed and assembled into the linear stiffness matrix of the structure. This matrix is assembled in banded format and due to symmetry only the upper semibandwidth is constructed.

6) A linear analysis of the arch is performed to obtain displacements and forces due to the applied loads. The displacements so determined are used to compute, for each element, the matrices [n1] and [n2] which are assembled also in banded format into the structure incremental stiffness matrices [N1] and [N2].

7) From the information input in 1 the type of eigenvalue problem that is to be solved is defined.

8) The lowest eigenvalue and its corresponding eigenvector for the specified arch and loading conditions are computed and output.

CHAPTER IV

NUMERICAL RESULTS

4.1 GENERAL

In this chapter a number of numerical examples of arch behavior were considered using the computer program which embodies the solution methods developed in the previous chapters. Initially a comparison of the finite element solutions of linear equilibrium problems was made with analytical solutions to test the reliability of the element.

The program is then used to solve both in-plane and out-of-plane buckling problems for circular and parabolic arches under several loading conditions using formulations of both linear and quadratic eigenproblems. Next, the buckling loads of the deck type and through type parabolic arches subjected to tilted loads are calculated. Comparisons of the numerical solutions obtained with available analytical data are presented. The effect of an out-of-plane transverse loading on the in-plane buckling load is then considered.

It should be noted that in all the numerical solutions presented in this chapter the degrees of freedom dw/ds and $d\beta/ds$ have been included since the formulation of the buckling problem does not allow condensation of any degrees of freedom.

4.2 LINEAR EQUILIBRIUM PROBLEMS

Two types of problems were solved. They are linear equilibrium problems for a concentrated load at the crown applied in the plane of the arch or normal to that plane. The examples were used to verify the reliability of the element and to consider the effect of the number of elements on the accuracy of the solution.

4.2.1 CONCENTRATED IN-PLANE LOAD (VERTICAL) AT CROWN

The solution was obtained for two types of arches, circular and parabolic. In both cases the symmetry of the load and of the structure were used to reduce the number of equations. Figure 4-1 shows, for different numbers of elements, for a semi-circular arch the difference between the computed displacement at the crown and the analytical solution. Two sets of data are shown in the figure. They differ in the treatment of the degree of freedom dw/ds at the support. In one case this d.o.f. is restrained, i.e., set equal to zero. For the other it is free, i.e., a corresponding "equilibrium equation" is formally assembled and it would generally take on a value different from zero.

It is seen that the two sets of data are the same except for cases including small numbers of elements.

Since dw/ds enters in the expression for axial strain (Equation (2-6)) it seems logical to let it be a free degree of freedom. The subsequent results presented herein all correspond to this specification of the boundary condition. Table 4-1 shows the numerical data for this case.

The data indicated that the differences with the analytical solution decrease rapidly with increase in the number of elements. However, at larger number of elements (say, greater than 10) the difference increases (for example, to 2.5% at 18 elements). It was first thought that the reason might be the sensitivity of the geometry coefficients b_2, b_3, and b_4 (Section 2.4.2). For a circular arch these coefficients should be zero but numerical calculations would sometimes produce non-zero values. However, when these coefficients were set equal to zero in the program, no difference in the results was observed. Thus, it appears that the sensitivity of these coefficients was not the cause. Such behavior would seem to be the result of the round-off errors accumulated from the increasing amount of computation as the number of elements was increased.

It may be noted that the use of the semi-circular arch in this case may be considered as a stringent test on the convergence behavior because it is a "deep arch" (2).

For shallower arches, the convergence should be better, and this is illustrated in the following.

Figure 4-2 (see also Table 4-2) shows similar pattern of results for a parabolic arch. It is observed that the solutions converge very rapidly from 0.61% for 2 elements to 0.0041% for 10 elements. However, as mentioned before, the results for larger number of elements, say greater than 12, are not as good as those for lesser number of elements.

4.2.2 CONCENTRATED TRANSVERSE LOAD (HORIZONTAL) AT CROWN

For this case, the solution was obtained only for the semi-circular arch as described in Figure 4-1. Figure 4-3 shows, for different numbers of elements, the difference between the computed transverse displacement at the crown and the analytical solution. For 2 elements the difference is 2.5% but decreases rapidly as the number of elements is increased (0.37% for 10 elements). Table 4-3 contains the numerical values for this case.

It may be mentioned in passing that the linear stiffness matrix, for this case of a circular beam subjected to out-of-plane loads, had been compared with that given in Reference (16) and found to be essentially the same.

4.3 BUCKLING PROBLEMS

The computation of the buckling load was discussed in Chapter III. It was pointed out that the load may be computed from the solution of a quadratic eigenvalue problem, Equation (3-4). If the quadratic term that involves the [N2] matrix is neglected, the buckling load may be computed from a linear eigenvalue problem, Equation (3-5).

4.3.1 LINEAR VERSUS QUADRATIC EIGENPROBLEM SOLUTIONS

To study the importance of the matrix [N2] in the solution of the eigenvalue problem, three types of problems were considered: first, the in-plane buckling of a 90^{o} circular arch under a uniformly distributed radial load, second, the out-of-plane buckling of the same type of arch under the same loading, and third, the out-of-plane buckling of a parabolic arch subjected to a uniformly distributed vertical load. All three arches were simply supported. For the out-of-plane cases, the rotation degrees of freedom about the x-axis (Figure 2-3) at the supports were restrained.

Table 4-4 shows the values and the ratios of the critical load as computed from the linear eigenproblem to that from the quadratic eigenproblem. The number of elements was kept constant at 12. It is seen that in each case the ratio is very close to unity. This would indicate that the inclusion of the matrix [N2] in the eigenvalue problem would not give significantly different results from the case where it is neglected. Since much computer time can be saved by ignoring [N2], this has been done in obtaining the following results.

4.3.2 IN-PLANE BUCKLING

To illustrate the effect of the number of elements on the computed values of the in-plane buckling loads, the critical load of a circular arch under a uniformly distributed radial load was calculated for different numbers of elements. Figure 4-4 shows the results in terms of percentage differences with respect to the analytical solution given in Reference (39). Table 4-5 lists the numerical values. The buckling mode of all solutions was that of antisymmetry or sidesway (Figure 1-1b). It is seen from Figure 4-4 that the results "converge" rapidly. However, instead of converging to the analytical buckling load they converge to a value approximately 6% higher than the analytical value. This discrepancy is thought to be due to the inherent difference of the methods that produced the results. This point will be discussed again in Chapter V.

4.3.3 OUT-OF-PLANE BUCKLING

The effect of the number of elements on the out-of-plane buckling of a simply supported parabolic arch (with the rotation d.o.f. about x-axis at supports restrained) under a uniformly distributed vertical load was considered.

The critical loads in terms of percentage differences with respect to the analytical values given in Reference (24) are shown in Figure 4-5 (see also Table 4-6). In this case, the symmetry of both the geometry of the arch and loading were utilized to halve the number of degrees of freedom for the problem. This can be done because the buckling mode is symmetric.

Again, as the number of elements is increased, a fast convergence is seen from 9.04% for 2 elements to 2.42% for 10 elements, beyond which some oscillation of the results was encountered.

4.4 BUCKLING OF PARABOLIC ARCHES SUBJECTED TO TILTED LOADS

The analysis for the buckling load of arches under tilted loads was given in Section 3.4. The numerical data obtained for this study involved all three cases of arch ribs: deck, through, and half-through as illustrated in Figure 3-2. It may be seen that basically, in the case of buckling the columns and/or hangers rotate about the deck as the rib undergoes out-of-plane buckling.

Table 4-7 shows for deck (HD = 0) and through arches, the values and the ratios of the buckling loads obtained in this research to the analytical values published in Reference (38). It is seen that the agreement between the two sets of results are quite good. It should be noted that the formulation developed in Section 3.4 has an advantage in that, for the deck arch, the deck is not required to be tangent at the crown of a symmetric arch (HD = 0, Figure 3-3). For the case of HD $\neq$ 0 a lower critical load and a symmetric buckling mode (contrary to an antisymmetric mode for HD = 0) would be expected. This was verified by using one of the arches of Table 4-7 (type 1, 20 elements), but with HD = 3 in., for which the critical load was found to be only 54% of that when HD = 0.

In addition, the analysis presented in Section 3.4 is also applicable to half-through arch ribs for which no analytical data is available. The buckling load for such a rib (type 1, 20 elements, HD = 2.4 in.) has been calculated to be 325.81 lb/in. which, as expected, is in between the buckling loads of 151 lb/in. for the deck type (HD = 0) and 566.39 lb/in. for the through type.

4.5 EFFECT OF HORIZONTAL TRANSVERSE LOAD ON THE VERTICAL BUCKLING LOAD OF ARCHES

The analysis for this loading case has been given in Section 3.5. Numerical data are presented here for a parabolic and a circular arch. Several levels of uniformly distributed transverse loads were considered. In each case a given level of transverse load was applied first, and then a uniformly distributed vertical load was added and the corresponding critical value of that vertical load was determined.

Figure 4.6 shows these results for a parabolic arch. The same data is also shown in Table 4-8. Since the dimensions of the arch considered correspond to those of a realistic arch bridge, it is of some interest to express the transverse load also in terms of wind velocities in mph.

For low wind velocity the vertical buckling load is little affected. As the velocity increases the rate of reduction in the vertical buckling load rapidly increases. A point is reached where the wind load would be enough to make the arch buckle in tension (at about 150 lb/ft). Subsequent increases in the wind load will result in negative vertical buckling load (upward) to keep the out-of-plane symmetrical buckled configuration. It must be noted, however, in real arch bridges, the two ribs would be braced together and the response could be quite different from that of a single arch.

Figure 4-7 shows similar kind of behavior for a circular arch. Note again that increasing the transverse load rapidly decreases the vertical buckling load of the arch. Once again a point is reached (at about 0.1034 lb/in) at which buckling would be achieved with tension in the rib. As in the case of the parabolic arch, the buckling mode was out-of-plane and symmetric.

CHAPTER V

CONCLUSION

5.1 DISCUSSION

In the preceding chapter results obtained using the finite element developed for this study were compared with those of analytical solutions. The differences were of the order of 6% which should be acceptable, at least for engineering design purposes. However, the fact that the numerical results did not in all cases converge to the analytical solution should be considered.

It should be noted that for all comparisons the analytical solutions correspond to the classical theory of elastic stability (39), i.e., at the buckling load an equilibrium configuration exists adjacent to the original undeformed configuration of the structure. In this sense, the elastic deformation prior to buckling is neglected.

The numerical method used herein may be regarded as an offshoot from a nonlinear equilibrium analysis (27). That is, if one formulates the tangent stiffness of the nonlinear structure (which is a function of the elastic deformation) and makes the assumption that the displacements increase linearly with the load, an eigenproblem for the buckling load is obtained as described in Chapter III. This approach does not appear to be identical to the class-cal theory, and hence the results should not be expected to be exactly the same. Although the differences of the buckling loads calculated certainly are not excessive, in-asmuch as this method is an approximate one and is applied here for the first time, as far as is known to the writer, it should be used only with caution.

Of course, like in most cases of finite elements, the method developed here may be used for problems intrac-table by analytical methods. The present method has the advantage of being able to represent practically any curved shape with prescribed curvatures as well as slopes at the ends. In addition the [N1] and [N2] matrices can be used for the more exact nonlinear equilibrium analysis to deter-mine the fundamental path (Figure 1-2). Also worthy of note is that the method can be applied to a structural system (including more than one type of element) in as straightforward a manner as in the linear structural analy-sis so far the assembling of the different matrices are concerned.

A significant contribution of this work is the formulation of the tilted load problem as presented. It would produce buckling loads not only for the deck and through types arch ribs but the half-through type also.

Another feature of this study is the formulation of the problem of the interaction of horizontal and vertical loads on the stability of arches.

Extension of this study should include an attempt to gain an in-depth understanding of the role of the matrices [N1] and [N2] as used herein. Further applications may include the stability of curved structures under different types of loading, the stability of ribs in half-through arch bridges, the stability of bridge systems (in which the ribs are braced together). More in-depth study should be made of the interaction of out-of-plane lateral load and in-plane load on the stability of arches. Finally, the more exact nonlinear equilibrium behavior may be studied using the [N1] and [N2] matrices developed herein.

5.2 SUMMARY

In the preceding chapters, the development of a three dimensional beam element curved in a plane has been presented. The solution method has been embodied in a general computer program written in FORTRAN.

The analysis uses the finite element method. A nonlinear formulation of the displacement-strain relations has been employed. Numerical integration was utilized to obtain the stiffness matrices of the curved element. The geometrically nonlinear effects are accounted for by the matrices [N1] and [N2] determined from the cubic and quartic parts of the strain energy expression, respectively.

The linear equilibrium solution of circular and parabolic arches under a concentrated vertical or horizontal transverse load at the crown was carried out by using the Gauss elimination procedure. Convergence, with respect to the number of elements used, was indicated when the results obtained with the computer program were compared with the analytical results.

The applications of the computer program have also included the computation of the buckling load of arches subjected to different loading conditions. Increasing number of elements were used to obtain a better measure of the reliability of the element developed in this report and also to provide guidance for selecting the number of elements to be used for later applications. It appears that effective solutions may be obtained with approximately 10 elements.

Comparative studies between the quadratic and the linear eigenvalue problems were carried out. The solutions did not differ by more than 0.5% which indicated that the inclusion of the matrix [N2] in the eigenvalue problem would not give significantly different results from those obtained from the linear eigenvalue problem for which the matrix [N2] was neglected.

The out-of-plane buckling of parabolic arches under tilted loads was also considered. Comparisons with data published for deck and through arch types in Reference (38) indicated reasonably good agreement. While earlier results were limited to cases in which the deck was tangent to the crown, it was shown that the formulation developed in this investigation could be used to compute the buckling load for the more realistic case when the deck is not tangent to the crown (HD $\neq$ 0, Figure 3-3), and also for the half-through type arch ribs.

Results on the influence of a uniformly distributed transverse load (normal to plane of arch) on the vertical buckling load of a parabolic and a circular arch were presented. It was found that while a small transverse load has little effect on the vertical buckling load, the latter decreases rapidly as the transverse load increases.

TABLE 4-1 LINEAR EQUILIBRIUM OF CIRCULAR ARCH SUBJECTED TO CONCENTRATED IN-PLANE LOAD AT CROWN

NUMBER OF ELEMENTS	DISPLACEMENT AT CROWN* (in.)	DIFFERENCE** (%)
2	0.0137649	6.20
3	0.0135398	7.73
4	0.0143005	2.55
5	0.0143899	1.94
6	0.0146852	0.03
8	0.0146179	0.39
10	0.0146035	0.48
12	0.0149135	-1.63
15	0.0145392	0.91
18	0.0142818	2.67
20	0.0144516	1.52

* Analytical Solution = 0.0146740 in.

** % Difference = $\dfrac{\text{Analytical-Numerical}}{\text{Analytical}}$

TABLE 4-2 LINEAR EQUILIBRIUM OF PARABOLIC ARCH SUBJECTED TO CONCENTRATED IN-PLANE LOAD AT CROWN

NUMBER OF ELEMENTS	DISPLACEMENT AT CROWN* (in. x 10^{-4})	DIFFERENCE** (%)
2	0.464243	0.61
3	0.465878	0.26
4	0.466563	0.11
5	0.466852	0.049
6	0.466870	0.046
8	0.466928	0.034
10	0.467066	0.0041
12	0.467152	-0.014
16	0.469314	-0.48
20	0.464965	0.45

* Analytical Solution = 0.467085 x 10^{-4} in.

** $\% \text{ Difference} = \dfrac{\text{Analytical-Numerical}}{\text{Analytical}}$

TABLE 4-3 LINEAR EQUILIBRIUM OF CIRCULAR ARCH SUBJECTED TO CONCENTRATED OUT-OF-PLANE LOAD AT CROWN

NUMBER OF ELEMENTS	DISPLACEMENT AT CROWN* (in.)	DIFFERENCE** (%)
2	2.755249	2.50
3	2.783664	1.50
4	2.795598	1.07
5	2.801654	0.86
6	2.806454	0.69
8	2.811643	0.51
10	2.815347	0.37
12	2.812205	0.49
15	2.816225	0.34
16	2.824705	0.04
18	2.822199	0.13
20	2.820483	0.19

* Analytical Solution = 2.825930 in.

** $\% \text{ Difference} = \dfrac{\text{Analytical-Numerical}}{\text{Analytical}}$

TABLE 4-4 LINEAR VERSUS QUADRATIC EIGENPROBLEM SOLUTIONS

TYPE OF ARCH*	TYPE OF LOAD	A (in^2)	Ixx (in^4)	Iyy (in^4)	CRITICAL LOAD (lb/in.)		
					LINEAR	QUADRATIC	RATIO
CIRCULAR IN-PLANE	RADIAL	0.1875	0.9765×10^{-3}	0.8789×10^{-2}	51.41	51.64	0.99
CIRCULAR OUT-OF-PLANE	RADIAL	1.00	0.5208×10^{-2}	1.3333	27.44	27.44	1.00
PARABOLIC OUT-OF-PLANE	VERTICAL	1.50	1.7578×10^{-2}	2.00	189.61	189.67	0.99

*Circular Arches: $R = 30$ in, $\alpha = 90^{\circ}$, $E = 10^{7}$ psi
Parabolic Arch: $H = 9.6$ in, $L = 48$ in, $E = 29 \times 10^{6}$ psi

TABLE 4-5 IN-PLANE BUCKLING OF CIRCULAR ARCH SUBJECTED TO UNIFORMLY DISTRIBUTED RADIAL LOAD

NUMBER OF ELEMENTS	CRITICAL LOAD* (lb/in)	DIFFERENCE** (%)
3	41.55	-14.91
4	48.43	- 0.82
5	50.95	4.34
6	51.69	5.86
8	51.95	6.39
9	51.85	6.18
12	51.41	5.28
15	52.05	6.59
18	51.25	4.96
20	51.42	5.30

* Analytical Solution = 48.83 lb/in

** % Difference = $\frac{\text{Numerical-Analytical}}{\text{Analytical}}$

TABLE 4-6 OUT-OF-PLANE BUCKLING OF PARABOLIC ARCH SUBJECTED TO UNIFORMLY DISTRIBUTED VERTICAL LOAD

NUMBER OF ELEMENTS	CRITICAL LOAD* (lb/in)	DIFFERENCE** (%)
2	206.07	9.04
3	197.82	4.67
4	195.59	3.49
5	194.67	3.01
6	194.43	2.88
8	193.79	2.54
10	193.57	2.42
12	189.61	0.33
16	192.79	2.01
20	193.56	2.42

* Analytical Solution = 188.99 lb/in

** % Difference = $\frac{\text{Numerical-Analytical}}{\text{Analytical}}$

TABLE 4-7 BUCKLING OF PARABOLIC ARCHES SUBJECTED TO TILTED LOADS

NUMBER OF ELEMENTS	CROSS SECTION TYPE*	H (in)	L (in)	CRITICAL LOAD (lb/in.) NUMERICAL	ANALYTICAL	RATIO
DECK ARCH (HD = 0)						
16	1	9.6	48	155.32	138.28	1.12
20	1	9.6	48	151.00	138.28	1.09
16	2	28.8	48	2079	1967	1.05
20	2	28.8	48	2034	1967	1.03
THROUGH ARCH						
16	1	9.6	48	568.16	534.69	1.06
20	1	9.6	48	566.39	534.69	1.05
16	2	28.8	48	5725	6136	0.93
20	2	28.8	48	6002	6136	0.97

*Type 1: $A = 1.50$ in^2, $Ixx = 1.7578 \times 10^{-2}$ in^4, $Iyy = 2.00$ in^4, $K_t = 0.6616 \times 10^{-1}$ in^4
Type 2: $A = 2.2564$ in^2, $Ixx = 0.30$ in^4, $Iyy = 0.60$ in^4, $K_t = 0.3783$ in^4
$E = 29 \times 10^6$ psi

TABLE 4-8 EFFECT OF HORIZONTAL TRANSVERSE LOAD ON VERTICAL BUCKLING LOAD OF A PARABOLIC ARCH*

HORIZONTAL LOAD (lb/ft)	HORIZONTAL LOAD WIND (mph)	VERTICAL BUCKLING LOAD (lb/ft)
0	0	3957
20	18	3887
40	26	3678
60	31	3330
80	36	2480
100	40	2210
120	44	1436
140	48	518
160	51	-526

* See Figure 4-6

TABLE 4-9 EFFECT OF HORIZONTAL TRANSVERSE LOAD ON VERTICAL BUCKLING LOAD OF A CIRCULAR ARCH

HORIZONTAL LOAD (lb/in)	VERTICAL BUCKLING LOAD (lb/in)
0	5.44
0.0272	5.06
0.0544	3.91
0.0816	2.00
0.1088	-0.68

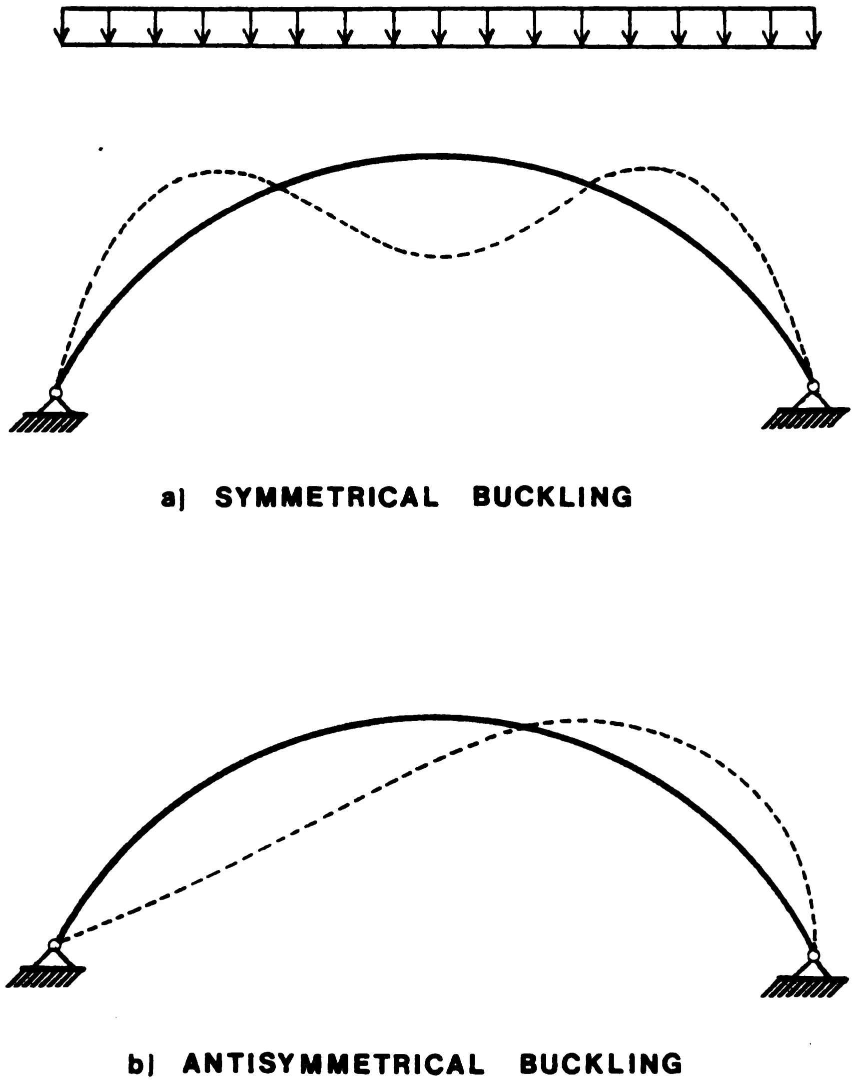


Figure 1-1. In-plane buckling under symmetrical loading.

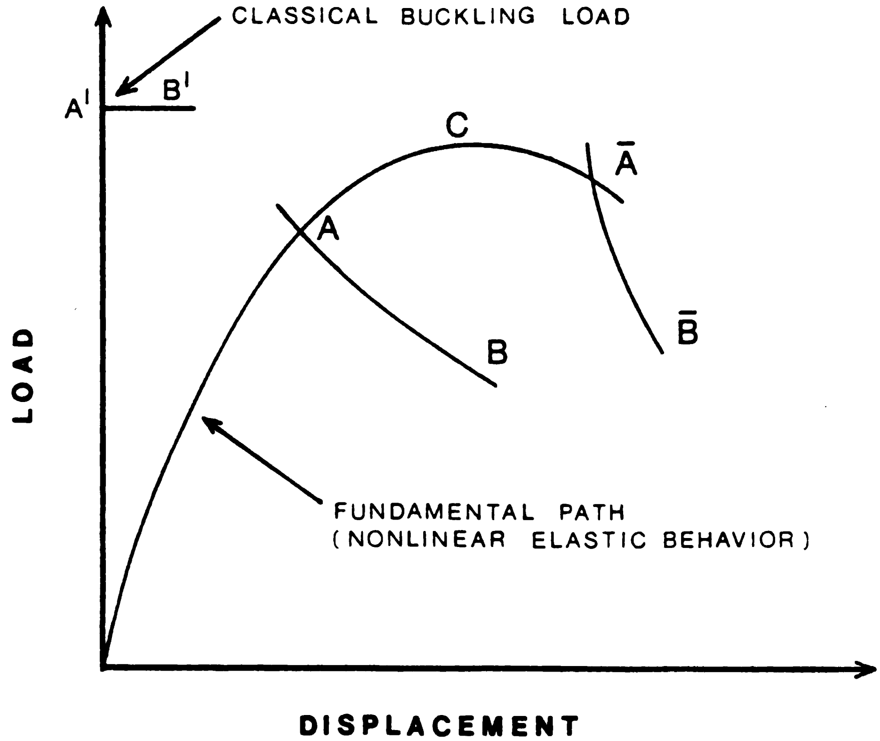


Figure 1-2. Load-deflection relation.

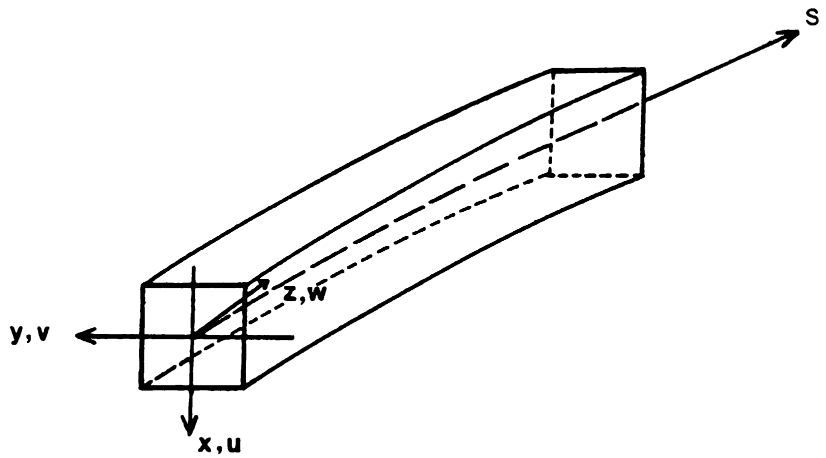


Figure 2-1. Beam element (curved in x-z plane).

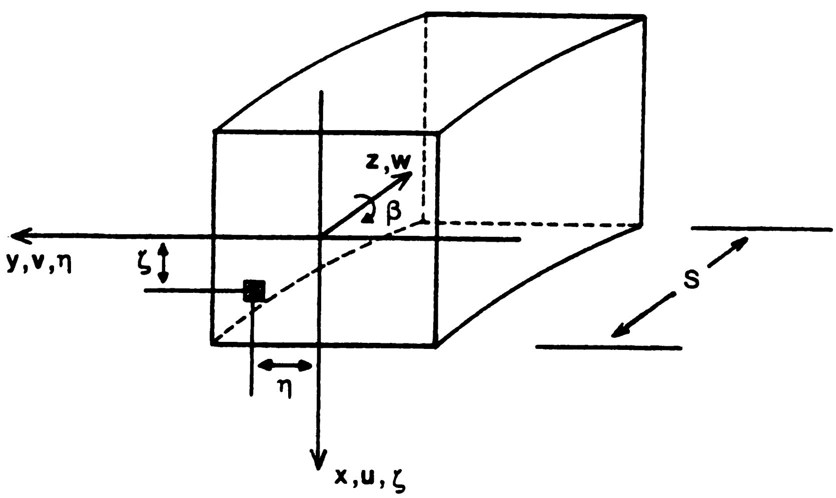


Figure 2-2. Cross-section of prismatic member.

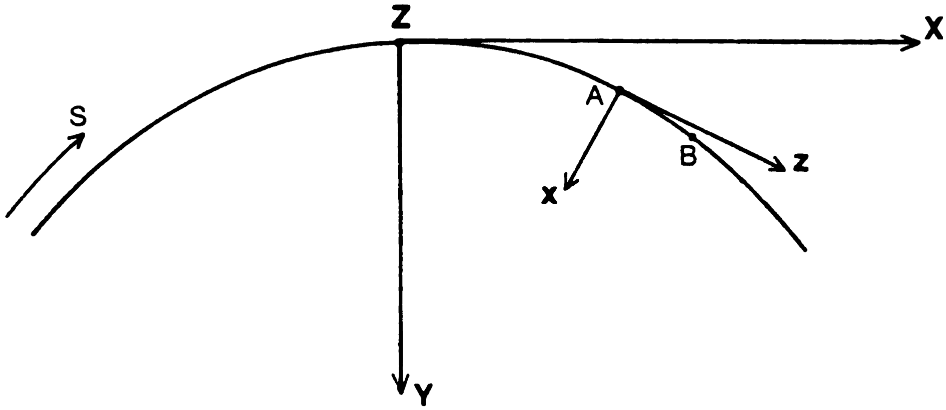


Figure 2-3. Coordinate systems.

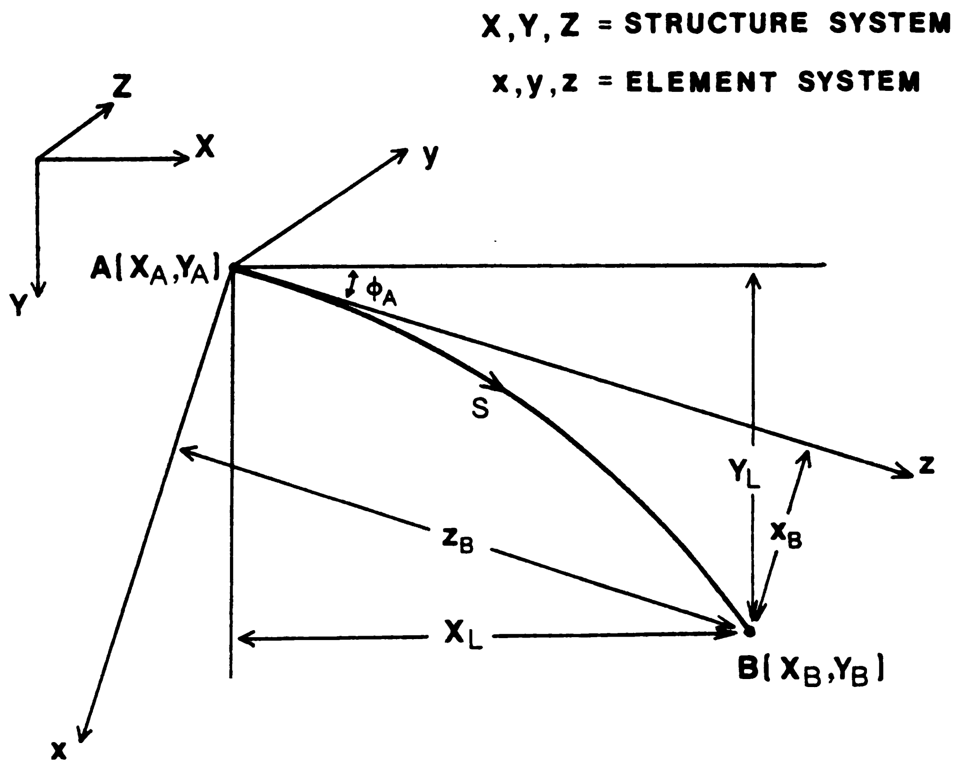


Figure 2-4. Typical element.

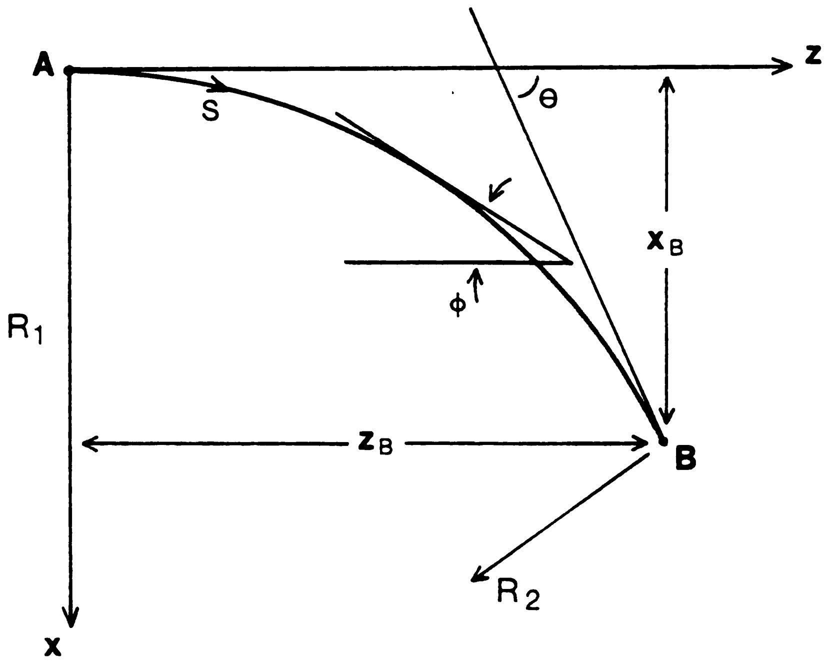


Figure 2-5. Typical element after transformation to element coordinate system.

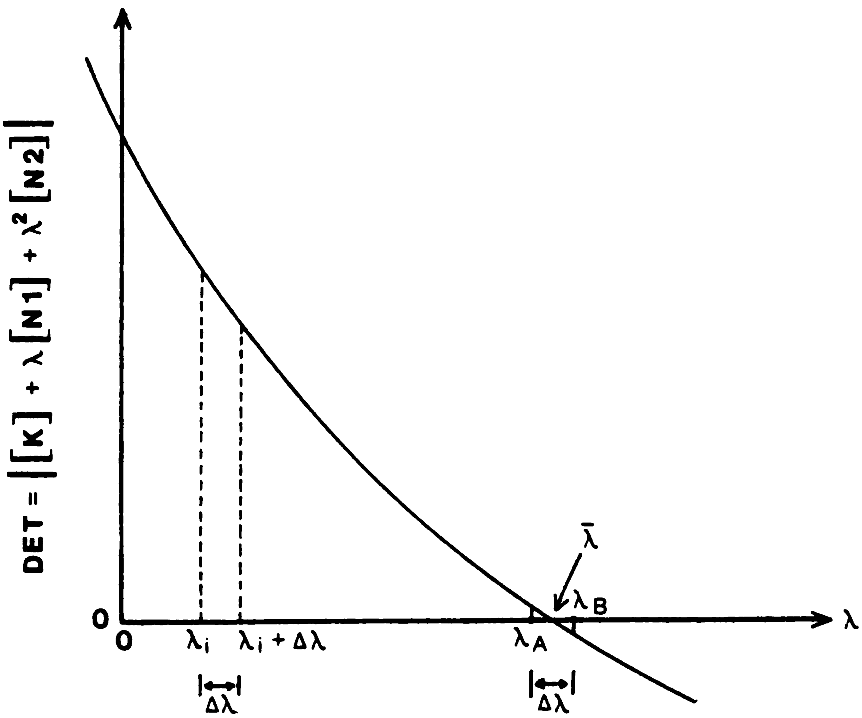
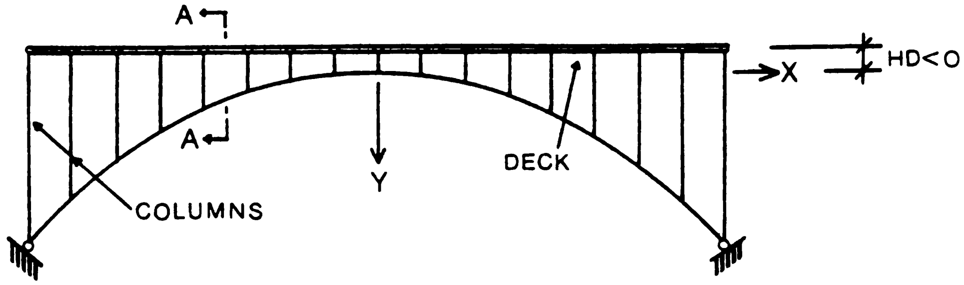
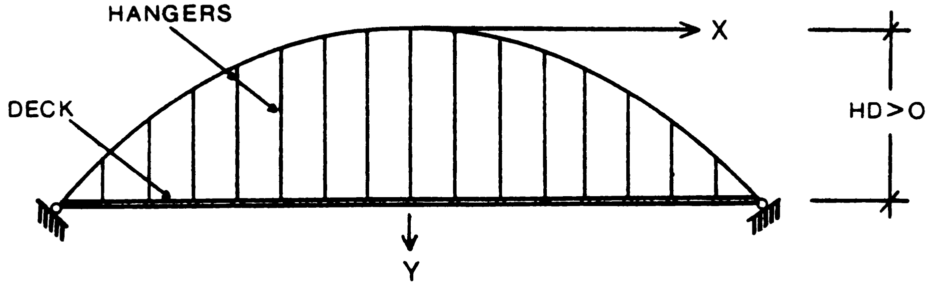


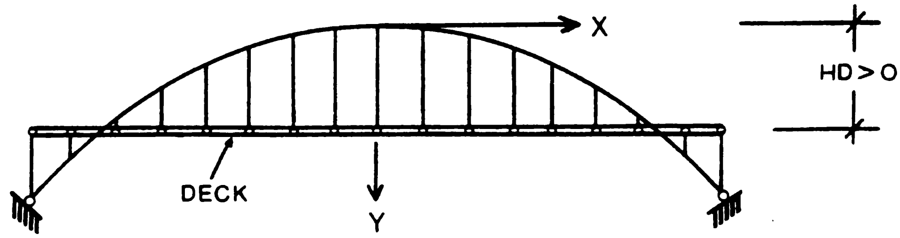
Figure 3-1. Determinant search method.



DECK ARCH BRIDGE



THROUGH ARCH BRIDGE



HALF-THROUGH ARCH BRIDGE

Figure 3-2. Typical arch bridges.

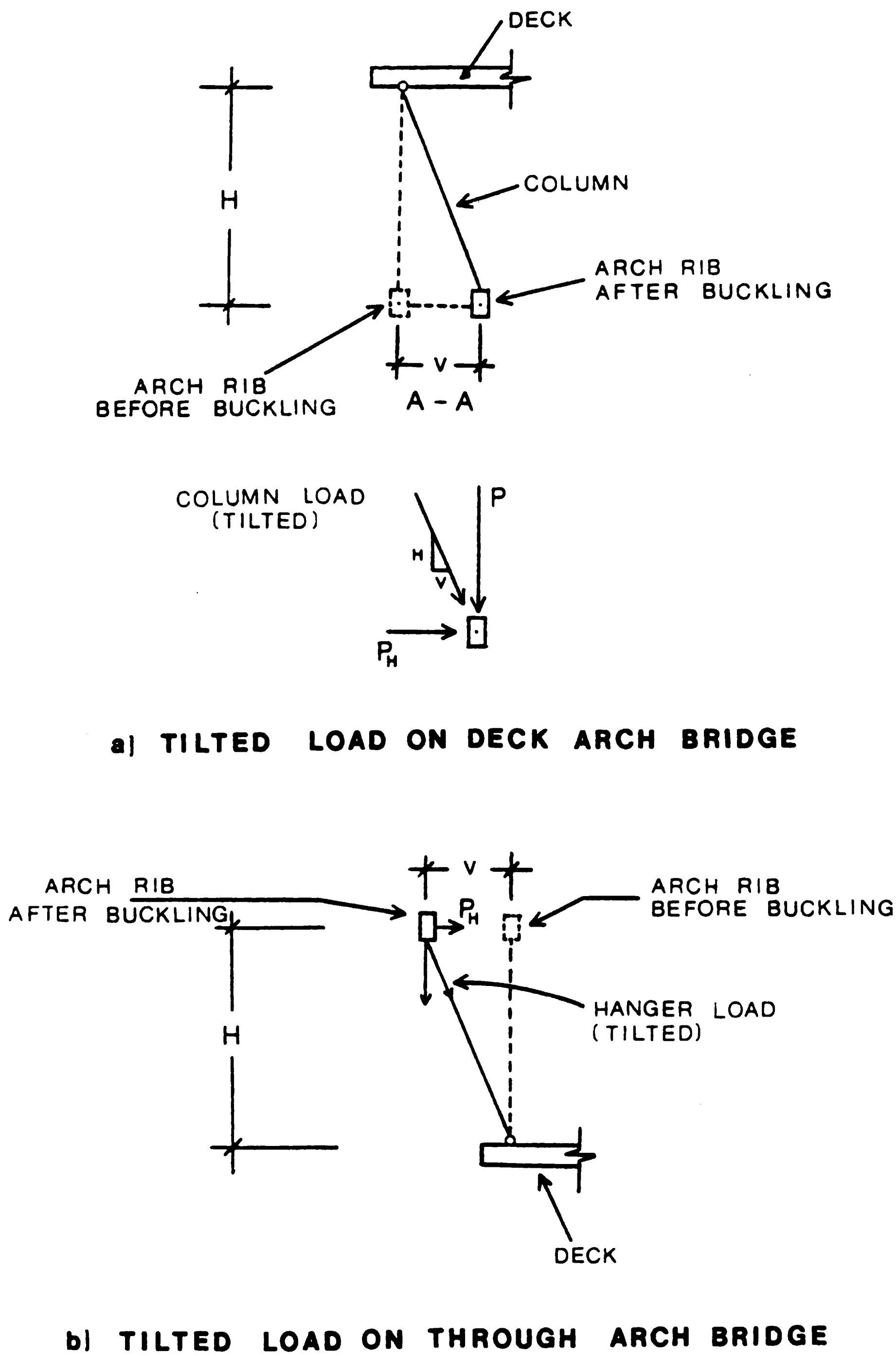


Figure 3-3. Tilted loads on ribs of deck and through bridges.

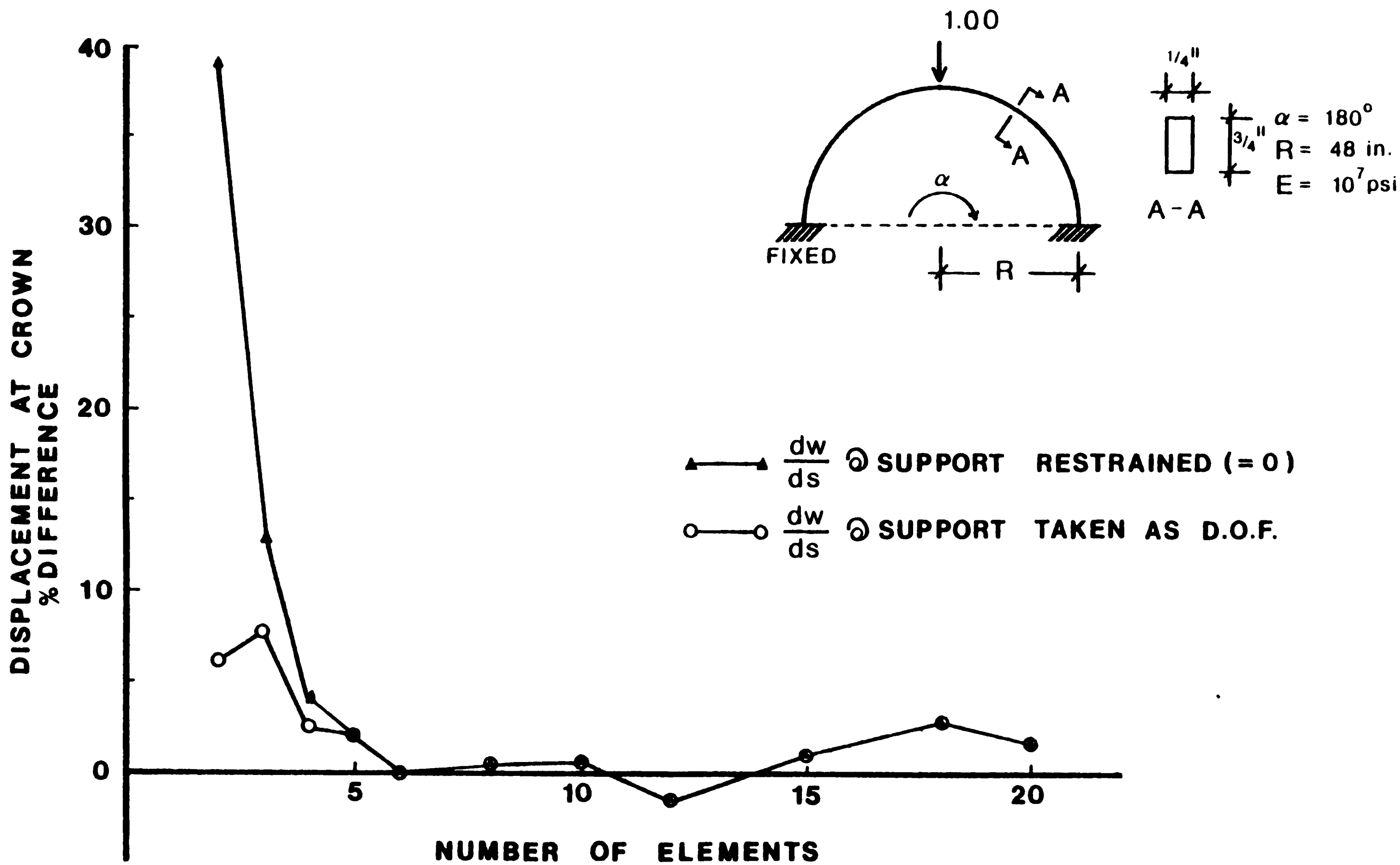


Figure 4-1. Linear equilibrium of circular arch subjected to concentrated in-plane load at crown.

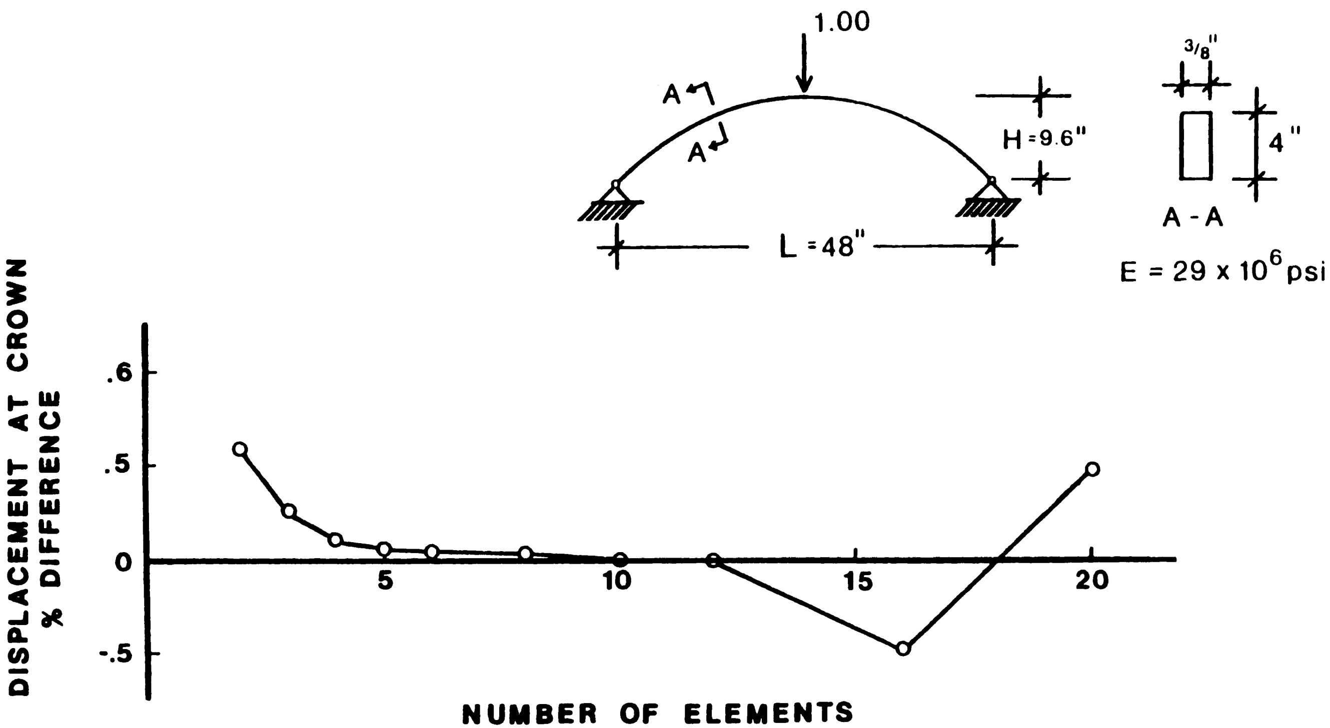


Figure 4-2. Linear equilibrium of parabolic arch subjected to concentrated in-plane load at crown.

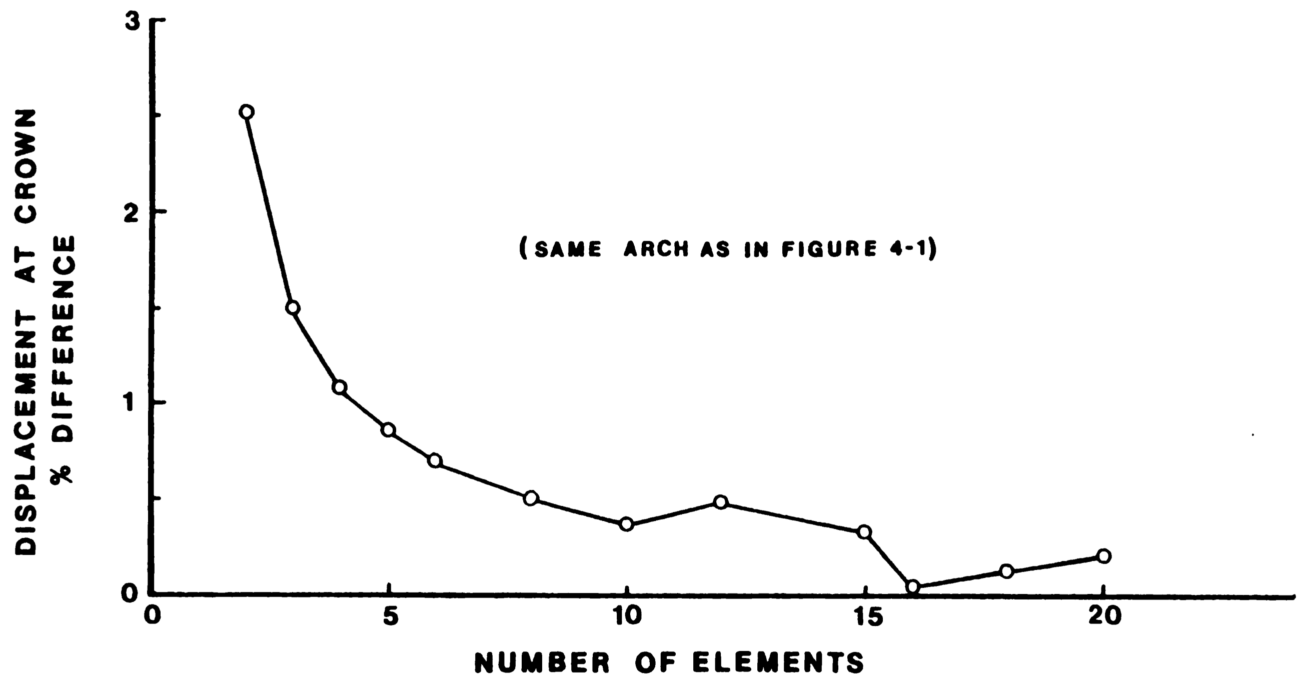


Figure 4-3. Linear equilibrium of circular arch subjected to concentrated out-of-plane load at crown.

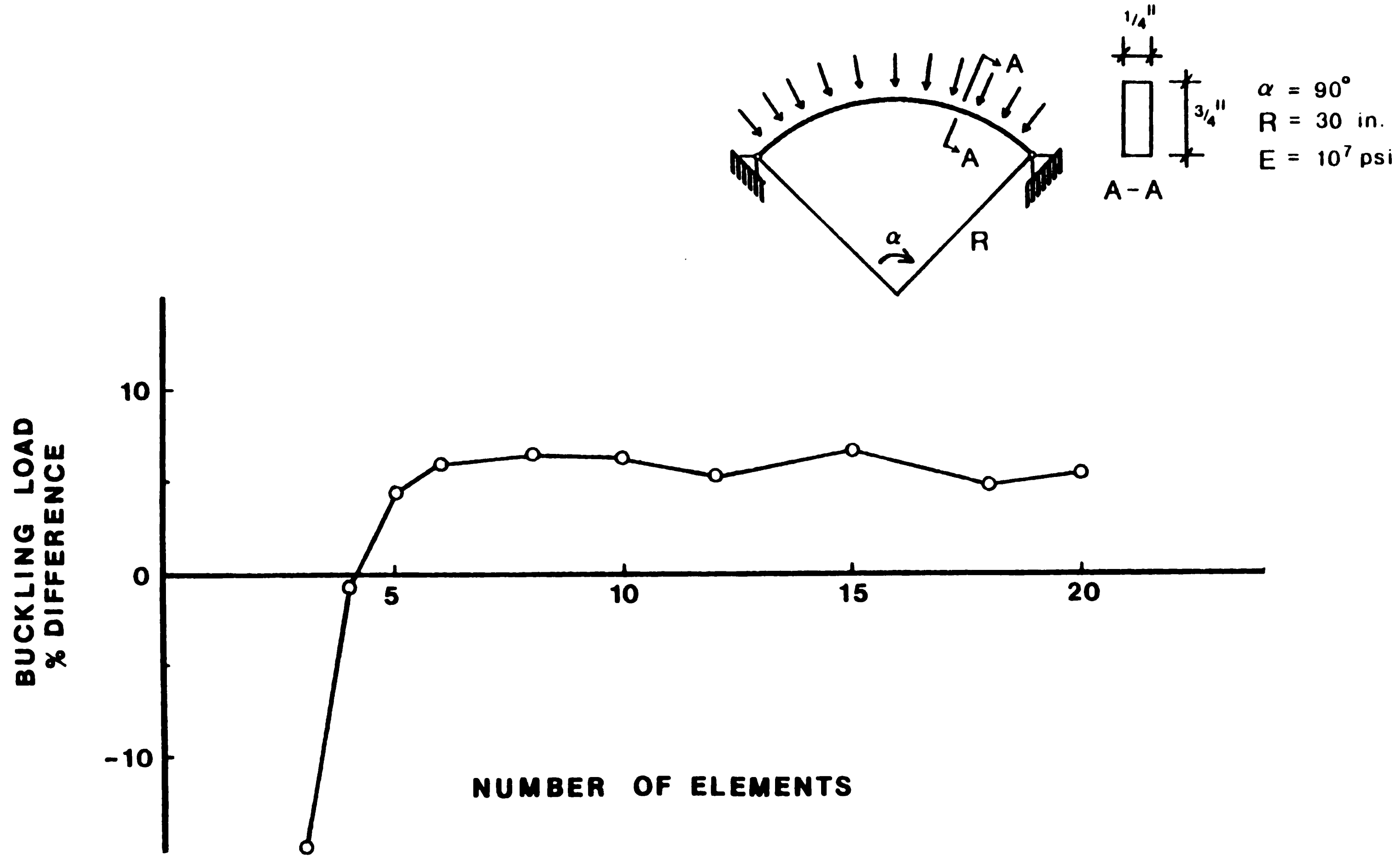


Figure 4-4. In-plane buckling of circular arch subjected to uniformly distributed radial load.

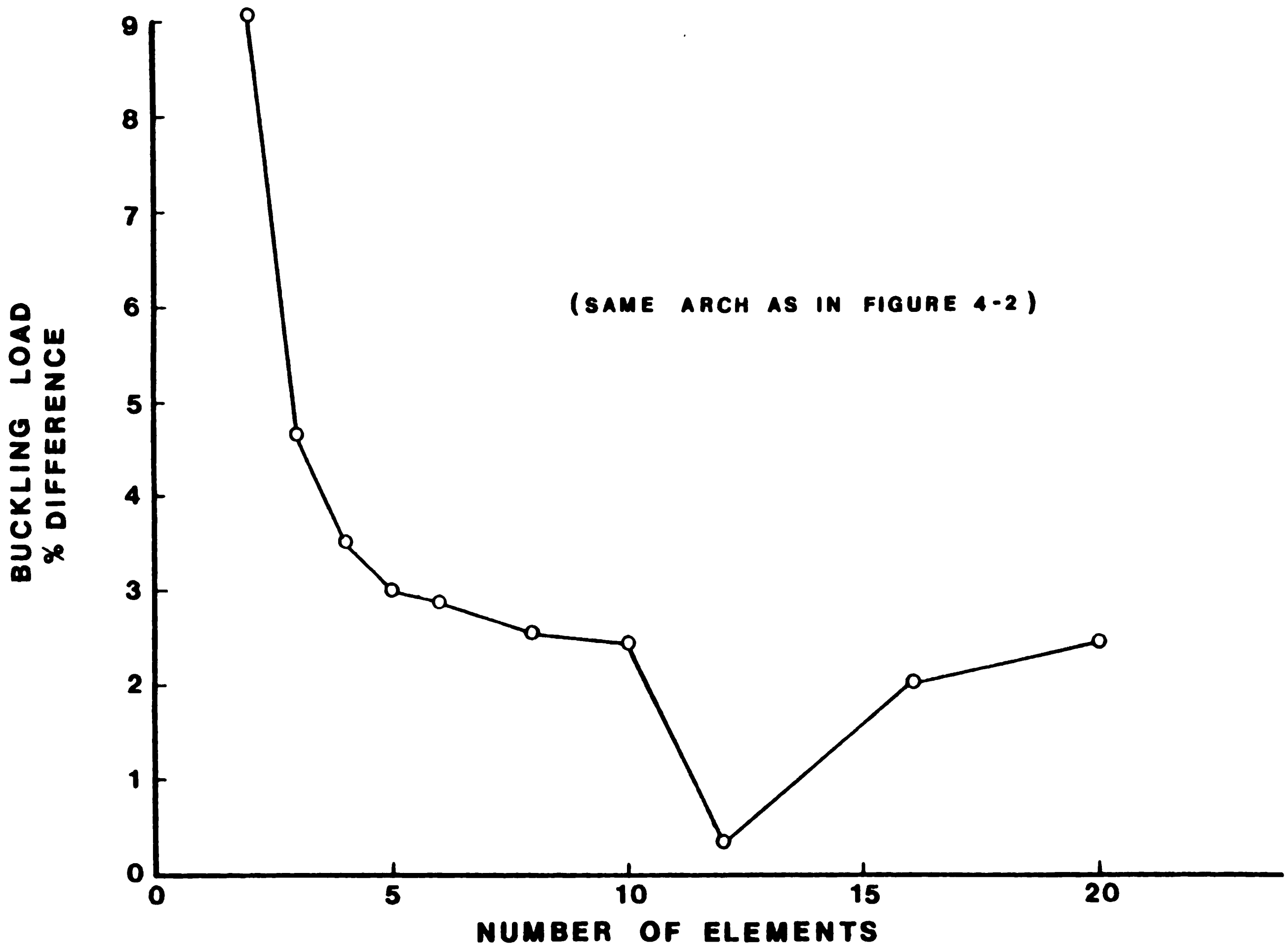


Figure 4-5. Out-of-plane buckling of parabolic arch subjected to uniformly distributed vertical load.

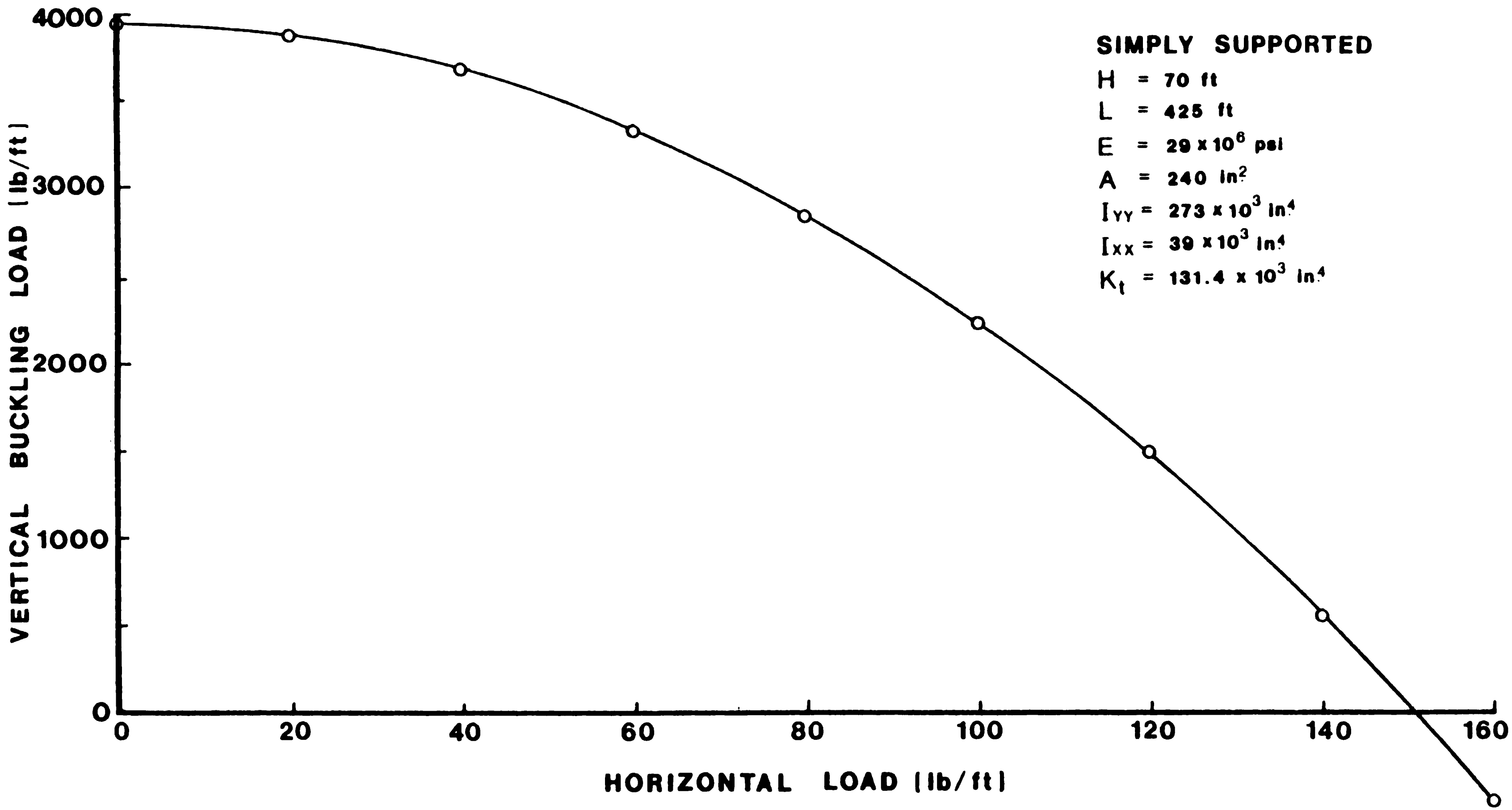


Figure 4-6. Effect of horizontal transverse load on vertical buckling load of a parabolic arch.

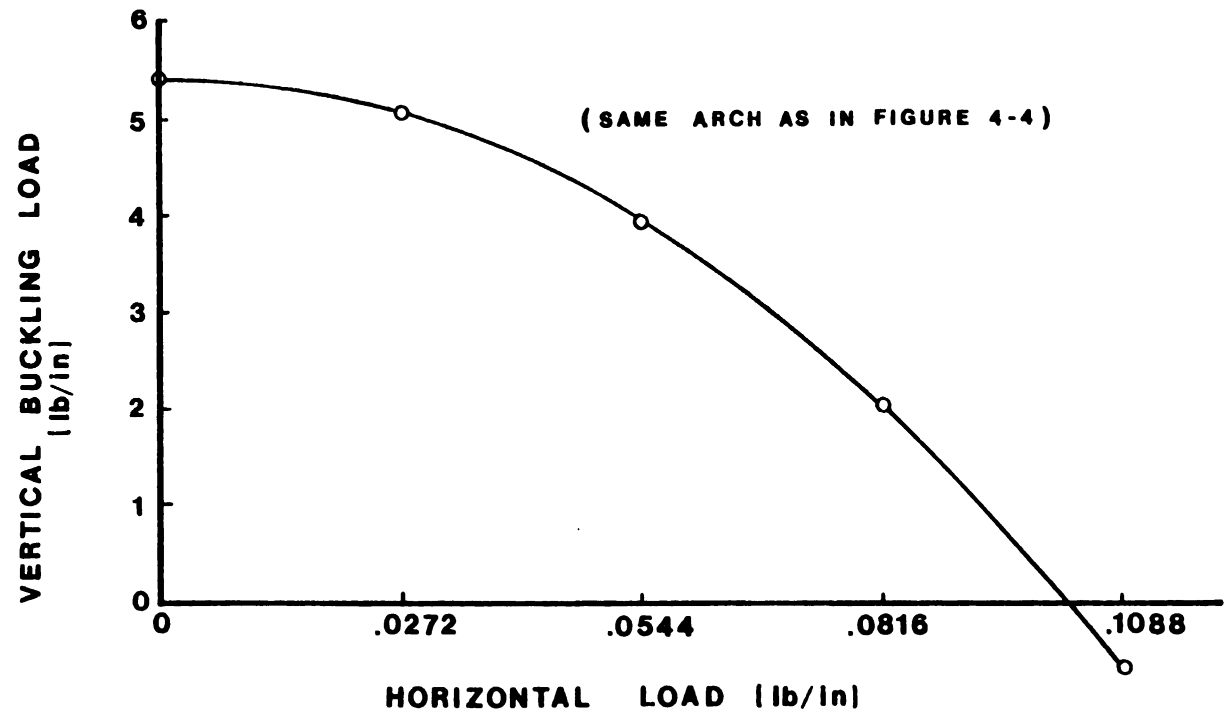


Figure 4-7. Effect of horizontal transverse load on vertical buckling load of a circular arch.

LIST OF REFERENCES

1. Ashwell, D. G., and Gallagher, R. H., Finite Elements for Thin Shells and Curved Elements, John Wiley & Sons, Inc., New York, 1976.

2. Ashwell, D. G., and Sabir, A. B., "Limitations of Certain Curved Finite Elements when Applied to Arches," International Journal of Mechanical Sciences, Vol. 13, No. 2, February, 1971, pp. 133-139.

3. Austin, W. J., "In-Plane Bending and Buckling of Arches," Journal of the Structural Division, ASCE, Vol. 97, ST5, May, 1971, pp. 1575-1592.

4. Austin, W. J., and Ross, T. J., "Elastic Buckling of Arches Under Symmetrical Loading," Journal of the Structural Division, ASCE, Vol. 102, No. ST5, May, 1976, pp. 1085-1095.

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

6. Brush, D. O., and Almroth, B. O., Buckling of Bars, Plates, and Shells, McGraw-Hill Book Company, New York, 1975.

7. Carnahan, B., Luther, H. A., and Wilkes, J. O., Applied Numerical Methods, John Wiley & Sons, Inc., New York, 1969.

8. Chaudhuri, S. K., and Shore, S., "Thin-Walled Curved Beam Finite Element," Journal of the Engineering Mechanics Division, ASCE, Vol. 103, No. EM5, October, 1977, pp. 921-937.

9. Conte, S. D., and DeBoor, C., Elementary Numerical Analysis an Algorithmic Approach, second edition, McGraw-Hill Book Company, New York, 1972.

10. Cook, R. D., Concepts and Applications of Finite Element Analysis, John Wiley & Sons, Inc., New York, 1974.

11. DaDeppo, D. A., and Schmidt, R., "Sidesway Buckling of Deep Circular Arches Under a Concentrated Load," Journal of Applied Mechanics, ASME. Vol. 36, June, 1969, pp. 325-327.

12. Dawe, D. J., "A Finite-Deflection Analysis of Shallow Arches by the Discrete Element Method," International Journal for Numerical Methods in Engineering, Vol. 3, 1971, pp. 529-552.

13. Dawe, D. J., "Numerical Studies Using Circular Arch Finite Elements," Computers and Structures, Vol. 4, 1974, pp. 729-740.

14. Donald, P., and Godden, W., "A Numerical Solution to the Curved Beam Problem," The Structural Engineer, Vol. 41, No. 6, June, 1963, pp. 179-186.

15. Donald, P., and Godden, W., "The Transverse Behavior of Laterally Unsupported Parabolic Arches," The Structural Engineer, Vol. 41, No. 6, June, 1963, pp. 187-191.

16. El-Amin, F. M., and Brotton, D. M., "Horizontally Curved Beam Finite Element Including Warping," International Journal for Numerical Methods in Engineering, Vol. 10, 1976, pp. 1397-1428.

17. El-Amin, F. M., and Kasem, M. A., "Higher-Order Horizontally-Curved Beam Finite Element Including Warping for Steel Bridges," International Journal for Numerical Methods in Engineering, Vol. 12, 1978, pp. 159-167.

18. Gallagher, R. H., Finite Element Analysis Fundamentals, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

19. Gallagher, R. H., Lien, S., and Mau, S. T., "Finite Element Plate and Shell Pre- and Post-Buckling Analysis," Cornell University.

20. Godden, W. G., "The Lateral Buckling of Tied Arches," Proceedings, Institution of Civil Engineers, London, Part 3, 1954, pp. 496-514.

21. Huddleston, J. V., "Finite Deflections and Snap-Through of High Circular Arches," Journal of Applied Mechanics, ASME, Vol. 35, December, 1968, pp. 763-769.

22. Huddleston, J. V., "Nonlinear Analysis of Steep, Compressible Arches of Any Shape," Journal of Applied Mechanics, Trans. ASME, Vol. 8, Ser. EN4, December, 1971, pp. 942-946.

23. Hurty, W. C., and Rubinstein, M. F., Dynamics of Structures, Prentice-Hall, Inc., 1964.

24. Johnston, B. G., editor, Guide to Stability Design Criteria for Metal Structures, third edition, John Wiley & Sons, Inc., New York, 1976.

25. Laursen, M., "A Curved Beam Equilibrium Element, Applicable in Standard Finite Element Program Systems," Technical University of Denmark, Structural Research Laboratory, Report No. R63, 1975.

26. Lee, H. P., "Generalized Stiffness Matrix of a Curved-Beam Element," AIAAJ, Vol. 7, No. 10, 1969, pp. 2043-2045.

27. Mallett, R. H., and Marcal, P. V., "Finite Element Analyses of Nonlinear Structures," Journal of the Structural Division, ASCE, Vol. 94, No. ST9, September, 1968, pp. 2081-2105.

28. Mak, C. K., and Kao, D., "Finite Element Analysis of Buckling and Post-Buckling Behaviors of Arches with Geometric Imperfections," Computers and Structures, Vol.3, 1973, pp. 149-161.

29. Mebane, P. M., and Stricklin, J. A., "Implicit Rigid Body Motion in Curved Finite Elements," Journal, American Institute of Aeronautics and Astronautics, Vol. 9, No. 2, 1971, pp. 344-345.

30. Newmark, N. M., "Numerical Procedure for Computing Deflections, Moments, and Buckling Loads," Transactions, ASCE, Vol. 108, Paper No. 2202, 1943, PP. 1161-1234.

31. Ojalvo, I. U., and Newman, M., "Buckling of Naturally Curved and Twisted Beams," Journal of the Engineering Mechanics Division, ASCE, Vol. 94, No. EM5, October, 1968, pp. 1067-1087.

32. Ojalvo, M., Demuts, E., and Tokarz, F., "Out-of-Plane Buckling of Curved Elements," Journal of the Structural Division, ASCE, Vol. 95, No. ST10, October, 1969, pp. 2305-2316.

33. Przemieniecki, J. S., Theory of Matrix Structural Analysis, McGraw-Hill Book Company, New York, 1968.

34. Rajasekaran, S., and Murray, D. W., "Incremental Finite Element Matrices," Journal of the Structural Division, ASCE, Vol. 99, No. ST12, December, 1973, pp. 2423-2438.

35. Sabir, A. B., and Lock, A. C., "Large Deflection, Geometrically Nonlinear Finite Element Analysis of Circular Arches," International Journal of Mechanical Sciences, Vol. 15, 1973, pp. 37-47.

36. Segerlind, L. J., Applied Finite Element Analysis, John Wiley & Sons, Inc., New York, 1976.

37. Seely, F. B., and Smith, J. O., Advanced Mechanics of Materials, 2nd Edition, John Wiley & Sons, Inc., 1952.

38. Shukla, S., and Ojalvo, M., "Lateral Buckling of Parabolic Arches with Tilting Loads," Journal of the Structural Division, ASCE, Vol. 97, No. ST6, June, 1971, pp. 1763-1773.

39. Timoshenko, S. P., and Gere, J. M., Theory of Elastic Stability, second edition, McGraw-Hill Book Company, New York, 1961.

40. Thornton, W. A., and Master, B. G., "Direct Stiffness Formulation for Horizontally Curved Beams," Journal of the Structural Division, ASCE, Technical Notes, Vol. 103, No. ST1, January, 1977, pp. 284-289.

41. Tokarz, F. J., Experimental Study of Lateral Buckling of Arches," Journal of the Structural Division, ASCE, Vo. 97, No. ST2, February, 1971, pp. 545-559.

42. Tokarz, F. J., and Sandhu, R. S., "Lateral-Torsional Buckling of Parabolic Arches," Journal of the Structural Division, ASCE, Vol. 98, No. ST5, May, 1972, pp. 1161-1179.

43. Walker, A. C., "A Nonlinear Finite Element Analysis of Shallow Arches," International Journal of Solids and Structures, Vol. 5, 1969, pp. 97-107.

44. Yabuki, T., and Kuranishi, S., "Out-of-Plane Behavior of Circular Arches Under Side Loadings," Japanese Society of Civil Engineers, Proceedings, No. 214, June, 1973, pp. 71-82.

45. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill, London, England, 1971.

APPENDIX A

METHODS FOR COMPUTING b_i IN EQUATION (2-17)

Three methods of solution were considered, i.e., closed form solution for the coefficients in single and double precision, and a normal application of the Cramer's rule method in single precision computation.

The closed form solution is as follows:

$$b_i = \frac{D_i}{D} \tag{A-1}$$

in which

$$D = 24\,\Theta\,[-\Theta^3 \sin\Theta - 4\,\Theta^2(2\cos\Theta + 1) + 24\,\Theta\sin\Theta + 24\,(\cos\Theta - 1)] \tag{A-2}$$

$$\begin{aligned} D_2 = {} & 12\,(R2 - R1)\,[-\Theta^4 - 2\,\Theta^3 \sin\Theta - 12\,\Theta^2\cos\Theta + \\ & 24\,\Theta\sin\Theta + 24\,(\cos\Theta - 1)] + \\ & 12\,(z_B - R1\sin\Theta)[\Theta^4 \sin\Theta + 2\,\Theta^3(2\cos\Theta + 1) - \\ & 6\,\Theta^2\sin\Theta\,] + 12\,[x_B - R1(1 - \cos\Theta)] \\ & [-\Theta^4\cos\Theta + 4\,\Theta^3\sin\Theta + 6\,\Theta^2(\cos\Theta - 1)] \end{aligned} \tag{A-3}$$

$$\begin{aligned} D_3 = {} & 8\,[x_B - R1(1 - \cos\Theta)][\Theta^3(2\cos\Theta + 1) - \\ & 6\,\Theta^2\sin\Theta - 6\,\Theta(\cos\Theta - 1)] + 8\,(R2 - R1) \\ & [\Theta^3(\cos\Theta + 2) - 3\,\Theta^2\sin\Theta] + 8\,[z_B - R1\sin\Theta] \\ & [-2\,\Theta^3\sin\theta - 6\,\theta^2\cos\Theta + 6\,\Theta\sin\Theta] \end{aligned} \tag{A-4}$$

$$D_4 = 6[z_B - R1 \sin \Theta][\Theta^2 \sin \Theta + 2\Theta(\cos \Theta - 1)] + 6(R2 - R1)[-\Theta^2(\cos \Theta + 1) + 4\Theta \sin \Theta + 4(\cos \Theta - 1)] + 6[x_B - R1(1 - \cos \Theta)][-\Theta^2(\cos \Theta + 1) + 2\Theta \sin \Theta] \qquad \text{(A-5)}$$

A comparison of solutions by the methods was made of a parabolic arch (H = 16.25 in, L = 96 in). The horizontal span L was divided into 16 equal intervals with element 1 next to the support and element 8 next to the crown. Table A-1 shows the values of the b_i coefficients and the computed lengths by each of the solution procedures. It may be seen that the normal application of the Cramer's rule in single precision gives as good results as those obtained from the double precision computation of the closed form expressions. Therefore, Cramer's rule was chosen to compute the coefficients. It may be seen also that the element lengths based on the polynomial approximation compare very well with the exact values.

TABLE A-1 SOLUTION PROCEDURES FOR COEFFICIENTS b_i

ELEMENT 1

	b_2	b_3	b_4	LENGTH*(in)
CLOSED FORM SINGLE PRECISION	-125.032539	152.003264	-133.257151	7.109667
CLOSED FORM DOUBLE PRECISION	-126.685134	171.832705	-149.414783	7.107792
CRAMER'S RULE SINGLE PRECISION	-126.685133	171.832679	-149.414572	7.107792

*EXACT SOLUTION = 7.107792 in.

ELEMENT 8

CLOSED FORM SINGLE PRECISION	- 9.163698	37.152620	- 4.213572	6.006985
CLOSED FORM DOUBLE PRECISION	- 9.090848	36.567125	- 4.153199	6.007155
CRAMER'S RULE SINGLE PRECISION	- 9.090848	36.567125	- 4.153201	6.007155

*EXACT SOLUTION = 6.007165 in

APPENDIX B

MODIFIED REGULA FALSI ITERATION TECHNIQUE

The technique used for the determinant search method (Equation (3-6)) is that described by Conte and de Boor (9). Figure B-1 illustrates the method graphically. If the curve f(x) is continuous in the interval $[a_o, b_o]$ then the first approximation to the root is b_1, obtained by a straight line from point F, $(a_o, f(a_o))$ to G, $(b_o, f(b_o))$. The next approximation, b_2, is obtained by joining G', $(b_1, f(b_1))$ and F', $(a_o, \frac{1}{2}f(a_o))$. Next time F'', $(a_o, \frac{1}{4}f(a_o))$ will be used and a_3 is obtained. It is seen to lie at the other side of the root. Now the procedure to approach the root is inversed by joining the value of the function at a_3 with $G''/2$, $(b_2, \frac{1}{2}f(b_2))$, and continued until convergence is achieved. The algorithm is given as follows: Given f(x) continuous on $[a_o, b_o]$ and such that $f(a_o)\ f(b_o) < 0$.

Set $F = f(a_o)$, $G = f(b_o)$, $W_o = a_o$

For n = 0, 1, 2,..., until satisfied, do:

Calculate $w_{n+1} = (Ga_n - Fb_n)/(G - F)$

If $f(a_n) f(w_{n+1}) \leq 0$, set $a_{n+1} = a_n$, $b_{n+1} = w_{n+1}$,

$G = f(w_{n+1})$

If also $f(w_n)\ f\ (w_{n+1}) > 0$, set $F = F/2$

Otherwise, set $a_{n+1} = w_{n+1}$, $F = f(w_{n+1})$

$b_{n+1} = b_n$

If also $f(w_n)\ f\ (w_{n+1}) > 0$, set $G = G/2$

Then $f(x)$ has a root in the interval $[a_{n+1}, b_{n+1}]$

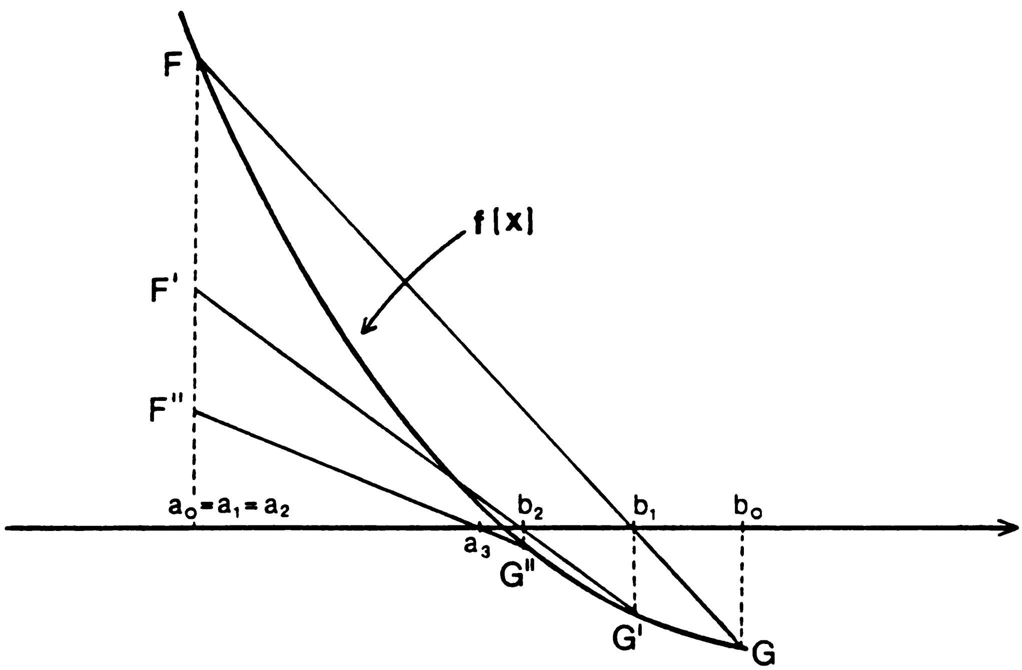


Figure B-1. Modified regula falsi iteration.

APPENDIX C

COMPUTER PROGRAM

C.1 DESCRIPTION OF SUBROUTINES

A general description of the computer program is given in Section 3.6. A listing of the program is presented at the end of this appendix with a considerable amount of "comment cards" to facilitate an understanding of the program. In the following a brief description of the subroutines is given.

The computer program consists of a main program called CONTROL, seventeen subroutines and one function subprogram. The program CONTROL directs the flow of the computations by calling the appropriate subroutines for each step of the solution procedure. The subroutine NODDATA reads data regarding the overall geometry of the arch and data regarding the nodal degrees of freedom. It generates the coordinates of the nodes and the equation numbers. The subroutine LOAD reads the location, magnitude, and direction of the external loads being applied to the arch. The subroutine BAND computes the semibandwidth, MBAND, that the stiffness matrix of the structure will have. This is done by obtaining the largest difference between the equation numbers of the nodes of any element.

The subroutine CURVED reads and directs all basic information concerning the curved elements, i.e., material properties, element and cross-section information, element geometric properties, and computation of the stiffness matrices of each element and their assembly into the structure stiffness matrices. The computation of the geometric properties of the curved elements is accomplished by the subroutine GEOMTRY. The subroutine NUMINT performs the numerical integration to obtain the linear and incremental stiffness matrices of each element. The assembly of these stiffnesses into the appropriate global stiffness matrices is accomplished with the subroutine ASEMBLE. The subroutines LINSOLN and GAUSSOL solve the system of linear equations by Gauss elimination. The solution of the linear eigenvalue problem and the quadratic eigenvalue problem is obtained by the subroutines EIGENVL and NLEIGNP respectively. The subroutine EIGENVL uses inverse vector iteration with Rayleigh quotient to obtain the lowest eigenvalue and corresponding eigenvector of the linear problem. For the solution of the quadratic problem, the subroutine NLEIGNP uses the modified regular falsi method of iteration by calling the subroutine MRGFLS and the function subprogram DET.

The solution of the buckling problem under tilted loads for either deck and through arches, is accomplished by the subroutine TILTED. For solutions of half-through arches, modifications should be made as explained in Section C.4.

C.2 VARIABLES USED IN THE COMPUTER PROGRAM

The variable names used in the program are listed below in alphabetical order:

Program CONTROL

A(M) = Area of the section of element M;

AC = Coefficient of parabola, $Y = (AC)X^2$, that defines arch;

A1, A2 = Limits of the numerical integration;

B2(M), B3(M), B4(M) = Coefficients defined by geometry of element M, they are used in the numerical integration;

D(I) = Displacement vector, found from the solution of the system S*D = R. I varies from 1 to NEQ;

DN(J) = Vector that identifies the displacements at the two nodes of an element. J varies from 1 to 16;

D10 = Dummy Variable;

E(N) = Modulus of elasticity of element group N;

G(N) = Shear Modulus of element group N;

IARCH = Variable that identifies the type of arch being studied. If EQ. 0, parabolic arch, if EQ. 1, circular arch;

IA(N,I) = "Boundary condition code" of node N for its Ith degree of freedom. Initially it is defined as follows:

IA(N, I) = 1 if constrained;

= 0   if free

After processing,

IA(N, I) = 0   if initially = 1;
= equation number for the d.o.f. if initially = 0;

IB(N,I) = "Additional boundary condition code."

IB(N, I) = 0   if free;
= N   if slave to node N;
= -1   if to be condensed.

After processing, IB(N, I) is unchanged except, IB(N, I) = -(condensation number for the d.o.f. if initially IB(N, I) = -1);

ICAL 1 = Variable controlling print-out. If EQ. 0, print element weight, nodal loads due to weight of elements, limits of integration and number of quadrature points, linear stiffness of each element, incremental stiffnesses of each element. If EQ. 1, skip;

ICAL 2 = Variable controlling print-out. If EQ. 0, print element geometric properties. If EQ. 1, skip;

ICAL 3 = Variable controlling print-out. If EQ. 0, print uncondensed load vector and uncondensed structure stiffnesses S, S1, and S2. If EQ. 1, skip;

ICAL 4 = Variable controlling print-out. If EQ. 0, print initial and nodal loads processed into load vector, R(I). If EQ. 1, skip;

ICAL 5 = Variable controlling print-out. If EQ. 0, print load vector R(I) for linear solution and the displacement vector, D(I). If EQ. 1, skip;

ICAL 6 = Variable controlling print-out.
If EQ. 0, print nodal displacements on each element.
If EQ. 1, skip;

ICAL 7 = Variable controlling print-out.
If EQ. 0, print data sent to the subroutine GAUSSOL on each iteration of the linear eigenvalue problem. Also print the intermediate values after each iteration.
If EQ. 1, skip;

IDATA = Variable for checking input data.
If EQ. 1, data check only, skip all computations.
If. EQ. 0, solve problem;

IDIRCN = Variable identifying if uniformly distributed load is vertical or horizontal.
If EQ. 0, distributed load is vertical.
If EQ. 1, distributed load is horizontal;

IEIGEN = Variable identifying the type of eigenvalue problem that is being solved.
If EQ. 0, linear.
If EQ. 1, quadratic
If EQ. 2, both linear and quadratic;

ILAT = Variable defining if the influence of initial horizontal load on the vertical buckling load is going to be computed (EQ. 1) or not (EQ. 0);

ILOAD = Variable identifying if the applied load is uniformly distributed (EQ. 0) or concentrated at the nodes (EQ. 1);

IPAR = Variable identifying different stages of the computation in the subroutine CURVED;
= 2, assemble and store in tape matrix K;
= 3, assemble and store in tape matrix N2;
= 4, assemble and store in tape matrix N1;

ISTRES = If EQ. 1, compute nodal forces and stresses in the structure. If EQ. 0, skip;

ITILT = Variable identifying whether the tilted load case is going to be studied or not.
If EQ. 1, compute eigenvalue for tilted case. If EQ. 0, skip;

IXX(M)	=	Moment of inertia about the y-axis of the cross-section of element M;
KT(M)	=	Torsion constant of element M. Depends on the shape and dimensions of the cross-section;
L(N,K)	=	Variable identifying the Kth element in the element group N;
LINEQL	=	If EQ. 0, execute complete buckling problem. If EQ. 1, compute linear equilibrium solution only;
LENGTH(M)	=	Length of element M;
MBAND	=	Semibandwidth of structure stiffness matrix;
MP	=	Number of points in the Gauss-Legendre quadrature formula;
NCOND	=	Identifies the total number of degrees of freedom to be condensed out;
NE	=	Total number of elements in the structure;
NE	=	Identifies the total number of equations that the system has;
NODEI(M)	=	Variable identifying the number of node I of element M;
NODEJ(M)	=	Variable identifying the number of node J of element M;
NSIZE	=	Identifies the total number of degrees of freedom, condensed and free, of the system. (NSIZE = NEQ + NCOND);
NTYPE(J)	=	Total number of elements of type J;
NUMEG	=	Total number of element groups;
NUMEL(I)	=	Total number of elements in element group I;
NUMND	=	Total number of nodal points;

PHII(M)	=	Angle ϕ_i that has tangent at node J of element M, with respect to the horizontal;
PHIJ(M)	=	Angle ϕ_j that has tangent at node J of element M, with respect to the horizontal;
PN(N,I)	=	Load applied at node N, in the I direction;
RAD	=	Radius of the arch in the case that it is circular;
R(I)	=	Load vector of the system;
RI(M)	=	Radius of curvature at node I of element M;
RJ(M)	=	Radius of curvature at node J of element M;
S(I,J)	=	I, Jth element of the structure stiffness matrix;
SE(I,J)	=	I, Jth element of the element stiffness matrix;
TI,...,T8	=	Variables for title of problem being solved;
TETA(M)	=	Angle between the radii of curvature at nodes I and J of element M;
U(N, I)	=	Variable identifying the displacement in the direction I of node N;
X(N)	=	Global X-coordinate of node N;
Y(N)	=	Global Y-coordinate of node N;
Z(N)	=	Global Z-coordinates of node N.

Subroutine NODDATA

DI	=	Total number of horizontal intervals in which the span of the parabolic arch is to be divided into;
H	=	Height of the arch;

L	=	Span covered by the arch;
XS	=	X-coordinate of the left most node of the arch.

Subroutine LOAD

W	=	Uniformly distributed load on the arch.

Subroutine CURVED

DM	=	Density of the material

Subroutine GEOMTRY

DY	=	First derivative of the curve defining the arch at a particular node N;
D2Y	=	Second derivative of the curve defining the arch at a particular node N;
XR,ZR	=	Nodal local coordinates of node J, after the rotation of a particular element M.

Subroutine NUMINT

JFIRST	=	Initial value of j, K_i ;
JLAST	=	Final value of j, $K_{i+1} -1$;
KEY	=	Vector K. Used to locate the roots and weight factors for the MP-point formula;
NPOINT	=	Vector P. Used to locate the roots and weight factors for the MP-point formula;
SUM	=	Accumulated sum to find the area under the curve;
WEIGHT	=	Vector of weight factors;
Z	=	Vector of Legendre polynomial roots.

Subroutine EIGENVL

EIGEN = Eigenvalue found from the inverse iteration solution of the linear eigen-problem;

EIGNVTR = Eigenvector corresponding to EIGEN;

EPSI = Variable determining the convergence criterion to the eigenvalue being sought;

MAX = Maximum number of iterations allowed;

RHO = Rayleigh quotient. Used to improve the inverse vector iteration method;

XB = Vector that stores the approximation to the eigenvector after each iteration.

Subroutine NLEIGNP

A, B = Variables defining the interval in which the eigenvalue is enclosed;

ERROR = Upper bound on the computation of the eigenvalue after convergence;

FL = Value of the determinant of the matrix S = K + L * N1 + L * L * N2 at the converged value of the eigenvalue;

FTOL = Convergence criterion for sufficiently small value of the determinant of eigen-value;

L = Converged value of the eigenvalue, L = (A + B)/2;

NTOL = Maximum number of iterations allowed;

XTOL = Convergence criterion for sufficiently small interval A, B , enclosing the eigenvalue.

Subroutine MRGFLS

IFLAG = Variable defining the status of the iteration. If EQ. 1, convergence was successful. If EQ. 2, no convergence after NTOL iterations. If EQ. 3, both endpoints, A and B, are on the same side of the root, hence method of iteration can not be used;

FA = Value of the determinant of matrix S at interval endpoint A;

FB = Value of the determinant of matrix S at interval endpoint B;

W = Weighted value of the root between interval endpoints A and B;

FW = Value of the determinant of matrix S at the weighted value W.

Function DET

DET = Value of the determinant of the matrix S = K +L * N1 + L * L * N2 at a particular value of L;

K = Part of element S(I, J) corresponding to linear stiffness K(I, J);

N1 = Part of element S(I, J) corresponding to matrix N1(I, J);

N2 = Part of element S(I, J) corresponding to matrix N2(I, J);

Subroutine TILTED

IDECK = Variable identifying a deck arch, EQ. 1, or a through arch, EQ. 0;

HD = Distance from the deck to the crown of the arch;

P = Load on each of the columns or on each of the hangers of either the through arch or the deck arch respectively.

C-3 COMPUTER PROGRAM

```
      PROGRAM CONTROL (INPUT,OUTPUT,TAPE60=INPUT,TAPE61=OUTPUT,TAPE1,
     +                 TAPE2,TAPE3,TAPE4,TAPE5,TAPE6,TAPE7,TAPE8,TAPE9)
C
C     ******************************************************************
C        THIS PROGRAM USES THE FINITE ELEMENT METHOD TO ANALYZE
C        A CURVED ELEMENT IN THREE DIMENSIONS.
C        OTHER ELEMENTS MAY BE ANALYZED BY ADDING A SUBROUTINE
C        FOR EACH NEW TYPE OF ELEMENT BEING USED.
C        NONLINEAR PROPERTIES ARE TAKEN INTO CONSIDERATION IN THE
C        CURVED ELEMENT.
C     ******************************************************************
C
      REAL IXX,IYY,KT,LENGTH
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +         ICAL4,ICAL5,ICAL6,ICAL7
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(37,8),IB(37,8),X(37),Y(37),Z(37),RAD,AC
      COMMON/4/SE(16,16)
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),A(36),IXX(36),IYY(36),
     +          KT(36),L(1,36)
      COMMON/6/A1,A2,MP,B2(36),B3(36),B4(36)
      COMMON/7/RI(36),RJ(36),PHII(36),PHIJ(36),TETA(36),LENGTH(36)
      COMMON/8/PN(37,8),R(296)
      COMMON/9/S(296,16)
      COMMON/10/D(296),D10(1184)
      COMMON/11/DN(16),U(37,8)
C
      READ(60,1010) T1,T2,T3,T4,T5,T6,T7,T8
      WRITE(61,2020)T1,T2,T3,T4,T5,T6,T7,T8
      READ(60,1015) NE,NUMNP,NUMEG,IDATA,ICAL1,ICAL2,ICAL3,ICAL4,
     +              ICAL5,ICAL6,ICAL7
      WRITE(61,2010)NE,NUMNP,NUMEG,IDATA,ICAL1,ICAL2,ICAL3,ICAL4,
```

```
     +                ICAL5,ICAL6,ICAL7
C
C........READ NODAL POINT DATA
C
      READ(60,1030) IARCH,ILOAD,ISTRESS,IEIGEN,SCALE,ILAT,ITILT,IDIRCN,
     +              LINEQL
      WRITE(61,2030)IARCH,ILOAD,ISTRESS,IEIGEN,SCALE,ILAT,ITILT,IDIRCN,
     +              LINEQL
      DX=0.
      CALL NODDATA (IARCH,DX)
C
C........READ AND STORE ELEMENT DATA
C
      IPAR=1
      DO 100 N=1,NUMEG
      READ(60,1020) NTYPE(N),NUMEL(N)
      CALL ELEMENT (N,IDATA,IARCH)
100   CONTINUE
C
C........READ AND STORE INITIAL LOAD DATA
C
      WRITE(61,2015)
      W=0.
      CALL LOAD (IARCH,ILOAD,IDIRCN,DX,W)
C
C........COMPUTE SEMIBANDWIDTH OF STRUCTURE STIFFNESS MATRIX
C
      CALL BAND
C
C........IF DATA CHECK ONLY SKIP ALL FURTHER CALCULATIONS
C
      IF(IDATA.NE.0) GO TO 900
```

```
C
C........ASSEMBLE INITIAL LOADS AND NODAL LOADS INTO LOAD VECTOR
C           SET ARRAYS -S- AND -R- EQUAL TO ZERO
C
102    CONTINUE
       DO 105 I=1,NSIZE
       R(I)=0.
       DO 105 J=1,MBAND
105    S(I,J)=0.
       CALL ASEMBLE (M)
C
C........COMPUTE ELEMENT LINEAR STIFFNESS AND ASSEMBLE INTO STRUCTURE
C           LINEAR STIFFNESS
C
       IPAR=2
       DO 110 N=1,NUMEG
       CALL ELEMENT (N,IDATA,IARCH)
110    CONTINUE
C
C........WRITE LINEAR STIFFNESS AND LOAD VECTOR OF STRUCTURE
C
       IF (ICAL3.EQ.0) CALL STCONDN
C
C........SOLVE SYSTEM OF LINEAR EQUATIONS S*D=R
C
       CALL LINSOLN
C
C........IDENTIFY DISPLACEMENTS FOUND FROM SOLUTION OF S*D=R
C
       CALL IDENT
C
C........COMPUTE NODAL FORCES AND STRESSES DUE TO LINEAR DISPLACEMENTS
```

```
C
      IF (ISTRESS.NE.0) CALL STRESS
C
C........CHECK IF ONLY WANT LINEAR EQUILIBRIUM SOLUTION
C
      IF (LINEQL.EQ.1) GO TO 900
C
C........FOR EACH CURVED ELEMENT COMPUTE SE1 AND SE2, STORE THEM,
C           AND ASSEMBLE THEM INTO STRUCTURE STIFFNESSES S1 AND S2.
C              IPAR=3, OBTAIN S1 FROM MATRICES SE1
C              IPAR=4, OBTAIN S2 FROM MATRICES SE2
C
      N=1
      IPAR=3
      DO 120 I=1,NSIZE
      DO 120 J=1,MBAND
  120 S(I,J)=0.
      CALL CURVED (N,IDATA,IARCH)
C
C........CHECK FOR PRINTING OF MATRIX -S1-
C
      IF (ICAL3.EQ.0) CALL STCONDN
      IPAR=4
      IF (IEIGEN.EQ.0.AND.IPAR.EQ.4) GO TO 130
      DO 125 I=1,NSIZE
      DO 125 J=1,MBAND
  125 S(I,J)=0.
      CALL CURVED (N,IDATA,IARCH)
C
C........CHECK FOR PRINTING OF MATRIX -S2-
C
      IF (ICAL3.EQ.0) CALL STCONDN
```

```
  130   CONTINUE
C
C........CHECK TYPE OF EIGENVALUE PROBLEM TO BE SOLVED
C
        IF (ILAT.EQ.1) GO TO 135
        IF (ITILT.EQ.1) GO TO 200
C
C........CHECK FOR SOLUTION OF QUADRATIC EIGENPROBLEM
C
        IF (IEIGEN.NE.0) CALL NLEIGNP (SCALE)
C
C........CHECK FOR SOLUTION OF LINEAR EIGENPROBLEM
C
        IF (IEIGEN.EQ.1) GO TO 900
        EIGEN=0.
        CALL EIGENVL (EIGEN,IDATA)
C
C........CHECK EFFECT OF INITIAL HORIZONTAL LOAD ON THE VERTICAL
C           BUCKLING LOAD OF THE ARCH OR VICEVERSA
C
  135   CONTINUE
        IF (ILAT.EQ.0) GO TO 200
C
C........COMBINE LINEAR STIFFNESS S AND NONLINEAR STIFFNESS S1
C           DUE TO EXISTING LOAD AND STORE IT
C
        DO 145 I=1,NEQ
        DO 140 J=1,MBAND
        READ(4,10) SK
        READ(5,10) SN1
        S(I,J)=SK+SN1
  140   CONTINUE
```

```
145   CONTINUE
      REWIND 4
      REWIND 5
      WRITE(7,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 7
C
C........READ SECOND LOADING ON ARCH
C        IF IDIRCN.EQ.0, SECOND LOADING IS HORIZONTAL
C        IF IDIRCN.EQ.1, SECOND LOADING IS VERTICAL
C
      IDR=IDIRCN
      IF (IDR.EQ.0) IDIRCN=1
      IF (IDR.EQ.1) IDIRCN=0
      W=0.
      CALL LOAD (IARCH,ILOAD,IDIRCN,DX,W)
C
C........COMPUTE MATRIX S1 FOR NEW APPLIED LOADING
C
      IEIGEN=0
      ILAT=0
      IDATA=1
      IPAR=1
      GO TO 102
200   CONTINUE
C
C........CHECK EFFECT OF TILTED LOAD ON BUCKLING LOAD
C
      IF (ITILT.EQ.0) GO TO 900
      CALL TILTED (IDATA,SCALE,DX,W)
C
900   CONTINUE
C
```

```
10    FORMAT(E21.6)
1010  FORMAT(A10,A10,A10,A10,A10,A10,A10,A10)
1015  FORMAT(11I5)
1020  FORMAT(2I5)
1030  FORMAT(4I5,E10.2,4I5)
2010  FORMAT(////7X,6HNE    =,I3//7X,6HNUMNP=,I3//7X,
     +       6HNUMEG=,I3//7X,6HIDATA=,I3//7X,6HICAL1=,I3//7X,6HICAL2=,
     +       I3//7X,6HICAL3=,I3//7X,6HICAL4=,I3//7X,6HICAL5=,I3//
     +       7X,6HICAL6=,I3//7X,6HICAL7=,I3)
2015  FORMAT(*1*,15H  INITIAL LOADS//7H   NODE,27X,14HLOAD DIRECTION//
     +       7H NUMBER,9X,1HU,9X,1HV,9X,1HW,9X,1HB,8X,2HTY,8X,2HTX//)
2020  FORMAT(*1*,A10,A10,A10,A10,A10,A10,A10,A10)
2030  FORMAT(////7X,8HIARCH   =,I3//7X,8HILOAD   =,I3//7X,8HISTRESS=,
     +       I3//7X,8HIEIGEN =,I3//7X,8HSCALE   =,E10.2//
     +       7X,8HILAT    =,I3//7X,8HITILT   =,I3//7X,8HIDIRCN =,I3//
     +       7X,8HLINEQL =,I3)
C
      END
```

```
      SUBROUTINE NODDATA (IARCH,DX)
C
C     ******************************************************************
C        TO READ AND PRINT NODAL POINT DATA
C        TO CALCULATE EQUATION NUMBERS AND CONDENSATION NUMBERS AND
C        STORE THEM IN ARRAYS  -IA- AND -IB- RESPECTIVELY
C     ******************************************************************
C
      REAL L
      COMMON/1/NE,NUMNP,D1(15)
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(37,8),IB(37,8),X(37),Y(37),Z(37),R,A
C
C.........READ NODAL POINT DATA
C
      WRITE(61,2000)
      WRITE(61,2010)
      WRITE(61,2015)
      IF (IARCH.EQ.0) GO TO 101
      READ(60,1010) R
100   READ(60,1000) N,(IA(N,I),I=1,8),(IB(N,I),I=1,8),T,Z(N)
      PI=4.*ATAN(1.)
      T=T*PI/180.
      X(N)=SIN(T)*R
      Y(N)=R*(1.-COS(T))
      WRITE(61,2020) N,(IA(N,I),I=1,8),(IB(N,I),I=1,8),X(N),Y(N),Z(N)
      IF (N.NE.NUMNP) GO TO 100
      GO TO 103
101   CONTINUE
      READ(60,1020) H,L,DI,XS
      A=H/(XS*XS)
      DX=L/DI
```

```
      DL=0.
102   READ(60,1000) N,(IA(N,I),I=1,8),(IB(N,I),I=1,8),T,Z(N)
      X(N)=XS+DL
      Y(N)=A*X(N)*X(N)
      WRITE(61,2020) N,(IA(N,I),I=1,8),(IB(N,I),I=1,8),X(N),Y(N),Z(N)
      DL=DL+DX
      IF (N.NE.NUMNP) GO TO 102
C
C........PROCESS ARRAYS -IA- AND -IB- TO FIND EQUATION NUMBERS AND
C        CONDENSATION NUMBERS. STORE NEQ"S AND NCOND"S IN ARRAYS IA AND
C        IB RESPECTIVELY.
C
103   NEQ=0
      NCOND=0
      DO 125 N=1,NUMNP
      DO 120 I=1,8
      IF(IA(N,I).NE.1) GO TO 105
      IA(N,I)=0
      GO TO 120
105   IA(N,I)=-1
      IF(IB(N,I)) 110,115,120
110   NCOND=NCOND+1
      IB(N,I)=-NCOND
      GO TO 120
115   NEQ=NEQ+1
      IA(N,I)=NEQ
120   CONTINUE
125   CONTINUE
      NSIZE=NEQ+NCOND
C
C........WRITE GENERATED NODAL POINT DATA
C
```

```
      WRITE(61,2030)
      WRITE(61,2040)
      WRITE(61,2050) (N,(IA(N,I),I=1,8),(IB(N,I),I=1,8),N=1,NUMNP)
      WRITE(61,2060) NSIZE,NEQ,NCOND
      RETURN
C
1000  FORMAT(I5,16I3,F15.10,F10.6)
1010  FORMAT(F15.9)
1020  FORMAT(4F10.5)
2000  FORMAT(*1*,33H N O D A L   P O I N T   D A T A   //)
2010  FORMAT(18H INPUT NODAL DATA //)
2015  FORMAT(7H   NODE,26X,36HNODAL POINT BOUNDARY CONDITION CODES,33X,
     +       23HNODAL POINT COORDINATES/7H NUMBER,21X,7HIA(N,I),33X,
     +       7HIB(N,I)/11X,2(4X,1HU,4X,1HV,4X,1HW,4X,1HB,3X,2HTY,3X,
     +       2HTX,10X),9X,4HX(N),8X,4HY(N),8X,4HZ(N))
2020  FORMAT(I5,6X,16I5,4X,3F12.3)
2030  FORMAT(///22H GENERATED NODAL DATA //)
2040  FORMAT(7H   NODE,16X,16HEQUATION NUMBERS,22X,
     +       20HCONDENSATION NUMBERS/7H NUMBER,21X,7HIA(N,I),33X,
     +       7HIB(N,I)/11X,2(4X,1HU,4X,1HV,4X,1HW,4X,1HB,3X,2HTY,3X,2HTX
     +       ,10X))
2050  FORMAT(I5,6X,16I5)
2060  FORMAT(*-*,6HNSIZE=,I3,3X,4HNEQ=,I3,3X,6HNCOND=,I3)
C
      END
```

```
      SUBROUTINE LOAD (IARCH,ILOAD,IDIRCN,DX,W)
C
C     ********************************************************************
C        TO READ AND STORE INITIAL LOAD DATA
C     ********************************************************************
C
      REAL LENGTH
      COMMON/1/NE,NUMNP,D1(15)
      COMMON/3/IA(37,8),IB(37,8),X(37),Y(37),Z(37),RAD,AC
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),D5(180)
      COMMON/7/RI(36),RJ(36),PHII(36),PHIJ(36),TETA(36),LENGTH(36)
      COMMON/8/PN(37,8),R(296)
C
C........CHECK TYPES OF LOADS TO BE READ
C          ILOAD.EQ.0 , LOAD IS UNIFORMLY DISTRIBUTED
C          ILOAD.EQ.1 , LOADS ARE CONCENTRATED
C
      IF (ILOAD.EQ.0) GO TO 200
100   READ(60,1015) N,(PN(N,I),I=1,8)
      WRITE(61,2020) N,(PN(N,I),I=1,8)
      IF (N.NE.NUMNP) GO TO 100
      RETURN
200   CONTINUE
      READ(60,1020) W
      IF (IDIRCN.EQ.0) WRITE(61,2030) W
      IF (IDIRCN.EQ.1) WRITE(61,2040) W
      DO 300 N=1,NUMNP
      DO 300 I=1,8
300   PN(N,I)=0.
      DO 400 M=1,NE
      NI=NODEI(M)
      NJ=NODEJ(M)
```

```
C
C........CHECK IF DISTRIBUTED LOAD IS VERTICAL OR HORIZONTAL
C           THEN CONCENTRATE IT AT THE NODES IN COMPONENTS
C             IDIRCN.EQ.0 , VERTICAL
C             IDIRCN.EQ.1 , HORIZONTAL
C
      IF (IDIRCN.EQ.1) GO TO 350
      IF (IARCH.EQ.1) DX=ABS(X(NJ)-X(NI))
      PN(NI,1)=PN(NI,1)+DX/2.*W*COS(PHII(M))
      PN(NI,3)=PN(NI,3)+DX/2.*W*SIN(PHII(M))
      PN(NJ,1)=PN(NJ,1)+DX/2.*W*COS(PHIJ(M))
      PN(NJ,3)=PN(NJ,3)+DX/2.*W*SIN(PHIJ(M))
      GO TO 400
350   CONTINUE
      PN(NI,2)=PN(NI,2)+W*LENGTH(M)/2.
      PN(NJ,2)=PN(NJ,2)+W*LENGTH(M)/2.
400   CONTINUE
      WRITE(61,2020) (N,(PN(N,I),I=1,8),N=1,NUMNP)
      RETURN
C
1015  FORMAT(I5,8F7.1)
1020  FORMAT(F10.5)
2020  FORMAT(I5,6X,8F10.3)
2030  FORMAT(*-*,38HUNIFORMLY DISTRIBUTED VERTICAL LOAD W=,F10.5//)
2040  FORMAT(*-*,40HUNIFORMLY DISTRIBUTED HORIZONTAL LOAD W=,F10.5//)
C
      END
```

```
      SUBROUTINE ELEMENT (N,IDATA,IARCH)
C
C     ******************************************************************
C        TO CALL THE APPROPRIATE ELEMENT SUBROUTINE
C     ******************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),D12(11)
C
      IF (NTYPE(N).GT.1) GO TO 200
      CALL CURVED (N,IDATA,IARCH)
      RETURN
200   RETURN
C
      END
```

```
      SUBROUTINE BAND
C
C     ******************************************************************
C        TO COMPUTE SEMIBANDWIDTH OF STRUCTURE STIFFNESS MATRIX
C        DONE BY FINDING THE MAXIMUMN DIFFERENCE BETWEEN THE
C        EQUATION NUMBERS ASSOTIATED WITH THE NODES OF A
C        PARTICULAR ELEMENT
C     ******************************************************************
C
      COMMON/1/NE,NUMNP,D1(15)
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(37,8),D3(409)
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),D5(180)
C
      MBAND=0
      DO 900 M=1,NE
      NI=NODEI(M)
      NJ=NODEJ(M)
      DO 800 I=1,8
      IF (IA(NI,I).LE.0) GO TO 800
      N1=IA(NI,I)
      DO 700 J=1,8
      IF (IA(NJ,J).LE.0) GO TO 700
      N2=IA(NJ,J)
      MB=N2-N1
      IF (MB.LT.0) MB=-MB+1
      IF (MB.GT.0) MB=MB+1
      IF (MB.GT.MBAND) MBAND=MB
  700 CONTINUE
  800 CONTINUE
  900 CONTINUE
      WRITE(61,2000) MBAND
```

```
      RETURN
C
2000  FORMAT(*1*,20HSEMIBANDWIDTH MBAND=,I3)
C
      END
```

```
      SUBROUTINE CURVED (N,IDATA,IARCH)
C
C     ******************************************************************
C        CURVED ELEMENT SUBROUTINE
C     ******************************************************************
C
      REAL IXX,IYY,KT,II,JJ,LENGTH
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +         ICAL4,ICAL5,ICAL6,ICAL7
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/4/SE(16,16)
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),A(36),IXX(36),IYY(36),
     +         KT(36),L(1,36)
      COMMON/6/A1,A2,MP,B2(36),B3(36),B4(36)
      COMMON/7/RI(36),RJ(36),PHII(36),PHIJ(36),TETA(36),LENGTH(36)
      COMMON/8/PN(37,8),R(296)
      COMMON/9/S(296,16)
      COMMON/10/D(296),D10(1184)
      COMMON/11/DN(16),U(37,8)
C
      GO TO (100,200,300,400),IPAR
C
C........READ MATERIAL INFORMATION
C
100   WRITE(61,2000) NTYPE(N)
      READ(60,1010) E(N),G(N),DM
      WRITE(61,2020) NUMEL(N),E(N),G(N),DM
C
C........READ ELEMENT AND CROSS SECTION INFORMATION
C
      WRITE(61,2021)
      K=0
```

```
105   READ(60,1020) M,NODEI(M),NODEJ(M),A(M),IXX(M),IYY(M),KT(M)
      WRITE(61,2022) M,A(M),IXX(M),IYY(M),KT(M)
      K=K+1
      L(N,K)=M
      IF(K.NE.NUMEL(N)) GO TO 105
C
C.........READ AND CALCULATE ELEMENT GEOMETRIC PROPERTIES
C
      IF (ICAL2.EQ.0) WRITE(61,2025)
      DO 110 KK=1,K
      M=L(N,KK)
      CALL GEOMTRY (M,IARCH)
110   CONTINUE
      RETURN
C
200   CONTINUE
C
C.........CALCULATE LINEAR STIFFNESS MATRIX OF EACH ELEMENT AND STORE IN
C           ARRAY SE(M,I,J). READ LIMITS OF INTEGRATION AND THE MP-POINTS
C           OF INTEGRATION TO BE USED IN THE GAUSS-LEGENDRE QUADRATURE
C
      IF (IDATA.EQ.0) READ(60,1030) A1,A2,MP
      IF (ICAL1.EQ.0) WRITE(61,2030) A1,A2,MP
C
C.........OBTAIN LINEAR STIFFNESS FOR EACH ELEMENT. INTEGRATE NUMERICALLY
C
      INUMEL=NUMEL(N)
      DO 220 K=1,INUMEL
      M=L(N,K)
      CALL NUMINT (N,M)
      IF(ICAL1.NE.0) GO TO 215
      WRITE(61,2032) M
```

```
      WRITE(61,2034) ((SE(I,J),J=1,8),I=1,8)
      WRITE(61,2036)
      WRITE(61,2034) ((SE(I,J),J=9,16),I=1,8)
      WRITE(61,2038)
      WRITE(61,2034) ((SE(I,J),J=1,8),I=9,16)
      WRITE(61,2040)
      WRITE(61,2034) ((SE(I,J),J=9,16),I=9,16)
215   WRITE(1,10) ((SE(I,J),J=1,16),I=1,16)
220   CONTINUE
      REWIND 1
C
C........ASSEMBLE LINEAR STIFFNESS OF EACH ELEMENT
C           INTO LINEAR STIFFNESS OF STRUCTURE
C
      DO 230 K=1,INUMEL
      M=L(N,K)
      READ(1,10) ((SE(I,J),J=1,16),I=1,16)
      CALL ASEMBLE (M)
230   CONTINUE
      REWIND 1
      REWIND 9
      REWIND 8
      WRITE(4,10) ((S(I,J),J=1,MBAND),I=1,NSIZE)
      REWIND 4
      RETURN
C
300   CONTINUE
C
C........OBTAIN NONLINEAR STIFFNESS -SE1- FOR EACH ELEMENT.
C           INTEGRATE NUMERICALLY.
C
      DO 320 K=1,INUMEL
```

```
      M=L(N,K)
      NI=NODEI(M)
      NJ=NODEJ(M)
      DO 305 ID=1,8
      DN(ID)=U(NI,ID)
  305 DN(ID+8)=U(NJ,ID)
      CALL NUMINT (N,M)
      IF(ICAL1.NE.0) GO TO 315
      WRITE(61,2050) M
      WRITE(61,2034) ((SE(I,J),J=1,8),I=1,8)
      WRITE(61,2036)
      WRITE(61,2034) ((SE(I,J),J=9,16),I=1,8)
      WRITE(61,2038)
      WRITE(61,2034) ((SE(I,J),J=1,8),I=9,16)
      WRITE(61,2040)
      WRITE(61,2034) ((SE(I,J),J=9,16),I=9,16)
  315 WRITE(2,10) ((SE(I,J),J=1,16),I=1,16)
  320 CONTINUE
      REWIND 2
C
C........ASSEMBLE NONLINEAR STIFFNESS SE1 OF EACH ELEMENT
C          INTO NONLINEAR STIFFNESS OF STRUCTURE, S1.
C          STORE MATRIX S1
C
      DO 330 K=1,INUMEL
      M=L(N,K)
      READ(2,10) ((SE(I,J),J=1,16),I=1,16)
      CALL ASEMBLE (M)
  330 CONTINUE
      REWIND 2
      WRITE(5,10) ((S(I,J),J=1,MBAND),I=1,NSIZE)
      REWIND 5
```

```
      RETURN
C
400   CONTINUE
C
C........OBTAIN NONLINEAR STIFFNESS -SE2- FOR EACH ELEMENT.
C          INTEGRATE NUMERICALLY.
C
      DO 420 K=1,INUMEL
      M=L(N,K)
      NI=NODEI(M)
      NJ=NODEJ(M)
      DO 405 ID=1,8
      DN(ID)=U(NI,ID)
405   DN(ID+8)=U(NJ,ID)
      CALL NUMINT (N,M)
      IF(ICAL1.NE.0) GO TO 415
      WRITE(61,2055) M
      WRITE(61,2034) ((SE(I,J),J=1,8),I=1,8)
      WRITE(61,2036)
      WRITE(61,2034) ((SE(I,J),J=9,16),I=1,8)
      WRITE(61,2038)
      WRITE(61,2034) ((SE(I,J),J=1,8),I=9,16)
      WRITE(61,2040)
      WRITE(61,2034) ((SE(I,J),J=9,16),I=9,16)
415   WRITE(3,10) ((SE(I,J),J=1,16),I=1,16)
420   CONTINUE
      REWIND 3
C
C........ASSEMBLE NONLINEAR STIFFNESS SE2 OF EACH ELEMENT
C          INTO NONLINEAR STIFFNESS OF STRUCTURE, S2.
C          STORE MATRIX S2
C
```

```
      DO 430 K=1,INUMEL
      M=L(N,K)
      READ(3,10) ((SE(I,J),J=1,16),I=1,16)
      CALL ASEMBLE (M)
430   CONTINUE
      REWIND 3
      WRITE(6,10) ((S(I,J),J=1,MBAND),I=1,NSIZE)
      REWIND 6
      RETURN
C
10    FORMAT(E21.6)
1010  FORMAT(3E10.2)
1020  FORMAT(3I5,4E15.6)
1030  FORMAT(2F5.2,I5)
2000  FORMAT(*1*,23HG R O U P  N U M B E R ,I2//2X,6HNUMBER,6X,7HMODULUS
     +       ,11X,5HSHEAR,8X,7HDENSITY/4X,2HOF,11X,2HOF,12X,7HMODULUS/
     +       1X,8HELEMENTS,4X,10HELASTICITY)
2020  FORMAT(I6,3E17.6)
2021  FORMAT(//8H ELEMENT,9X,4HA(M),10X,6HIXX(M),9X,6HIYY(M),9X,5HKT(M))
2022  FORMAT(I6,5X,4E15.6)
2025  FORMAT(//8H ELEMENT,3X,8HNODEI(M),3X,8HNODEJ(M),9X,
     +       19HRADIUS OF CURVATURE,18X,17HNODAL SLOPE ANGLE/8H  NUMBER/
     +       39X,5HRI(M),10X,5HRJ(M),15X,7HPHII(M),8X,7HPHIJ(M),7X,
     +       7HTETA(M))
2030  FORMAT(*1*,21HLIMITS OF INTEGRATION,3X,3HA1=,F3.1,3X,3HA2=,F3.1//
     +       26H QUADRATURE FORMULA POINTS,3X,3HMP=,I2)
2032  FORMAT(*1*,44HUNCONDENSED LINEAR STIFFNESS (SE) OF ELEMENT,I3///
     +       9H BLOCK II)
2034  FORMAT(//1X,8F15.6)
2036  FORMAT(///9H BLOCK IJ)
2038  FORMAT(*1*//9H BLOCK JI)
2040  FORMAT(///9H BLOCK JJ)
```

```
2050  FORMAT(*1*,48HUNCONDENSED NONLINEAR STIFFNESS (SE1) OF ELEMENT,I3
     +       ///9H BLOCK II)
2055  FORMAT(*1*,48HUNCONDENSED NONLINEAR STIFFNESS (SE2) OF ELEMENT,I3
     +       ///9H BLOCK II)
C
      END
```

```
      SUBROUTINE GEOMTRY (M,IARCH)
C
C     ******************************************************************
C        TO CALCULATE AND STORE GEOMETRIC PROPERTIES OF CURVED ELEMENTS.
C     ******************************************************************
C
      REAL IXX,IYY,KT,II,JJ,LENGTH
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +         ICAL4,ICAL5,ICAL6,ICAL7
      COMMON/3/IA(37,8),IB(37,8),X(37),Y(37),Z(37),R,AC
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),A(36),IXX(36),IYY(36),
     +         KT(36),L(1,36)
      COMMON/6/A1,A2,MP,B2(36),B3(36),B4(36)
      COMMON/7/RI(36),RJ(36),PHII(36),PHIJ(36),TETA(36),LENGTH(36)
C
      XI=X(NODEI(M))
      YI=Y(NODEI(M))
      XJ=X(NODEJ(M))
      YJ=Y(NODEJ(M))
C
C........READ ELEMENT GEOMETRIC PROPERTIES
C
      IF (IARCH.EQ.0) GO TO 100
      RI(M)=RJ(M)=R
      DY=XI/SQRT(R*R-XI*XI)
      PHII(M)=ATAN(DY)
      DY=XJ/SQRT(R*R-XJ*XJ)
      PHIJ(M)=ATAN(DY)
      GO TO 200
100   CONTINUE
      D2Y=2.*AC
      DY=2.*AC*XI
```

```
      RI(M)=(1.+DY*DY)**1.5/D2Y
      PHII(M)=ATAN(DY)
      DY=2.*AC*XJ
      RJ(M)=(1.+DY*DY)**1.5/D2Y
      PHIJ(M)=ATAN(DY)
200   CONTINUE
      XL=XJ-XI
      YL=YJ-YI
      TETA(M)=ABS(PHII(M)-PHIJ(M))
      T=TETA(M)
C
C........CALCULATE NODAL LOCAL COORDINATES AFTER ROTATION
C
      XR=XL*COS(PHII(M))+YL*SIN(PHII(M))
      ZR=-XL*SIN(PHII(M))+YL*COS(PHII(M))
C
C........CHECK DATA GENERATION
C
      IF (ICAL2.EQ.0) WRITE(61,2010) M,NODEI(M),NODEJ(M),RI(M),RJ(M),
     +                               PHII(M),PHIJ(M),T
C
C........SOLVE SYSTEM OF EQUATIONS BY CRAMER"S RULE (NON-CLOSED FORM,
C        SINGLE PRECISSION), TO OBTAIN VARIABLES B2, B3, B4 AND LENGTH
C        FIRST CALCULATE COEFFICIENTS OF THE VARIABLES
C
      A11=2.*T
      A12=3.*T**2
      A13=4.*T**3
      A21=2.*(SIN(T)-T*COS(T))
      A22=3.*(-T**2*COS(T)+2.*T*SIN(T)+2.*COS(T)-2.)
      A23=4.*(-T**3*COS(T)+3.*T**2*SIN(T)+6.*T*COS(T)-6.*SIN(T))
      A31=2.*(T*SIN(T)+COS(T)-1.)
```

```
      A32=3.*(T**2*SIN(T)+2.*T*COS(T)-2.*SIN(T))
      A33=4.*(T**3*SIN(T)+3.*T**2*COS(T)-6.*T*SIN(T)-6.*COS(T)+6.)
C
C........CALCULATE INDEPENDENT COEFFICIENTS
C
      C1=RJ(M)-RI(M)
      C2=ZR-RI(M)*(1.-COS(T))
      C3=XR-RI(M)*SIN(T)
C
C........CALCULATE DETERMINANTS OF CRAMER"S RULE
C
      D=(A11*A22*A33)+(A21*A32*A13)+(A12*A23*A31)-
     +  (A31*A22*A13)-(A12*A21*A33)-(A32*A23*A11)
      D2=(C1*A22*A33)+(C2*A32*A13)+(A12*A23*C3)-
     +   (C3*A22*A13)-(A12*C2*A33)-(A32*A23*C1)
      D3=(A11*C2*A33)+(A21*C3*A13)+(C1*A23*A31)-
     +   (A31*C2*A13)-(C1*A21*A33)-(C3*A23*A11)
      D4=(A11*A22*C3)+(A21*A32*C1)+(A12*C2*A31)-
     +   (A31*A22*C1)-(A12*A21*C3)-(A32*C2*A11)
C
C........CALCULATE B2,B3,B4,LENGTH
C
      B2(M)=D2/D
      B3(M)=D3/D
      B4(M)=D4/D
      LENGTH(M)=RI(M)*T+B2(M)*T**2+B3(M)*T**3+B4(M)*T**4
C
C........CHECK DATA GENERATION
C
      IF(ICAL2.EQ.0) WRITE(61,2020) XR,ZR,B2(M),B3(M),B4(M),LENGTH(M)
      RETURN
C
```

```
2010  FORMAT(//I6,5X,I5,6X,I5,6X,2F15.6,6X,3F15.6)
2020  FORMAT(/10X,3HXR=,F15.10,3X,3HZR=,F15.10//10X,6HB2(M)=,E15.9,3X,
     +       6HB3(M)=,E15.9,3X,6HB4(M)=,E15.9,3X,10HLENGTH(M)=,F13.6)
C
      END
```

```
      SUBROUTINE NUMINT (N,M)
C
C     *******************************************************************
C        TO INTEGRATE NUMERICALLY THE TERMS OF THE CURVED ELEMENT
C        STIFFNESS MATRICES SE, SE1, SE2, IT USES THE GAUSS-LEGENDRE
C        QUADRATURE FORMULA.
C        THE ROUTINE NUMINT USES THE MP-POINT GAUSS-LEGENDRE QUADRATURE
C        FORMULA TO COMPUTE THE INTEGRAL OF FUNCTN(GM)*DGM BETWEEN
C        INTEGRATION LIMITS A1 AND A2. THE ROOTS OF SEVEN LEGENDRE
C        POLYNOMIALS AND THE WEIGHT FACTORS FOR CORRESPONDING
C        QUADRATURES ARE STORED IN THE Z AND WEIGHT ARRAYS RESPECTIVELY.
C        MP MAY ASSUME VALUES 2, 3, 4, 5, 6, 10, AND 15 ONLY. THE
C        APPROPRIATE VALUES FOR THE MP-POINT FORMULA ARE LOCATED IN
C        ELEMENTS Z(KEY(I))...Z(KEY(I+1)-1) AND WEIGHT(KEY(I))...
C        WEIGHT(KEY(I+1)-1) WHERE THE PROPER VALUE FOR I IS DETERMINED
C        BY FINDING THE SUBSCRIPT OF THE ELEMENT OF THE ARRAY NPOINT
C        WHICH HAS THE VALUE MP. IF AN INVALID VALUE OF MP IS USED, A
C        TRUE ZERO IS RETURNED AS THE VALUE OF GAUSS.
C     *******************************************************************
C
      REAL IXX,IYY,KT,II,JJ,LENGTH,L1,L2,K
      DIMENSION NPOINT(7),KEY(8),Z(24),WEIGHT(24),K(16,16)
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +        ICAL4,ICAL5,ICAL6,ICAL7
      COMMON/4/SE(16,16)
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),A(36),IXX(36),IYY(36),
     +        KT(36),L(1,36)
      COMMON/6/A1,A2,MP,B2(36),B3(36),B4(36)
      COMMON/7/RI(36),RJ(36),PHII(36),PHIJ(36),TETA(36),LENGTH(36)
      COMMON/11/DN(16),U(37,8)
      DATA NPOINT/ 2, 3, 4, 5, 6, 10, 15/
      DATA KEY/ 1, 2, 4, 6, 9, 12, 17, 25/
```

```
      DATA Z              / 0.577350269,0.0        ,0.774596669,
     1          0.339981044,0.861136312,0.0        ,0.538469310,
     2          0.906179846,0.238619186,0.661209387,0.932469514,
     3          0.148874339,0.433395394,0.679409568,0.865063367,
     4          0.973906529,0.0        ,0.201194094,0.394151347,
     5          0.570972173,0.724417731,0.848206583,0.937273392,
     6          0.987992518 /
      DATA WEIGHT         / 1.0        ,0.888888889,0.555555556,
     1          0.652145155,0.347854845,0.568888889,0.478628671,
     2          0.236926885,0.467913935,0.360761573,0.171324493,
     3          0.295524225,0.269266719,0.219086363,0.149451349,
     4          0.066671344,0.202578242,0.198431485,0.186161000,
     5          0.166269206,0.139570678,0.107159221,0.070366047,
     6          0.030753242 /
C
      T=TETA(M)
      R1=RI(M)
      R2=RJ(M)
      L1=R1*T
      L2=R2*T
C
C........FIND SUBSCRIPT OF FIRST Z AND WEIGHT VALUE
C
      DO 100 I=1,7
      IF(MP.EQ.NPOINT(I)) GO TO 200
100   CONTINUE
C
C........INVALID MP USED
C
      GAUSS=0.0
      WRITE(61,2000) GAUSS
      RETURN
```

```
C
C.........SET UP INITIAL PARAMETERS
C
200   JFIRST=KEY(I)
      JLAST=KEY(I+1)-1
      C=(A2-A1)/2.
      D=(A2+A1)/2.
C
C.........ACCUMULATE THE SUM IN THE MP-POINT FORMULA
C
      DO 250 I=1,16
      DO 250 J=I,16
250   K(I,J)=K(J,I)=0.
      DO 500 J=JFIRST,JLAST
      I=0
      IF (Z(J).EQ.0.) GO TO 350
300   I=I+1
      IF (I.EQ.1) GM=Z(J)*C+D
      IF (I.EQ.2) GM=-Z(J)*C+D
      GO TO 360
350   GM=D
360   AA=6.*GM**2-6.*GM
      BB=3.*GM**2-4.*GM+1.
      CC=3.*GM**2-2.*GM
      DD=12.*GM-6.
      EE=6.*GM-4.
      FF=6.*GM-2.
      GG=2.*GM**3-3.*GM**2+1.
      HH=GM**3-2.*GM**2+GM
      II=-2.*GM**3+3.*GM**2
      JJ=GM**3-GM**2
      R=R1+2.*B2(M)*T*GM+3.*B3(M)*T**2*GM**2+4.*B4(M)*T**3*GM**3
```

```
      GMSS=(-1./(R**3*T))*(2.*B2(M)+6.*B3(M)*T*GM+12.*B4(M)*T**2*GM**2)
      GMSG=R*T*GMSS
C
C........CHECK WHICH PART OF THE STIFFNESS MATRIX IS BEING COMPUTED
C                  IPAR=2, COMPUTE ARRAY SE
C                  IPAR=3, COMPUTE ARRAY SE1
C                  IPAR=4, COMPUTE ARRAY SE2
C
      GO TO (370,370,390,410),IPAR
C
370   CONTINUE
C
C........INTEGRANDS OF CURVED ELEMENT LINEAR STIFFNESS (SYMMETRIC)
C
      C1=-E(N)*A(M)*GG/R
      C2=(E(N)*IYY(M)/(R**3*T**3))*(DD+AA*GMSS*R**2*T**2)
      C3=(E(N)*IXX(M)*T/R**3)*(-DD/T**2-GMSS*R**2*AA)
      C4=G(N)*KT(M)*AA/(R**3*T)
      C5=(E(N)*A(M)/(R*T))*(AA+T**2*HH)
      C6=(E(N)*IYY(M)/(R**3*T**3))*(-T*EE-GMSS*R**2*T**3*BB+T*AA+
     +   GMSG*R*T**2*GG)
      C7=E(N)*IXX(M)*T*GG/R**2
      C8=G(N)*KT(M)*AA/(R**2*T)
      C9=-E(N)*A(M)*L1*HH/R
      C10=(E(N)*IYY(M)/(R**3*T**3))*(L1*EE+GMSS*R**2*T**2*L1*BB)
      C11=(E(N)*IXX(M)*T/R**3)*(L1*EE/T**2+GMSS*R**2*L1*BB)
      C12=-G(N)*KT(M)*L1*BB/(R**3*T)
      C13=E(N)*A(M)*R1*BB/(R*T)
      C14=(E(N)*IYY(M)/(R**3*T**3))*(T*R1*BB+GMSG*R*T**2*R1*HH)
      C15=E(N)*IXX(M)*T*L1*HH/R**2
      C16=G(N)*KT(M)*L1*BB/(R**2*T)
      C17=-E(N)*A(M)*II/R
```

```
      C18=(E(N)*IYY(M)/(R**3*T**3))*(-DD-GMSS*R**2*T**2*AA)
      C19=-C3
      C20=-C4
      C21=(E(N)*A(M)/(R*T))*(-AA+T**2*JJ)
      C22=(E(N)*IYY(M)/(R**3*T**3))*(-T*FF-GMSS*R**2*T**3*CC-T*AA+
     +    GMSG*R*T**2*II)
      C23=E(N)*IXX(M)*T*II/R**2
      C24=-C8
      C25=(-E(N)*A(M)*L2*JJ)/R
      C26=(E(N)*IYY(M)/(R**3*T**3))*(L2*FF+GMSS*R**2*T**2*L2*CC)
      C27=(E(N)*IXX(M)*T/R**3)*(L2*FF/T**2+GMSS*R**2*L2*CC)
      C28=-G(N)*KT(M)*L2*CC/(R**3*T)
      C29=E(N)*A(M)*R2*CC/(R*T)
      C30=(E(N)*IYY(M)/(R**3*T**3))*(T*R2*CC+GMSG*R*T**2*R2*JJ)
      C31=E(N)*IXX(M)*T*L2*JJ/R**2
      C32=G(N)*KT(M)*L2*CC/(R**2*T)
C
        SE(1,1)=C1*T*(-GG)+C2*(DD+AA*GMSS*R**2*T**2)
        SE(1,2)=SE(1,4)=SE(1,6)=SE(1,8)=SE(1,10)=SE(1,12)=SE(1,14)=
     +          SE(1,16)=0.0
        SE(1,3)=C1*(AA+T**2*HH)+C2*(-T*EE-GMSS*R**2*T**3*BB+T*AA+
     +          GMSG*R*T**2*GG)
        SE(1,5)=C1*T*(-L1)*HH+C2*L1*(EE+BB*GMSS*R**2*T**2)
        SE(1,7)=C1*R1*BB+C2*(T*R1*BB+GMSG*R*T**2*R1*HH)
        SE(1,9)=C1*(-T*II)-C2*(DD+AA*GMSS*R**2*T**2)
        SE(1,11)=C1*(-AA+T**2*JJ)+C2*(-T*FF-GMSS*R**2*T**3*CC-T*AA+
     +           GMSG*R*T**2*II)
        SE(1,13)=C1*(-T*L2*JJ)+C2*(L2*FF+GMSS*R**2*T**2*L2*CC)
        SE(1,15)=C1*(R2*CC)+C2*(T*R2*CC+GMSG*R*T**2*JJ)
C
        SE(2,2)=C3*(-DD/T**2-GMSS*R**2*AA)+C4*AA
        SE(2,3)=SE(2,5)=SE(2,7)=SE(2,9)=SE(2,11)=SE(2,13)=SE(2,15)=0.0
```

```
      SE(2,4)=C3*R*GG+C4*R*AA
      SE(2,6)=C3*(L1*EE/T**2+GMSS*R**2*L1*BB)-C4*L1*BB
      SE(2,8)=C3*R*L1*HH+C4*R*L1*BB
      SE(2,10)=C3*(DD/T**2+GMSS*R**2*AA)-C4*AA
      SE(2,12)=C3*R*II-C4*R*AA
      SE(2,14)=C3*(L2*FF/T**2+GMSS*R**2*L2*CC)-C4*L2*CC
      SE(2,16)=C3*R*L2*JJ+C4*R*L2*CC
C
      SE(3,3)=C5*(AA+T**2*HH)+C6*(-T*EE-GMSS*R**2*T**3*BB+T*AA+
     +         GMSG*R*T**2*GG)
      SE(3,4)=SE(3,6)=SE(3,8)=SE(3,10)=SE(3,12)=SE(3,14)=SE(3,16)=0.0
      SE(3,5)=C5*(-T*L1*HH)+C6*(L1*EE+GMSS*R**2*T**2*L1*BB)
      SE(3,7)=C5*R1*BB+C6*(T*R1*BB+GMSG*R*T**2*R1*HH)
      SE(3,9)=C5*(-T*II)+C6*(-DD-GMSS*R**2*T**2*AA)
      SE(3,11)=C5*(-AA+T**2*JJ)+C6*(-T*FF-GMSS*R**2*T**3*CC-T*AA+
     +          GMSG*R*T**2*II)
      SE(3,13)=C5*(-T*L2*JJ)+C6*(L2*FF+GMSS*R**2*T**2*L2*CC)
      SE(3,15)=C5*R2*CC+C6*(T*R2*CC+GMSG*R*T**2*JJ)
C
      SE(4,4)=C7*R*GG+C8*R*AA
      SE(4,5)=SE(4,7)=SE(4,9)=SE(4,11)=SE(4,13)=SE(4,15)=0.0
      SE(4,6)=C7*(L1*EE/T**2+GMSS*R**2*L1*BB)+C8*(-L1*BB)
      SE(4,8)=C7*R*L1*HH+C8*R*L1*BB
      SE(4,10)=C7*(DD/T**2+GMSS*R**2*AA)+C8*(-AA)
      SE(4,12)=C7*R*II+C8*(-R*AA)
      SE(4,14)=C7*(L2*FF/T**2+GMSS*R**2*L2*CC)+C8*(-L2*CC)
      SE(4,16)=C7*R*L2*JJ+C8*R*L2*CC
C
      SE(5,5)=C9*(-T*L1*HH)+C10*(L1*EE+GMSS*R**2*T**2*L1*BB)
      SE(5,6)=SE(5,8)=SE(5,10)=SE(5,12)=SE(5,14)=SE(5,16)=0.0
      SE(5,7)=C9*R1*BB+C10*(T*R1*BB+GMSG*R*T**2*R1*HH)
      SE(5,9)=C9*(-T*II)+C10*(-DD-GMSS*R**2*T**2*AA)
```

```
      SE(5,11)=C9*(-AA+T**2*JJ)+C10*(-T*FF-GMSS*R**2*T**3*CC-T*AA+
     +        GMSG*R*T**2*II)
      SE(5,13)=C9*(-T*L2*JJ)+C10*(L2*FF+GMSS*R**2*T**2*L2*CC)
     SE(5,15)=C9*R2*CC+C10*(T*R2*CC+GMSG*R*T**2*R2*JJ)
C
      SE(6,6)=C11*(L1*EE/T**2+GMSS*R**2*L1*BB)+C12*(-L1*BB)
      SE(6,7)=SE(6,9)=SE(6,11)=SE(6,13)=SE(6,15)=0.0
      SE(6,8)=C11*R*L1*HH+C12*R*L1*BB
      SE(6,10)=C11*(DD/T**2+GMSS*R**2*AA)+C12*(-AA)
      SE(6,12)=C11*R*II+C12*(-R*AA)
      SE(6,14)=C11*(L2*FF/T**2+GMSS*R**2*L2*CC)+C12*(-L2*CC)
      SE(6,16)=C11*R*L2*JJ+C12*R*L2*CC
C
      SE(7,7)=C13*R1*BB+C14*(T*R1*BB+GMSG*R*T**2*R1*HH)
      SE(7,8)=SE(7,10)=SE(7,12)=SE(7,14)=SE(7,16)=0.0
      SE(7,9)=C13*(-T*II)+C14*(-DD-GMSS*R**2*T**2*AA)
      SE(7,11)=C13*(-AA+T**2*JJ)+C14*(-T*FF-GMSS*R**2*T**3*CC-T*AA+
     +        GMSG*R*T**2*II)
      SE(7,13)=C13*(-T*L2*JJ)+C14*(L2*FF+GMSS*R**2*T**2*L2*CC)
      SE(7,15)=C13*R2*CC+C14*(T*R2*CC+GMSG*R*T**2*R2*JJ)
C
      SE(8,8)=C15*R*L1*HH+C16*R*L1*BB
      SE(8,9)=SE(8,11)=SE(8,13)=SE(8,15)=0.0
      SE(8,10)=C15*(DD/T**2+GMSS*R**2*AA)+C16*(-AA)
      SE(8,12)=C15*R*II+C16*(-R*AA)
      SE(8,14)=C15*(L2*FF/T**2+GMSS*R**2*L2*CC)+C16*(-L2*CC)
      SE(8,16)=C15*R*L2*JJ+C16*R*L2*CC
C
      SE(9,9)=C17*T*(-II)+C18*(-DD-GMSS*R**2*T**2*AA)
      SE(9,10)=SE(9,12)=SE(9,14)=SE(9,16)=0.0
      SE(9,11)=C17*(-AA+T**2*JJ)+C18*(-T*FF-GMSS*R**2*T**3*CC-T*AA+
     +        GMSG*R*T**2*II)
```

```
      SE(9,13)=C17*(-T*L2*JJ)+C18*(L2*FF+GMSS*R**2*T**2*L2*CC)
      SE(9,15)=C17*R2*CC+C18*(T*R2*CC+GMSG*R*T**2*JJ)
C
      SE(10,10)=C19*(DD/T**2+GMSS*R**2*AA)+C20*(-AA)
      SE(10,11)=SE(10,13)=SE(10,15)=0.0
      SE(10,12)=C19*R*II+C20*(-R*AA)
      SE(10,14)=C19*(L2*FF/T**2+GMSS*R**2*L2*CC)+C20*(-L2*CC)
      SE(10,16)=C19*R*L2*JJ+C20*R*L2*CC
C
      SE(11,11)=C21*(-AA+T**2*JJ)+C22*(-T*FF-GMSS*R**2*T**3*CC-T*AA+
     +          GMSG*R*T**2*II)
      SE(11,12)=SE(11,14)=SE(11,16)=0.0
      SE(11,13)=C21*(-T*L2*JJ)+C22*(L2*FF+GMSS*R**2*T**2*L2*CC)
      SE(11,15)=C21*R2*CC+C22*(T*R2*CC+GMSG*R*T**2*R2*JJ)
C
      SE(12,12)=C23*R*II+C24*(-R*AA)
      SE(12,13)=SE(12,15)=0.0
      SE(12,14)=C23*(L2*FF/T**2+GMSS*R**2*L2*CC)+C24*(-L2*CC)
      SE(12,16)=C23*R*L2*JJ+C24*R*L2*CC
C
      SE(13,13)=C25*(-T*L2*JJ)+C26*(L2*FF+GMSS*R**2*T**2*L2*CC)
      SE(13,14)=SE(13,16)=0.0
      SE(13,15)=C25*R2*CC+C26*(T*R2*CC+GMSG*R*T**2*R2*JJ)
C
      SE(14,14)=C27*(L2*FF/T**2+GMSS*R**2*L2*CC)+C28*(-L2*CC)
      SE(14,15)=0.0
      SE(14,16)=C27*R*L2*JJ+C28*R*L2*CC
C
      SE(15,15)=C29*R2*CC+C30*(T*R2*CC+GMSG*R*T**2*R2*JJ)
      SE(15,16)=0.0
C
      SE(16,16)=C31*R*L2*JJ+C32*R*L2*CC
```

```
C
      DO 380 IE=1,16
      DO 380 JE=IE,16
  380 K(IE,JE)=K(IE,JE)+WEIGHT(J)*SE(IE,JE)
      IF (I.EQ.1) GO TO 300
      GO TO 500
C
  390 CONTINUE
C
C........INTEGRANDS OF CURVED ELEMENT NONLINEAR STIFFNESS SE1
C
      UD=GG*DN(1)+HH*(L1*DN(5)-T*DN(3))+II*DN(9)+JJ*(L2*DN(13)-T*DN(11))
      WD=GG*DN(3)+HH*R1*DN(7)+II*DN(11)+JJ*R2*DN(15)
      UG=AA*(DN(1)-DN(9))+BB*(L1*DN(5)-T*DN(3))+CC*(L2*DN(13)-T*DN(11))
      VG=AA*(DN(2)-DN(10))-BB*L1*DN(6)-CC*L2*DN(14)
      WG=AA*(DN(3)-DN(11))+BB*R1*DN(7)+CC*R2*DN(15)
C
      C1=-E(N)*A(M)*T*GG/(R**2*T**2)
      C2=E(N)*A(M)*AA/(R**2*T**2)
      C3=UG+T*WD
      C4=WG-T*UD
C
      SE(1,1)=C1*C3*AA+C2*(C3*(-T*GG)+C4*AA)
      SE(1,2)=C1*VG*AA
      SE(1,3)=C1*C3*(-T*BB+T*GG)+C2*(C3*(AA+T**2*HH)+C4*(-T*BB+T*GG))
      SE(1,4)=0.0
      SE(1,5)=C1*C3*L1*BB+C2*(C3*(-T*L1*HH)+C4*L1*BB)
      SE(1,6)=C1*VG*(-L1*BB)
      SE(1,7)=C1*C3*T*R1*HH+C2*(C3*R1*BB+C4*T*R1*HH)
      SE(1,8)=0.0
      SE(1,9)=C1*C3*(-AA)+C2*(C3*(-T*II)+C4*(-AA))
      SE(1,10)=-SE(1,2)
```

```
      SE(1,11)=C1*C3*(-T*CC+T*II)+C2*(C3*(-AA+T**2*JJ)+C4*(-T*CC+T*II))
      SE(1,12)=0.0
      SE(1,13)=C1*C3*L2*CC+C2*(C3*(-T*L2*JJ)+C4*L2*CC)
      SE(1,14)=C1*VG*(-L2*CC)
      SE(1,15)=C1*C3*T*R2*JJ+C2*(C3*R2*CC+C4*T*R2*JJ)
      SE(1,16)=0.0
C
      C5=(E(N)*A(M)*AA/(R**2*T**2))*C4
      C6=(E(N)*A(M)*AA/(R**2*T**2))*VG
C
      SE(2,2)=C5*AA
      SE(2,3)=C6*(AA+T**2*HH)
      SE(2,4)=0.0
      SE(2,5)=C6*(-T*L1*HH)
      SE(2,6)=C5*(-L1*BB)
      SE(2,7)=C6*R1*BB
      SE(2,8)=0.0
      SE(2,9)=C6*(-T*II)
      SE(2,10)=-SE(2,2)
      SE(2,11)=C6*(-AA+T**2*JJ)
      SE(2,12)=0.0
      SE(2,13)=C6*(-T*L2*JJ)
      SE(2,14)=C5*(-L2*CC)
      SE(2,15)=C6*R2*CC
      SE(2,16)=0.0
C
      C7=(E(N)*A(M)*(AA+T**2*HH)/(R**2*T**2))*C3
      C8=E(N)*A(M)*(-T*BB+T*GG)/(R**2*T**2)
      C9=(E(N)*A(M)*(AA+T**2*HH)/(R**2*T**2))*VG
C
      SE(3,3)=C7*(-T*BB+T*GG)+C8*(C3*(AA+T**2*HH)+C4*(-T*BB+T*GG))
      SE(3,4)=0.0
```

```
      SE(3,5)=C7*L1*BB+C8*(C3*(-T*L1*HH)+C4*L1*BB)
      SE(3,6)=C9*(-L1*BB)
      SE(3,7)=C7*T*R1*HH+C8*(C3*R1*BB+C4*T*R1*HH)
      SE(3,8)=0.0
      SE(3,9)=C7*(-AA)+C8*(C3*(-T*II)+C4*(-AA))
      SE(3,10)=C9*(-AA)
      SE(3,11)=C7*(-T*CC+T*II)+C8*(C3*(-AA+T**2*JJ)+C4*(-T*CC+T*II))
      SE(3,12)=0.0
      SE(3,13)=C7*L2*CC+C8*(C3*(-T*L2*JJ)+C4*L2*CC)
      SE(3,14)=C9*(-L2*CC)
      SE(3,15)=C7*T*R2*JJ+C8*(C3*R2*CC+C4*T*R2*JJ)
      SE(3,16)=0.0
C
      SE(4,4)=SE(4,5)=SE(4,6)=SE(4,7)=SE(4,8)=SE(4,9)=SE(4,10)=0.0
      SE(4,11)=SE(4,12)=SE(4,13)=SE(4,14)=SE(4,15)=SE(4,16)=0.0
C
      C10=E(N)*A(M)*(-T*L1*HH)/(R**2*T**2)
      C11=C10*C3
      C12=E(N)*A(M)*L1*BB/(R**2*T**2)
      C13=C10*VG
C
      SE(5,5)=C11*L1*BB+C12*(C3*(-T*L1*HH)+C4*L1*BB)
      SE(5,6)=C13*(-L1*BB)
      SE(5,7)=C11*T*R1*HH+C12*(C3*R1*BB+C4*T*R1*HH)
      SE(5,8)=0.0
      SE(5,9)=C11*(-AA)+C12*(C3*(-T*II)+C4*(-AA))
      SE(5,10)=-SE(2,5)
      SE(5,11)=C11*(-T*CC+T*II)+C12*(C3*(-AA+T**2*JJ)+C4*(-T*CC+T*II))
      SE(5,12)=0.0
      SE(5,13)=C11*L2*CC+C12*(C3*(-T*L2*JJ)+C4*L2*CC)
      SE(5,14)=C13*(-L2*CC)
      SE(5,15)=C11*T*R2*JJ+C12*(C3*R2*CC+C4*T*R2*JJ)
```

```
      SE(5,16)=0.0
C
      C14=-C12*C4
      C15=-C12*VG
C
      SE(6,6)=C14*(-L1*BB)
      SE(6,7)=C15*R1*BB
      SE(6,8)=0.0
      SE(6,9)=C15*(-T*II)
      SE(6,10)=-SE(2,6)
      SE(6,11)=C15*(-AA+T**2*JJ)
      SE(6,12)=0.0
      SE(6,13)=C15*(-T*L2*JJ)
      SE(6,14)=C14*(-L2*CC)
      SE(6,15)=C15*R2*CC
      SE(6,16)=0.0
C
      C16=E(N)*A(M)*R1*BB/(R**2*T**2)
      C17=C16*C3
      C18=E(N)*A(M)*T*R1*HH/(R**2*T**2)
      C19=C16*VG
C
      SE(7,7)=C17*T*R1*HH+C18*(C3*R1*BB+C4*T*R1*HH)
      SE(7,8)=0.0
      SE(7,9)=C17*(-AA)+C18*(C3*(-T*II)+C4*(-AA))
      SE(7,10)=C19*(-AA)
      SE(7,11)=C17*(-T*CC+T*II)+C18*(C3*(-AA+T**2*JJ)+C4*(-T*CC+T*II))
      SE(7,12)=0.0
      SE(7,13)=C17*L2*CC+C18*(C3*(-T*L2*JJ)+C4*L2*CC)
      SE(7,14)=C19*(-L2*CC)
      SE(7,15)=C17*T*R2*JJ+C18*(C3*R2*CC+C4*T*R2*JJ)
      SE(7,16)=0.0
```

```
C
      SE(8,8)=SE(8,9)=SE(8,10)=SE(8,11)=SE(8,12)=0.0
      SE(8,13)=SE(8,14)=SE(8,15)=SE(8,16)=0.0
C
      C20=E(N)*A(M)*(-T*II)/(R**2*T**2)
      C21=C20*C3
      C22=E(N)*A(M)*AA/(R**2*T**2)
      C23=C20*VG
C
      SE(9,9)=C21*(-AA)-C22*(C3*(-T*II)+C4*(-AA))
      SE(9,10)=-SE(2,9)
      SE(9,11)=C21*(-T*CC+T*II)-C22*(C3*(-AA+T**2*JJ)+C4*(-T*CC+T*II))
      SE(9,12)=0.0
      SE(9,13)=C21*L2*CC-C22*(C3*(-T*L2*JJ)+C4*L2*CC)
      SE(9,14)=C23*(-L2*CC)
      SE(9,15)=C21*T*R2*JJ-C22*(C3*R2*CC+C4*T*R2*JJ)
      SE(9,16)=0.0
C
      SE(10,10)=SE(2,2)
      SE(10,11)=-SE(2,11)
      SE(10,12)=0.0
      SE(10,13)=-SE(2,13)
      SE(10,14)=-SE(2,14)
      SE(10,15)=-SE(2,15)
      SE(10,16)=0.0
C
      C25=E(N)*A(M)*(-AA+T**2*JJ)/(R**2*T**2)
      C26=E(N)*A(M)*(-T*CC+T*II)/(R**2*T**2)
C
      SE(11,11)=C25*C3*T*(-CC+II)+C26*(C3*(-AA+T**2*JJ)+C4*T*(-CC+II))
      SE(11,12)=0.0
      SE(11,13)=C25*C3*L2*CC+C26*(C3*(-T*L2*JJ)+C4*L2*CC)
```

```
      SE(11,14)=C25*VG*(-L2*CC)
      SE(11,15)=C25*C3*T*R2*JJ+C26*(C3*R2*CC+C4*T*R2*JJ)
      SE(11,16)=0.0
C
      SE(12,12)=SE(12,13)=SE(12,14)=SE(12,15)=SE(12,16)=0.0
C
      C27=E(N)*A(M)*(-T*L2*JJ)/(R**2*T**2)
      C28=E(N)*A(M)*L2*CC/(R**2*T**2)
C
      SE(13,13)=C27*C3*L2*CC+C28*(C3*(-T*L2*JJ)+C4*L2*CC)
      SE(13,14)=C27*VG*(-L2*CC)
      SE(13,15)=C27*C3*T*R2*JJ+C28*(C3*R2*CC+C4*T*R2*JJ)
      SE(13,16)=0.0
C
      C29=E(N)*A(M)*(-L2*CC)/(R**2*T**2)
C
      SE(14,14)=C29*C4*(-L2*CC)
      SE(14,15)=C29*VG*R2*CC
      SE(14,16)=0.0
C
      C30=E(N)*A(M)*R2*CC/(R**2*T**2)
      C31=E(N)*A(M)*T*R2*JJ/(R**2*T**2)
C
      SE(15,15)=C30*C3*T*R2*JJ+C31*(C3*R2*CC+C4*T*R2*JJ)
      SE(15,16)=0.0
C
      SE(16,16)=0.0
C
      DO 400 IE=1,16
      DO 400 JE=IE,16
  400 K(IE,JE)=K(IE,JE)+WEIGHT(J)*SE(IE,JE)
      IF (I.EQ.1) GO TO 300
```

```
      GO TO 500
C
410   CONTINUE
C
C.........INTEGRANDS ON CURVED ELEMENT NONLINEAR STIFFNESS SE2
C
      WD=GG*DN(3)+HH*R1*DN(7)+II*DN(11)+JJ*R2*DN(15)
      UG=AA*(DN(1)-DN(9))+BB*(L1*DN(5)-T*DN(3))+CC*(L2*DN(13)-T*DN(11))
      VG=AA*(DN(2)-DN(10))-BB*L1*DN(6)-CC*L2*DN(14)
C
      C1=E(N)*A(M)*AA/(2.*R**3*T**3)
      C2=3.*(UG+T*WD)**2+VG**2
      C3=2.*VG*(UG+T*WD)
C
      SE(1,1)=C1*C2*AA
      SE(1,2)=C1*C3*AA
      SE(1,3)=C1*C2*(-T*BB+T*GG)
      SE(1,4)=0.0
      SE(1,5)=C1*C2*L1*BB
      SE(1,6)=C1*C3*(-L1*BB)
      SE(1,7)=C1*C2*T*R1*HH
      SE(1,8)=0.0
      SE(1,9)=-SE(1,1)
      SE(1,10)=-SE(1,2)
      SE(1,11)=C1*C2*(-T*CC+T*II)
      SE(1,12)=0.0
      SE(1,13)=C1*C2*L2*CC
      SE(1,14)=C1*C3*(-L2*CC)
      SE(1,15)=C1*C2*T*R2*JJ
      SE(1,16)=0.0
C
      C4=3.*VG**2+(UG+T*WD)**2
```

```
C
      SE(2,2)=C1*C4*AA
      SE(2,3)=C1*C3*(-T*BB+T*GG)
      SE(2,4)=0.0
      SE(2,5)=-SE(1,6)
      SE(2,6)=C1*C4*(-L1*BB)
      SE(2,7)=C1*C3*T*R1*HH
      SE(2,8)=0.0
      SE(2,9)=-SE(1,2)
      SE(2,10)=-SE(2,2)
      SE(2,11)=C1*C3*(-T*CC+T*II)
      SE(2,12)=0.0
      SE(2,13)=-SE(1,14)
      SE(2,14)=C1*C4*(-L2*CC)
      SE(2,15)=C1*C3*T*R2*JJ
      SE(2,16)=0.0
C
      C5=E(N)*A(M)*T*(-BB+GG)/(2.*R**3*T**3)
C
      SE(3,3)=C5*C2*T*(-BB+GG)
      SE(3,4)=0.0
      SE(3,5)=C5*C2*L1*BB
      SE(3,6)=C5*C3*(-L1*BB)
      SE(3,7)=C5*C2*T*R1*HH
      SE(3,8)=0.0
      SE(3,9)=-SE(1,3)
      SE(3,10)=-SE(2,3)
      SE(3,11)=C5*C2*T*(-CC+II)
      SE(3,12)=0.0
      SE(3,13)=C5*C2*L2*CC
      SE(3,14)=C5*C3*(-L2*CC)
      SE(3,15)=C5*C2*T*R2*JJ
```

```
      SE(3,16)=0.0
C
      SE(4,4)=SE(4,5)=SE(4,6)=SE(4,7)=SE(4,8)=SE(4,9)=SE(4,10)=0.0
      SE(4,11)=SE(4,12)=SE(4,13)=SE(4,14)=SE(4,15)=SE(4,16)=0.0
C
      C6=E(N)*A(M)*L1*BB/(2.*R**3*T**3)
C
      SE(5,5)=C6*C2*L1*BB
      SE(5,6)=C6*C3*(-L1*BB)
      SE(5,7)=C6*C2*T*R1*HH
      SE(5,8)=0.0
      SE(5,9)=C6*C2*(-AA)
      SE(5,10)=C6*C3*(-AA)
      SE(5,11)=C6*C2*(-T*CC+T*II)
      SE(5,12)=0.0
      SE(5,13)=C6*C2*L2*CC
      SE(5,14)=C6*C3*(-L2*CC)
      SE(5,15)=C6*C2*T*R2*JJ
      SE(5,16)=0.0
C
      SE(6,6)=C6*C4*L1*BB
      SE(6,7)=-C6*C3*T*R1*HH
      SE(6,8)=0.0
      SE(6,9)=-SE(5,10)
      SE(6,10)=-SE(2,6)
      SE(6,11)=-C6*C3*(-T*CC+T*II)
      SE(6,12)=0.0
      SE(6,13)=SE(5,14)
      SE(6,14)=C6*C4*L2*CC
      SE(6,15)=-C6*C3*T*R2*JJ
      SE(6,16)=0.0
C
```

```
      C7=E(N)*A(M)*T*R1*HH/(2.*R**3*T**3)
C
      SE(7,7)=C7*C2*T*R1*HH
      SE(7,8)=0.0
      SE(7,9)=-SE(1,7)
      SE(7,10)=-SE(2,7)
      SE(7,11)=C7*C2*(-T*CC+T*II)
      SE(7,12)=0.0
      SE(7,13)=C7*C2*L2*CC
      SE(7,14)=C7*C3*(-L2*CC)
      SE(7,15)=C7*C2*T*R2*JJ
      SE(7,16)=0.0
C
      SE(8,8)=SE(8,9)=SE(8,10)=SE(8,11)=SE(8,12)=0.0
      SE(8,13)=SE(8,14)=SE(8,15)=SE(8,16)=0.0
C
      SE(9,9)=SE(1,1)
      SE(9,10)=SE(1,2)
      SE(9,11)=-SE(1,11)
      SE(9,12)=0.0
      SE(9,13)=-SE(1,13)
      SE(9,14)=-SE(1,14)
      SE(9,15)=-SE(1,15)
      SE(9,16)=0.0
C
      SE(10,10)=SE(2,2)
      SE(10,11)=-SE(2,11)
      SE(10,12)=0.0
      SE(10,13)=-SE(2,13)
      SE(10,14)=-SE(2,14)
      SE(10,15)=-SE(2,15)
      SE(10,16)=0.0
```

```
C
      C8=E(N)*A(M)*(-T*CC+T*II)/(2.*R**3*T**3)
C
      SE(11,11)=C8*C2*(-T*CC+T*II)
      SE(11,12)=0.0
      SE(11,13)=C8*C2*L2*CC
      SE(11,14)=C8*C3*(-L2*CC)
      SE(11,15)=C8*C2*T*R2*JJ
      SE(11,16)=0.0
C
      SE(12,12)=SE(12,13)=SE(12,14)=SE(12,15)=SE(12,16)=0.0
C
      C9=E(N)*A(M)*L2*CC/(2.*R**3*T**3)
C
      SE(13,13)=C9*C2*L2*CC
      SE(13,14)=C9*C3*(-L2*CC)
      SE(13,15)=C9*C2*T*R2*JJ
      SE(13,16)=0.0
C
      SE(14,14)=C9*C4*L2*CC
      SE(14,15)=-C9*C3*T*R2*JJ
      SE(14,16)=0.0
C
      C10=E(N)*A(M)*T*R2*JJ/(2.*R**3*T**3)
C
      SE(15,15)=C10*C2*T*R2*JJ
      SE(15,16)=0.0
C
      SE(16,16)=0.0
C
      DO 420 IE=1,16
      DO 420 JE=IE,16
```

```
420   K(IE,JE)=K(IE,JE)+WEIGHT(J)*SE(IE,JE)
      IF (I.EQ.1) GO TO 300
500   CONTINUE
C
C........MAKE INTERVAL CORRECTION AND RETURN
C
      DO 550 I=1,16
      DO 550 J=I,16
      SE(I,J)=C*K(I,J)
550   SE(J,I)=SE(I,J)
      RETURN
C
2000  FORMAT(*1*,15HINVALID MP USED///7H GAUSS=,F4.1)
C
      END
```

```
      SUBROUTINE ASEMBLE (M)
C
C     ******************************************************************
C        TO PROCESS AND ASSEMBLE ELEMENT STIFFNESS MATRICES AND NODAL
C        LOAD VECTORS INTO THEIR CORRESPONDING STRUCTURE ARRAYS.
C     ******************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +         ICAL4,ICAL5,ICAL6,ICAL7
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(37,8),IB(37,8),D31(113)
      COMMON/4/SE(16,16)
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),D5(180)
      COMMON/8/PN(37,8),R(296)
      COMMON/9/S(296,16)
C
      IF (IPAR.NE.1) GO TO 90
C
C........PROCESSING OF INITIAL LOADS AND NODAL LOADS INTO LOAD VECTOR
C
      IF(ICAL4.EQ.0) WRITE(61,2000)
      DO 80 N=1,NUMNP
      DO 70 I=1,8
      IF(IA(N,I)) 20,70,10
10    II=IA(N,I)
      GO TO 60
20    IF (IB(N,I).LT.0) GO TO 30
      NN=IB(N,I)
      GO TO 35
30    II=-IB(N,I)+NEQ
      GO TO 60
35    IF (IA(NN,I)) 40,70,50
```

```
40     II=-IB(NN,I)+NEQ
       GO TO 60
50     II=IA(NN,I)
60     R(II)=PN(N,I)
       IF(ICAL4.EQ.0) WRITE(61,2010) II,N,I,R(II)
70     CONTINUE
80     CONTINUE
       RETURN
C
C........ASSEMBLE ELEMENT STIFFNESS INTO STRUCTURE STIFFNESS
C
90     NI=NODEI(M)
       NJ=NODEJ(M)
       DO 165 K1=1,2
       IF(K1.EQ.1) NP=NI
       IF(K1.EQ.2) NP=NJ
       DO 160 I=1,8
       IF(IA(NP,I)) 105,160,100
100    II=IA(NP,I)
       GO TO 115
105    IF (IB(NP,I).LT.0) GO TO 110
       NN=IB(NP,I)
       GO TO 111
110    II=-IB(NP,I)+NEQ
       GO TO 115
111    IF (IA(NN,I)) 112,160,113
112    II=-IB(NN,I)+NEQ
       GO TO 115
113    II=IA(NN,I)
115    CONTINUE
       DO 155 K2=1,2
       IF(K2.EQ.1) ND=NI
```

```
      IF(K2.EQ.2) ND=NJ
      DO 150 J=1,8
      IF(IA(ND,J)) 125,150,120
120   JJ=IA(ND,J)
      GO TO 145
125   IF (IB(ND,J).LT.0) GO TO 130
      NN=IB(ND,J)
      GO TO 132
130   JJ=-IB(ND,J)+NEQ
      GO TO 145
132   IF (IA(NN,J)) 135,150,140
135   JJ=-IB(NN,J)+NEQ
      GO TO 145
140   JJ=IA(NN,I)
145   CONTINUE
C
C........FILL-IN STRUCTURE STIFFNESS MATRIX IN BANDED FORMAT
C         ONLY UPPER SEMIBANDWIDTH INCLUDING DIAGONAL
C
      IF (JJ.LT.II) GO TO 150
      IF(K1.EQ.1) IE=I
      IF(K1.EQ.2) IE=I+8
      IF(K2.EQ.1) JE=J
      IF(K2.EQ.2) JE=J+8
C
C........CHANGE -JJ- SUBSCRIPT OF FULL MATRIX TO -JJ- SUBSCRIPT
C         OF BANDED FORMAT. LOOP OVER TERMS OUTSIDE OF BAND
C
      JJ=JJ-II+1
      S(II,JJ)=S(II,JJ)+SE(IE,JE)
150   CONTINUE
155   CONTINUE
```

```
160   CONTINUE
165   CONTINUE
      RETURN
C
2000  FORMAT(*1*,43HINITIAL AND NODAL LOADS PROCESSED INTO LOAD,
     +       12H VECTOR R(I)//)
2010  FORMAT(*0*,2HR(,I3,4H)=P(,I2,1H,,I2,2H)=,F16.6)
C
      END
```

```
      SUBROUTINE STCONDN
C
C     ******************************************************************
C        TO WRITE THE UNCONDENSED STRUCTURE STIFFNESS ACCORDING
C        TO THE VALUES OF IPAR=2,3,4 FOR S, S1, S2 RESPECTIVELY
C     ******************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +         ICAL4,ICAL5,ICAL6,ICAL7
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/8/PN(37,8),R(296)
      COMMON/9/S(296,16)
C
      IF (IPAR.NE.2) GO TO 40
      IF (ICAL3.EQ.0) WRITE(61,2050)
      IF (ICAL3.EQ.0) WRITE(61,2060) (I,R(I),I=1,NSIZE)
C
C.........WRITE UNCONDENSED STRUCTURE LINEAR STIFFNESS -S- OR UNCONDENSED
C        NONLINEAR STIFFNESS S1 OR S2 DEPENDING ON VALUE OF IPAR
C
40    IF (ICAL3.NE.0) GO TO 90
      IF(IPAR.EQ.2) WRITE(61,2030)
      IF (IPAR.EQ.3) WRITE(61,2040)
      IF (IPAR.EQ.4) WRITE(61,2045)
      K1=1
      K2=8
      K3=MBAND-K1
      IF (K3.LE.7) GO TO 60
50    WRITE(61,2015) K1,K2
      WRITE(61,2020) ((S(I,J),J=K1,K2),I=1,NSIZE)
      K1=K1+8
      K2=K2+8
```

```
      K3=MBAND-K1
      IF(K3.LE.7) GO TO 60
      GO TO 50
60    WRITE(61,2015) K1,MBAND
      IF (K3.EQ.0) WRITE(61,2027) ((S(I,J),J=K1,MBAND),I=1,NSIZE)
      IF (K3.EQ.1) WRITE(61,2021) ((S(I,J),J=K1,MBAND),I=1,NSIZE)
      IF (K3.EQ.2) WRITE(61,2022) ((S(I,J),J=K1,MBAND),I=1,NSIZE)
      IF (K3.EQ.3) WRITE(61,2023) ((S(I,J),J=K1,MBAND),I=1,NSIZE)
      IF (K3.EQ.4) WRITE(61,2024) ((S(I,J),J=K1,MBAND),I=1,NSIZE)
      IF (K3.EQ.5) WRITE(61,2025) ((S(I,J),J=K1,MBAND),I=1,NSIZE)
      IF (K3.EQ.6) WRITE(61,2026) ((S(I,J),J=K1,MBAND),I=1,NSIZE)
      IF (K3.EQ.7) WRITE(61,2020) ((S(I,J),J=K1,MBAND),I=1,NSIZE)
90    CONTINUE
      REWIND 4
      REWIND 5
      REWIND 6
      RETURN
C
2015  FORMAT(*-*,7HCOLUMNS,I4,7HTHROUGH,I4)
2020  FORMAT(*0*,8E16.5)
2021  FORMAT(*0*,2E16.5)
2022  FORMAT(*0*,3E16.5)
2023  FORMAT(*0*,4E16.5)
2024  FORMAT(*0*,5E16.5)
2025  FORMAT(*0*,6E16.5)
2026  FORMAT(*0*,7E16.5)
2027  FORMAT(*0*,E16.5)
2030  FORMAT(*1*,45HUNCONDENSED LINEAR STIFFNESS OF STRUCTURE (S))
2040  FORMAT(*1*,49HUNCONDENSED NONLINEAR STIFFNESS OF STRUCTURE (S1))
2045  FORMAT(*1*,49HUNCONDENSED NONLINEAR STIFFNESS OF STRUCTURE (S2))
2050  FORMAT(*1*,28HUNCONDENSED LOAD VECTOR R(I)//)
2060  FORMAT(* *,2HR(,I3,2H)=,F16.6)
C
      END
```

```
      SUBROUTINE LINSOLN
C
C     ******************************************************************
C        TO SOLVE SYSTEM OF LINEAR EQUATIONS S*D=R BY CALLING THE
C        APPROPRIATE SUBROUTINE
C            S= STUCTURE"S LINEAR STIFFNESS
C            D= VECTOR OF D.O.F."S
C            R= LOAD VECTOR
C     ******************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +         ICAL4,ICAL5,ICAL6,ICAL7
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/8/PN(37,8),R(296)
      COMMON/9/S(296,16)
      COMMON/10/D(296),D10(1184)
C
C........FILL-IN ARRAY D(I) WITH VALUES OF LOAD VECTOR R(I)
C        AFTER SOLUTION D(I) WILL CONTAIN THE DISPLACEMENT VALUES
C
      DO 110 I=1,NEQ
110   D(I)=R(I)
C
C........CHECK DATA GENERATION FOR SOLUTION OF EQUATIONS
C
      IF(ICAL5.EQ.0) WRITE(61,2020)
      IF (ICAL5.EQ.0) WRITE(61,2010) (I,D(I),I=1,NEQ)
C
C........SOLVE SYSTEM OF -NEQ- LINEAR EQUATIONS
C
      CALL GAUSSOL
C
```

```
C........CHECK DATA GENERATION
C
      IF (ICAL5.NE.0) GO TO 140
      WRITE(61,2000)
      WRITE(61,2010) (I,D(I),I=1,NEQ)
140   RETURN
C
2000  FORMAT(*1*,34HDISPLACEMENTS FROM LINEAR SOLUTION//)
2010  FORMAT(* *,2HD(,I3,2H)=,E25.15)
2020  FORMAT(*1*,31HLOAD VECTOR FOR LINEAR SOLUTION//)
C
      END
```

```
      SUBROUTINE GAUSSOL
C
C     ******************************************************************
C        GAUSS ELIMINATION EQUATION SOLVER, BANDED FORMAT
C        FROM BOOK BY ROBERT D. COOK, FIG. 2.8.1., PAGE 45
C        CONCEPTS AND APPLICATIONS OF FINITE ELEMENT ANALYSIS
C     ******************************************************************
C
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/9/S(296,16)
      COMMON/10/D(296),D10(1184)
C
C.........FORWARD REDUCTION OF MATRIX (GAUSS ELIMINATION)
C
      DO 790 N=1,NEQ
      DO 780 L=2,MBAND
      IF (S(N,L).EQ.0.) GO TO 780
      I=N+L-1
      C=S(N,L)/S(N,1)
      J=0
      DO 750 K=L,MBAND
      J=J+1
750   S(I,J)=S(I,J)-C*S(N,K)
      S(N,L)=C
780   CONTINUE
790   CONTINUE
C
C.........FORWARD REDUCTION OF CONSTANTS (GAUSS ELIMINATION)
C
      DO 830 N=1,NEQ
      DO 820 L=2,MBAND
      IF (S(N,L).EQ.0.) GO TO 820
```

```
      I=N+L-1
      D(I)=D(I)-S(N,L)*D(N)
820   CONTINUE
830   D(N)=D(N)/S(N,1)
C
C.........SOLVE FOR UNKNOWNS BY BACK SUBSTITUTION
C
      DO 860 M=2,NEQ
      N=NEQ+1-M
      DO 850 L=2,MBAND
      IF (S(N,L).EQ.0.)GO TO 850
      K=N+L-1
      D(N)=D(N)-S(N,L)*D(K)
850   CONTINUE
860   CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE IDENT
C
C     *****************************************************************
C        TO IDENTFY THE DISPLACEMENTS FOUND IN THE SOLUTION OF
C        EQUATIONS S*D=R AND THE ONES FOUND IN THE RECOVERY PROCESS
C     *****************************************************************
C
      COMMON/1/NL,NUMNP,NUMEG,NTYPE(3),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +         ICAL4,ICAL5,ICAL6,ICAL7
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(37,8),IB(37,8),D31(113)
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),A(36),IXX(36),IYY(36),
     +         KT(36),L(1,36)
      COMMON/10/D(296),D10(1184)
      COMMON/11/DN(16),U(37,8)
C
C.........IDENTIFICATION OF DISPLACEMENTS
C
      IF (ICAL6.EQ.0) WRITE(61,2000)
      DO 230 NN=1,NUMEG
      INUMEL=NUMEL(NN)
      DO 230 K=1,INUMEL
      M=L(NN,K)
      IF (ICAL6.EQ.0) WRITE(61,2010) M
      NI=NODEI(M)
      NJ=NODEJ(M)
      DO 230 K1=1,2
      IF(K1.EQ.1) NP=NI
      IF(K1.EQ.2) NP=NJ
      DO 220 I=1,8
      IF(IA(NP,I)) 160,155,150
150   NE=IA(NP,I)
```

```
      U(NP,I)=D(NE)
      IF (ICAL6.EQ.0) WRITE(61,2020) NE,NP,I,U(NP,I)
      GO TO 220
155   U(NP,I)=0.
      IF (ICAL6.EQ.0) WRITE(61,2020) NE,NP,I,U(NP,I)
      GO TO 220
160   IF (IB(NP,I).LT.0) GO TO 170
      NM=IB(NP,I)
      GO TO 180
170   NE=-IB(NP,I)+NEQ
      U(NP,I)=D(NE)
      IF (ICAL6.EQ.0) WRITE(61,2020) NE,NP,I,U(NP,I)
      GO TO 220
180   IF (IA(NM,I)) 190,200,210
190   NE=-IB(NM,I)+NEQ
      U(NP,I)=D(NE)
      IF (ICAL6.EQ.0) WRITE(61,2020) NE,NP,I,U(NP,I)
      GO TO 220
200   U(NP,I)=0.
      IF (ICAL6.EQ.0) WRITE(61,2020) NE,NP,I,U(NP,I)
      GO TO 220
210   NE=IA(NM,I)
      U(NP,I)=D(NE)
      IF (ICAL6.EQ.0) WRITE(61,2020) NE,NP,I,U(NP,I)
220   CONTINUE
230   CONTINUE
      RETURN
C
2000  FORMAT(*1*,35HNODAL DISPLACEMENTS ON EACH ELEMENT)
2010  FORMAT(*-*,7HELEMENT,I3//)
2020  FORMAT(* *,2HD(,I3,1H),5X,2HU(,I2,1H,,I1,2H)=,E25.15)
C
      END
```

```
      SUBROUTINE STRESS
C
C     ****************************************************************
C        TO COMPUTE NODAL FORCES AND STRESSES IN THE STRUCTURE
C     ****************************************************************
C
      DIMENSION D(16)
      COMMON/1/NE,NUMNP,NUMEG,NTYPE(3),NUMEL(3),D11(8)
      COMMON/4/SE(16,16)
      COMMON/5/E(3),G(3),NODEI(36),NODEJ(36),A(36),IXX(36),IYY(36),
     +         KT(36),L(1,36)
      COMMON/8/PN(37,8),R(296)
      COMMON/11/DN(16),U(37,8)
C
      WRITE(61,2000)
      DO 100 N=1,NUMNP
      DO 100 I=1,8
100   PN(N,I)=0.
C
C........PROCESS EVERY ELEMENT OF EACH ELEMENT GROUP
C
      DO 200 K=1,NUMEG
      INUMEL=NUMEL(K)
      DO 190 KK=1,INUMEL
      M=L(K,KK)
      NI=NODEI(M)
      NJ=NODEJ(M)
      READ(1,10) ((SE(I,J),J=1,16),I=1,16)
C
C........IDENTIFY NODAL DISPLACEMENTS ON EACH ELEMENT
C
      N=NI
```

```
      DO 110 I=1,8
110   D(I)=U(N,I)
      N=NJ
      DO 120 I=9,16
120   D(I)=U(NI,I-8)
C
C........OBTAIN RESULTANT LOADS
C
      N=NI
      DO 145 I=1,8
      DN(I)=0.
      DO 140 J=1,16
140   DN(I)=DN(I)+SE(I,J)*D(J)
145   PN(N,I)=PN(N,I)+DN(I)
      N=NJ
      DO 160 I=9,16
      DN(I)=0.
      DO 155 J=1,16
155   DN(I)=DN(I)+SE(I,J)*D(J)
160   PN(N,I-8)=PN(N,I-8)+DN(I)
C
C........WRITE RESULTANT LOADS OF THE NODES OF EACH ELEMENT
C
      WRITE(61,2010) M,(DN(I),I=1,8),(DN(I),I=9,16)
190   CONTINUE
200   CONTINUE
C
C........WRITE NODAL RESULTANTS OF STRUCTURE
C
      WRITE(61,2020)
      WRITE(61,2030) (N,(PN(N,I),I=1,8),N=1,NUMNP)
      REWIND 8
```

```
      RETURN
C
10    FORMAT(E21.6)
2000  FORMAT(*1*,31HRESULTANT LOADS ON EACH ELEMENT///)
2010  FORMAT(*-*,8H ELEMENT,I3//4X,6HNODE-I,8E15.9//
     +             4X,6HNODE-J,8E15.9)
2020  FORMAT(*1*,48HSTRUCTURE RESULTANTS DUE TO LINEAR DISPLACEMENTS///
     +       1X,4HNODE,15X,3HPN1,15X,3HPN2,15X,3HPN3,15X,3HPN4,
     +       15X,3HPN5,15X,3HPN6//)
2030  FORMAT(*0*,I5,4X,8E15.9)
C
      END
```

```
      SUBROUTINE EIGENVL (EIGEN,IDATA)
C
C     ****************************************************************
C        TO SOLVE EIGENVALUE PROBLEM S*X=-(LAMBDA)*S1*X
C        WILL OBTAIN ONLY THE LOWEST EIGENVALUE AND CORRESPONDING
C        EIGENVECTOR. USES INVERSE VECTOR ITERATION WITH THE
C        RAYLEIGH QUOTIENT.
C     ****************************************************************
C
      COMMON/1/DUMMY(16),ICAL7
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/9/S(296,16)
      COMMON/10/XB(296),YB(296),X(296),Y(296),EIGNVTR(296)
C
C........ASSUME STARTING SHIFT, STARTING VECTOR, AND
C        MAXIMUN NUMBER OF ITERATIONS ALLOWED.
C
      WRITE(61,2010)
      READ(60,1000) MAX,EPSI,RHO
      WRITE(61,2000)MAX,EPSI,RHO
      DO 100 I=1,NEQ
100   X(I)=1.
C
C........OBTAIN VECTOR Y(I) FROM Y(I)=S1(I,J)*X(I)
C        FIRST CHANGE SIGN OF MATRIX S1
C
      READ(5,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 5
      DO 107 I=1,NEQ
      DO 105 J=1,MBAND
      S(I,J)=-S(I,J)
105   CONTINUE
```

```
107   CONTINUE
      WRITE(5,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 5
C
C........HORIZONTAL SWEEP OF S1(I,J)*X(I), DIAGONAL NOT INCLUDED
C
      DO 130 I=1,NEQ
      Y(I)=0.
      II=I+1
      IF (II.GT.NEQ) GO TO 130
      DO 120 J=2,MBAND
      IF (S(I,J).EQ.0.) GO TO 110
      Y(I)=Y(I)+S(I,J)*X(II)
110   II=II+1
      IF (II.GT.NEQ) GO TO 130
120   CONTINUE
130   CONTINUE
C
C........DIAGONAL SWEEP OF S1(I,J)*X(I)
C
      DO 160 I=1,NEQ
      II=I
      JJ=1
140   IF (S(II,JJ).EQ.0.) GO TO 150
      Y(I)=Y(I)+S(II,JJ)*X(II)
150   II=II-1
      JJ=JJ+1
      IF (II.EQ.0) GO TO 160
      IF (JJ.GT.MBAND) GO TO 160
      GO TO 140
160   CONTINUE
C
```

```
C........START ITERATION PROCEDURE BY STABLISHING THE SYSTEM OF
C         EQUATIONS S(I,J)*XB(I)=Y(I) AND SOLVING FOR XB(I).
C         STORE VALUES OF Y(I) INTO XB(I) FOR GAUSS SOLUTION.
C
      DO 300 K = 1,MAX
      DO 165 I=1,NEQ
165   XB(I)=Y(I)
      IF (IDATA.EQ.0) READ(4,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      IF (IDATA.EQ.1) READ(7,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 4
      REWIND 7
      IF (ICAL7.NE.0) GO TO 176
C
C........PRINT DATA SENT TO SUBROUTINE GAUSSOL
C
      WRITE(61,2100) K
      K1=1
      K2=8
      K3=MBAND-K1
      IF (K3.LE.7) GO TO 174
172   WRITE(61,2110) K1,K2
      WRITE(61,2115) ((S(I,J),J=K1,K2),I=1,NEQ)
      K1=K1+8
      K2=K2+8
      K3=MBAND-K1
      IF (K3.LE.7) GO TO 174
      GO TO 172
174   WRITE(61,2110) K1,MBAND
      IF (K3.EQ.0) WRITE(61,2120) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF (K3.EQ.1) WRITE(61,2121) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF (K3.EQ.2) WRITE(61,2122) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF (K3.EQ.3) WRITE(61,2123) ((S(I,J),J=K1,MBAND),I=1,NEQ)
```

```
      IF (K3.EQ.4) WRITE(61,2124) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF (K3.EQ.5) WRITE(61,2125) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF (K3.EQ.6) WRITE(61,2126) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF (K3.EQ.7) WRITE(61,2115) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      WRITE(61,2130)
      WRITE(61,2135) (XB(I),I=1,NEQ)
176   CONTINUE
C
C........SOLVE SYSTEM OF EQUATIONS S(I,J)*XB(I)=Y(I)
C
      CALL GAUSSOL
      IF (ICAL7.EQ.0) WRITE(61,2030)
C
C........OBTAIN VECTOR YB(I) FROM YB(I)=S1(I,J)*XB(I)
C
      READ(5,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 5
C
C........HORIZONTAL SWEEP OF S1(I,J)*XB(I), DIAGONAL NOT INCLUDED
C
      DO 200 I=1,NEQ
      YB(I)=0.
      II=I+1
      IF (II.GT.NEQ) GO TO 200
      DO 190 J=2,MBAND
      IF (S(I,J).EQ.0.) GO TO 180
      YB(I)=YB(I)+S(I,J)*XB(II)
180   II=II+1
      IF (II.GT.NEQ) GO TO 200
190   CONTINUE
200   CONTINUE
C
```

```
C........DIAGONAL SWEEP OF S1(I,J)*XB(I)
C
      DO 230 I=1,NEQ
      II=I
      JJ=1
  210 IF (S(II,JJ).EQ.0.) GO TO 220
      YB(I)=YB(I)+S(II,JJ)*XB(II)
  220 II=II-1
      JJ=JJ+1
      IF (II.EQ.0) GO TO 230
      IF (JJ.GT.MBAND) GO TO 230
      GO TO 210
  230 CONTINUE
C
C........COMPUTE RAYLEIGH QUOTIENT
C
      RQ=RHO
      Q1=Q2=0.
      DO 240 I=1,NEQ
      Q1=Q1+XB(I)*Y(I)
  240 Q2=Q2+XB(I)*YB(I)
      RHO=Q1/Q2
      DO 250 I=1,NEQ
  250 Y(I)=YB(I)/(Q2**.5)
C
C........CHECK CONVERGENCE TO DESIRED EIGENVALUE
C
      CHECK=ABS(RHO-RQ)/RHO
      IF (CHECK.LE.EPSI) GO TO 310
      EIGEN=RHO
      DO 260 I=1,NEQ
  260 EIGNVTR(I)=XB(I)/(Q2**.5)
```

```
      IF (ICAL7.NE.0) WRITE(61,2035) K,EIGEN
      IF (ICAL7.NE.0) GO TO 300
      WRITE(61,2040) K,RHO,CHECK,EIGEN
      WRITE(61,2050) (XB(I),YB(I),Y(I),EIGNVTR(I),I=1,NEQ)
300   CONTINUE
C
C........OBTAIN EIGENVALUE AND CORRESPONDING EIGENVECTOR
C
310   EIGEN=RHO
      DO 320 I=1,NEQ
320   EIGNVTR(I)=XB(I)/(Q2**.5)
      ILAST=K
      WRITE(61,2070) ILAST
      WRITE(61,2080) EIGEN
      WRITE(61,2090) (EIGNVTR(I),I=1,NEQ)
      RETURN
C
10    FORMAT(E21.6)
1000  FORMAT(I5,2F10.6)
2000  FORMAT(*-*,4HMAX=,I3///6H EPSI=,F10.6///6H  RHO=,F10.6)
2010  FORMAT(*1*,45HLINEAR EIGENVALUE PROBLEM (INVERSE ITERATION)//)
2030  FORMAT(*1*,38HINVERSE VECTOR ITERATION WITH SHIFTING///2X,1HK,9X,
     +       2HXB,16X,2HYB,16X,3HRHO,14X,5HCHECK,15X,1HY,15X,5HEIGEN,
     +       12X,7HEIGNVTR///)
2035  FORMAT(*-*,2HK=,I3,5X,6HEIGEN=,E15.9)
2040  FORMAT(*-*,I3,39X,E15.9,3X,E15.9,21X,E15.9)
2050  FORMAT(* *,6X,E15.9,3X,E15.9,39X,E15.9,21X,E15.9)
2100  FORMAT(*1*,34HDATA FOR GAUSSOL S(I,J) AND XB(I)//1X,2HK=,I3//)
2110  FORMAT(*-*,7HCOLUMNS,I4,10H   THROUGH,I4)
2115  FORMAT(*0*,8E16.8)
2120  FORMAT(*0*,E16.8)
2121  FORMAT(*0*,2E16.8)
```

```
2122  FORMAT(*0*,3E16.8)
2123  FORMAT(*0*,4E16.8)
2124  FORMAT(*0*,5E16.8)
2125  FORMAT(*0*,6E16.8)
2126  FORMAT(*0*,7E16.8)
2130  FORMAT(*-*,37HVECTOR Y(I), SENT TO GAUSSOL AS XB(I)//)
2135  FORMAT(*0*,10X,E15.9)
2070  FORMAT(*1*,5X,10HEIGENVALUE,9X,11HEIGENVECTOR,5X,6HILAST=,I3)
2080  FORMAT(*-*,E15.9)
2090  FORMAT(* *,20X,E15.9)
C
      END
```

```
      SUBROUTINE NLEIGNP (SCALE)
C
C     *************************************************************
C        THIS ROUTINE WILL COMPUTE THE EIGENVALUE OF THE
C        QUADRATIC EIGENVALUE PROBLEM (K+L*N1+L*L*N2)*X=0
C        IT USES THE MODIFIED REGULA FALSI METHOD
C     *************************************************************
C
      EXTERNAL DET
      REAL L
C
      READ(60,1000) XTOL,FTOL,NTOL,DINCR
      WRITE(61,2010) XTOL,FTOL,NTOL,DINCR
      WRITE(61,2030)
      A=0.
  100 FA=DET(A,SCALE)
      WRITE(61,2020) A,FA
      IF (FA.LT.0.) GO TO 110
      A=A+DINCR
      GO TO 100
  110 CONTINUE
      B=A
      A=A-DINCR
      CALL MRGFLS (DET,A,B,XTOL,FTOL,NTOL,IFLAG,SCALE)
      IF (IFLAG.GT.2) GO TO 500
      L=(A+B)/2.
      ERROR=ABS(B-A)/2.
      FL=DET(L,SCALE)
      WRITE(61,2000) L,ERROR,FL
  500 CONTINUE
      RETURN
C
```

```
1000  FORMAT(2E10.2,I10,F10.5)
2000  FORMAT(////14H  THE ROOT IS ,E25.15,10X,12H PLUS/MINUS ,E25.15//
     +        15H  DETERMINANT =,E25.15)
2010  FORMAT(*1*,28HQUADRATIC EIGENVALUE PROBLEM///6H XTOL=,E10.2///
     +        6H FTOL=,E10.2///6H NTOL=,I3///7H DINCR=,F10.5)
2020  FORMAT(*-*,E25.15,5X,E25.15//)
2030  FORMAT(////13X,6HLAMBDA,17X,11HDETERMINANT//)
C
      END
```

```
      SUBROUTINE MRGFLS (F,A,B,XTOL,FTOL,NTOL,IFLAG,SCALE)
C
C     ****************************************************************
C        ITERATES TO A SUFFICIENTLY SMALL VALUE OF THE DETERMINANT
C        OR TO A SUFFICIENTLY SMALL INTERVAL WHERE THE ROOT MAY
C        BE FOUND
C     ****************************************************************
C
      IFLAG=0
      FA=F(A,SCALE)
      SIGNFA=FA/ABS(FA)
      FB=F(B,SCALE)
C
C.........CHECK FOR SIGN CHANGE
C
      IF (SIGNFA*FB.LE.0.) GO TO 100
      IFLAG=3
      WRITE(61,2010) A,B
      RETURN
C
100   W=A
      FW=FA
      DO 400 N=1,NTOL
C
C.........CHECK FOR SUFFICIENTLY SMALL INTERVAL
C
      IF (ABS(B-A)/2..LE.XTOL) RETURN
C
C.........CHECK FOR SUFFICIENTLY SMALL DETERMINANT VALUE
C
      IF (ABS(FW).GT.FTOL) GO TO 200
      A=W
```

```
      B=W
      IFLAG=1
      RETURN
200   W=(FA*B-FB*A)/(FA-FB)
      PREVFW=FW/ABS(FW)
      FW=F(W,SCALE)
C
C........TEMPORARY PRINT OUT
C
      NM1=N-1
      WRITE(61,2020) NM1,A,W,B,FA,FW,FB
C
C........CHANGE TO NEW INTERVAL
C
      IF (SIGNFA*FW.LT.0.) GO TO 300
      A=W
      FA=FW
      IF (FW*PREVFW.GT.0.) FB=FB/2.
      GO TO 400
300   B=W
      FB=FW
      IF (FW*PREVFW.GT.0.) FA=FA/2.
400   CONTINUE
      IFLAG=2
      WRITE(61,2030) NTOL
      RETURN
C
2010  FORMAT(////43H F(X) IS OF SAME SIGN AT THE TWO ENDPOINTS ,
     +       2E25.15)
2020  FORMAT(*-*,I3,9H L-VALUES,3E25.15//4X,9H F-VALUES,3E25.15//)
2030  FORMAT(////19H  NO CONVERGENCE IN,I5,11H ITERATIONS)
C
      END
```

```
      FUNCTION DET (L,SCALE)
C
C     **************************************************************
C        THIS FUNCTION COMPUTES THE VALUE OF THE DETERMINANT
C        OF THE MATRIX S = K + L * N1 + L * L * N2
C            K=LINEAR STIFFNESS OF STRUCTURE
C            N1=NONLINEAR STIFFNESS OF STRUCTURE (CUBIC TERMS)
C            N2=NONLINEAR STIFFNESS OF STRUCTURE (QUARTIC TERMS)
C            L=VALUE OF LAMBDA FOR WHICH S IS COMPUTED
C     **************************************************************
C
      REAL K,N1,N2,L
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/9/S(296,16)
C
C........COMPUTE MATRIX S=K+L*N1+L*L*N2 (BANDED FORMAT)
C
      IF (L.EQ.0.) GO TO 220
      DO 210 I=1,NEQ
      DO 200 J=1,MBAND
      READ(4,10) K
      IF (IEIGEN.EQ.1) READ(5,10) N1
      IF (IEIGEN.EQ.2) READ(7,10) N1
      READ(6,10) N2
      S(I,J)=K+L*N1+L*L*N2
200   CONTINUE
210   CONTINUE
      GO TO 230
220   READ(4,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
230   REWIND 4
      REWIND 5
      REWIND 6
```

```
      REWIND 7
C
C........FORWARD REDUCTION OF MATRIX (GAUSS ELIMINATION)
C
      DO 390 N=1,NEQ
      DO 380 LL=2,MBAND
      IF (S(N,LL).EQ.0.) GO TO 380
      I=N+LL-1
      C=S(N,LL)/S(N,1)
      J=0
      DO 350 KK=LL,MBAND
      J=J+1
  350 S(I,J)=S(I,J)-C*S(N,KK)
      S(N,LL)=C
  380 CONTINUE
  390 CONTINUE
C
C........COMPUTE DETERMINANT OF MATRIX S
C        SCALE DOWN "DET" BY A "SCALE" VALUE AFTER EACH STEP
C
      DT=1.
      DO 400 I=1,NEQ
      DT=DT*S(I,1)/SCALE
  400 CONTINUE
      DET=DT
      RETURN
C
   10 FORMAT(E21.6)
C
      END
```

```
      SUBROUTINE TILTED (IDATA,SCALE,DX,W)
C
C     *************************************************************
C        TO COMPUTE THE BUCKLING LOAD OF A DECK BRIDGE OR
C        A THROUGH BRIDGE DUE TO TILTED LOADS
C     *************************************************************
C
      COMMON/1/NE,NUMNP,D1(15)
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(37,8),IB(37,8),X(37),Y(37),H(37),RAD,AC
      COMMON/9/S(296,16)
C
C........CHECK TYPE OF BRIDGE BEING CONSIDERED
C           IDECK.EQ.0 , THROUGH BRIDGE
C           IDECK.EQ.1 , DECK BRIDGE
C
      READ(60,1010) IDECK,HD
      WRITE(61,2000)
      IF (IDECK.EQ.0) WRITE(61,2020) IDECK,HD
      IF (IDECK.EQ.1) WRITE(61,2010) IDECK,HD
C
C........MODIFY MATRIX S1 BY PARAMETER P/H ACCORDING TO
C        EQUATION NUMBERS IN ARRAY IA(N,I). STORE MATRIX S1
C
      READ(5,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 5
      P=W*DX
      WRITE(61,2030) P
      DO 110 N=1,NUMNP
      IF (IA(N,2).LE.0) GO TO 110
      I=IA(N,2)
      IF (IDECK.EQ.0) GO TO 100
```

```
      IF (IDECK.EQ.1) GO TO 105
100   H(N)=HD-Y(N)
      WRITE(61,2040) N,H(N)
      WRITE(61,2050) S(I,1)
      S(I,1)=S(I,1)+P/H(N)
      WRITE(61,2060) S(I,1)
      GO TO 110
105   H(N)=HD+Y(N)
      WRITE(61,2040) N,H(N)
      WRITE(61,2050) S(I,1)
      S(I,1)=S(I,1)-P/H(N)
      WRITE(61,2060) S(I,1)
110   CONTINUE
      IF (IEIGEN.EQ.0) WRITE(5,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      IF (IEIGEN.EQ.1) WRITE(5,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      IF (IEIGEN.EQ.2) WRITE(7,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 5
      REWIND 7
C
C........SOLVE EIGENVALUE PROBLEM
C          IF IEIGEN.EQ.0 , SOLVE LINEAR CASE
C          IF IEIGEN.EQ.1 , SOLVE QUADRATIC CASE
C          IF IEIGEN.EQ.2 , SOLVE BOTH
C
      EIGEN=0.
      IF (IEIGEN.EQ.0) CALL EIGENVL (EIGEN,IDATA)
      IF (IEIGEN.EQ.1) CALL NLEIGNP (SCALE)
      IF (IEIGEN.EQ.2) GO TO 120
      RETURN
120   CALL NLEIGNP (SCALE)
      CALL EIGENVL (EIGEN,IDATA)
      RETURN
```

```
C
10    FORMAT(E21.6)
1010  FORMAT(I5,F10.5)
2000  FORMAT(*1*,18H TILTED LOAD CASE //)
2010  FORMAT(////13H  DECK BRIDGE///9H  IDECK =,I2///6H  HD =,F10.5)
2020  FORMAT(////16H  THROUGH BRIDGE///9H  IDECK =,I2///6H  HD =,F10.5)
2030  FORMAT(////25H  LOAD ON EACH COLUMN IS ,F10.5////
     +       16H  COLUMN LENGTHS//)
2040  FORMAT(*0*,7H     H(,I2,3H) =,F7.3)
C
2050  FORMAT(*+*,25X,E25.15)
2060  FORMAT(*+*,56X,E25.15)
      END
```

C.4 TILTED LOAD SOLUTION FOR HALF-THROUGH ARCHES

In this case statements 32 to 44 in the subroutine TILTED should be replaced by the following:

```
      H(N) = Y(N) - HD
      WRITE (61, 2040) N, H(N)
      IF (H(N). EQ. 0.) GO TO 100
      S(I, 1) = S(I, 1) + P/H(N)
100   CONTINUE
```