This is to certify that the

dissertation entitled

Stability and Nonlinear Response
of Deck-type Arch Bridges

presented by

Khaled Yagoob Medallah

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Civil Engineering

R. K. Wen
Major professor

Date March 26, 1984

STABILITY AND NONLINEAR RESPONSE OF DECK-TYPE ARCH BRIDGES

By

Khaled Yagoob Medallah

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

1984

ABSTRACT

STABILITY AND NONLINEAR RESPONSE OF DECK-TYPE ARCH BRIDGES

By

Khaled Yagoob Medallah

The nonlinar response in three dimensions of deck-type bridges was considered by an "amplification factor method" which has a form similar to that used in the design of beam-columns. For use in the amplification factor method the eigenvalues (and corresponding eigenvectors) or buckling loads of the structures were studied. Two three-dimensional numerical model bridges were constructed from actual designed bridges. The finite element formulation was used in the analysis using stright beam and truss elements; only geometric nonlinearity was considered.

Factors affecting the elastic stability in three dimensional space studied included the patterns and the amount of rib bracing, the deck-ribs connections, and the stiffnesses of the towers, deck, and the transverse bracing.

Predictions of nonlinear responses using the amplification factor method were compared with nonlinear equilibrium solutions for lateral, longitudinal, and vertical loadings. The comparisons in general indicated good agreement.

ACKNOWLEDGMENTS

The author would like to express his sincere appreciation to those who made the completion of this research possible. Special appreciation is extended to my dissertation advisor, Dr. Robert K. Wen, for suggesting this topic and for his continuous advice and encouragement throughout the work. Thanks and gratitude are also due to the other members of the Guidance Committee: Dr. William Bradley, Dr. Charles Cutts and Dr. Nicholas Altiero. The support and encouragement of Dr. William Taylor, Chairman of the Department of Civil and Sanitary Engineering, are sincerely appreciated. Also, thanks are due to the Saudi Arabian Educational Mission for supporting the research and to Vicki Switzer and Debbie Lange for their dedicated typing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Object and Scope.

Arch bridges are known for their aesthetical lines and the large distances they can span. They are usually built of concrete or steel to overpass rivers, to connect roads over valleys in mountain areas, to bridge highways, and sometimes even to support special structures. According to the deck location arch bridges are classified as "deck bridge," "half-through bridge" or "through bridge." Figure 1-1 illustrates the different types of arch bridges.

An arch bridge subjected to a vertical uniformly distributed load would develop mainly thrust actions, a situation similar to that of a column under axial compression. Dead loads are essentially uniformly distributed vertical loads. Live loads, on the other hand, may cover only a portion of the span to produce maximum bending. Wind load (and earthquake loads) are applied laterally or longitudinally. Due to the initial compression in the arch in response to the dead loads, the structure behavior under additional live load (or wind load) will be nonlinear. This is a situation similar to the case of a beam subjected to an axial load and lateral load.

Starting about the late forties, with increasing needs for more economical and slender arches the deflection effect, or the geometric nonlinearity effect on their response became an important issue. Many attempts have been made to address the problem. While design engineers used relatively simple approaches, researchers studied it in theoretically more rigorous frameworks. (See section on "Literature Review.") Today, a fully nonlinear solution is possible with the help of finite element codes and advanced computing facilities. Such a solution is very expensive and not yet readily available to many designers. A less involved and simpler approach is desirable to facilitate, at least, preliminary designs.

A procedure that had been used to estimate nonlinear response, R_n, from the linear response, R_L, is as follows:

$$R_n = R_L \frac{1}{1 - \frac{P}{P_c}} \tag{1.1}$$

where P is the compression load on a structure and P_c is the critical buckling load of the same structure. Equation (1.1) may be written as:

$$R_n = R_L \, AF$$

in which AF (stands for "amplification factor") = $1/(1-P/P_c)$. This procedure will be referred to in this thesis as the "amplification factor method."

The major objective of this work is to examine the validity of the amplification factor method for estimating nonlinear responses of arch bridges. Since AF involves P_c, obviously the buckling load is of great importance in this work. In the literature the effects of some factors such as the torsional rigidity of the ribs and the flexural rigidity of the bracing members on the buckling load have been reported. It seems that there are other factors such as the deck rigidity and the bracing which may be important factors also. The effect of such factors on the buckling loads of arch bridges will also be considered here.

The study was based on a nonlinear elastic beam finite element as reported in Reference 23. The computer program used for this study was obtained by modifications and expansion of two available programs CURVEL and FRAL3D (21,22). The modifications comprised of the treatment of larger systems including the loading vectors, the addition of a nonlinear elastic truss element, solution for multiple eigenpairs (eigenvalues and eigenvectors), and the graphic output of the buckling modes. The computer program was extensively checked by comparing results with known correct ones.

Two bridge models were used for this research. One, designated as MCSCB, was a modified version of the Cold Spring Canyon Bridge in California (20), and the second, designated as FHAB, was taken from a Federal Highway Administration Report on arch bridges (11). The use of such model structures of practical design should increase the significance of the data and the value of the results.

The most important finding of this study is perhaps that, so far as elastic stability is concerned, the arch bridge should be considered as a system. In the past, much attention had been focussed on the stability of arch ribs. The results of the study showed that other components such as the deck, longitudinal bracing, transverse bracing, as well as rib bracing, had very significant effects on the stability of the bridge. For example, an addition of nominal transverse bracing members at the crown could change the buckling mode and correspondingly significantly increase the buckling load.

The amplification factor method was considered for lateral, longitudinal and vertical loadings. The values of the maximum responses compared well with those obtained from solutions of the nonlinear equilibrium equations if the loads are, say, no greater than one half of the buckling load. In regard to buckling load, the value used in Eq. 1.1 must be that which corresponds to a buckling mode compatible to the deflection shape caused by the applied load.

1.2 Literature Review.

The problem of out-of-plane buckling of a curved beam was studied by Ojalvo and Newman (1). The authors presented linearized perturbation equations about a reference state, which satisfy the nonlinear beam equilibrium equations. Equations for the determination of the buckling load could be solved by a "shooting" procedure (a boundary value problem solved as an initial value problem). Ojalvo, Demuts and Tokarz (2) presented two equations for determining the out-of-plane buckling of a planar curved member initially in a plane under pull and thrust, with the load also initially in plane. Wen and Lange (3,21) presented a curved beam finite element model including geometric nonlinear effects. The element is initially curved in a plane but may deform out of or in the initial plane. Members of various shapes can be studied due to the flexibility of the geometry of the finite element model. Presented data for in-plane and out-of-plane buckling showed that for symmetric and symmetrically loaded arches using the linear or quadratic eigenvalue formulations made only insignificant differences in the buckling loads. Two matrices were considered for the first nonlinear stiffness, one with rotation terms included and one without. Results showed that the two agree for problems with little bending, but for problems involving significant bending the latter should be used.

Tokarz (4) presented experimental results on the lateral buckling of parabolic arches. Tokarz and Sundhu (5) presented a pair of equations governing the torsional buckling of arches under uniform vertical loading based on linear buckling theory. Both papers used a free standing rib, two ribs braced at the crown, and used uniform bracing only in the experimental work (4). Factors affecting the torsional buckling such as the rise to span ratio, the torsional rigidity to the out-of-plane bending rigidity of the arch, and "tilted load effects" (see section 2.5) were considered. Some of the tabulated results will be compared with solutions obtained by the finite element model used herein.

Donald and Godden (6) presented a solution for the problem of a curved beam with transverse loading only and transverse with axial loading using a numerical forward integration method (a "shooting" procedure). Results correlated well with experimental work. An amplification factor method related to the thrust in the arch gave predictions with accuracy to one percent. Only symmetric lateral buckling was studied.

The problem of tied arches (for through bridges) was studied by Godden (7) and Godden and Thompson (8). Theoretical and experimental work were presented for various factors affecting the lateral stability of unbraced tied arch bridges, such as flexural rigidities, torsional rigidities, rise to span ratio and the stiffness of the hangers connected to the tie. The hanger effect was found to be very significant. A good correlation between theoretical and experimental results was noted.

Shukla and Ojalvo (9) presented a numerical solution, based on a forward marching integration, for the theory developed in (1). It was found that the buckling load for through arches is three to four times that for deck type arches. Optimum rise to span ratio with torsional to flexural rigidities are presented. Only single rib arches were used in the study.

In the book Guide to Stability Design Criteria of Metal Structures, third edition (10), were summarized available data in the form of formulas and tables for arch design for in-plane and out-of-plane elastic stability under different loading conditions. In Nettleton's work (11) related to the design of steel and concrete bridges, suggestions for the practitioners were presented regarding local and overall stability design. Moment magnification was related to the thrust at supports. Wind load design procedures were suggested for both the single and double lateral systems. Based on data in Reference (10), the effective slenderness ratios for the braced arches similar to the concept of column design were proposed for different rise to span ratios with the effective length given by Bleich (12). He also suggested that a continuous roadway would contribute to the stiffness of the arch in proportion to the ratio of the floor member depth to the depth of the rib.

Ostlund (13) was the first to study two ribs braced with cross beams. Two arch forms (one "deep," one "shallow") with varying properties were studied. Factors considered included the flexural rigidity of transverse bars, rise of arch, spacing of the two ribs, number of transverse bars, torsional stiffness of ribs, the "tilted load" effects, and connection of ribs to the crown. A good correlation between the theoretical and experimental results were found by the author. The arches used had the out-of-plane mode as the lowest buckling mode. In practice it would appear that the in-plane buckling load would be the lowest.

Wastlund (14) presented several works done by several researchers under his supervision. A method analogous to that of the beam-column was suggested for arches with vertical buckling. A simple formula for additional moment is presented (amplification factor based on the arch thrust). A method for calculating the vertical symmetrical and antisymmetrical buckling load was also discussed. Tests results which correlated well with theoretical data were presented. For out-of-plane buckling of braced arches, Wastlund suggested an approximate method which stated that arches should be straightened out so that the ribs and the bracing would lie in one plane. The buckling load then would be computed as for a column with battens. This method ignores the rise to span ratio, the torsional stiffness of the rib and the vertical stiffness of the bracing.

Almeida (15) presented numerical solutions of the curved beam theory (2) for two parabolic ribs braced with tranverse beams. The cases of the deck situated above, below, and no decks were considered. Factors affecting the buckling load were studied. Almeida's data checked with the work of Tokarz fairly well and with Ostlund satisfactorily.

Sakimoto and Namita (16) used the transform matrix method to obtain the eigenvalues of the differential equation governing the buckling load of framed structures. Only circular arches under uniformly distributed radial forces were considered. The authors found that in certain cases the location of the transverse bars was more important than the number of them and that to increase the strength of the arch it was better to constrain the out-of-plane flexure rigidity of the arch than the torsional deformation, and that a slight loosening of the fixed end about the out-of-plane may reduce the out-of-plane buckling of the arch considerably.

Sakimoto and Komatsu (17,18) studied the ultimate load-carrying capacity of arch systems under vertical loading. The papers considered the overall slenderness ratio of the structure as a major parameter. The second paper presented for through bridges gave three effective length values to be used under certain conditions for buckling load estimation. A small initial deflection was assumed to induce buckling. A design formula for the preliminary design of through bridges was suggested.

Yabuki and Vinnakota (19) did a literature review that covered a very broad area of arch stability including planar and spatial, linear, nonlinear (geometrical and material nonlinearity) and some parameters affecting the stability problem. A look at the German and Japanese specifications with suggestions based on the presented data was also included.

1.3 Nomenclature.

A	= Cross-sectional area;
A^*	= Cross-sectional area of a deck beam;
A, B	= End nodes of an element;
A_o	= Cross-sectional area of the modelled column (MCSCB);
A_c, A_c^*	= Cross-sectional area of chord in original and modelled bridges;
A_{CB}	= Cross-sectional area of composite beam;
A_D	= Cross-sectional area of D-truss bracing;
A_d, A_d^*	= Cross-sectional area of diagonal member in original and modelled bridges;
A_s	= Cross-sectional area of spandrel bracing;
A_{slab}	= Cross-sectional area of deck slab;
A_T	= Cross-sectional area of transverse bracing between ribs (beam bracing);
A_t	= Cross-sectional area of truss member;
A_v, A_v^*	= Cross-sectional area of vertical member in original and modelled bridges;
a	= Chord length in truss (CSCB);
$a_1 \ldots a_{12}$	= Parameters used for definition of shape functions;
E	= Modulus of elasticity;
f	= Rise;
G	= Shear modulus;
h	= Height of crown column;

h_o = Original height of crown column as modelled for (MCSCB);

I_o = Moment of inertia of column as used for (MSCSB);

I_B = Moment of inertia due to beam action in beam-truss tower;

I_{BT} = Moment of inertia for beam-truss tower;

I_c = Moment of inertia due to truss action in truss and beam-truss towers;

I_{xx}, I_{zz} = Moment of inertia of cross-section (Figure 3-5);

I_{Y-Y} = Moment of inertia of deck about global Y-Y axis;

J = Torsional constant;

KT = Torsional constant of cross-section;

[k], [K] = Element and structural linear stiffness matrices;

L = Arch bridge span;

ℓ = Length of element;

ℓ_d, $\ell_d{*}$ = Length of diagonal member in truss (original and modelled bridges);

ℓ_v, $\ell_v{*}$ = Length of vertical member in truss (original and modelled bridges);

N = Total number of panels;

$[n_1]$, $[N_1]$ = Element and structural first order nonlinear stiffness matrices;

$[n_1{*}]$, $[N_1{*}]$ = Element and structural first order geometric stiffness matrices;

$[n_2{*}]$, $[N_2{*}]$ = Element and structural second order nonlinear stiffness matrices;

P = Applied concentrated load;

{P} = External load vector;

$\{P_{ref}\}$ = Reference external load vector;

$\{P_c\}$	=	Critical value of applied load;
P_c	=	Critical (buckling) load;
Q	=	Lateral concentrated load;
$\{Q\}$	=	Structural generalized displacement vector;
$\{Q_o\}$	=	Unit vector;
$\{Q_1\}$	=	First eigenvector;
$\{\bar{Q}_o\}$	=	$\{Q_o\} - \alpha_1 \{Q_1\}$
$\{Q_{ref}\}$	=	Reference structural generalized displacement vector;
q_o	=	Axial compression at crown in rib;
$\{q\}$	=	$[q_1, q_2, \ldots q_6, q_7, q_8, \ldots q_{12}]^T$;
R	=	Response;
R_L	=	Linear response;
R_n	=	Nonlinear response;
S	=	Spacing between ribs or deck width;
S_D	=	Force in diagonal member in truss;
S_v	=	Force in vertical member in truss;
$[S_s]$	=	Structural secant stiffness matrix;
$[S_T]$	=	Structural tangent stiffness matrix;
u, v, w	=	Displacements along local x, y, z axes, respectively;
u_1, v_1, w_1 and u_2, v_2, w_2	=	Displacement for nodes 1 and 2 of beam element along x, y, z axes, respectively;
u_1, u_2	=	Axial displacements of ends of truss element, local coordinates;
U_ε	=	Strain energy of the element;
U_t	=	Torsional strain energy of the element;
U	=	Total strain energy of the element;

U_2, U_3, U_4 = Quadratic, cubic and quartic parts of strain energy;

V = Total volume of bracing members between ribs;

w = Nonuniform distributed load;

w_a = Additional applied load (distributed on portion of the span to produce maximum response);

w_c = Critical uniform load;

w_F = Fixed load (uniformly distributed);

w_y = Uniformly distributed yield load.

W = w_c/w_y

X, Y, Z = Global coordinates of the bridge (see Figure 2-2);

x, z = Local coordinates of element (see Figure 3-5);

α = w_F/w_c;

α_1 = Constant, see equation 2.4.9;

ε = Longitudinal strain;

ϕ, ψ, θ = Rotation about x, y, z axes, respectively;

ϕ_1, ψ_1, θ_1 and ϕ_2, ψ_2, θ_2 = Rotation about x, y, z axes for nodes 1 and 2, respectively;

Φ_p = Total potential energy;

λ = Buckling load parameter;

Δ = Incremental operator;

Δ_s, Δ_s* = Shear displacement in original and modelled bridge;

Δ_B, Δ_B* = Bending displacement in original and modelled bridge;

τ = Shear stress;

θ, θ* = Angle of inclination of diagonal truss member in original and modelled bridge;

η, ζ	= **Coordinates of a point with respect to principal axes;**
σ_y	= **Yield stress.**

CHAPTER II

THEORETICAL BACKGROUND

2.1 Introduction.

In this chapter the basis of the amplification factor method is explained in detail. For the sake of completeness, an outline of the derivation of the equilibrium and eigenvalue equation (21,22) is presented next, followed by a description of the procedure used for the solution of the eigenvalues and eigenvectors. The chapter concludes with a description of the so-called "tilted load effects."

2.2 The "Amplification Factor Method".

It was mentioned in the preceding chapter that the "Amplification Factor", AF, was introduced by Timoshenko (27) when considering a simple beam problem subjected to a combined lateral and axial load (see Figure 2-1). Using the principle of superposition a trigonometric series for the beam deflection was obtained. The first term in the series for the mid-span deflection, R, is

$$R = \frac{2\ Q\ \ell^3}{\pi^4\ EI}\ \left(\frac{1}{1-\alpha}\right) \qquad (2.2.1)$$

where Q is the lateral load, EI is the flexural rigidity, ℓ is the beam length, and $\alpha = P/P_c$, in which P is the applied axial load on the beam, and P_c is its lowest buckling or critical value.

The value of the linear deflection, R_L, is approximately equal to the first factor in equation 2.2.1, i.e.,

$$\frac{Q\ell^3}{48EI} \doteq \frac{2\ Q\ \ell^3}{\pi^4\ EI} = \frac{Q\ell^3}{48.7\ EI} \tag{2.2.2}$$

Thus equation 2.2.1 may be rewritten as:

$$R \doteq R_L \ (\frac{1}{1-\alpha}) \tag{2.2.3}$$

The term $(1/1-\alpha)$, as mentioned earlier, is the amplification factor, AF.

The preceding concept had been employed to estimate the nonlinear response of other types of structures. For example, for a single arch, the factor α was expressed as the ratio of the actual thrust and the critical value of the thrust for the arch (see, for example, (6)).

For a three dimensional arch bridge it is more convenient to use the AF as a function of the applied loads. If the vertical fixed load (usually a uniformly distributed "dead load") is denoted by w_F, and the additional load, "live or wind load" by w_a, then the nonlinear response, R_n, due to w_a may be estimated by

$$R_n = R_L \frac{1}{1-\alpha} \qquad (2.2.4)$$

where R_L is the linear response due to w_a and $\alpha = w_F/w_c$. The quantity w_c is the lowest "compatible buckling load." The preceding term is used here to define the buckling or critical value of the vertical load the associated buckling mode of which is conformable or compatible with the deflection shape of the bridge due to the added load under consideration, i.e., w_a. A space structure has an infinite number of w_c, each corresponding to a different buckling mode, and so care is required to use the correct w_c for the AF method.

The global coordinate system is presented in Figure 2-2. The combinations of loading conditions are presented in Figure 2-3, and two typical buckled shapes are presented in Figure 2-4. For response in the X or Y direction, generally w_c that causes the buckled shape illustrated in Figure 2-4a is to be used, for response in the Z direction, generally w_c that causes the buckled shape illustrated in Figure 2-4b is to be used. It is very essential to note that depending on the structure-load system, the compatible buckling mode shape may be different from those given in Figure 2-4. This will be pointed out in Chapter V.

2.3 Equilibrium and Eigenvalue Equations.

The nonlinear equilibrium solutions and the buckling loads and modes used herein are based on the nonlinear finite element model described in references 22 and 23. For the sake of completeness, the method is outlined in the following.

The following assumptions were made for the derivation of the model:

(1) The material of the elements is linearly elastic.

(2) Plane sections remain plane.

(3) The cross-section of the element is constant and has two axes of symmetry.

(4) The effect of torsional deformation on normal strain is negligible.

(5) The axial strain due to the transverse displacement is averaged over the element length.

Consider a beam element in space with x, y and z coordinates and u, v and w displacements, ϕ, ψ and θ are rotations about x, y and z axes respectively. An assumed linear shape function for each of u and ϕ and a cubic shape function for each of v and w are:

$$\begin{aligned} u &= a_1 + a_2 x \\ v &= a_3 + a_4 x + a_5 x^2 + a_6 x^3 \\ w &= a_7 + a_8 x + a_9 x^2 + a_{10} x^3 \\ \phi &= a_{11} + a_{12} x \end{aligned} \tag{2.3.1}$$

Referring to Figure 2-5 the boundary conditions are:

at $x = 0$

$$u = u_1,\ v = v_1,\ w = w_1,\ \frac{dv}{dx} = \theta_1 \tag{2.3.2a}$$

$$\frac{dw}{dx} = -\psi_1 \text{ and } \phi = \phi_1.$$

and at $x = \ell$

$$u = u_2,\ v = v_2,\ w = w_2,\ \frac{dv}{dx} = \theta_2 \tag{2.3.2b}$$

$$\frac{dw}{dx} = -\psi_2 \text{ and } \phi = \phi_2$$

Substituting equation (2.3.1) into equation (2.3.2), a system of linear equations would be obtained. When solved, the values of u, v, w and ϕ as functions of the generalized coordinates are found.

Using beam theory where plane sections remain plane, the longitudinal strain of beam elements is:

$$\varepsilon(x,\eta,\zeta) = \varepsilon_a(x) + \eta\frac{d^2v}{dx^2} + \zeta\frac{d^2w}{dx^2} \tag{2.3.3}$$

in which η and ζ are the coordinates of the point from the cross-section centroid at which the strain is evaluated (see Figure 2-6), and $\varepsilon_a(x)$ is the axial strain at the centroid which, when using the average strain assumption, is

$$\varepsilon_a(x) = \frac{du}{dx} + \frac{1}{\ell}\int_o^\ell \frac{1}{2}\left(\frac{dv}{dx}\right)^2 dx + \frac{1}{\ell}\int_o^\ell \frac{1}{2}\left(\frac{dw}{dx}\right)^2 dx. \tag{2.3.4}$$

From equation (2.3.3) and (2.3.4) the strain at each point of the section can be determined readily.

The strain energy due to normal strain is,

$$U_\varepsilon = \int_{vol} \frac{1}{2} E\,[\xi(x,\eta,\zeta)^2]\,dvol \tag{2.3.5a}$$

and due to torsion

$$U_t = \frac{1}{2} \int_0^\ell GJ \left(\frac{d\phi}{dx}\right)^2 dx \tag{2.3.5b}$$

upon substitution of equations (2.3.3) and (2.3.4) in equation (2.3.5a), and $\frac{d\phi}{DX} = \frac{\phi_2 - \phi_1}{\ell}$ in equation (2.3.5b) then the strain energy, U, can be found:

$$U = U_\varepsilon + U_t \tag{2.3.6}$$

The stiffness matrices can be derived from the strain energy equation (2.3.6). Equation (2.3.6) can be divided into three parts, i.e.;

$$U = U_2 + U_3 + U_4 \tag{2.3.7}$$

in which, U_2 contains only quadratic terms, and U_3 and U_4 contain cubic and quartic terms respectively. The stiffness matrices can be derived as follows:

$$\begin{aligned}
[k] &= [(k)_{i,j}] = \left[\frac{\partial^2 U_2}{\partial q_i\ \partial q_j}\right] \\
[n_1] &= [(n_1)_{i,j}] = \left[\frac{\partial^2 U_3}{\partial q_i\ \partial q_j}\right] \\
[n_2] &= [(n_2)_{i,j}] = \left[\frac{\partial^2 U_4}{\partial q_i\ \partial q_j}\right]
\end{aligned} \tag{2.3.8}$$

in which q_i and q_j represent the generalized coordinates such as u_1, v_1, w_1, ..., etc., [k] is the linear stiffness matrix, $[n_1]$ and $[n_2]$ are the first and second order nonlinear stiffness matrices, containing, respectively, linear and quadratic terms of the displacements. If terms containing rotational displacements in $[n_1]$ are eliminated, the resulting matrix is denoted by $[n_1{*}]$. Details of the entries of the various stiffness matrices can be found in the above-cited references.

The linear and nonlinear stiffness matrices for a truss member were developed in the same manner as for the beam. The matrices are listed in Appendix A. (A linear shape factor was assumed.)

If the stiffness matrices of the elements are transformed into global coordinates, assembled, and denoted by [K], $[N_1]$ and $[N_2]$ for the linear, first and second order system stiffness matrices, and denoting the generalized displacement vector and external load vector by {Q} and {P}, the total strain energy U and the potential energy Φ_p can be written as follows (see Mallet and Marcal (25)):

$$U = \{Q\}[\frac{1}{2}[K] + \frac{1}{6}[N_1] + \frac{1}{12}[N_2]]\{Q\} \qquad (2.3.9)$$

and

$$\phi_p = U - \{Q\}\{P\} \tag{2.3.10}$$

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

$$[S_s]\ \{Q\} = \{P\} \tag{2.3.11}$$

where

$$[S_s] = [K] + \frac{1}{2}[N_1] + \frac{1}{3}[N_2] \tag{2.3.12}$$

The second variation of the potential equation gives the incremental equilibrium equation

$$[S_T]\ \{\Delta Q\} = \{\Delta P\} \tag{2.3.13}$$

where $\{\Delta Q\}$ and $\{\Delta P\}$ are the incremental displacement and loads respectively. $[S_T]$ is the tangent stiffness matrix which is given by:

$$[S_T] = [K] + [N_1] + [N_2] \tag{2.3.14}$$

from equation (2.3.14), the incremental equilibrium equation is

$$([K] + [N_1] + [N_2])\ \{\Delta Q\} = \{\Delta P\} \tag{2.3.15}$$

The nonlinear equilibrium responses obtained for this study were solutions of the preceding equation. The buckling loads and modes were obtained by setting $\{\Delta P\}=\{0\}$ in the preceding equation, i.e.,

$$([K] + [N_1] + [N_2])\ \{\Delta Q\} = \{0\} \tag{2.3.16}$$

The solution of equation (2.3.16) is considered in the next section.

2.4 Eigenvalue Solution.

2.4.1 Introduction.

The eigenvalue solution is a mathematical formulation used for the determination of critical loads. If a load on a structure is increased proportionally, the structure reaches a load called the buckling load at which the response could become indefinite.

The solution of equation 2.3.16 is difficult due to the fact that the matrices $[N_1]$ and $[N_2]$ are functions of the displacement variables $\{q\}$. Assuming that the displacements are linear functions of the applied loads up to the point of buckling, then:

$$\{Q_{ref}\} = [k]^{-1} \{P_{ref}\} \tag{2.4.1}$$

where $\{P_{ref}\}$ is an arbitrary reference load vector and $\{Q_{ref}\}$ is the corresponding linear response. Since $[N_1]$ and $[N_2]$ are linear and quadratic functions of the displacements, writing

$$\{P\} = \lambda \{P_{ref}\} \tag{2.4.2}$$

one obtains

$$[N_1(\{Q\})] = [N_1(\{Q_{ref}\})]\lambda \tag{2.4.3}$$

and

$$[N_2(\{Q\})] = [N_2(\{Q_{ref}\})\lambda^2 \qquad (2.4.4)$$

in which λ is a scaler parameter.

Equation 2.3.16 can then be rewritten as

$$([K] + \lambda_c[N_1] + \lambda^2{}_c\ [N_2])_{\{Q_{ref}\}}\ \{\Delta Q\} = 0 \qquad (2.4.5)$$

Equation (2.4.5) is a quadratic eigenvalue equation. For sufficiently small displacement, the $[N_2]$ matrix can be neglected, i.e.,

$$([K] + \lambda_c[N_1])_{\{Q_{ref}\}}\ \{\Delta Q\} = 0 \qquad (2.4.6)$$

Equation (2.4.6) is a linear eigenvalue equation. A solution of the eigenvalue problem defined by equations (2.4.5) or (2.4.6) yields the value of λ_c, and consequently the critical load, $\{P_c\}$, may be found using equation (2.4.2), i.e.,

$$\{P_c\} = \lambda_c\ \{P_{ref}\} \qquad (2.4.7)$$

2.4.2 Solution Procedure for Eigenpairs.

To solve for the first eigenpair (eigenvalue and eigenvector), the well-known method of "inverse vector iteration" was used using a unit starting vector, i.e., $\{Q_o\}^T = [1\ \ 1\ \ 1\ \ \ldots.\ 1]$. The next eigenpair was obtained by "deflating" the iteration vector or sweeping out from the latter the eigenvector just computed, i.e.,

$$\{\bar{Q}_o\} = \{Q_o\} - \alpha_1\{Q_1\} \qquad (2.4.8)$$

in which $\{Q_1\}$ is the first eigenvector just found.

If one writes:

$$Q_0 = \alpha_1 Q_1 + \alpha_2 Q_2 + \alpha_3 Q_3 + \cdots \tag{2.4.9}$$

the value of α_1 may be found using the orthogonality of the Q_i vectors with respect to $[-N_1]$, i.e.;

$$\begin{aligned} \{Q_n\} \; [-N_1] \; \{Q_m\} &= 1 \qquad n = m \\ &= 0 \qquad n \neq m \end{aligned}$$

It follows:

$$\alpha_1 = \{Q_1\}^T [-N_1] \{Q_o\} \tag{2.4.10}$$

Iteration starting with $\{\bar{Q}_o\}$ would produce a second eigenpair. Similarly, a third eigenpair can be obtained after sweeping the two eigenvectors from the unit vector and use it for starting the iterative procedure. The details may be found in such works as reference 24.

2.5 Tilted Load Effects.

In earlier research works on the effect of the bridge deck on the lateral buckling load, the deck was assumed to be rigid in that direction. Therefore, the deck vertical loads transferred to the arch ribs would be tilted on account of the lateral displacement. The phenomenon, illustrated in Figure 2-7, is similar to the "P-Δ" effect considered in building structures. It will also be referred to herein as "rigid or classical deck effect." For a finite element formulation, the tilted load effect had been considered in reference 21 for a single arch rib.

The same approach is used herein for the arch bridge system.

CHAPTER III

BRIDGES, MODELLING, COMPUTER PROGRAMS AND VALIDATION

3.1 Introduction.

It was mentioned earlier that two bridges were used for this study: the FHAB and the MCSCB. The FHAB was taken from a research report by Nettleton (11) and used for this research essentially as it was given therein. The MCSCB was obtained by modifying or simplifying the model for the Cold Spring Canyon Bridge. Two reasons make the simplification necessary. One is the computer solution cost. The other is the central computer memory. This analysis used 340,000 words, the full memory of the MSU CYB 750 has 377,000 words. Thus, the present study with the simplifying modelling already used about ninety percent of the central memory. The theoretical basis for the modelling is presented in section 3.3.

Two computer programs were used for this study. Program NEAMAH and EIGGRAPH. Program NEAMAH is for the solutions of linear and nonlinear equilibrium problems and eigenvalue problems; program EIGGRAPH is for the plotting of the buckled shapes. Many examples have been solved and some compared with available data in the literature in order to check and validate the program and the numerical procedures. The sensitivity of the eigenvalue solutions to the tolerance used in the numerical procedure was also studied.

3.2 The FHAB.

The FHAB bridge consists of two braced parabolic ribs (without a deck). The structure has thirteen panels with 28 nodes and 38 straight beam elements. The two ribs are supported by four hinges where no translations are allowed and only rotation about the transverse axis is permitted. The details of the structure are presented in Figures 3-1 and 3-2. The Figures include the cross-sectional area A in sq. ft., the moment of inertia about the horizontal (major principal) axis, I_{xx}, the minor principal axis, I_{zz}, and the torsional constant, KT, all in ft^4 (see Figure 3-5 for local coordinates).

The transverse beam bracing between the two ribs is so oriented that the slope of its major principal axis is the average of the slopes of the two adjacent ribs (see reference 28, pp.. 288-291).

3.3 The MCSCB Bridge and Modelling.

3.3.1 Introduction.

The Cold Spring Canyon Bridge (reference 20, pp. 13-22) is a deck-bridge located about 13.5 miles north of city limit of Santa Barbara, California. The bridge has eleven panels with a 700 ft. span , and 119 ft. rise. The ribs are spaced at 26 ft. apart. The original bridge is not symmetric with respect to the crown. The model MCSCB is symmetric with the following overall dimensions and material properties:

span = 700 ft; rise = 121.25 ft

spacing between the two ribs = 26 ft

deck elevation = 133.50 ft; deck width = 28 ft

modulus of elasticity $E = 0.4175 \times 10^7$ ksf

Poisson's ratio = 0.3

A complete three dimensional model of the bridge must include the two ribs, the rib bracing, the deck, and its bracing, the columns, the towers, and sometimes, the spandrel and transverse bracings. A large number of elements are involved. In this case it has been practical to use only four panels for the bridge system. For the MCSCB four panel model, the number of elements is 54, the total number of nodes is 22, the total number of degrees of freedom is 90, and the bandwidth is 33. Eight panels can be and were used when the deck was eliminated or only the in-plane behavior of the bridge with the deck was considered.

3.3.2 Rib Cross-Sectional Properties.

For four and eight panels the ribs are illustrated in Figure 3-4 and tabulated in Tables 3-1 and 3-2.

In the following the theoretical basis for the modelling of the different components of the MCSCB is presented.

3.3.3 Modelling of the Deck.

The deck was modelled by two beams braced with x-truss bracing (see Figure 3-3). The actual deck is mainly built of a slab carried by four stringers (see Figure 3-7). The moment of inertia for each beam in the model about x-x is calculated from simple statics for two composite stringers in the actual deck. For the torsional rigidity of the two beams, a three cell box is modelled to represent the four stringers, the deck slab, and the bottom lacing (see Figure 3-6). The torsional rigidity for each beam is taken to be 1/2 of that of the three cell model.

The moment of inertia about the z-z axis for the modelled beam representing the deck and the cross-sectional area is calculated to provide a moment of inertia equal to that of the actual deck about the global Y axis. The latter moment of inertia of the actual deck, I_{Y-Y}, which is illustrated in Figure 3-7b, may be obtained as

$$I_{Y-Y} = 2.A_{CB} \left((S/_2)^2 + (S/6)^2 \right) \tag{3.3.1}$$

where A_{CB} is the cross-sectional area of the composite beam and S is the spacing between the two exterior stringers. Since I_{Y-Y} is to be provided by the two beams in the model, then I_{Y-Y} is equal to;

$$I_{Y-Y} = 2(I_{z-z} + \frac{S^2}{4} A^*) \tag{3.3.2}$$

where I_{z-z} is the moment of inertia about the z-z local axis parallel to the global Y axis. Taking the latter to be twice the local moment of inertia about the local z-z axis of a composite beam, then A^*, the cross-sectional area of a model deck beam, can be calculated readily.

The deck slab can be modelled by an x-truss bracing to provide the same shear stiffness as that of the slab (see Figure 3-8). Denote the shear displacement of the slab in Figure 3-8a by Δs, then

$$\Delta_s = \text{shearing strain} \cdot \ell \qquad (3.3.3)$$

$$\Delta_s = \frac{\tau}{G} \cdot \ell$$

where τ is the shear stress and G is the shear modulus, or, for a unit load,

$$\Delta_s = \frac{1 \times \ell}{A_{slab}\, G} \qquad (3.3.4)$$

The x-truss bracing shear displacement is

$$\Delta_s^* = \frac{2\, S_D^2\, \ell_d}{A_d^*\, E} \qquad (3.3.5)$$

where S_D is the force in the diagonal member, ℓ_d is the length of the diagonal member, A^*_d is the diagonal member cross-sectional area, and E is the modulus of elasticity. Substitute for each value in equation 3.3.5.

$$\Delta^*_s = [1/2 \frac{(\ell^2+S^2)^{1/2}}{S}]^2 \frac{(\ell^2+S^2)^{1/2}}{A^*_d E} \times 2 \qquad (3.3.6)$$

The bracing and slab deck are to have the same shear displacement for the same (unit) shear load. Hence,

$$\Delta_s = \Delta^*_s \qquad (3.3.7)$$

Substituting for Δ_s and Δ^*_s in 3.3.7 and simplifying, one obtains:

$$\frac{A^*_d}{A_{slab}} = \frac{G}{2E}[(\frac{\ell}{S})^2 + \frac{1}{(\frac{\ell}{S})}] \qquad (3.3.8)$$

<u>3.3.4 Modelling of Rib Bracing.</u>

The K-truss bracing for a panel between the two ribs of the real bridge is shown in Figure 3-9a. The x-truss bracing that is to replace the K-truss bracing is illustrated in Figure 3-9b. Under a unit shear load, the two trusses are to have equal displacements (sum of bending displacement Δ_B or Δ^*_B and shear displacements Δ_s or Δ^*_s), i.e.

$$\Delta_B + \Delta_s = \Delta^*_B + \Delta^*_s \qquad (3.3.9)$$

If S_c is the force in the chord members, ℓ_c is the member's length, E is the modulus of elasticity, and A_c is the chord cross-sectional area.

$$\Delta_B = 2 \sum_{n=1}^{N} \frac{S_c^2 \ell_c}{EA_c} \qquad n = 1,2,3,\ldots N \quad (3.3.10)$$

where N stands for the total number of panels - i.e.

$$\Delta_B = \sum_{n=1}^{N} \left(\frac{n}{2}\right)^2 \frac{a \text{ COT}^2\theta}{E\ A_c}\ 2 \quad (3.3.11)$$

$$= \frac{2\ a\ \text{COT}^2\theta}{E\ A_c} \sum_{n=1}^{N} \left(\frac{n}{2}\right)^2$$

where θ is the angle of the inclination for the diagonal as illustrated in Figure 3-9a; the chord length is a. For the single panel x-truss:

$$\Delta_B^* = \frac{Na\ \text{COT}^2\theta^*}{2\ A_c^*\ E} \quad (3.3.12)$$

in which A^*_c is the chord area and θ^* is illustrated in Figure 3-9b.

The shear displacement of the K-truss can be represented by the following:

$$\Delta_s = \left(\frac{2\ S_D^2 \ell_d}{A_d\ E} + \frac{2\ S_v^2\ \ell_v}{A_v E}\right) N \quad (3.3.13)$$

where the subscript d stands for the diagaonal member and v for the vertical, if S is the spacing between the two ribs. Substitute for each value in equation 3.3.13, i.e.:

$$\Delta_s = \left[2\ \frac{(1/2\ \text{CSC}\theta)^2}{A_d\ E} \ell_d + \frac{2\ (1/2)^2\ s/2}{A_v\ E}\right] N \quad (3.3.14)$$

and similarly for the x-truss bracing, i.e.

$$\Delta_s^* = (1/2\ \csc\theta^*)\frac{2\ \ell^*_d}{A^*_d\ E}\ 2 \qquad (3.3.15)$$

where;

$$\ell^*_d = (N^2a^2 + S^2)^{1/2}, \quad \ell_d = (a^2 + (S/2)^2)^{1/2}$$

$$\csc\theta = \frac{\ell_d}{S/2}, \quad \csc\theta^* = \frac{\ell^*_d}{Na}$$

Setting $A_c^*=A_c$ and substituting in equation 3.3.9, the value of A^*_d can be readily found.

A test problem involving modelling of the K-truss by X-truss was solved. The K-truss had four panels which was modelled by a one panel X-truss. The displacement of the free end of both cantilever trusses checked well which gives support to the validity of such modelling.

3.3.5 Modelling of Towers and Columns.

The tower is, in actual practice, a pier in the form of a plane frame (see Figure 3-10). It is used to support the deck vertically, by transferring the loads to the foundation directly, and laterally, by providing a stiffness in the lateral direction. For the vertical stiffness of the tower, the tower is assumed to be perfectly rigid, which can be modelled by having a support in the Y-direction (see Figure 3-10). The lateral stiffness can be estimated by evaluating the actual stiffness of the frame which amounts to 1022 kips/ft. This is represented by a truss element with length 20 ft. and cross-sectional area of 0.00489 ft^2 (see Figure 3-3 for the modelled tower). For the deck supports, only translation in the z-direction and rotations about the z and y axis are allowed. One of the two supports is a roller, as illustrated in Figure 3-3, which allows x-displacement, too.

The column cross-sectional areas for the model were increased in proportion to the increased spacing between the columns in the model over the spacing in the original bridge.

3.4 Computer Programs.

Two programs have been prepared for this study: Program NEAMAH and Program EIGGRAPH. NEAMAH may be used to evaluate the nonlinear equilibrium response and the n-eigenvalues and corresponding eigenvectors of a general space framed structure.

The program was an extension of one originally developed at Michigan State University by Jose Lange (21) for the lowest eigenvalue of a curved beam deformable in three dimensional space. It was extended for three dimensional nonlinear equilibrium problems using beam finite elements solution by Jalil Rahimadeh (22).

The author's contribution lies mainly in the increased capability and efficiency in handling load inputs, the solution for more than one eigenvalue, and the incorporation of the space truss element in the program. Subroutine BAND was rewritten for program NEAMAH, and for the "Lagrange updated solution," the rotation of the major principal axis was modified. The program coding is listed in Appendix B.

Program EIGGRAPH was developed with the aid of the consultants at the MSU Computer Laboratory to plot the eigenvectors. Plots obtained using this program can be seen in chapter IV.

All plots are plotted with scale 1:20 unless otherwise mentioned in the figure.

3.5 Validation.

3.5.1 Introduction.

The purpose of this section is to present data for the validation of the computer program (NEAMAH) and the procedures which it embodies. Comparisons with published results in the literature are made where possible. A check of the truss linear stiffness with SAPIV is made. The buckling loads of a plane truss-tower, a plane truss and beam tower are compared with the buckling of a cantilever column. The effects of tolerance on the eigenproblem solutions were also studied.

3.5.2 Vertical Stability.

For a validation of the program to solve problems involving vertical stability, the FHAB bridge was used. Equilibrium solution was obtained by use of the "updated-Lagrange" procedure. Plotted in Figure 3-11 as functions of the loading are the quarter point deflection and the determinant of the stiffness matrix. The equilibrium solution may be used to identify the bounds of the buckling load for the structure. Instability may be assumed to occur when the determinant vanishes. It is not possible using the available equilibrium solution procedure to actually pinpoint the exact buckling load due to the use of finite load increments. Instability occurs between two successive load increments in which the first is stable and the second is not. Using smaller load increments would decrease the bounds, yet the solution cost would increase very significantly. The bounds are plotted in Figure 3-11 as a dotted line.

On the other hand, an eigenvalue solution was obtained for this problem and plotted as a horizontal straight line in the figure. It is seen that the eigenvalue solution lies within the bounds of the equilibrium solution. Since the two solutions were essentially independent, the results lend credibility to both.

3.5.3 Lateral Stability

Attempts were made to compare the lateral buckling load using the eigenvalue solution with existing data available in the literature. The comparisons included results given by: (i) Equation 16.25 in Reference 10 for out-of-plane buckling of single arches, (ii) Lars Ostlund (13) for the buckling of two ribs braced with beam element, and (iii) Tokarz (4) and Almeida (15) for two braced ribs.

3.5.3.1 Equation 16.25 in Reference 10

In the book: "Guide to Structural Stability of Metal Structures" (GSSMS (10)) edited by B. Johnston, Equation 16.25 was presented for estimating the critical load for the out-of-plane buckling of a single parabolic arch rib loaded uniformly. The effect of the in-plane flexural rigidity is not considered.

Three arches were used for comparison. (i) Arch-A as presented in Figure 3-12, (ii) the FHAB, and (iii) the MCSCB (all as single arches). The results are given in Table 3-3. The equilibrium solution of arch-A is also presented in Figure 3-13. The results are seen to compare well.

3.5.3.2 Ostlund's Data

Two examples were considered, both for structures with two ribs braced with cross beams only (Vierendeel type). Figure 3-14 gives the arch dimensions, the loading condition, and material properties. The properties of the cross-sections of the various elements are presented in Table 3-4. The comparison of results is listed in Table 3-5 for examples one and two. The tabulated values are for $c = q_o L^2/EI_o$ where q_o is the critical axial compression at the crown. The three values of c that appear in Table 3-5 are due to different approximations employed by Ostlund as noted in Table 3-5.

Note that three types of "bridges" were considered. The "No-Deck" type represents the two braced ribs with loads applied at the panel points of the ribs. The "Through Bridge" type denotes the case in which the loads are applied to the arch ribs through rigid hangers that are jointed to a laterally rigid deck (from which the vertical load is supposed to be applied). The "Deck Bridge" type denotes the case in which the loads are applied to the arch ribs through rigid columns that are joined to a laterally rigid deck. Analyses of the latter two types are known as "tilted load" effects as discussed previously.

It is seen that Ostlund's results are generally not sensitive to the approximations used. Where the approximation resulted in appreciable differences, it is interesting to note that the results obtained using NEAMAH lies in the range given by Ostlund's approximations.

It may be noted that a substantial difference exists in values of the first buckling load for the "Through Bridge" between Ostlund's data and the NEAMAH results. The difference may be explained by the fact that the much lower value given by NEAMAH corresponded to a mixed buckling mode in a truly three dimensional solution. Ostlund's solution corresponded to a prescribed purely lateral buckling mode; i.e., it probably missed this lower mode.

3.5.3.3 The Data of Tokarz and Almeida.

Many tests were run by Tokarz (4) for a single rib, two ribs braced at the crown with one beam and two ribs braced at a number of panel points. The results of two tests were compared herein. They correspond to test number

27 and number 33 in reference 4. Each test was conducted on two aluminum ribs, 2024-T3 with modulus of elasticity E = 10.7 x 10^6 psi and Poisson's ratio = 0.3. The ribs were 0.192 in. thick and 1.5 in. deep. The ribs were fixed at the two end supports and real deck was constructed. Test number 27 was braced with 15 equally spaced round bars of 3/32 in. diameter and test number 33 was braced with 15 equally spaced round bars of 5/32 in. diameter. The two structures were also analyzed by Almeida (15). Comparison of the results is listed in Table 3-6. It is seen that the agreement is generally quite good.

3.5.4 Truss Elements.

To check the program for the addition of the truss elements SAPIV was used. Two problems were solved. The first was a system built of fourteen truss elements and eight nodes subjected to one concentrated load. Program NEAMAH and SAPIV gave the same results for both linear displacement and axial forces. Secondly, a system consisted of four columns (beam elements) supporting a horizontal truss grid of five elements subjected to one concentrated load at one of the corner nodes. Again the agreement was total.

For the study of the truss elements in a buckling problem, the buckling load of a tower built of truss elements (see Figure 3-15) was calculated by the eigenvalue program and compared with the Euler buckling load of a cantilever column. Then the columns in the tower were replaced by beam elements and the buckling loads computed again.

For the truss tower (see Figure 3-15), its buckling load may be estimated as that of a column with a moment of inertia $I_t = 2A_t(S/2)^2$ where S is the width of the truss. For beam-truss tower, similarly the buckling load may be estimated as that of a column with a moment of inertia $I_{BT} = I_t + I_B$ where I_B is twice the moment of inertia of the beam element.

Table 3-7 presents the buckling loads obtained by NEAMAH of the truss tower and the truss-beam tower as described in Figure 3-15 and the equivalent Euler column buckling loads. The buckled shape of the truss-beam tower is illustrated in Figure 3-16. The buckled shape of the truss tower is similar.

3.5.5 Tolerance Effect.

3.5.5.1 Equilibrium Solutions.

Three different tolerances were used for the purpose of checking the tolerance effect on equilibrium solutions. The values of the tolerances are $1x10^{-2}$, $1x10^{-5}$, and $1x10^{-7}$. A single rib of the MCSCB was used, subjected to a constant uniform load of 5.858 kips/ft over the full span and an incremental load of 0.571 kips/ft on one half of the bridge span. The obtained data showed that the nonlinear response was not affected by the tolerance values used.

3.5.5.2 Eigenvalue Solutions.

For this study, the buckling load of a pin-ended column was considered. The column is 120 ft. long with I_{xx} = 35.99 ft^4, I_{zz} = 3.947 ft^4, and KT = 10.97 ft^4. It was represented by 12 beam elements.

The results indicated that the eigenvalues as such were not affected by the value of the tolerance used in the solution. However, the sequence in which they were successively generated by the procedure was. Thus the smaller the value of the tolerance, the better or more reliable the sequence of the eigenvalues is. Table 3-8 lists the first five eigenpairs corresponding to two values of tolerances used.

Note that for the case of the higher tolerance, the third mode was obtained ahead of the second mode, the fourth mode was not even in the picture. For the lower tolerance, the first three modes were produced in the proper order. The fifth was generated before the fourth. For this simple example structure, the proper values for the modes are known a priori. This is not generally the case for a complex structure for which the eigenvalues are being sought. Therefore, one must use the method with some caution, although for all cases considered herein the first mode was generated first. It should also be mentioned that irrespective of the order of generation, the mode shape obtained always corresponds to the correct buckling load. Of course, the buckling mode itself contains much information.

CHAPTER IV

BUCKLING LOADS AND MODES

4.1 Introduction.

Elastic buckling loads, as a type of limit load, are of interest by themselves. In addition, as was explained in Chapter II, they are needed for the amplification factor method which may be used to estimate the nonlinear response.

As discussed earlier in Chapter I, extensive data are available on factors affecting the buckling loads. However, previous researchers had studied either lateral buckling or in-plane buckling, i.e., where each case is treated separately. Furthermore, all previous works on three dimensional elastic stability involved cross beam bracing and associated parameters only.

The new aspects of the stability problem of arch bridges that are considered in this research include: (1) the general stability of arch bridge in three dimensional space without having to limit the problem a priori to either in-plane or lateral buckling; (2) the effect of the finite stiffness of the deck on in-plane and out-of-plane stability; (3) the different types of bracing in the bridge system and their effects on the overall stability.

The data obtained here will be given in terms of $\bar{w} = w_c/w_y$, where w_c is the uniformly distributed buckling load, and w_y is the uniformly distributed yield load evaluated as follows:

$$w_y = \frac{8f\sigma_y A}{L^2}$$

where f is the rise, σ_y is the yield stress, A is the average cross-sectional area of the arch rib, and L is the span. The tolerance used in the eigenvalue solution is 1 x 10^{-6}.

4.2 Bracing Between Ribs - Bracing Patterns.

Different types of bracing patterns have been used in practice. In this study, several different common patterns of bracing are compared based on equal volumes for all patterns.

The different patterns used for this study are defined in Figure 4-1, which includes (a) X-truss, (b) K-truss, (c) Diagonal, and (d) Transverse bracings. The arch ribs used for this study were modelled from the Cold Spring Canyon Bridge as described in Chapter III. The total volume, V, of the bracing members for the X-truss bracing case was derived using the same modelling approach presented in that chapter using eight panels. This resulted in cross-sectional area of diagonal members, A_D = 0.341 ft.^{2} and transverse members, A_T = 0.341 ft.^{2} for the eight-panel model, the volume of the X-truss case is:

$$V = 16\ A_D\sqrt{s^2 + \ell^2} + 7\ A_T\ s = 778.46\ \text{ft.}^3 \qquad (4.2.1)$$

in which s (= 70 ft.) and ℓ (= 87.5 ft.) are the spacing between the two ribs and the panel length, respectively. For the K-truss bracing case, the volume is:

$$V = 16\ A_D \sqrt{(s/2)^2 + \ell^2} + 7\ A_T\ s = 778.46\ \text{ft.}^3 \tag{4.2.2}$$

Using the same value for A_T as for the above X-truss case, A_D evaluated from equation 4.2 is equal to 0.4055 ft.^{2}

Following a similar procedure, the volume of the diagonal bracing case is given by:

$$V = 16\ A_D \sqrt{s^2 + \ell^2} \tag{4.2.3}$$

for equal volume, A_D = 0.4342 ft.^{2} For transverse bracing, the volume is $V = 7A_TS$ and the cross-sectional area of each transverse beam is A_T = 1.5887 ft.^{2} The beam flexural and torsional stiffnesses were calculated assuming that the cross-sectional depth is twice as much as its width and its thickness equal to 1/12 its width.

Five buckling loads were obtained for each bracing pattern. The results are presented in Table 4-1. The buckling loads are given in terms of w_c/w_y in which, as previously, w_c is the uniform buckling load and w_y is the uniform yield load. Some of the buckled shapes are illustrated in Figures 4-2 through 4-13. The buckled shapes are also summarized in Table 4-1. The following abbreviations were used for that purpose.

In-plane:	The arch buckled "mainly" in the x-y plane. ("mainly" implies that the maximum displacement in that plane is at least one order of magnitude larger than those in the other orthogonal planes.)
Out-of-plane:	The arch mainly buckled in the z-direction.
Lat.-tors:	The arch buckled in a lateral-torsional mode.
3-D mixed:	The buckled shape is three-dimensional and of a nondescript form.
Sym.:	Symmetric.
Anti-sym.:	Anti-symmetric.

An examination of the data presented indicates the following.

1. The lowest in-plane buckling loads and corresponding modes are essentially the same for all cases. This shows that the lowest in-plane buckling load is independent of lateral bracing.
2. The lowest in-plane buckling mode is the first mode (among all modes--in-plane and otherwise) in all cases except for the beam-bracing case where it is the fifth.
3. For the beam bracing case, the first four modes are all of the lateral buckling type with buckling loads less than the lowest in-plane buckling load. Thus the beam bracing is the weakest pattern.
4. The second in-plane buckling loads are also largely independent of lateral bracings. For the X-bracing, the second in-plane buckling load corresponds to the second mode, for the K- and D-bracing it corresponds to the

third mode. The lowest lateral-torsional mode is the third for the X-bracing ($w_c/w_y = 1.447$) and second for the K- and D-bracings ($w_c/w_y = 1.271$, and 1.218 respectively). These data would lead to the observation that whenever lateral displacements are involved in the buckling mode the beneficial effects of the bracing seem to increase in the order of D-truss, K-truss to X-truss.

In summary:

1. The pattern of lateral bracing is essentially negligible if the buckling mode is in-plane.
2. If the buckling mode involves appreciable lateral displacements, then the effects of the bracing would increase in the order of B-bracing, D-truss, K-truss, and X-truss patterns. The differences between beam and truss bracings are substantially larger than those among the different truss-bracings themselves.

4.3 Bracing Between Ribs - Amount of Bracing.

The preceding section - 4.2 considered the effect of bracing pattern on the buckling behavior of two braced ribs. In that study the amount of bracing (as referenced by the total volume) of bracing material was fixed. In this section, the effect of the amount of bracing is considered. Two bracing patterns are considered: (a) X-truss bracing for the MCSCB ribs, and (b) Beam-bracing for the FHAB ribs. For each case the amount of bracing will be varied.

4.3.1 MCSCB Ribs (with X-truss bracing).

For the case of truss-bracing, a study of the available data on several existing bridges (see reference 20, chapter 13), showed that the ratio of the bracing cross-sectional area, A_{bo}, to the rib cross-sectional area, A_R, is approximately 1/6. Using A_{bo} as a reference cross-sectional area, the following ratios were employed: $A_b/A_{bo} = 0.001$, 0.01, 0.1, 0.25, 0.5, 1.0, and 2.0. Since, as indicated by the preceding section, lateral bracing has little effect on in-plane buckling, and to save computing time, the in-plane moment of inertia of the ribs were increased by 20 times in order to force the out-of-plane buckling to be the lowest mode of buckling.

The results of the MCSCB X-truss bracing are plotted in Figure 4-14 and the buckled shapes are presented in Figures 4-15 through 4-20. Observation of the presented data leads to the following conclusions:

1. Figure 4-14 shows that practical variation of the bracings stiffness does not affect significantly the out-of-plane buckling load.
2. As the (A_b/A_{bo}) ratio is reduced the buckling load of the bridge approach the case of a single rib. Such reduction is very significant when the ratio is less than 0.1 and the buckled shape changes from anti-symmetric out-of-plane to symmetric out-of-plane.
3. In the range of $A_b/A_{bo} = 0.1$ to 1.0 the buckling load increases linearly with bracing. In the parameter range

of 1.0 to 2.0 the buckling load increases at a higher rate.

4. Comparing the buckling load of an X-truss braced bridge in Table 4-1, the out-of-plane anti-symmetric mode were (w_c/w_y = 1.45) and that of Figure 4-14 were (w_c/w_y = 1.864), one notices that the latter is higher even for less amount of bracing. This is merely due to the increase in the in-plane stiffness of the bridge. Such an effect is not considered in the formula given by reference (10), as discussed in Chapter III.

5. Within practical range of bracing, for a bridge braced with X-truss bracing the lowest out-of-plane buckling mode is anti-symmetric, whereas, for beam-braced bridge the lowest out-of-plane buckling mode is symmetric. (see Ostlund, Tokarz, and the FHAB). This may be due to: (1) For the symmetric mode the outer rib would be subjected to tension and the inner one to compression, while the anti-symmetric mode could take place inextensibly, and (2) for the case of diagonal bracing, it would appear that the diagonals would be strained to a greater degree in the symmetric mode than the anti-symmetric mode (See Figure 4-21).

4.3.2 FHAB Ribs (with Beam Bracing).

In this section, the amount of bracing and spacing effect on the bridge buckling is studied. The bridge is not forced to buckle out-of-plane. As presented earlier in Chapter III the lowest buckling load is in-plane. The data for various amounts of bracing is compared with the single rib where no bracing is used. Results are presented in Table 4-2. Observation of the data leads to the following:

1. The data presented in Table 4-1 and 4-2 shows that the lowest buckling load for a properly designed bridge is in-plane.
2. The table showed that small variation in the amount of bracing did not affect the in-plane buckling until the bracing was reduced to 1.25%. Then the bridge buckled out-of-plane.
3. Increasing the spacing by three times did not affect the in-plane buckling.
4. With 1.25% reduction in the bracing the out-of-plane buckling load was still higher than that of a single rib.

4.4 In-plane Effect of Deck.

As was mentioned in Chapter II, a deck situated above the arch rib reduces the out-of-plane buckling load. No data are available on the tilted load effect on in-plane stability, and no actual study has been reported on the finite deck stiffness effect. The latter factors will be studied in this section.

To conduct the study, the MCSCB, 8-panels, single rib, with and without a deck was used. Five cases of two braced ribs were examined:

1. A reference case of no deck.
2. A truss deck, load applied on the deck--Only tilted load effect is expected, since the truss deck has no in-plane (with reference to arch ribs) stiffness.

3. Beam deck, load applied on the deck--Both tilted load and deck stiffness effects are expected.
4. Beam deck, load applied on the rib--Only the deck stiffness effect is expected.
5. Truss deck, load applied on rib--No deck effect is expected.

Results are presented in Table 4-2 in terms of $\bar{w}$. An examination of the data leads to the following observations:

A. Tilted Load Effect

1. From cases one and two, the tilted load effect is estimated as

$$T_1 = 0.6714 - 0.5184 = 0.1530$$

2. From cases three and four,

$$T_2 = 0.7297 - 0.5612 = 0.1685$$

Thus, the tilted load effects is for T_1, 0.1530/0.6174 = 24.8% and, for T_2, 0.1685/0.7297 = 23.1%

B. Deck Stiffness Effect

1. From cases two and three the effect of deck stiffness is estimated as

$$S_1 = 0.5612 - 0.5184 = 0.0428$$

2. From cases four and one, deck stiffness effect may be estimated as

$$S_2 = 0.7297 - 0.6714 = 0.0583$$

Thus, the deck stiffness effect is, for S_1, 0.0428/0.5184 = 8.3% and, for S_2, 0.0583/0.6714 = 8.7%. It may be interesting to note that the ratio of the deck to the rib flexural rigidity is 0.10. It had been mentioned (19) that the buckling load should be increased in direct proportion to the sum of the bending rigidities of the ribs and the deck. The preceding data indicated that the increase is somewhat less than that.

4.5 Effects of Type of Deck-Rib Connections on In-Plane Buckling

In the preceding section, the deck effect on the in-plane buckling load of the arch bridge was studied. All columns were assumed pinned. Therefore, no shear transfer is assumed between the deck and the arch ribs in the longitudinal direction. In practice, arch bridges are designed so that such shear transfer would take place. This, of course, depends on the connection between the ribs and the deck. Such connection is studied in this section.

Shear connections usually used in practice are:

1. Rigid "moment" connection at one or more intersection panel point (see Fig. 4-22a).
2. In-plane spandrel bracing at one or more panels (see Fig. 4-22b), and
3. The ribs and the deck are rigidly connected at the crown, i.e., for this case the deck is at the same elevation as the crown is.

The MCSCB, 8-panels, with deck as presented in Chapter III was used.

The results obtained for rigidly connected columns and spandrel bracing are shown in Table 4-4 and also plotted in Figure 4-23. Some of the buckled shapes are presented in Figures 4-24 through 4-28. In Table 4-4 case one denotes the reference case, which is the case of no shear transfer. Cases two and three represent perfectly rigid connection of the crown column and all columns respectively. For this study the moment stiffness of each rigid column was varied.

Studying the present results indicate the following:

1. The buckling load almost doubled with only the crown column rigidly connected.
2. An increase of the bending stiffness by one order of magnitude would force the lowest buckling mode to the symmetric mode.
3. As expected, by rigidly joining all ends of all columns, the lowest buckling load increased by 40% over the case of only the crown column is rigidly connected.
4. Cases four and five refer to spandrel bracing of one and two middle panels. The case A_D/A_o = .325 corresponds to the use of a 1 5/8 in. rope. It is obvious that the use of spandrel bracing increases the buckling load (P_c) but there is no substantial difference between the different cross-sectional area or when the spandrel bracing is extended to the case of two braced middle panels.

The effect of column heights is shown in Table 4-5. The column heights are defined by the crown column with all other columns varying to conform to a horizontal deck. All columns are pinned unless otherwise noted.

It is seen that as the columns shorten P_c decreases. This is due to the P-Δ effect. The shorter the column, the greater the de-stabilizing effect or the P-Δ effect. When the ribs and deck are joined together with no column at the crown the P_c increases because the bridge is forced into a symmetric mode, likewise for the case the $h/h_o = 0.08$ but the crown column is rigidly connected where essentially the same P_c is achieved with the same symmetric buckling mode.

4.6 Effects of Towers, Deck, and Transverse Bracing on Lateral Buckling.

The two towers in a deck bridge support the deck which is situated above the ribs and meet with the two ribs at the foundation. The two towers are part of the structural system and their rigidity should be significant to the overall stability of the bridge. For vertical buckling, the towers are very rigid and no significant effect is expected. For lateral buckling the two towers work as laterally loaded frames. This makes lateral stiffness of the bridge affected by the lateral stiffness of the tower. The deck in-plane effect was discussed in section 4-4. The out-of-plane effect of the deck has been extensively studied in the literature, yet all works looked at the deck as very rigid and correspondingly only tilted load effects were considered. The tilted load (P-Δ) effect is very significant but the deck stiffness would seem to be a very important part of the deck effect. The latter effect can only be studied by assuming that the deck is not rigid, and that it actually undergoes elastic deformations in three dimension.

Transverse bracing (see Figure 3-3) is often used for steel bridges. Transverse bracing adds to the global stiffness of the bridge.

The bridge used for this study is the MCSCB, 4 panels (see chapter III - Table 3.1). The deck and the tower was modelled and included in the same table. The transverse bracing was studied once by bracing the MCSCB at the crown intersection panel only and by uniformly bracing the bridge (i.e., bracing every intersection panel except at supports.).

To study the deck lateral stiffness the flexural rigidity of the deck, I_{yy} (as modelled in Chapter III), was reduced by 10, and increased by 10. Similarly the lateral stiffness of the tower was reduced by 10, and increased by 10.

The results are available in Table 4-6 and buckling is presented as $\bar{w}$. The buckled shapes are presented in Figure 4-31 through Figure 4-40 and the deck lateral stiffness and the tower lateral stiffness as compared to the classical rigid deck effect are illustrated in Figure 4-41. A study of the data leads to the following conclusions:

1. The first row in Table 4-6 shows the basic reference case, or the case with no transverse bracing, and hence no shear transfer.

   The lowest buckling load (P_c) is $\bar{w} = 0.169$ and both the deck and the rib buckled of the same magnitude, the second P_c is $\bar{w} = 0.613$ which is in-plane anti-symmetric including deck effect (P-Δ and deck stiffness). The third P_c is $\bar{w} = 0.899$ out-of-plane anti-symmetric with no in-plane buckling and the tower is involved in some lateral deformation with the deck. The fourth P_c is $\bar{w} = 1.155$ and is just like the third except with additional in-plane buckling which makes it generally three-dimensional. The fifth mode has a similar configuration.

2. The rows 2a and 2b correspond to transverse bracing at the crown intersection panel only and at all intersection panels, respectively. The previous half wave symmetric mode out-of-plane ($\bar{w} = 0.169$) was eliminated by the bracing and the lowest P_c ($\bar{w} = 0.613$) corresponds to an in-plane mode. The lowest out-of-plane is anti-symmetric with $\bar{w} = 0.9$.

3. The effect of the deck lateral stiffness is shown by rows 3a, 3b, and 3c, which correspond to lateral stiffness of 1/10, 10, and perfectly rigid (using the classical solution known as the tilted load effect). All cases are without transverse bracing.

   In conjunction with case one, the deck lateral stiffness increase the lateral P_c but even with infinitely rigid deck the lateral P_c is smaller than that of in-plane buckling.

4. The effect of tower lateral stiffness is shown in row 4a, and 4b. As expected, the tower lateral stiffness increases P_c. However, the relation is not linear. The results are presented in Figure 4-41 which shows that increasing the tower lateral stiffness by 10 times, P_c is increased by only 12%.

In summary:

Transverse bracings are very effective and it seems important to have at least one intersection-panel braced.

Deck and tower stiffness do matter. But the influence is not linear. Existing designs seem to be effective and large increase of their lateral stiffness would not greatly increase the lateral stability of the existing designs.

CHAPTER V

NONLINEAR RESPONSES

5.1 Introduction.

The amplification factor method as a means of estimating the nonlinear response to loads in the lateral, longitudinal, or vertical direction is studied in this chapter. Results are compared with nonlinear equilibrium solutions as obtained by use of program NEAMAH.

5.2 Lateral Response.

5.2.1 Nonlinear Equilibrium Solution.

Using the FHAB bridge (Chapter III), a solution was obtained for a constant uniformly distributed vertical load, w_f, and variable uniformly distributed lateral load, w_a, (see Figure 2-3c). Results are presented in Figure 5-1 and 5-2.

Figure 5-1 shows that at the crown, for the fixed w_f and increasing lateral load w_a, the vertical displacement increased nonlinearly, but the lateral displacement was linear. Figure 5-2 shows that the latter type of linearity held not only at the crown but also at other points along the bridge. It should be noted that the above-mentioned linearity is with respect to w_a. For a given w_a, the response with respect to varying w_f would obviously be nonlinear.

5.2.2 Nonlinear Lateral Response by the AF Method.

The accuracy of the amplification factor method may be considered using the same combinations of lateral and vertical loading conditions as in the preceding section. Three different cases were employed, each for a different value of fixed w_f/w_c and variable w_a. The amplified responses were compared with "actual" nonlinear response (obtained from equilibrium solutions).

The AF was computed from the following:

$$AF = \frac{1}{1 - \alpha} \tag{5.1}$$

where $\alpha = w_f/w_c$, w_f is the fixed vertical uniform load, and w_c is the "compatible buckling load," (see Chapter II). For the FHAB used for this analysis the lateral buckling load was $w_c/w_y = 2.264$. This is not the lowest P_c for the bridge model. It is the second.

Knowing the values of w_f and w_c, the lateral nonlinear response R_n, due to the additional load w_a, was computed from:

$$R_n = R_L \times AF \tag{5.2}$$

in which R_L is the linear response.

A reference lateral load due to a wind pressure at 100 mph wind velocity was used for this presentation. Results are presented in Table 5-1. The presented data shows the values of w_f/w_c, the Amplification Factor, the linear response, the nonlinear response, and the ratio of the estimated to the "actual" nonlinear response. The data show that the amplification factor method yielded good estimates of the nonlinear responses in the lateral direction.

5.3 Longitudinal Response.

5.3.1 Nonlinear Equilibrium Solution.

In a similar manner to that presented in the preceding section, the nonlinear equilibrium solutions for longitudinal displacements were obtained. The FHAB bridge was used with several cases of fixed load, w_f, each accompanied by variable longitudinal additional load, w_a (see Figure 2-3b).

Results are presented in Figure 5-3 for the crown point. It is seen that for a given w_f, and increasing w_a, the vertical displacement increased nonlinearly, but the longitudinal displacement was linear. This situation is similar to that of the lateral loading as discussed in the preceding section.

5.3.2 Nonlinear Longitudinal Response by the AF Method.

Using the FHAB bridge model and the procedure outlined in section 5.2.2 for estimating the nonlinear response, the amplification factor method for nonlinear longitudinal response was considered. In this case the compatible buckling mode is the lowest in-plane anti-symmetric mode with a corresponding buckling load equal to $w_f/w_c = 1.052$. This is the lowest critical buckling load for the FHAB bridge model.

The results are presented in Table 5-2 where the same reference wind pressure load was used as previously.

The data shows that the amplification factor method again provided good estimates of the maximum longitudinal responses for the arch bridge model.

5.4 Vertical Response.

5.4.1 General.

In the preceding sections, w_f, was applied in the vertical direction but the responses considered were orthogonal to the vertical direction, i.e., the linear and nonlinear responses were in the same direction of w_a. Since the response considered in this section is vertical, the effect of w_f is to be included in the computation of the linear response, i.e.,

$$R_L = R_L(w_a) + R_L(w_f) \tag{5.3}$$

Depending on the loading pattern, the compatible buckling load would be different. In Chapter IV it was found that the deck-rib connection condition significantly affected the buckling mode and load. Using the MCSĆB bridge, eight panels, with deck, three cases were considered for studying the amplification factor method as applied to vertical loading. They are discussed in the following.

5.4.2 All Columns Pinned.

In this case the lowest buckling mode was anti-symmetric, $w_c/w_y = 0.561$, α values used were 0.15, 0.25, and 0.5

and the loading condition was presented in Figure 5-4a. The results are presented in Table 5-3 for the quarter point under the additonal load w_a. It is seen that the AF method provided good estimates with an "error" generally less than 10%. The AF method tended to underestimate the maximum response as the value of w_a increased.

5.4.3 Crown Column Rigidly Connected and All Other Columns Pinned.

For this type of connection, two subcases were considered depending on the loading pattern.

5.4.3.1 Uniform Loading.

The loading condition for this case is shown in Figure 5-4b. The corresponding buckling load was $w_c/w_y = 1.065$. The values of α used were also 0.15, 0.25, and 0.5.

The results are presented in Table 5-4 for the quarter point under the additional load w_a. It is seen that the accuracy of the estimates provided by the AF method was somewhat lower than the previous case. But, they may still be considered good if the loads w_a/w_c are not greater than, say, 0.10.

5.4.3.2 Nonuniform Loading.

The loading condition as shown in Figure 5-4c, with the ratio of the left half-span loading to the right half-span loading equal to 1.2, and the applied load parameter w is incremented. Note that w is the only applied load (comparable to $w_a + w_f$), and $\alpha = w/w_c$. This means that the "amplification factor," AF, is variable for each load increment. The range of α used is 0.15 to 0.55.

The results are shown in Table 5-5 and Figure 5-5 also for the quarter point under the larger loading. The presented data show that at least for this pattern of nonuniform loading the amplification factor method produces good conservative approximations to the nonlinear responses. The range of validity in this case is larger than the previous cases.

<u>5.4.4 Ribs and Deck Rigidly Connected</u>.

The loading pattern for this case is presented in Figure 5-4b. The compatible buckling mode is symmetric (also corresponds to the lowest buckling load, w_c/w_y = 1.468). The values of α used were 0.147, 0.254, and 0.423.

The results are presented in Table 5-6, for the crown, which shows that the amplification factor method gives good estimations for the nonlinear responses. For α = 0.147, the error was about 8% to 10%; for α = 0.254, it was between 2% to 9%, and for α = 0.423 it was between 4% and 16%.

CHAPTER VI

SUMMARY AND CONCLUSION

6.1 Summary.

6.1.1 General.

The objective of this research was to examine the problem of elastic stability of arch bridges and to consider a simple approximate method for estimating the maximum nonlinear response. The method, which uses the linear response and an amplification factor (function of the buckling loads) is called the "amplification factor method."

Computer modelling was employed to compute the buckling loads and modes and to obtain the nonlinear equilibrium solutions needed for checking the above mentioned amplification factor method. The bridge models used were three dimensional and quite complete, each including two ribs, bracings between ribs, deck system, longitudinal and lateral bracings between the deck and the ribs, and end towers.

A computer program, NEAMAH, was developed for use in this study through modification and expansion of certain available ones. The program was validated by extensive corroborations with known data.

The obtained results were in two groups, summarized as follows.

6.1.2 Buckling Loads and Modes.

(i) Effect of rib bracing--It was found that truss bracing, as compared to beam bracing (Vierendeel), could increase the lateral buckling load by as much as three times for the same amount of bracing material.

(ii) Longitudinal shear transfer--Providing a rigidly connected column to transfer shear between the deck and the ribs, rigidly connecting the deck and the ribs, or use of spandrel bracing increased the in-plane buckling load two to three fold. Such shear transfer should be provided for the stability of the bridge in the vertical plane.

(iii) Effects of the deck--A deck situated above the ribs has a positive and a negative effect on in-plane buckling. It provides an additional stiffness in the vertical plane. This added stiffness increased the buckling load. But the increase was a little less than what the ratio of the deck to the ribs flexural rigidities would indicate, as was suggested by some investigators. The "negative" effect of the deck is analogous to the "tilted load" effect for lateral stability (similar to the so-called "P-Δ effect"). For the MCSCB model considered, such effect was about 23 to 25% of the overall rib buckling load. The stiffening effect was 8-9%. Thus, the softening effect due to the deck was much greater than the stiffening effect.

(iv) Lateral stiffness of the tower and deck--Corresponding to a reduction and an increase of the deck lateral stiffness by ten-fold, the buckling load was reduced by 80% and increased by 55%, respectively. Similarly for the tower, the lateral stiffness was reduced and increased by ten-fold and the buckling load was reduced by 50% and increased by 12% respectively.

(v) Effect of transverse bracing--Transverse bracing between the ribs and the deck increased the lateral buckling strength markedly. For even with one transversely braced panel, the buckling load would be increased by more than three-fold over the case with no transverse bracing.

6.1.3 Accuracy of the Amplification Factor Method.

The lateral nonlinear response was presented for w_f/w_c = 0.0627-0.2547 and w_a/w_{100} = 0.5-83. For the wide range of data presented for the FHAB (beam bracing only) the "error" was no more than 2%.

The longitudinal nonlinear response, obtained for w_f/w_c = 0.0489-0.4888 and w_a/w_{100} = 3.3-52.3, had errors that were no more than 2% in each case. The error can be 20% for w_a/w_{100} higher than 52.3.

The vertical response was studied for three cases of deck-ribs connections.

(i) All Columns Pinned--For w_f/w_c = 0.15-0.5 and w_a/w_c = 0.0325-0.1625, the error in estimating the vertical nonlinear response was not more than 10%.

(ii) Crown Column Rigidly Connected (All Other Columns Pinned)--Two subcases were considered.

(a) For a uniform w_f/w_c = 0.15-0.5 and w_a/w_c = 0.0325 - 0.1625, the data were not as good as previously. The error approached, in some cases, 20%. However, for

w_a/w_c less than 0.10, it was less than 10%. For the case where $w_f/w_c = 0.5$ and $w_a/w_c = 0.1625$, the error jumped to 38%.

(b) In this case, the buckling load was based on the same nonuniform load pattern--As sampled for a range of $w/w_c = 0.1509\text{-}0.5460$ all the amplified responses were conservative and the maximum error was about 11%.

(iii) Deck and Ribs Rigidly Connected--The data were obtained for $w_f/w_c = 0.147\text{-}0.423$ and $w_a/w_c = 0.0325\text{-}0.1625$. the error was not more than 10% except that for the case $w_f/w_c = 0.423$ and $w_a/w_c = 0.0325$, the error increased to 16%.

In the application of the amplification factor method, it is essential that the amplification factor be computed using the "compatible buckling load" (the buckling load that corresponds to a buckling mode conformable to the response under consideration).

6.2 Concluding Remarks.

Because of cost, only two bridge models were used in this study, but the qualitative aspects of the results should be applicable to deck arch bridges in general. The buckling loads and modes indicated that the problem should be considered as one of a three-dimensional system so as not to miss any mixed mode buckling load that cannot be predicted by a formulation that rules out mixed mode a priori.

Current design practice seems to provide some form of shear transfer between the deck and ribs, for example, using bracing members or rigid connections. The practice seems adequate so far as elastic buckling is concerned. The stiffness of the end towers also seems adequate.

The use of the amplification factor method for the estimation of the nonlinear response appears to be quite promising for practically all types of loading. Of course, more data are needed to extend and/or establish the range of validity.

This study has focused on geometric nonlinearity of the response of deck type arch bridges. The critical buckling loads have been presented in terms of the yield load. It was obvious that in some cases yield would occur before elastic buckling could take place. Therefore, it should be natural that material nonlinearity be considered for future research.

Table 3-1 Cross-sectional Properties of MCSCB Four Panels, with Deck (for element number identification refer to Figure 3-4 a, b, and c.)

Element number	A	I_{xx}	I_{zz}	KT
1	2.6559	35.99	3.9390	21.0100
2	2.9058	41.58	4.1260	24.7500
3	0.7860	3.72	2.1800	1.3600
4	0.9000	--	--	--
5	1.0891	1.27	0.4165	0.9661
6	1.0891	--	--	--
7	14.7800	--	--	--
8	14.7800	--	--	--
9	0.00489	--	--	--

Table 3-2 Cross-sectional Properties of MCSCB - Eight Panels (for element number identification, refer to Figure 3-4 d).

Element number	A	I_{xx}	I_{zz}	KT
1	2.4000	30.28	3.7400	24.88
2	2.9058	41.58	4.1260	25.00
3	3.0300	44.52	4.2200	25.30
4	2.6559	35.99	3.9390	24.95
5	0.7860	3.72	2.1800	1.36
6	0.3264	--	--	--

Table 3-3 Values of Critical Loads for Lateral Buckling (Kips/ft.)

Structure	By equilibrium solution bounds	By eigen value solution	GSSMS (Eq. 16.25)	Ratio of eigen value to GSSMS
ARCH-A	8.00 & 9.116	7.8670	8.2600	0.92
FHAB	--	4.5576	4.8738	0.94
MCSCB	--	1.8381	1.9929	0.92

Table 3-4 Cross-sectional Properties for Ostlund's Arches (see Figure 3-14).

Elements	A	I_{xx}	I_{zz}	KT
1, 2, 7, 8	111.803	1,164.6172	931.6946	1,716.9380
3, 4, 5, 6	180.278	4,882.5528	1,502.3167	3,925.8346
9, 10, 11	60.500	152.5104	610.0417	418.8794
1, 2, 7, 8	50.990	441.9100	106.2300	294.3000
3, 4, 5, 6	58.309	660.8240	121.4770	355.0300
9, 10, 11	12.000	4.0000	36.0000	13.9861

Transverse beam bracing minor principal axes orientation coincides with the average angle of the two adjacent rib angles of inclination.

Table 3-5 Comparison with Ostlund's Data (Tabulated are values of $C = q_o L^2/EI_o$)

	No Deck		Through bridge		Deck Bridge: Clamped at crown		Deck Bridge: Hinged at crown	
	1st mode	2nd mode	1st mode	2nd mode	1st mode	2nd mode	1st mode	2nd mode
			Example 1					
Ostlund								
(a)	38		95	113	47		38	
(b)	37	44	92	91	43	--	35	--
(c)	38		94	100	44		39	
	symmetric	anti-symmetric	symmetric	anti-symmetric	anti-symmetric		anti-symmetric	
Neamah	37.62 symmetric	44.7 anti-symmetric	68.19 mixed mode	--	46.59 anti-symmetric	--	38.52 anti-symmetric	--
	Example 2 (for buckled shape see above description).							
Ostlund								
(a)	55		121		50		49	
(b)	52	85	95	--	70	--	35	--
(c)	52		117		73		47	
Neamah	53.94	83.62	117.9	140.97	59.48	--	38.37	59.13

(a) "Beam-column" effect of individual elements neglected.
(b) The flexural rigidity in the vertical plane of the arches disregarded.
(c) Same as (a) and (b) and, in addition, the axial shortening of the arches disregarded.

Table 3-6 Comparison with Data of Tokarz and Almeida. Tabulated are the values of $w_c L^3/EI$ where w_c is the critical distributed load per rib

Test No.	Source	First mode (symmetric)	Second mode (anti-symmetric)
	Tokarz	53.4	--
27	Almeida	46.7	97.70
	Neamah	46.92	99.45
	Tokarz	79.7	--
33	Almeida	63.7	114.3
	Neamah	77.33	129.99

Table 3-7 Results for Cantilever Column and Tower (All values are in Kips)

		First Mode	Second Mode
Truss Tower	Neamah	0.448×10^6	3.440×10^6
	Euler	0.457×10^6	4.113×10^6
Truss-Beam Tower	Neamah	0.905×10^6	7.510×10^6
	Euler	0.914×10^6	8.226×10^6

Table 3-8 Tolerance Effect on Eigenvalue Solutions

	Sequence of Eigensolutions as Obtained				
Tolerance	1	2	3	4	5
1×10^{-6}	1[a] 484.46	3 4,364.61	2 1,938.21	5 12,202.34	7 24,435.82
1×10^{-18}	1 484.46	2 1,938.21	3 4,364.61	5 12,202.33	4 7,776.04

a - These numbers are the actual number of the mode.

Table 4-1 Buckling Loads for Different Rib Bracing Patterns (Values of w_c/w_y and Buckled Shapes).

Bracing Pattern	First Mode	Second Mode	Third Mode	Fourth Mode	Fifth Mode
X-truss	<u>0.673</u>	<u>1.406</u>	<u>1.447</u>	<u>1.736</u>	<u>1.973</u>
	in-plane anti-sym. Fig. 4-2	in-plane sym. Fig. 4-3	lat.-tors. sym. Fig. 4-4	3-D mixed	lat.-tors. sym.
K-truss	<u>0.672</u>	<u>1.271</u>	<u>1.397</u>	<u>1.652</u>	<u>1.837</u>
	in-plane anti-sym.	lat.-tors. sym.	in-plane sym.	3-D mixed	out-of-plane sym.
D-truss	<u>0.671</u>	<u>1.218</u>	<u>1.396</u>	<u>1.595</u>	<u>1.610</u>
	in-plane anti-sym. Fig. 4-5	lat.-tors. sym. Fig. 4-6	in-plane sym. Fig. 4-7	local buckling Fig. 4-8	out-of-plane mixed
Transverse	<u>0.237</u>	<u>0.300</u>	<u>0.437</u>	<u>0.574</u>	<u>0.671</u>
	out-of-plane sym. Fig. 4-9	out-of-plane anti-sym. Fig. 4-10	out-of-plane sym. Fig. 4-11	out-of-plane anti-sym. Fig. 4.12	in-plane anti-sym. Fig. 4.13

Table 4-2 Effect of Bracing Cross-Sectional Properties on Buckling Load, FHAB Bridge.

Cross-sectional Properties					w_c/w_y	Buckled Shape
A	I_{zz}	I_{yy}	KT	Spacing		
0.8316	7.997	1.428	1.976	28	1.053	In-plane anti-sym
0.6316	3.441	1.428	1.976	28	1.053	In-plane anti-sym
0.4000	2.000	0.750	1.000	28	1.053	In-plane anti-sym
0.8316	7.997	1.428	1.976	84	1.053	In-plane anti-sym
0.0100	0.010	0.010	0.010	28	0.184	Out-of-plane sym.
Single arch rib (no rib-bracing)					0.153	Out-of-plane sym.

Table 4-3 Tilted Load and Deck Stiffness Effect
All values are presented in terms of $\bar{w}$

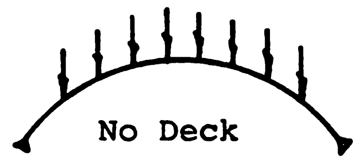
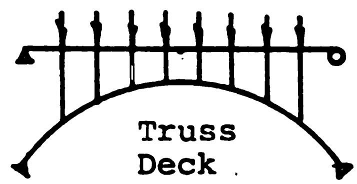
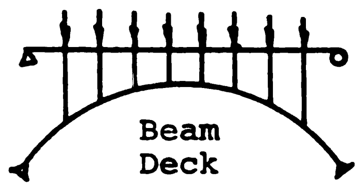
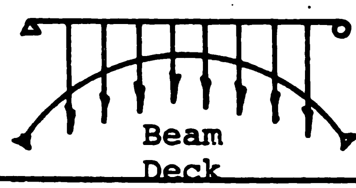
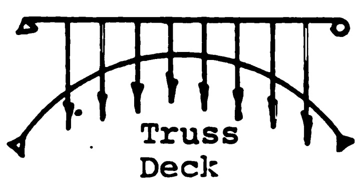
Case	Load and Deck Conditions	$\bar{w}$
1		0.6714
2		0.5184
3		0.5612
4		0.7297
5		0.6714

Table 4-4 Effects of Column Connections on In-Plane Buckling Load.

Case	Column connection type	Column Stiffness A/A_o, I/I_o and A_D/A_o	Mode 1 $\bar{W}$	Mode 1 Shape	Mode 2 $\bar{W}$	Mode 2 Shape	Figure
1	All columns pinned	1.0	0.561	ANT	1.480	SYM	4-24
2 a	Columns at	1.0	1.065	ANT	1.480	SYM	4-25
b	crown section	10.0	1.480	SYM	1.634	ANT	4-26
c	are rigidly	100.0	1.480	SYM	1.855	ANT	
d	connected others are pinned	1,000.0	1.480	SYM	1.895	ANT	
3 a	All columns	1.0	1.423	ANT	1.549	SYM	4-27
b	are rigidly	10.0	1.887	SYM	2.435	ANT	
c	connected	100.0	2.422	SYM	2.889	ANT	
d		1,000.0	2.741	SYM	3.087	ANT	
4 a	Spandrel	0.325	1.325	ND	1.623	ND	4-28
b	bracing for one middle panel	1.000	1.426	ND	2.025	ND	
5 a	Spandrel	0.325	1.560	SYM	1.805	ANT	
b	bracing for two middle panels	1.000	1.656	SYM	2.214	ANT	

A_o: Cross-sectional area of column (as used for the MCSCB).

A: Cross-sectioal area of column (as used for the present study).

A_D: Cross-sectional area of spandrel bracing.

I_o: Moment of inertia for column (as used for the MCSCB).

I: Moment of inertia for column (as used for the present study).

ANT: Anti-symmetric mode.

SYM: Symmetric.

ND: Nondescript.

Table 4-5 Effect of Column Heights on In-plane buckling Load.

Case	h/h_o	Column End conditions	Mode 1 $\bar{w}$	Mode 1 Shape	Mode 2 $\bar{w}$	Mode 2 Shape	Figure
1 a	1.0	(pinned)	0.561	ANT	1.480	SYM	4-24
b	0.5	(pinned)	0.503	ANT	1.475	SYM	
2 a	0.08	(pinned)	0.297	ANT	1.469	SYM	4-29
b	0.08	(rigid)	1.469	SYM	2.485	ANT	
3	0.0		1.468	SYM	2.505	ANT	4-30

h_o: Original column height as used for MCSCB.

h: Column height used for the present study.

ANT: Anti-symmetric mode.

SYM: Symmetric.

Table 4-6 Tower, Deck, and Transverse Bracing Effect on the Bridge Stability. Values of $\bar{W} = w_c/w_y$.

Case	Description of case study	First mode	Second mode	Third mode	Fourth mode	Fifth mode
1	Basic Model "No Transverse Bracing"	0.169 out-of-plane sym. Fig. 4-31	0.613 in-plane anti-sym. Fig. 4-32	0.899 out-of-plane anti-sym.	1.155 3-D anti-sym. Fig. 4-33	1.347 3-D anti-sym. Fig. 3-34
2a	Transverse Bracing at Crown Inter-section Panel	0.613 in-plane anti-sym. Fig. 4-35	0.899 out-of-plane anti-sym.	1.148 3-D anti-sym. Fig. 3-36		
2b	Transverse Bracing at Every Inter-section Panel	0.613 in-plane anti-sym.	0.903 out-of-plane anti-sym.	1.242 out-of-plane anti-sym. Fig. 4-37	1.436 out-of-plane sym.	1.844 out-of-plane sym. Fig. 4-38
3a	Deck Lateral Stiffness Reduced by 10.	0.030 out-of-plane sym. Fig. 4-39				
3b	Deck Lateral Stiffness Increased by 10.	0.262 out-of-plane sym.				
3c	Classical Rigid Deck	0.356 out-of-plane sym.				
4a	Tower Lateral Stiffness Reduced by 10.	0.080 axial deformation in the modelled tower Fig. 4-40				
4b	Tower Lateral Stiffness increased by 10.	0.189 out-of-plane sym. Fig. 4-41				

Table 5-1 "Lateral" Nonlinear Responses by Equilibrium and Amplification Factor Method.

w_f/w_c	AF	w_a/w_{100}	R_L ft.	R_n ft.	$\frac{R_L \cdot AF}{R_n}$
0.0627	1.0669	0.517	0.1399	0.1488	1.003
		2.580	0.6995	0.7444	1.003
		20.672	5.5960	6.0085	0.994
		113.680	30.7856	32.2625	1.018
0.1274	1.1459	20.668	5.5960	6.4492	0.994
		41.344	11.1920	12.8798	0.996
		124.000	33.5760	38.2913	1.005
0.2547	1.3417	20.668	5.5960	7.5632	0.993
		41.344	11.1920	15.1227	0.993
		82.668	22.3840	30.3560	0.989

Table 5-2 "Longitudinal" Nonlinear Responses by Equilibrium and Amplification Factor Method

w_f/w_c	AF	w_a/w_{100}	R_L ft.	R_n ft.	$\frac{R_L \cdot AF}{R_n}$
0.0489	1.0514	6.604	0.2100	0.2204	1.051
		13.208	0.4188	0.4402	1.000
0.0978	1.1084	3.302	0.1067	0.1174	1.008
		6.604	0.2100	0.2327	1.000
		13.208	0.4188	0.4639	1.001
0.1955	1.2430	3.302	0.1067	0.1330	0.997
		6.604	0.2100	0.2616	0.998
0.2933	1.4150	6.604	0.2100	0.2981	0.997
		13.208	0.4188	0.5905	1.004
0.4888	1.9562	3.302	0.1067	0.2104	0.992
		13.208	0.4188	0.8135	1.007
		52.316	1.6588	3.1955	1.016
		105.663	3.3504	5.8073	1.129
		132.078	4.1879	6.7760	1.209

Table 5-3 "Vertical" Nonlinear Responses by Equilibrium and by Amplification Factor Method* ("All Columns Pinned")

w_f/w_c	AF	w_a/w_c	R_L ft.	R_n ft.	$\frac{R_L \cdot AF}{R_n}$
0.15	1.1765	0.0325	0.3977	0.4620	1.013
		0.0650	0.7111	0.8575	0.976
		0.0975	1.0246	1.2717	0.948
		0.1300	1.3381	1.7058	0.923
		0.1625	1.6515	2.1609	0.899
0.25	1.3333	0.0325	0.4538	0.5707	1.060
		0.0650	0.7673	1.0233	1.000
		0.0975	1.0808	1.4996	0.961
		0.1300	1.3942	2.0014	0.929
		0.1625	1.7077	2.5302	0.900
0.50	2.00	0.0325	0.5942	0.9442	1.259
		0.0650	0.9077	1.6532	1.098
		0.0975	1.2211	2.4123	1.012
		0.1300	1.5346	3.2248	0.952
		0.1625	1.8480	4.0945	0.903

* For quarter point deflection under w_a.

Table 5-4 "Vertical" Nonlinear Responses by Equilibrium and by Amplification Factor Method * (Crown Column Rigidly Connected; All Other Columns Pinned, w_f Uniform)

w_f/w_c	AF	w_a/w_c	R_L ft.	R_n ft.	$\frac{R_L \cdot AF}{R_n}$
0.15	1.1765	0.0325	0.5086	0.5845	1.024
		0.0650	0.8574	1.0421	0.968
		0.0975	1.2062	1.5385	0.922
		0.1300	1.5550	2.0800	0.880
		0.1625	1.9037	2.6753	0.837
0.25	1.3333	0.0325	0.6152	0.7511	1.092
		0.0650	0.9640	1.2804	1.004
		0.0975	1.3128	1.8653	0.938
		0.1300	1.6616	2.5178	0.880
		0.1625	2.0104	3.2548	0.824
0.5	2.00	0.0325	0.8816	1.2947	1.362
		0.6500	1.2304	2.1992	1.119
		0.0975	1.5791	3.3300	0.948
		0.1300	1.9279	4.8523	0.794
		0.1625	2.2767	7.3116	0.623

* For quarter point deflection under w_a.

Table 5-5 "Vertical" Nonlinear Responses by Equilibrium and by Amplification Factor Method* (Crown Column Rigidly Connected; All Other Columns Pinned, Non-uniform Loading)

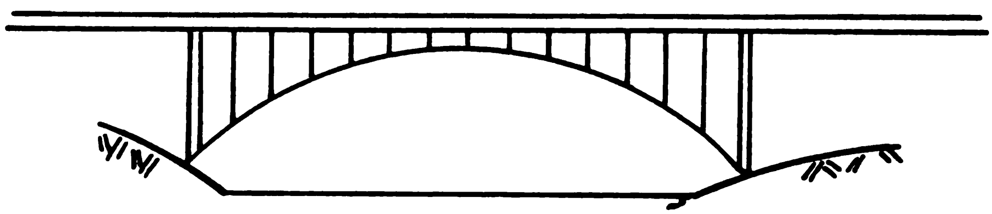
w/w_c	AF	R_L	R_n	$\frac{R_L \cdot AF}{R_n}$
0.1509	1.1778	0.4291	0.4799	1.053
0.1778	1.2162	0.5056	0.5789	1.062
0.2047	1.2574	0.5821	0.6831	1.071
0.2316	1.3014	0.6586	0.7932	1.081
0.2598	1.3510	0.7388	0.9206	1.084
0.2867	1.4019	0.8153	1.0454	1.093
0.3136	1.4569	0.8918	1.1790	1.102
0.3405	1.5163	0.9683	1.3227	1.110
0.4384	1.7806	1.2467	1.9970	1.112
0.4653	1.8702	1.3232	2.2205	1.114
0.4922	1.9693	1.3997	2.4730	1.115
0.5191	2.0794	1.4762	2.7627	1.111
0.5460	2.2026	1.5527	3.1019	1.103

* For quarter point deflection under w_a.

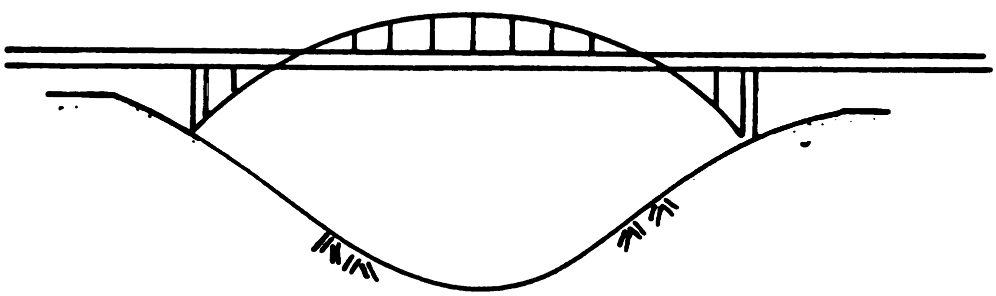
Table 5-6 "Vertical" Nonlinear Responses by Equilibrium and by Amplification Factor Method* (Ribs and Deck Rigidly Connected)

w_f/w_c	AF	w_a/w_c	R_L ft.	R_n ft.	$\frac{R_L \cdot AF}{R_n}$
0.147	1.1723	0.0325	0.4958	0.6316	0.920
		0.0650	0.7705	0.9694	0.932
		0.0975	1.0453	1.3239	0.926
		0.1300	1.3199	1.6966	0.912
		0.1625	1.5947	2.0889	0.895
0.254	1.340	0.0325	0.6485	0.8884	0.979
		0.0650	0.9241	1.2738	0.973
		0.0975	1.1998	1.6818	0.956
		0.1300	1.4754	2.1147	0.935
		0.1625	1.7511	2.5750	0.912
0.423	1.733	0.0325	0.8971	1.3352	1.164
		0.0650	1.1725	1.8332	1.109
		0.0975	1.4484	2.3718	1.058
		0.1300	1.7240	2.9565	1.011
		0.1625	1.9997	3.5943	0.964

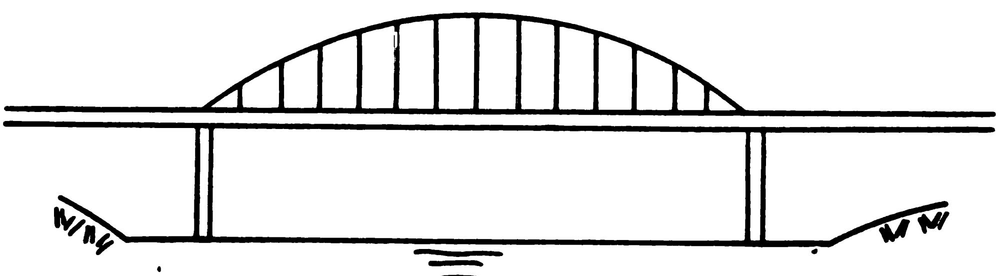
* For crown deflection.



(a) Deck Bridge



(b) Half Through Bridge



(c) Through Bridge

Figure 1-1. Types of Arch Bridges.

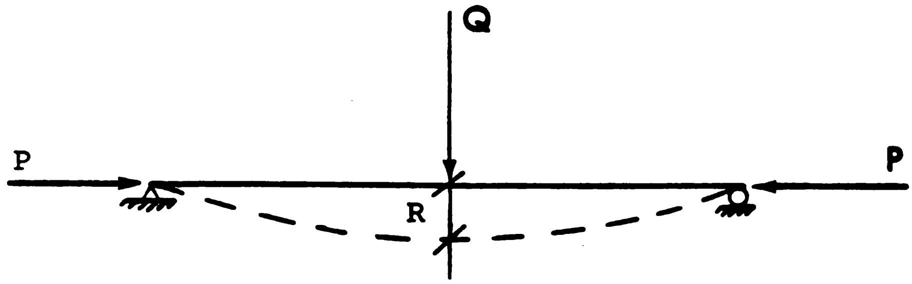
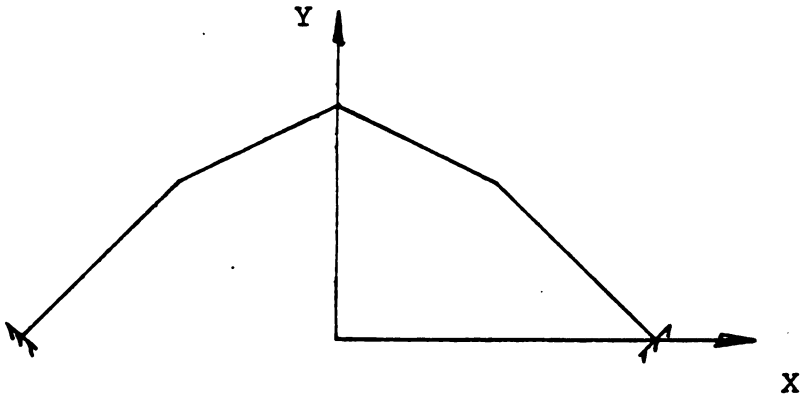
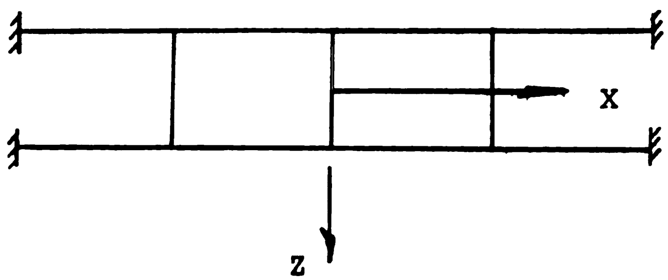


Figure 2-1. Beam Subjected to Combined Axial and Lateral Load.

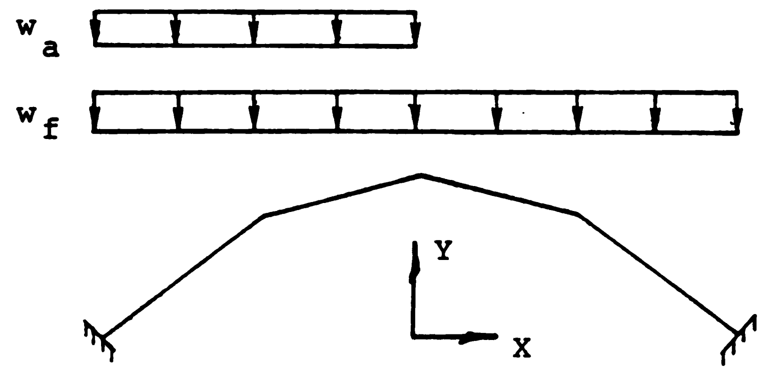


(a) X-Y Coordinates.

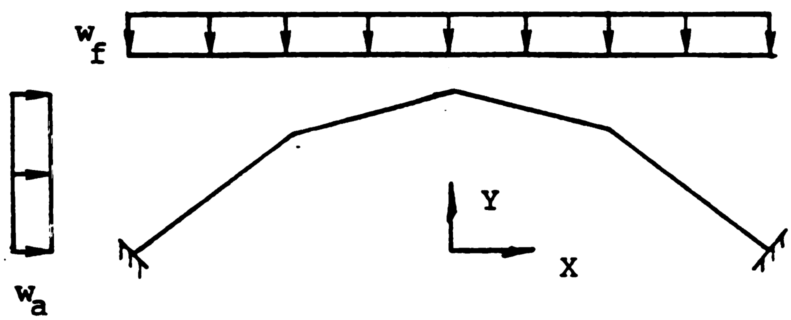


(b) X-Z Coordinates.

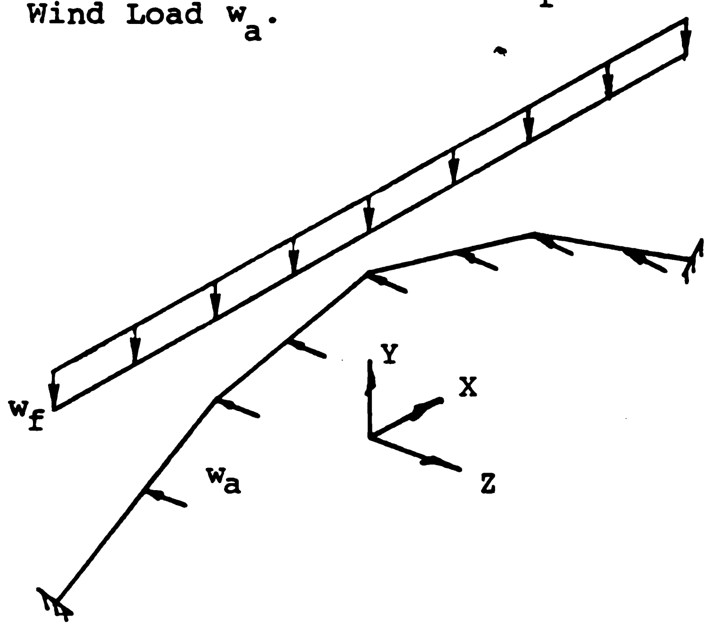
Figure 2-2. Coordinate System for Arch Bridge.



(a) Combination of Dead Load w_f and Additional Vertical Load w_a.

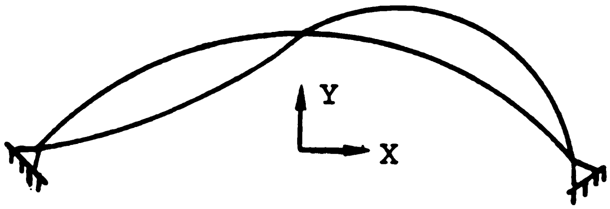


(b) Combination of Dead Load w_f and Additional Longitudinal Wind Load w_a.

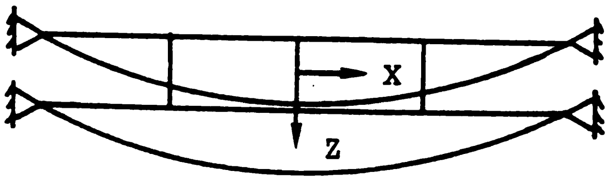


(C) Combination of Dead Load w_f and Lateral Wind Load w_a.

Figure 2-3. Loading Conditions.



(a) Anti-symmetric In-plane Buckling Mode.



(b) Symmetric Out-of-plane Buckling Mode.

Figure 2-4. Typical Buckling Modes.

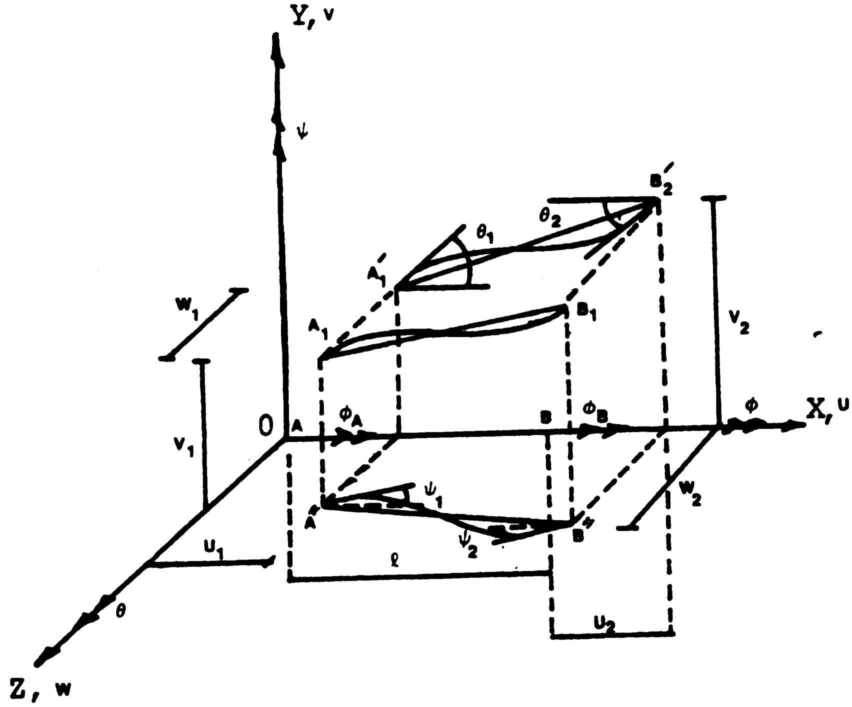


Figure 2-5. End Displacement of Three Dimensional Beam Element.

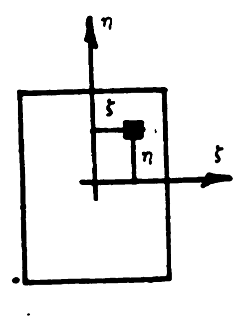
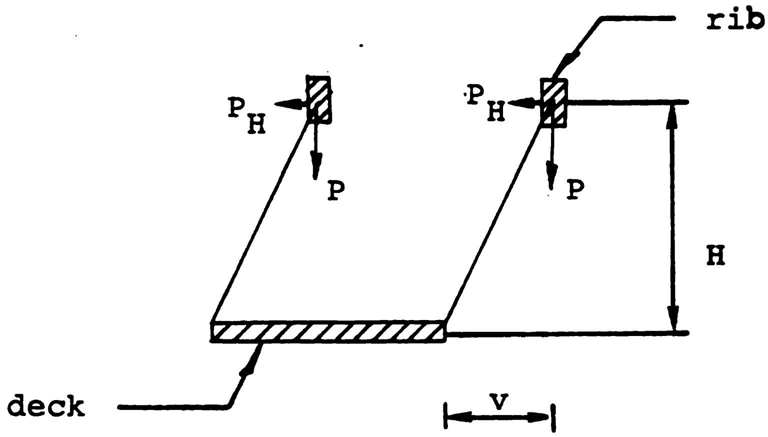
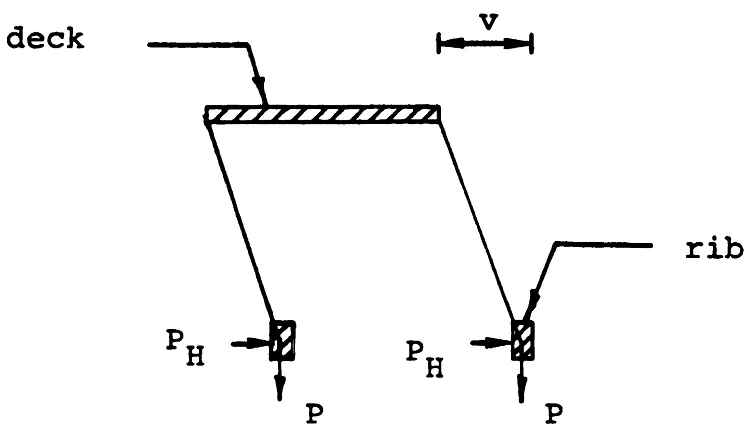


Figure 2-6. Cross-section of Beam Element.



(a) Deck Situated Below Ribs.



(b) Deck Situated Above Ribs.

Figure 2-7. Tilted Load Effect.

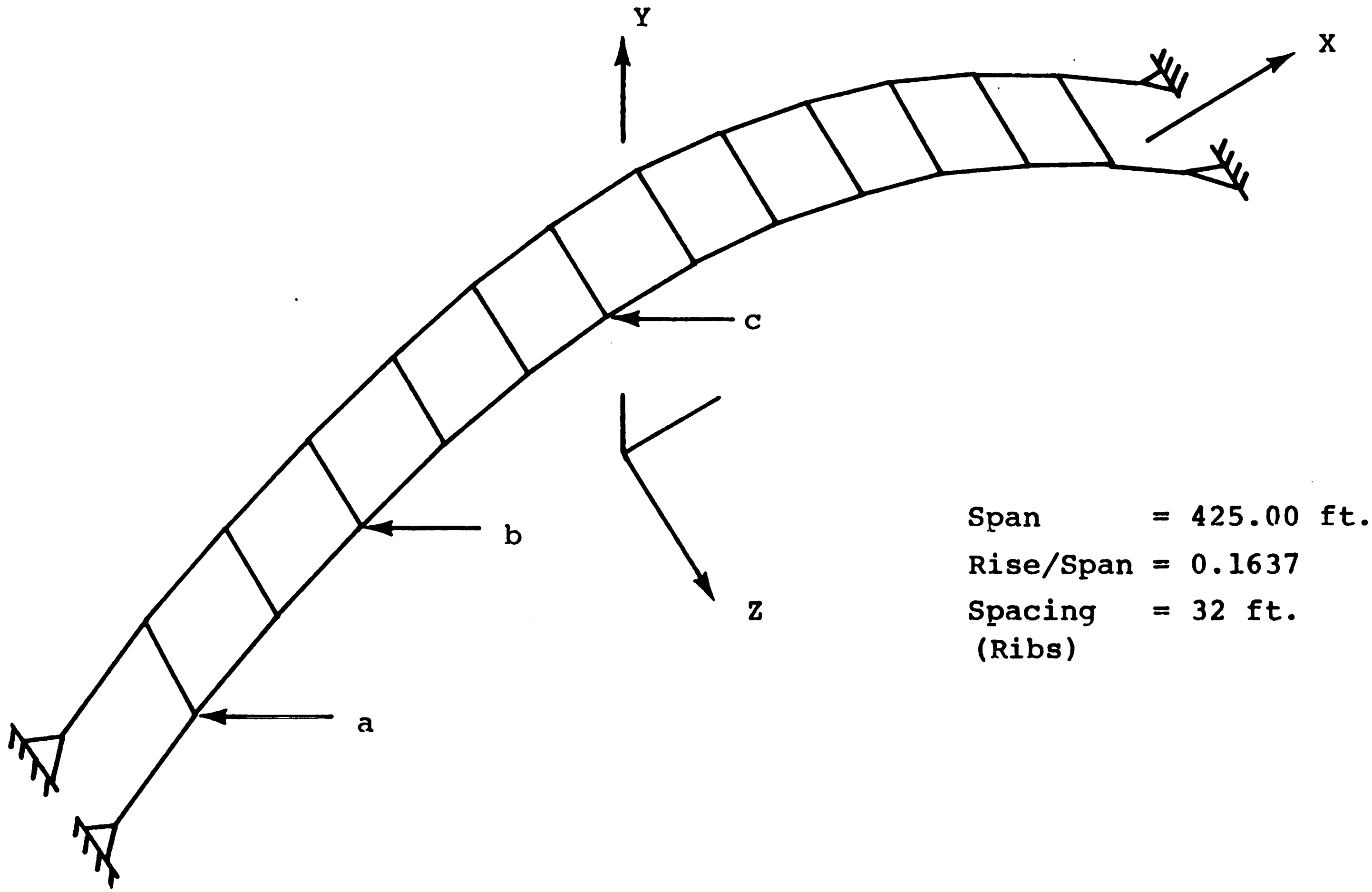
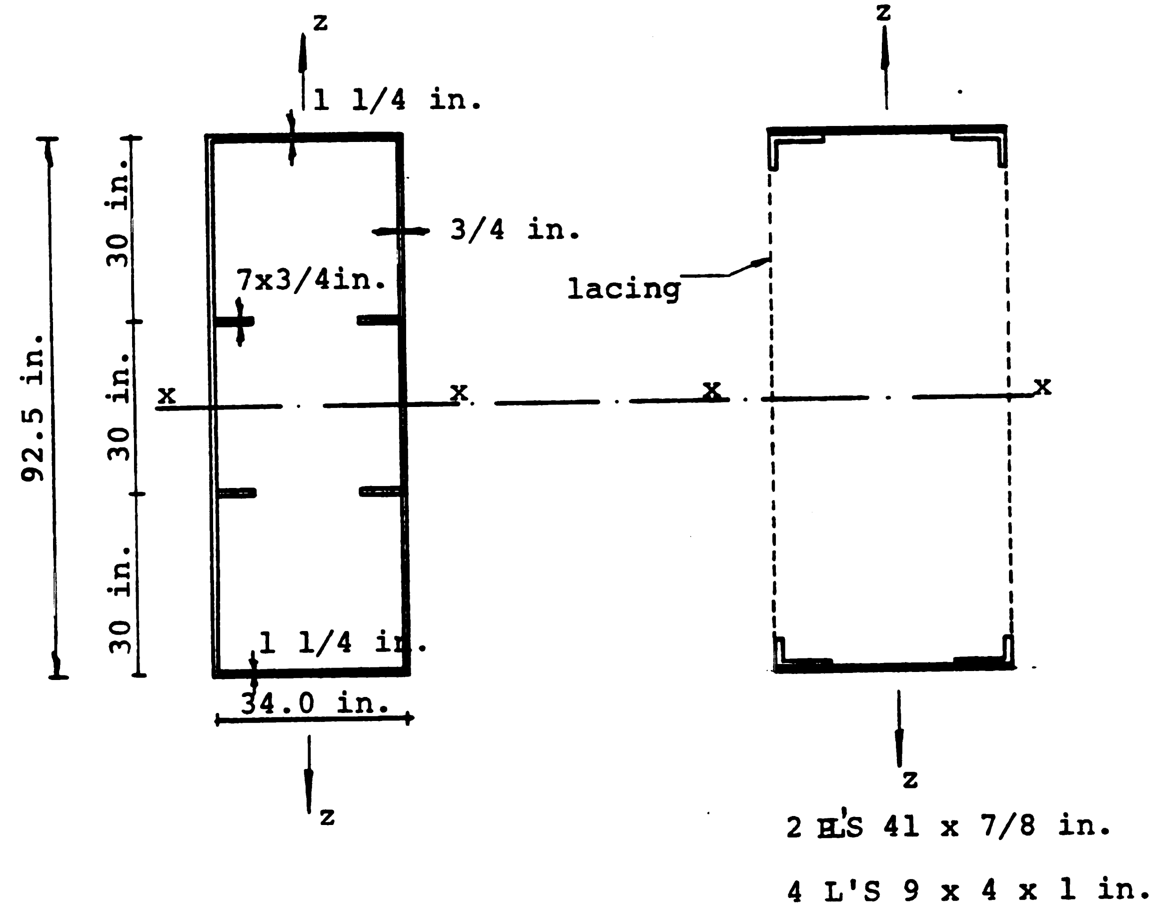


Figure 3-1. Overall Dimensions of FHAB Bridge.



<u>Ribs Properties</u>

$A = 1.6736$ ft.^{2}
$I_{xx} = 13.8395$ ft.^{4}
$I_{zz} = 2.1956$ ft.^{4}
$KT = 5.9880$ ft.^{4}

<u>Bracing Properties</u>

$A = 0.8316$ ft.^{2}
$I_{xx} = 7.9965$ ft.^{4}
$I_{zz} = 1.4275$ ft.^{4}
$KT = 1.9759$ ft.^{4}

$E = 0.4175 \times 10^7$ ksf; $\mu = 0.3$; $\sigma_y = 40$ ksf

Figure 3-2. Cross-sectional and Material Properties of FHAB.

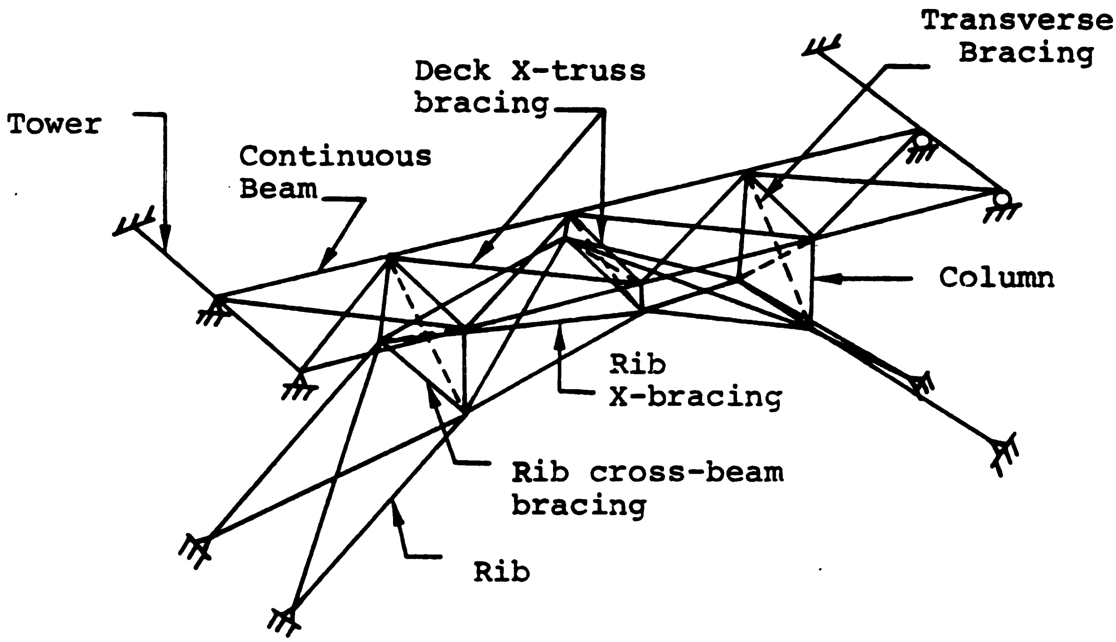
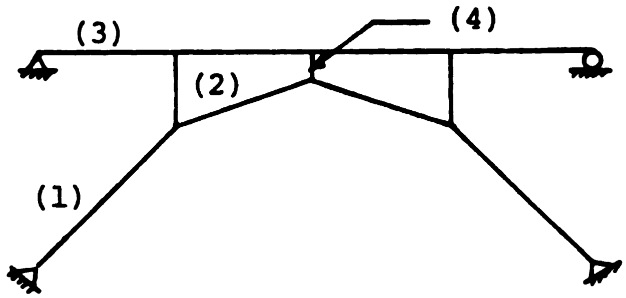
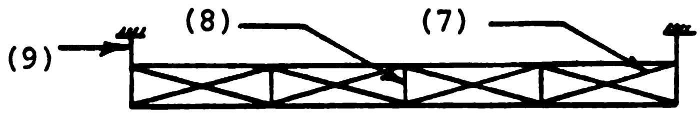


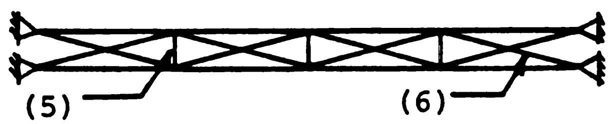
Figure 3-3. Model for MCSCB Bridge.



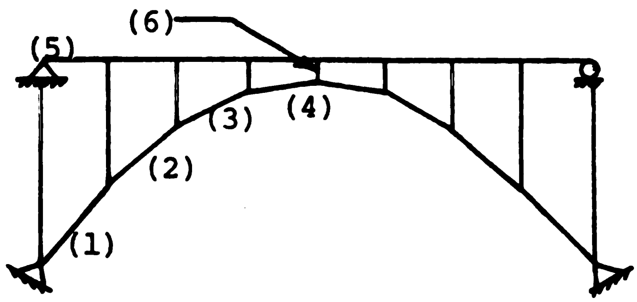
(a) Front View of Four Panel Model.



(b) Plan for Deck and Towers.



(c) Plan for Ribs X-truss Bracing.



(d) Front View of Eight Panels Model.

Figure 3-4. 4- and 8-Panel Models for MCSCB Bridge.

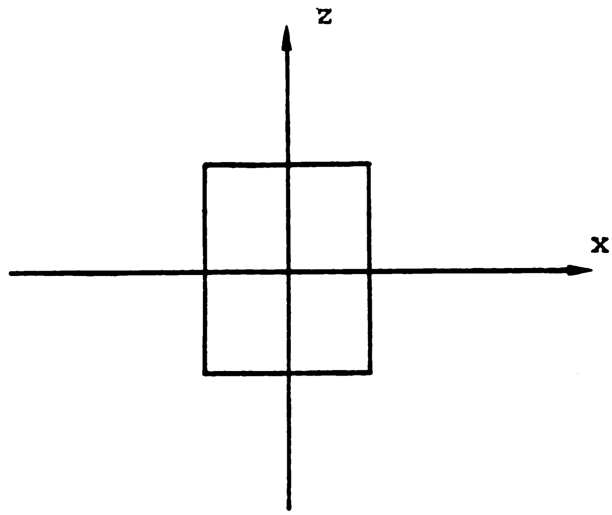
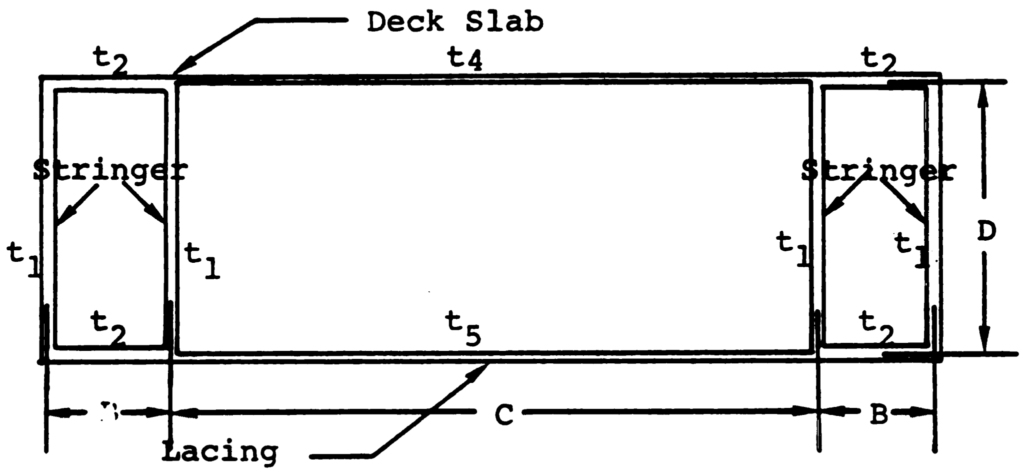


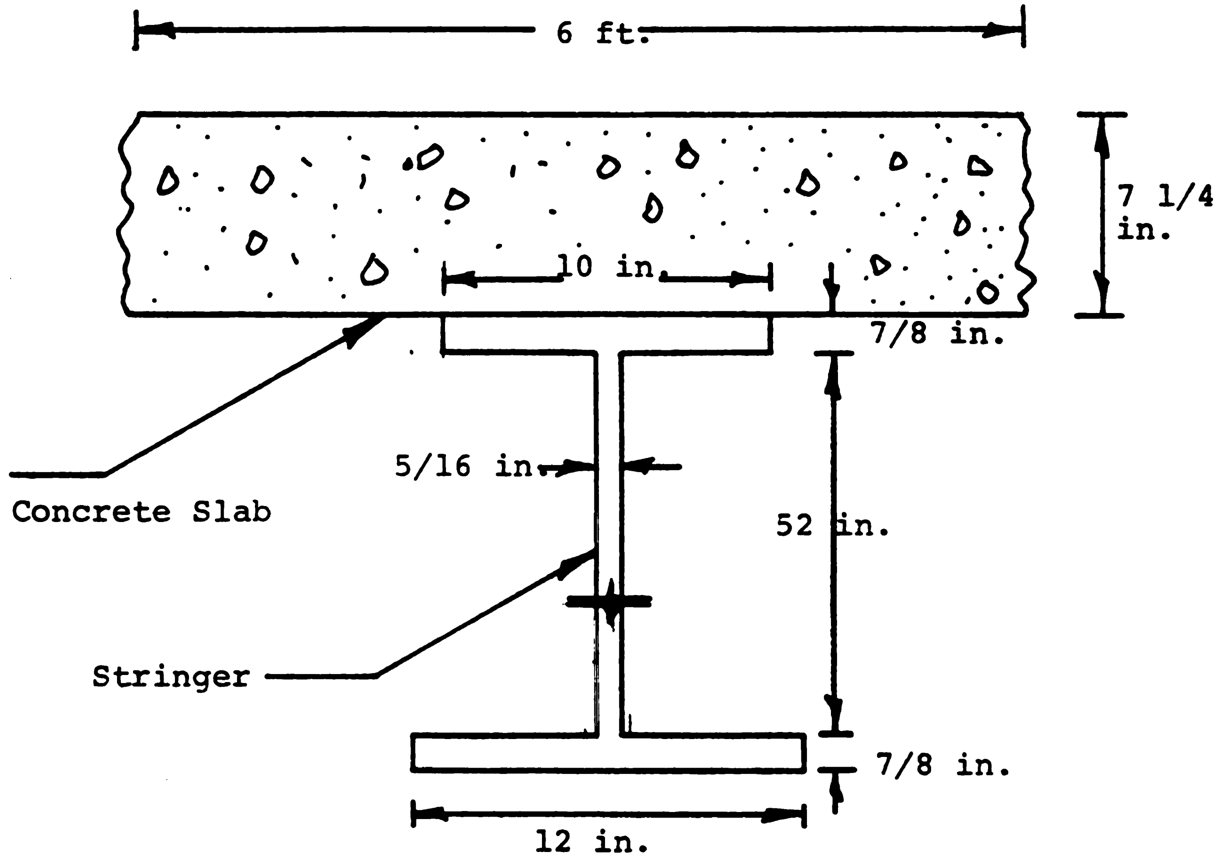
Figure 3-5. Local X and Z Coordinates for Truss and Beam Members.



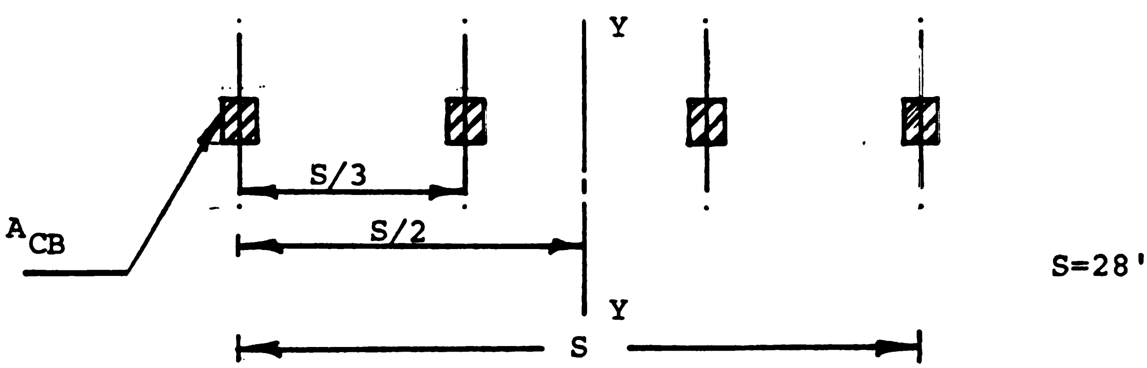
D = 9 ft. B = 2.92 ft. C = 23.0 ft. t_1 = 15/16 in.

$t_2 = t_3 = 3$ in. $t_4 = t_5 = 0.073$ in.

Figure 3-6. Three Cell Box Model for Torsional Stiffness of Deck Beams.

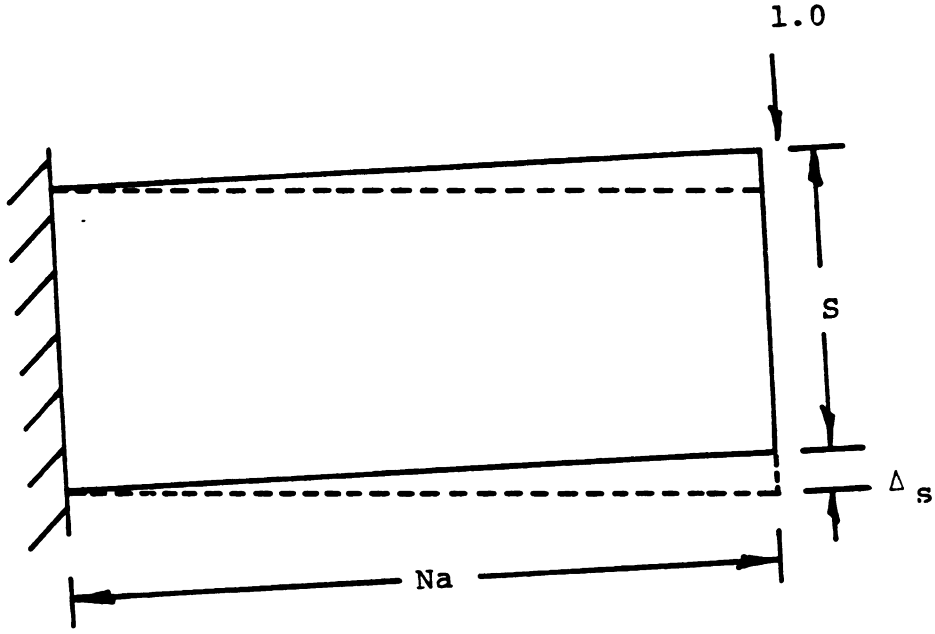


(a) Composite Beam Deck Slab and Stringer of Actual Bridge.

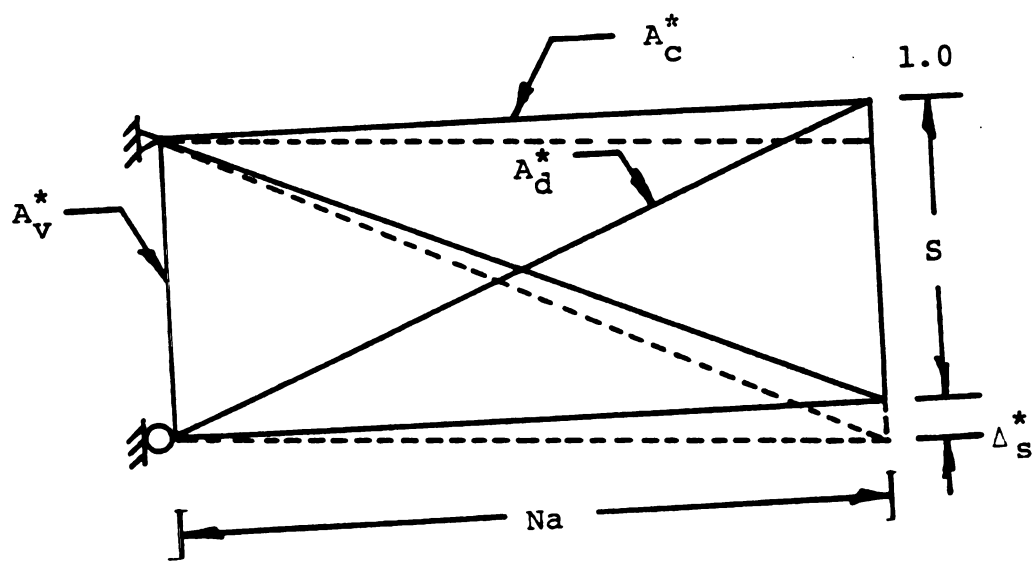


(b) Spacing Between Composite Beams.

Figure 3-7. Components of Actual Bridge Deck.

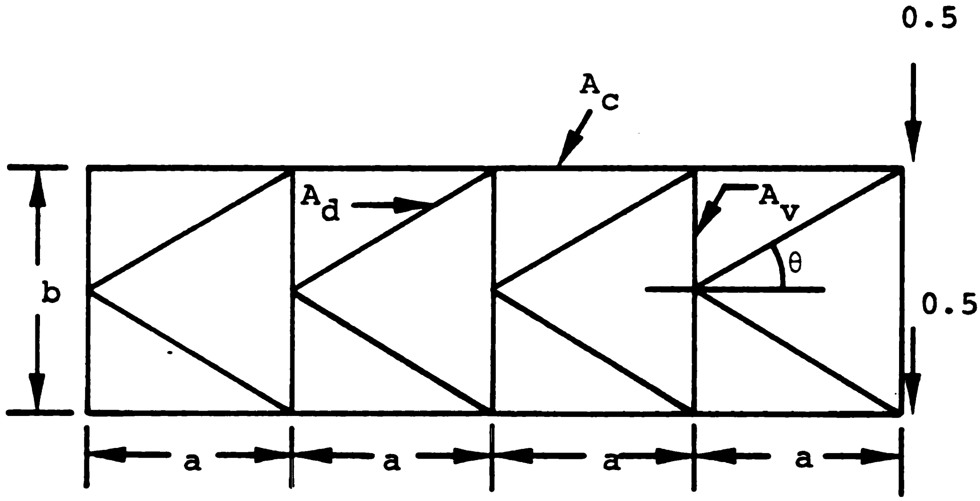


(a) Shear Deformation of Slab.

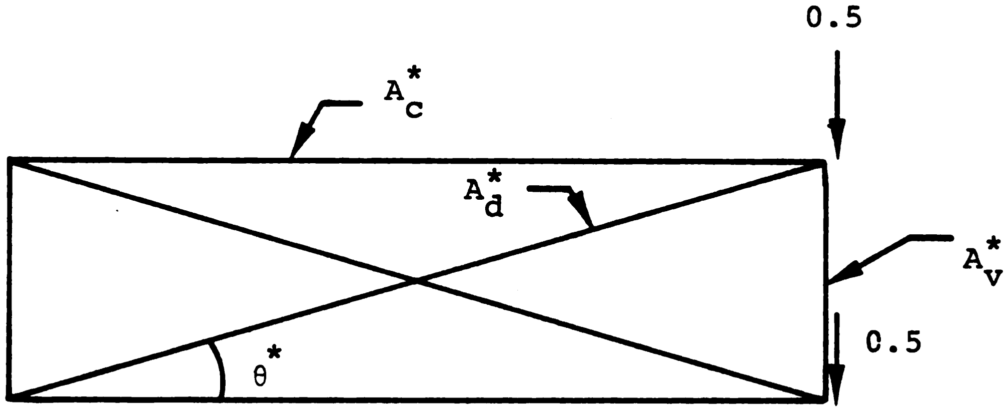


(b) Shear Deformation in X-truss Bracing

Figure 3-8. Modeling of Deck Slab.



(a) K-truss Bracing in Actual Bridge.



(b) X-truss Bracing in Bridge Model.

Figure 3-9. Modelling of K-truss Bracing Between Ribs.

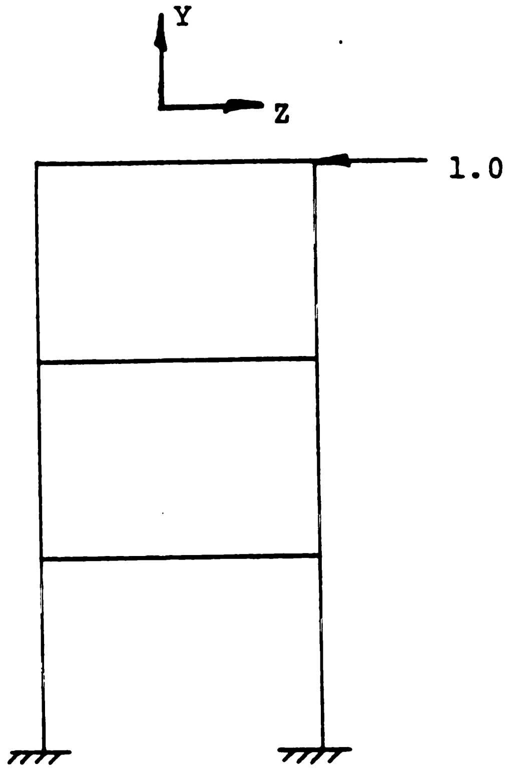


Figure 3-10. Sketch of Actual Tower.

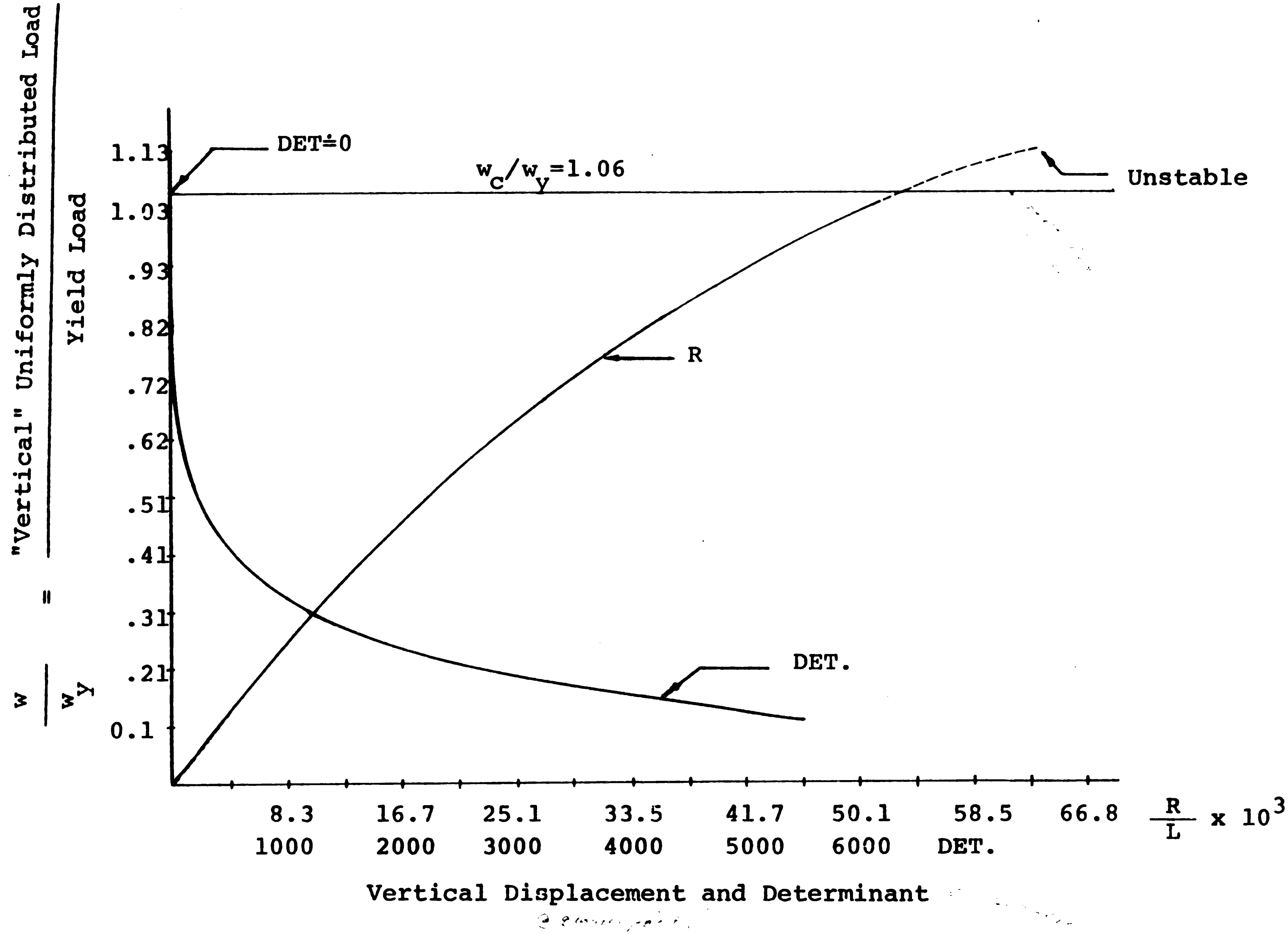
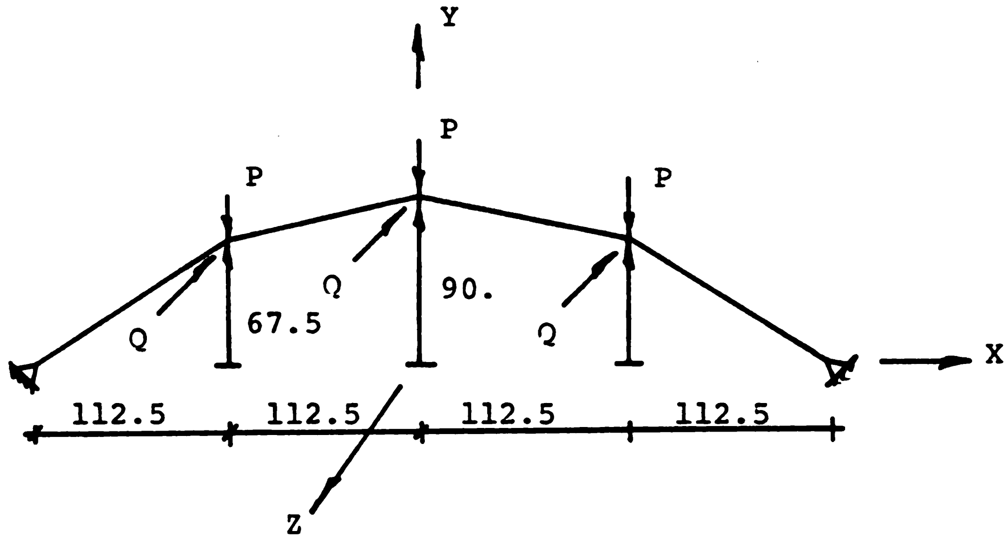


Figure 3-11. Equilibrium and Eigen value Solutions of FHAB Bridge.



All Dimensions are in ft.

For Equilibrium Solution, P and Q are Load Increments;

P = 112.5 Kips, and Q = 0.001 P.

For Eigenvalue Solutions

P = 1.0 Kips, and Q = 0.0.

Cross-sectional Properties;

$A = 2.7$ ft.^{2}, $I_{xx} = 32.5$ ft.^{4}, $I_{zz} = 4.45$ ft.^{4}, and

$KT = 5.785$ ft.^{4}

Material Properties;

$E = 0.4176 \times 10^7$ Ksf, $\mu = 0.3$, and $\sigma_y = 40$ Ksf.

Figure 3-12. Dimensions and Properties of Arch-A.

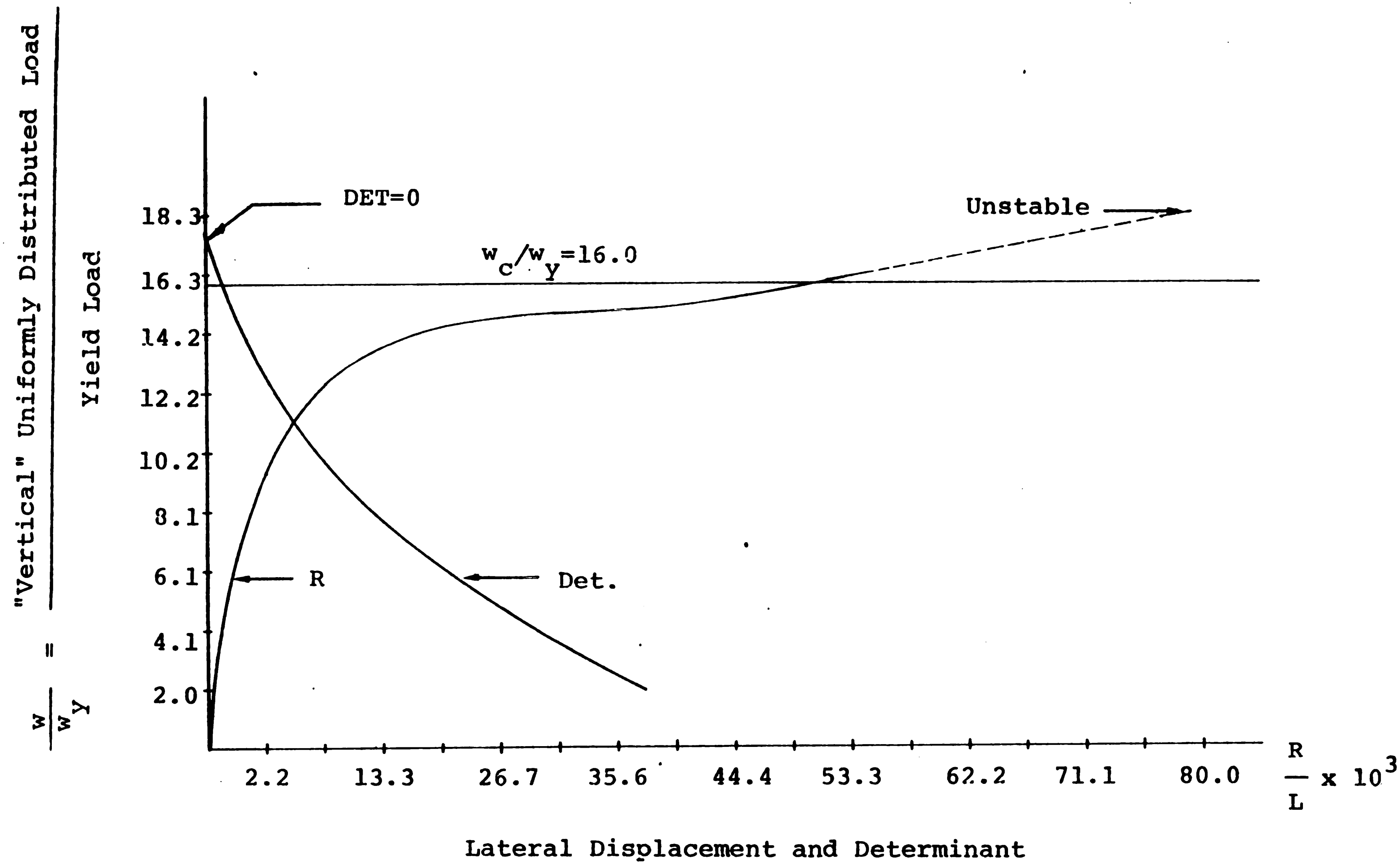
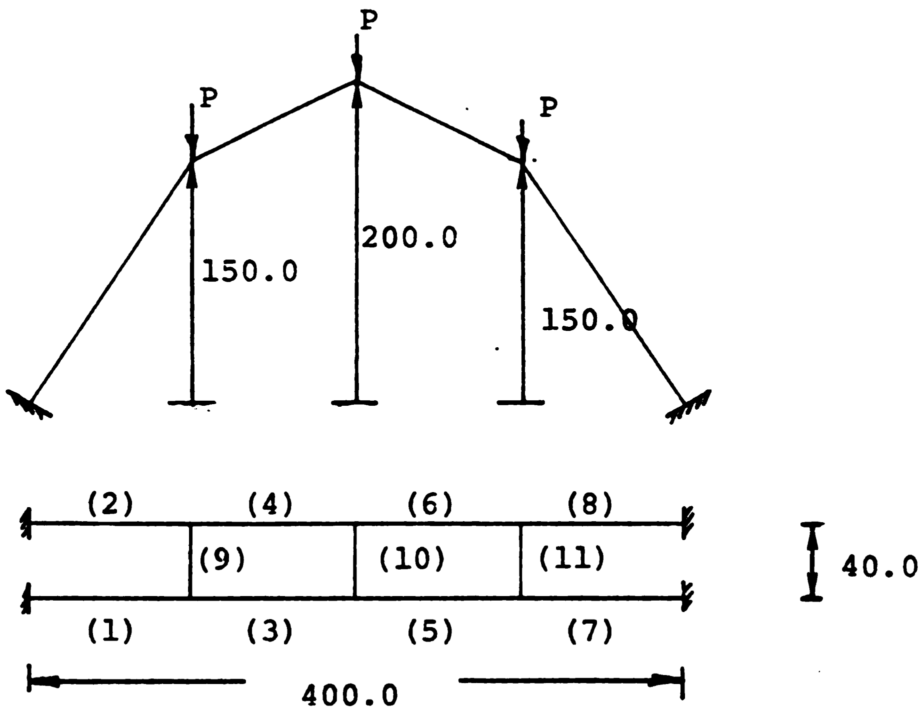
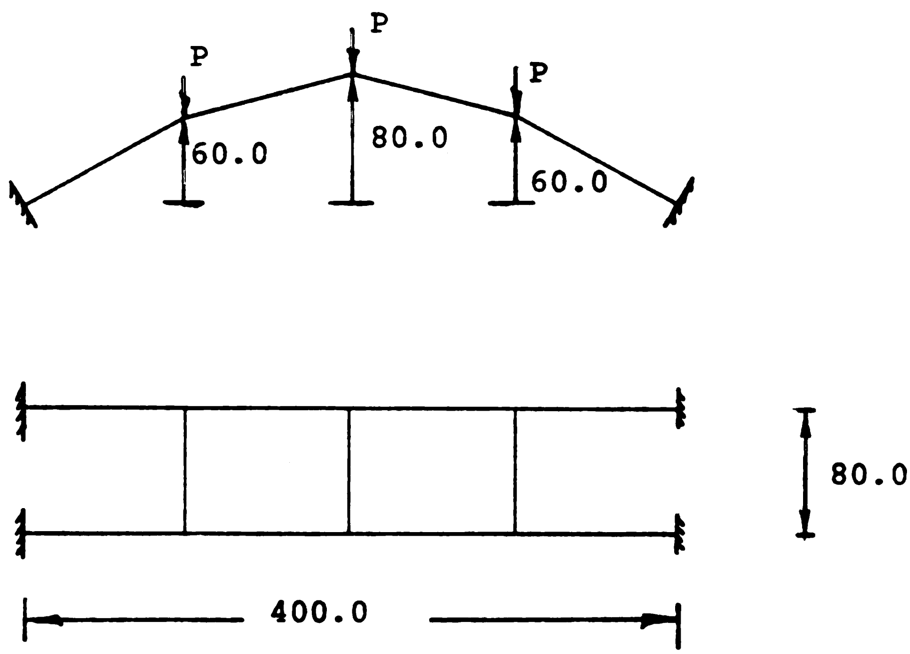


Figure 3-13. Equilibrium and Eigenvalue Solution of Arch-A.



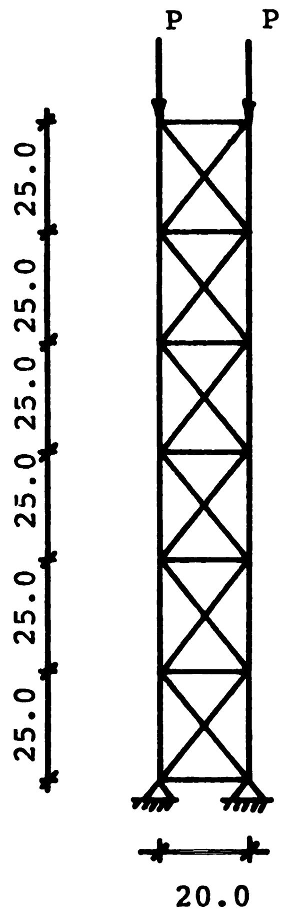
(a) Example One.



(b) Example Two.

$E = 0.4176 \times 10^7$ Ksf. $\mu = 0.3$

Figure 3-14. Arch Bridges Considered by Ostlund.



Truss Tower

all cross-sectional areas = 5.0 ft.^{2}

Truss and Beam Tower

X and cross members are truss elements cross-sectional area = 5.0 ft.^{2}

Columns are beam elements with the following cross-sectional properties;

A = 5.0 ft.^{2}

I_{xx} = 500.0 ft.^{4}

I_{zz} = 0.01 ft.^{4}

KT = 0.05 ft.^{4}

E = 0.4176×10^7 Ksf, and

Figure 3-15. Properties of Truss and Truss-Beam Towers.

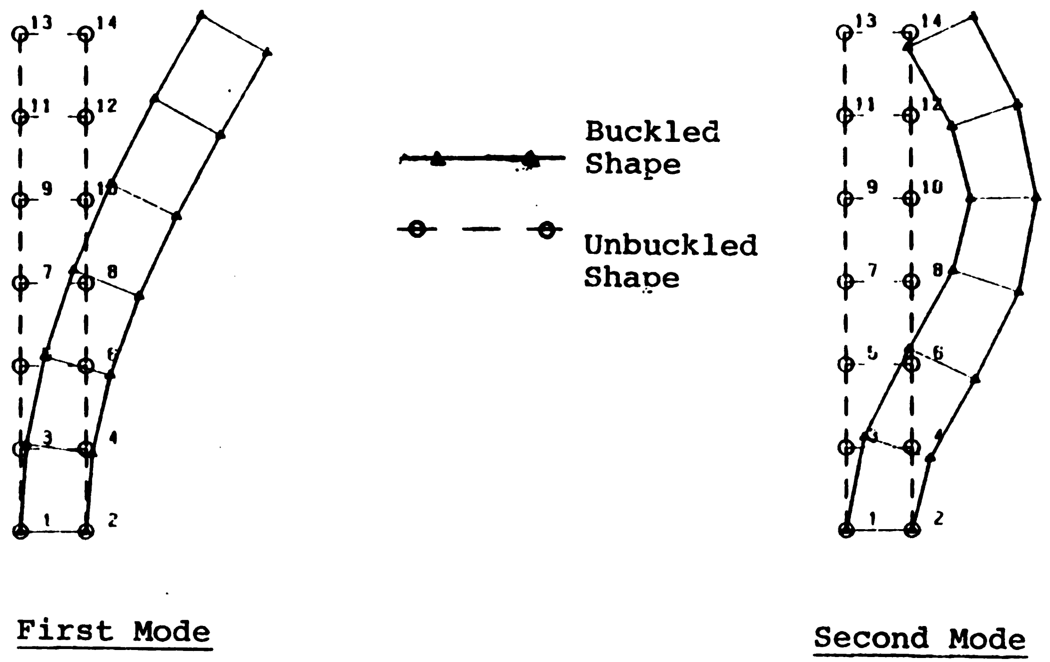
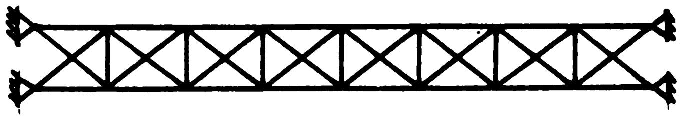
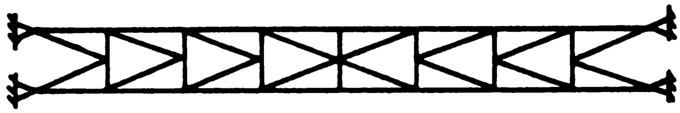


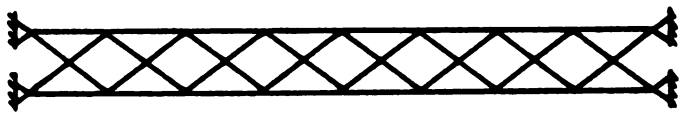
Figure 3-16. Buckled Shapes of a Beam-Truss Tower.



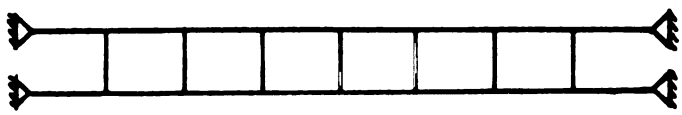
(a) X-truss Bracing



(b) K-truss Bracing



(c) Diagonal Bracing



(d) Transverse Bracing

Figure 4-1. Patterns of Rib Bracing.

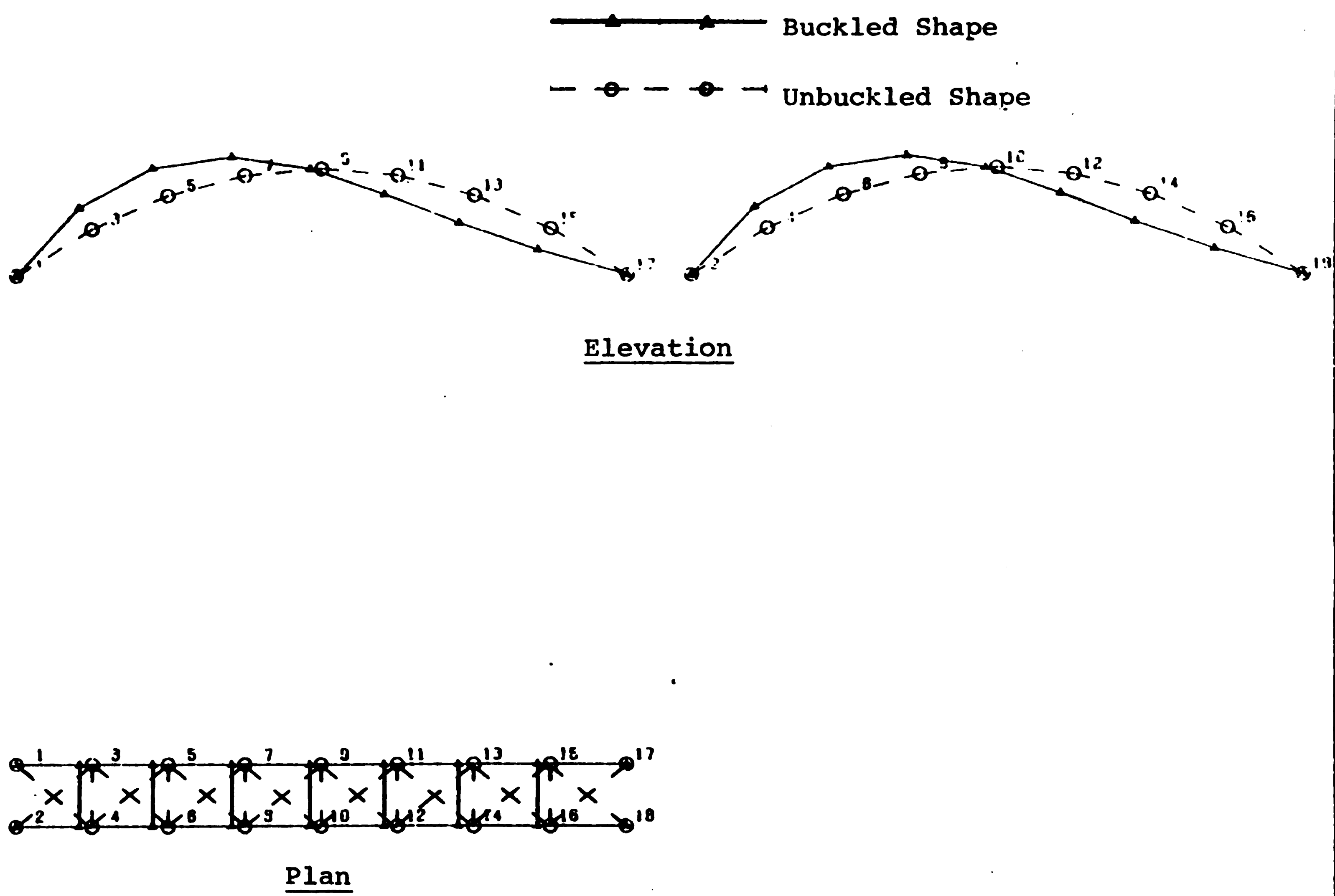


Figure 4-2. Buckled Shape of MCSCB Ribs, X-truss Bracing (First Mode).

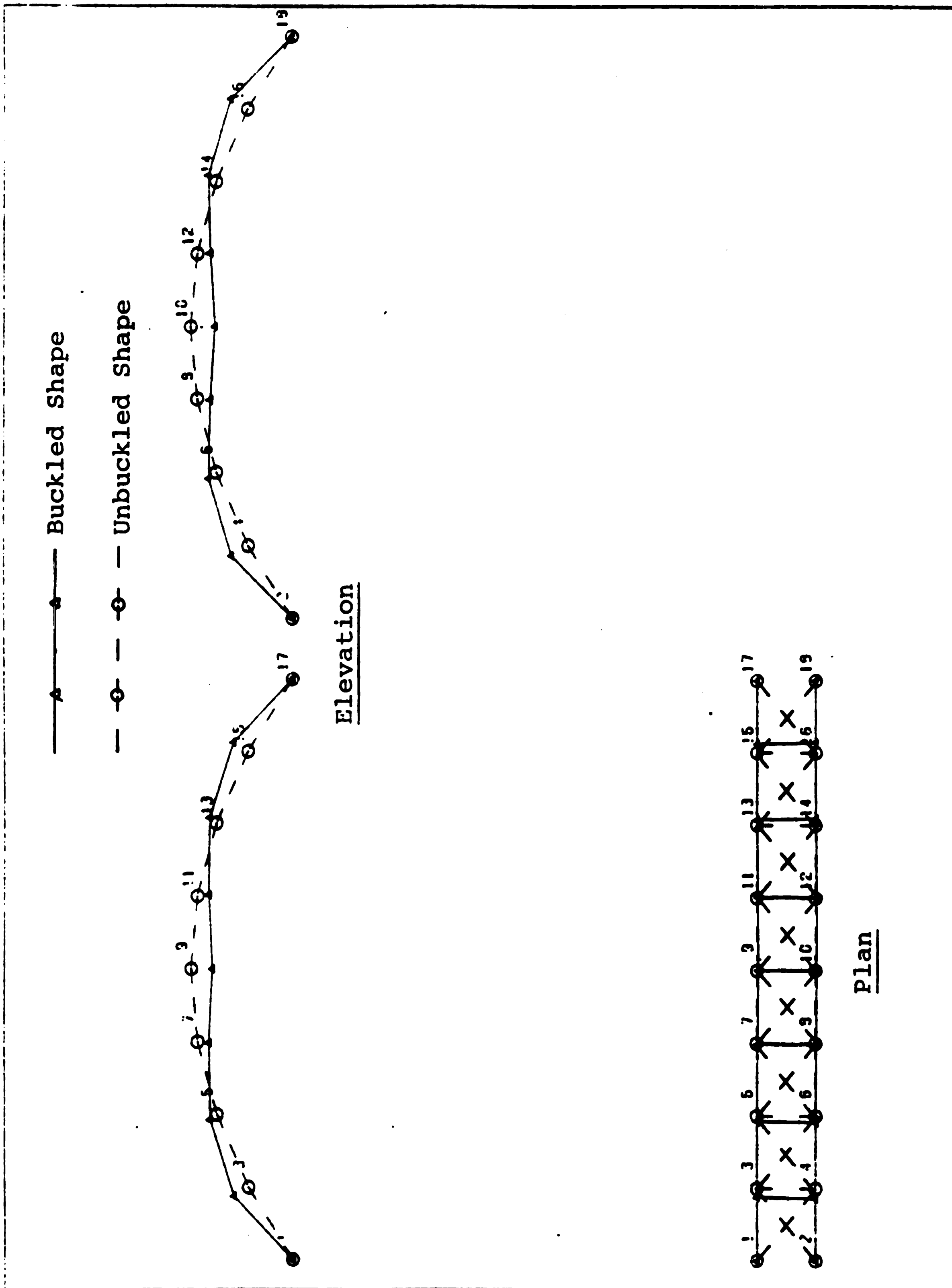


Figure 4-3. Buckled Shape of MCSCB Ribs, X-truss Bracing (Second Mode).

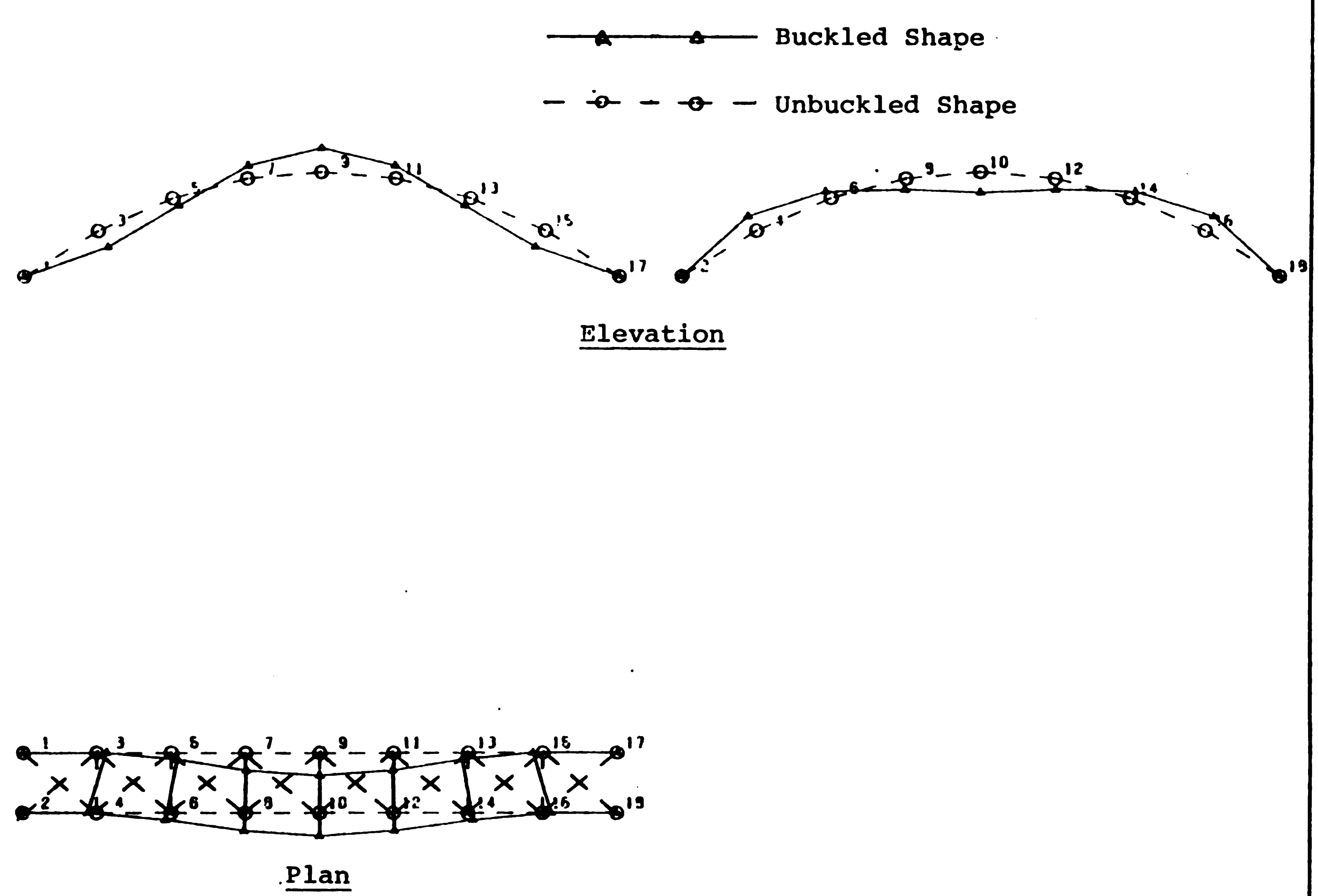


Figure 4-4. Buckled Shape of MCSCB Ribs, X-truss Bracing (Third Mode).

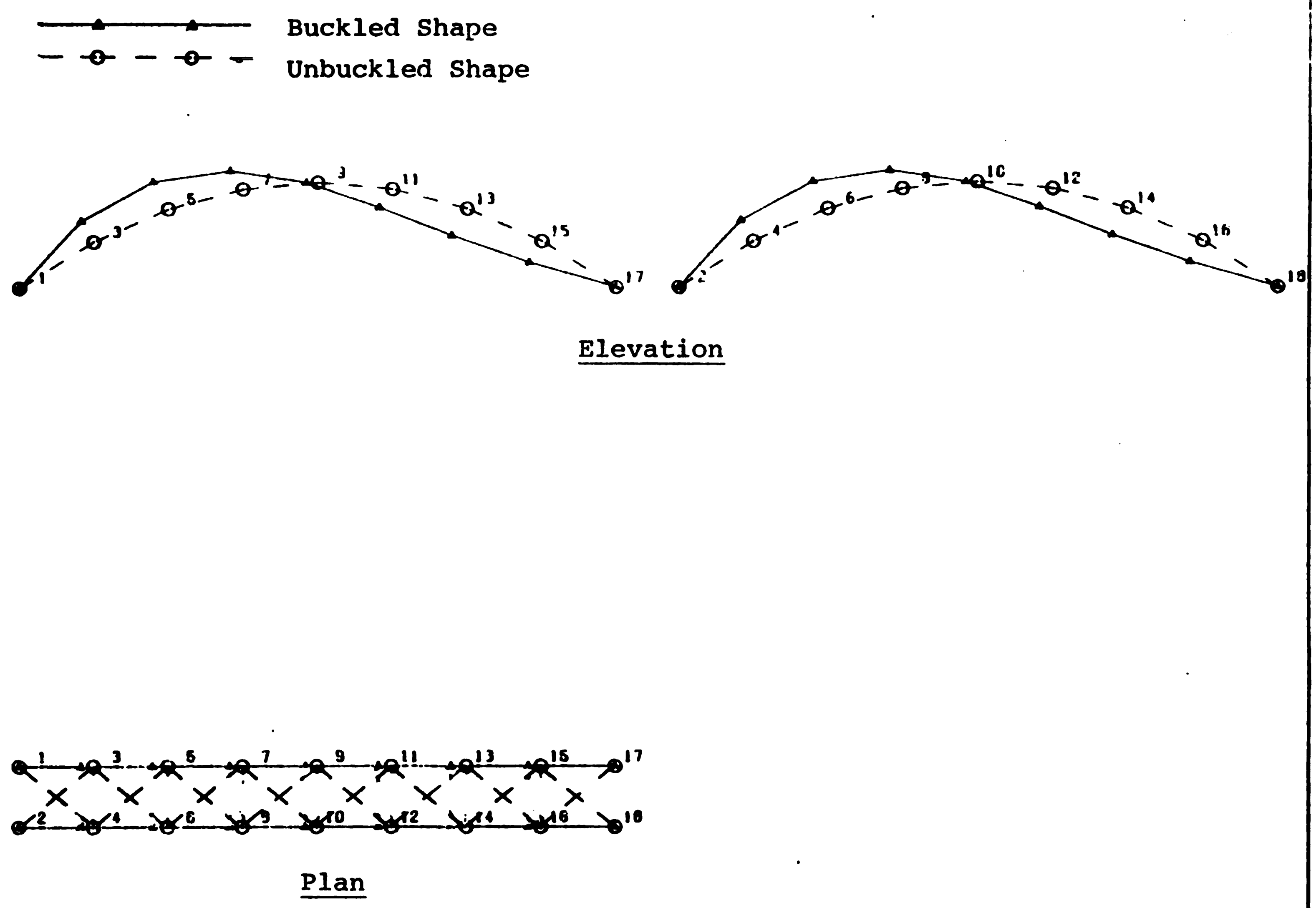


Figure 4-5. Buckled Shape of MCSCB Ribs, Diagonal Bracing (First Mode).

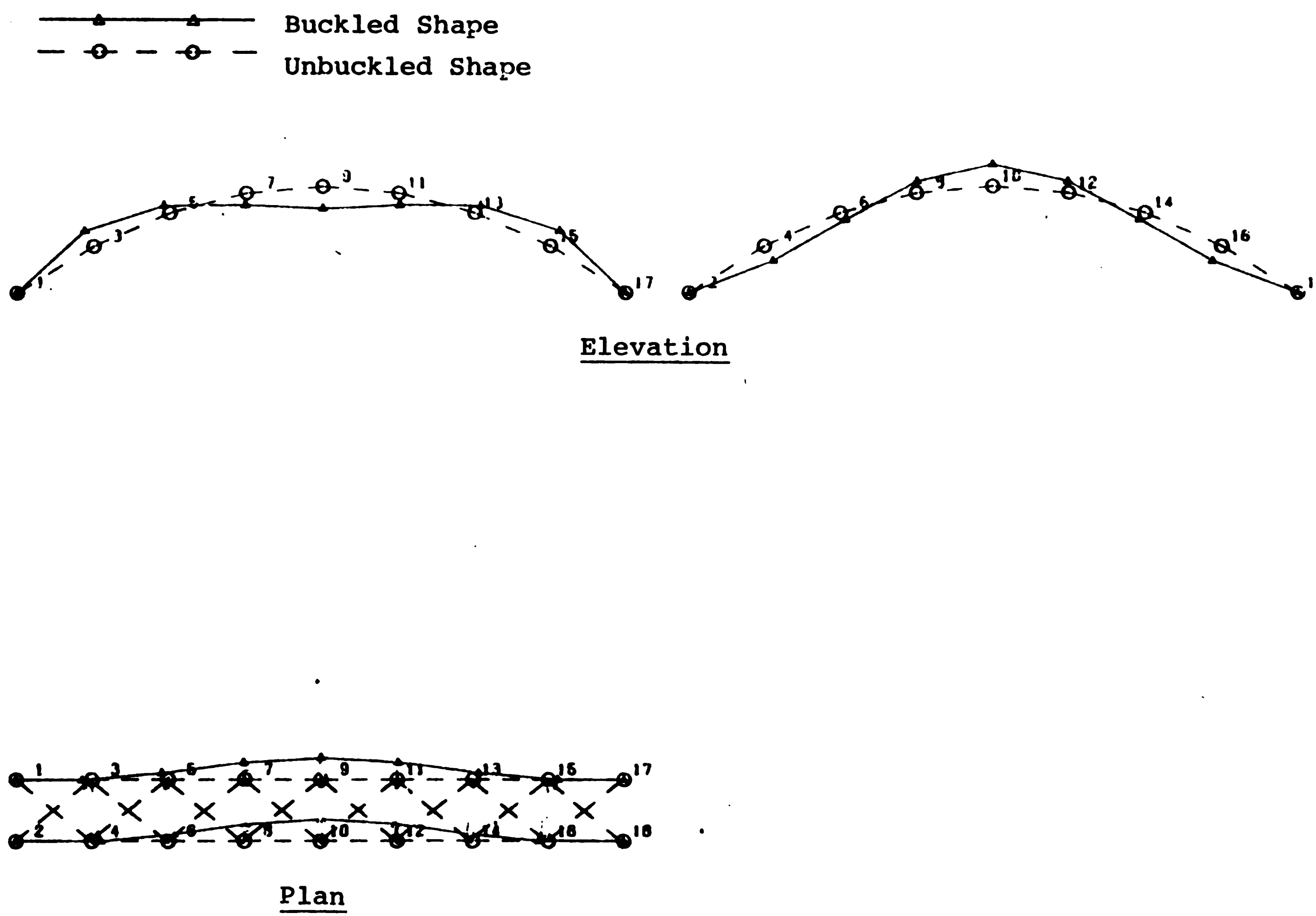


Figure 4-6. Buckled Shape of MCSCB Ribs, Diagonal Bracing (Second Mode).

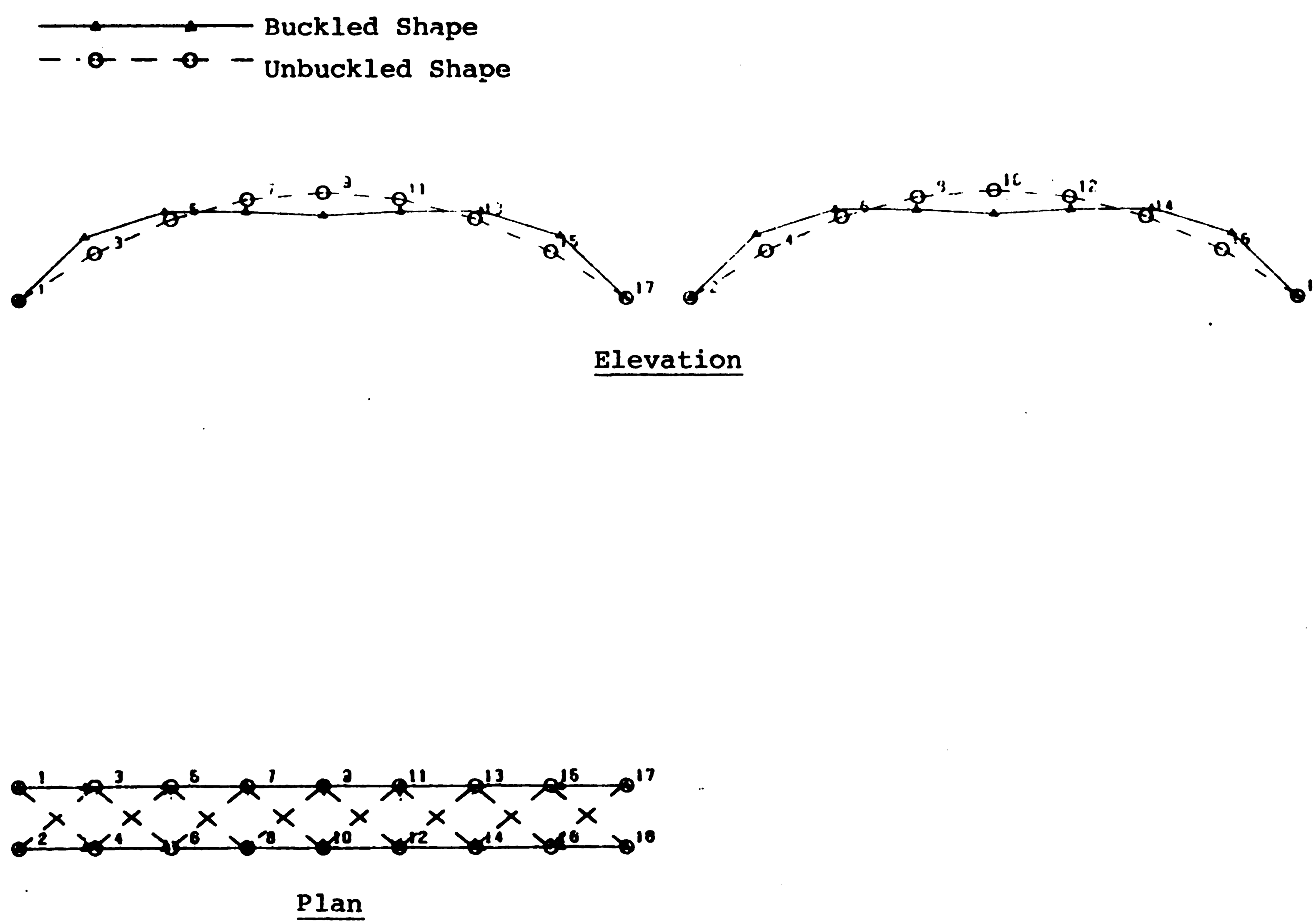


Figure 4-7. Buckled Shape of MCSCB Ribs, Diagonal Bracing (Third Mode).

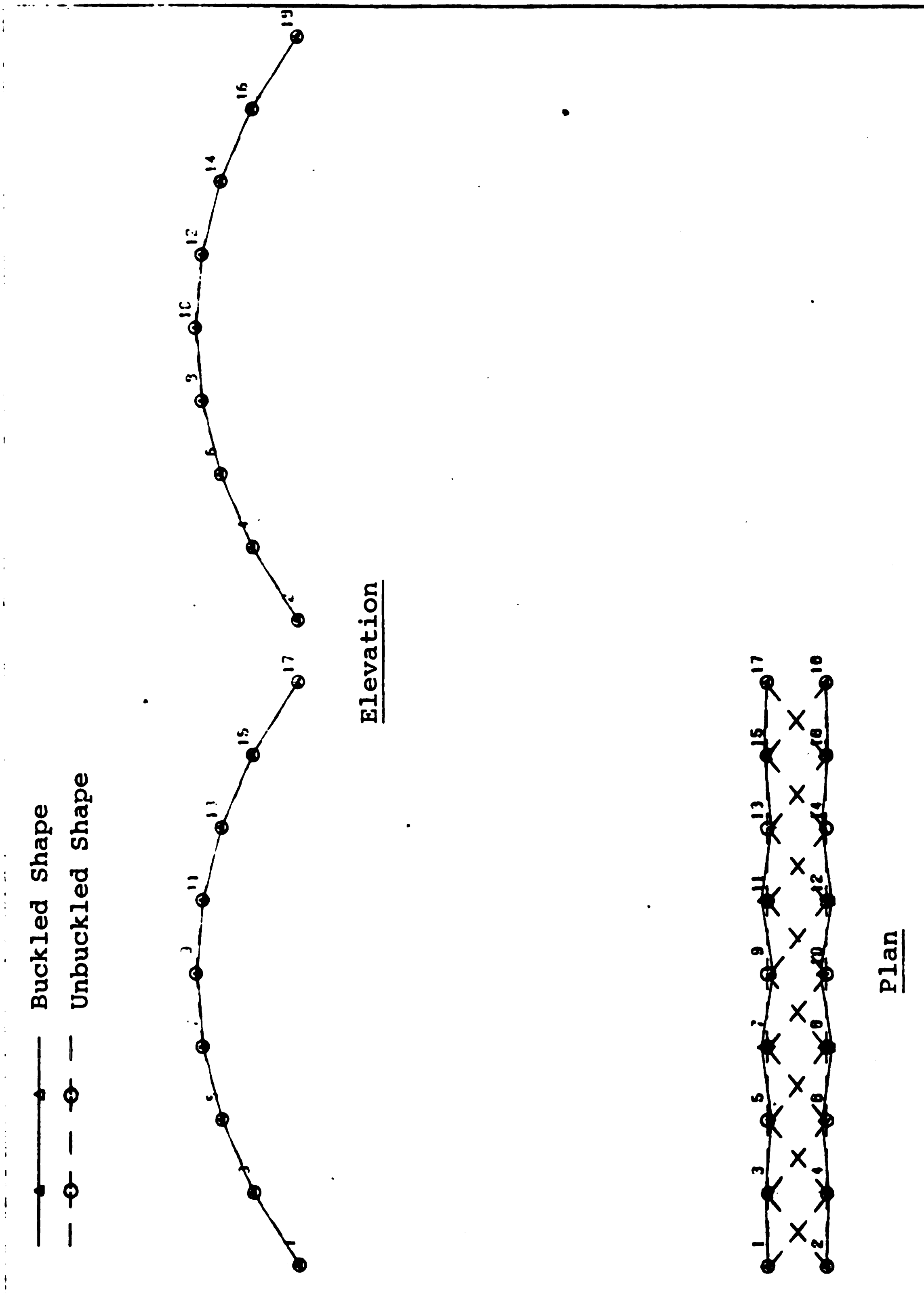


Figure 4-8. Buckled Shape of MCSCB Ribs, Diagonal Bracing (Fourth Mode).

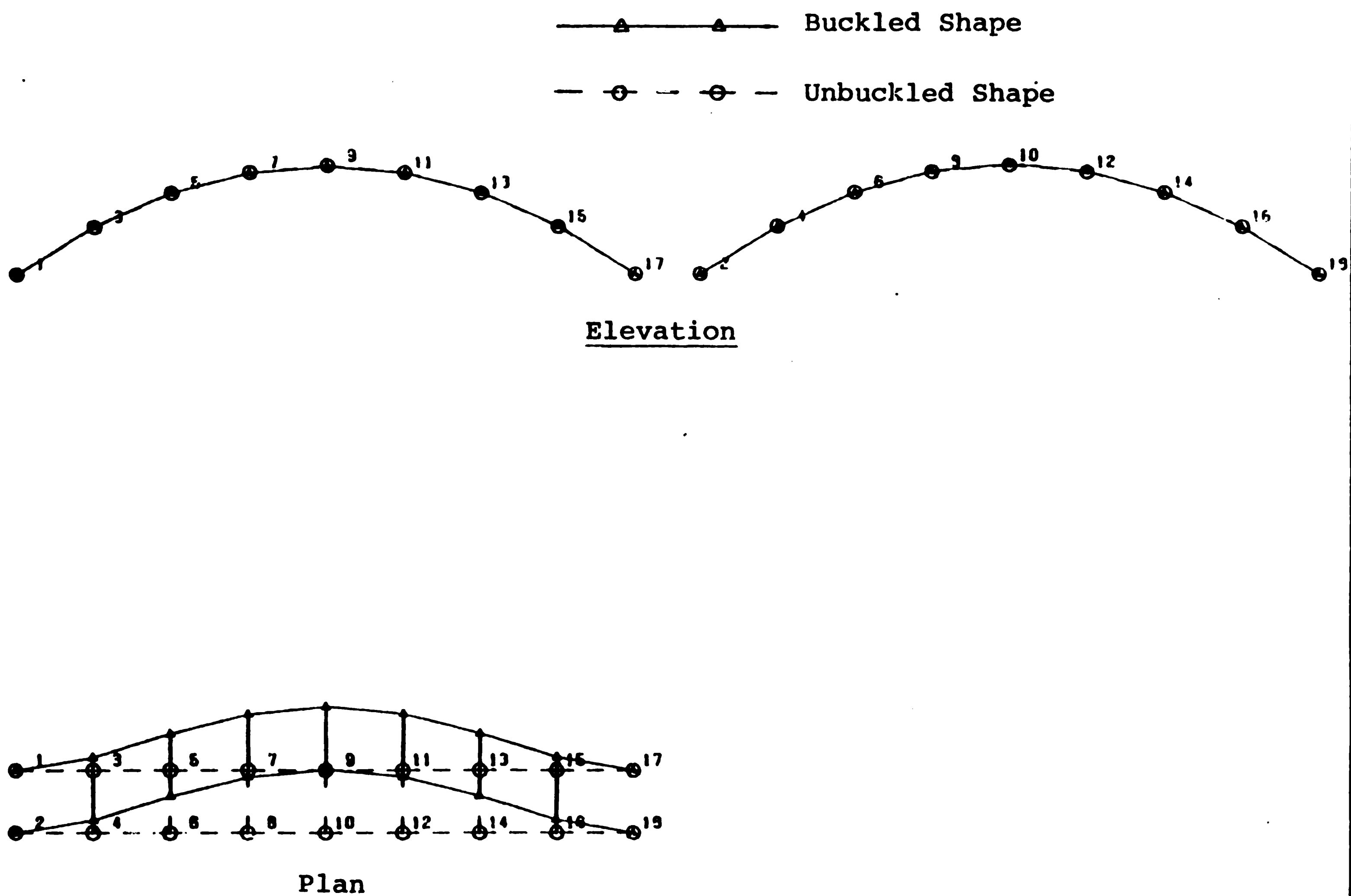


Figure 4-9. Buckled Shape of MCSCB Ribs, Transverse Bracing (First Mode).

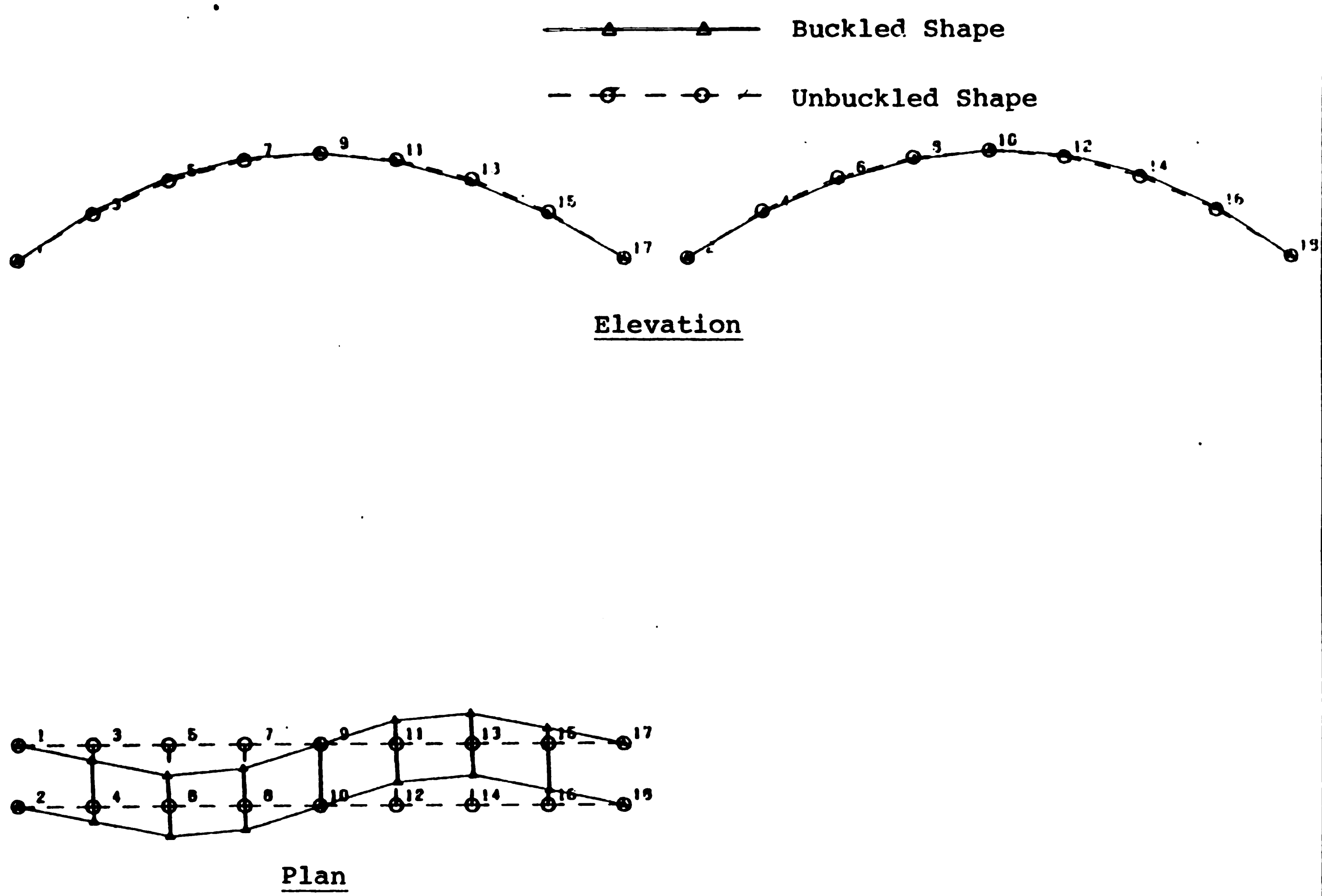


Figure 4-10. Buckled Shape of MCSCB Ribs, Transverse Bracing (Second Mode).

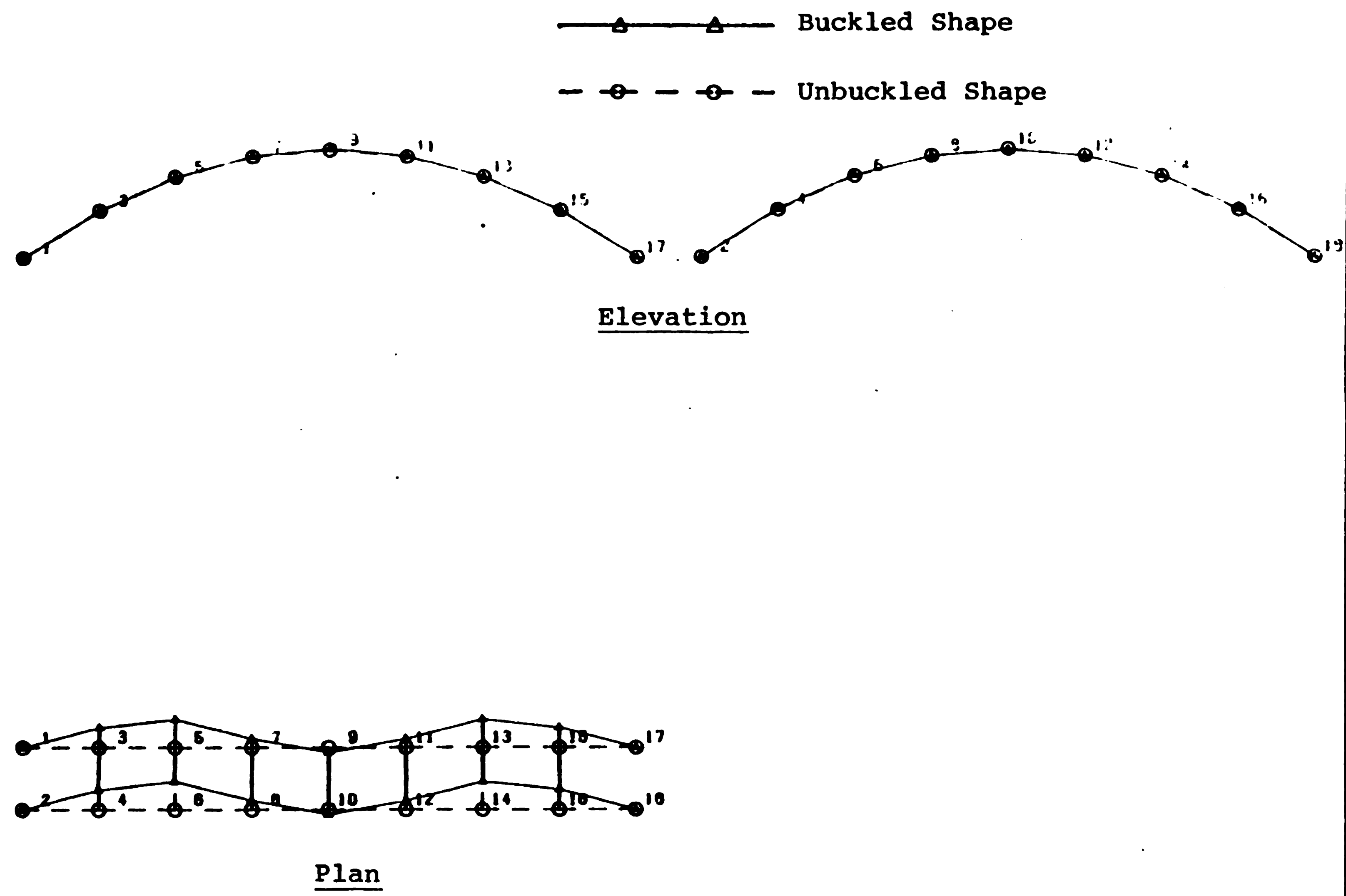


Figure 4-11. Buckled Shape of MCSCB Ribs, Transverse Bracing (Third Mode).

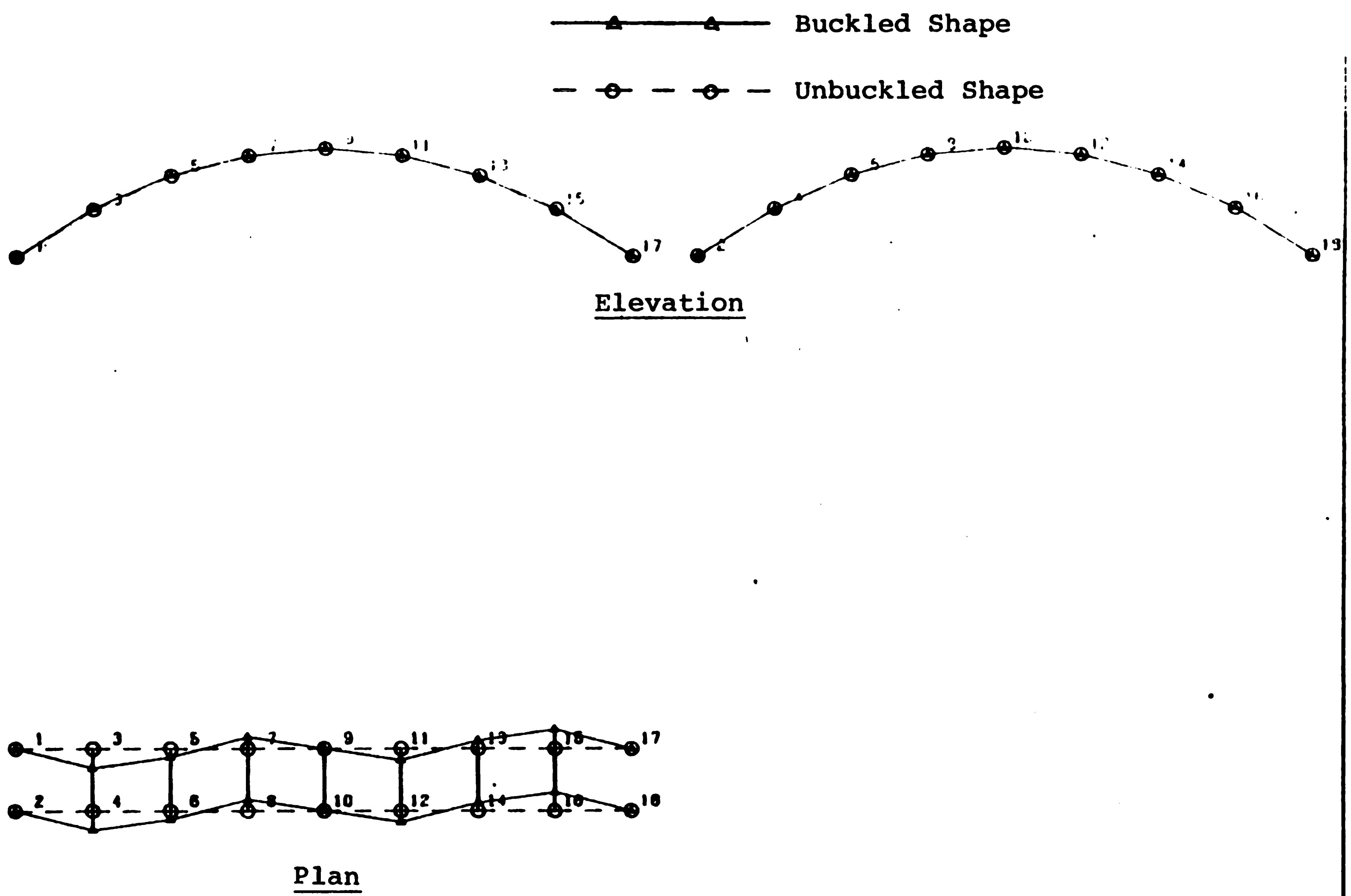


Figure 4-12. Buckled Shape of MCSCB Ribs, Transverse Bracing (Fourth Mode).

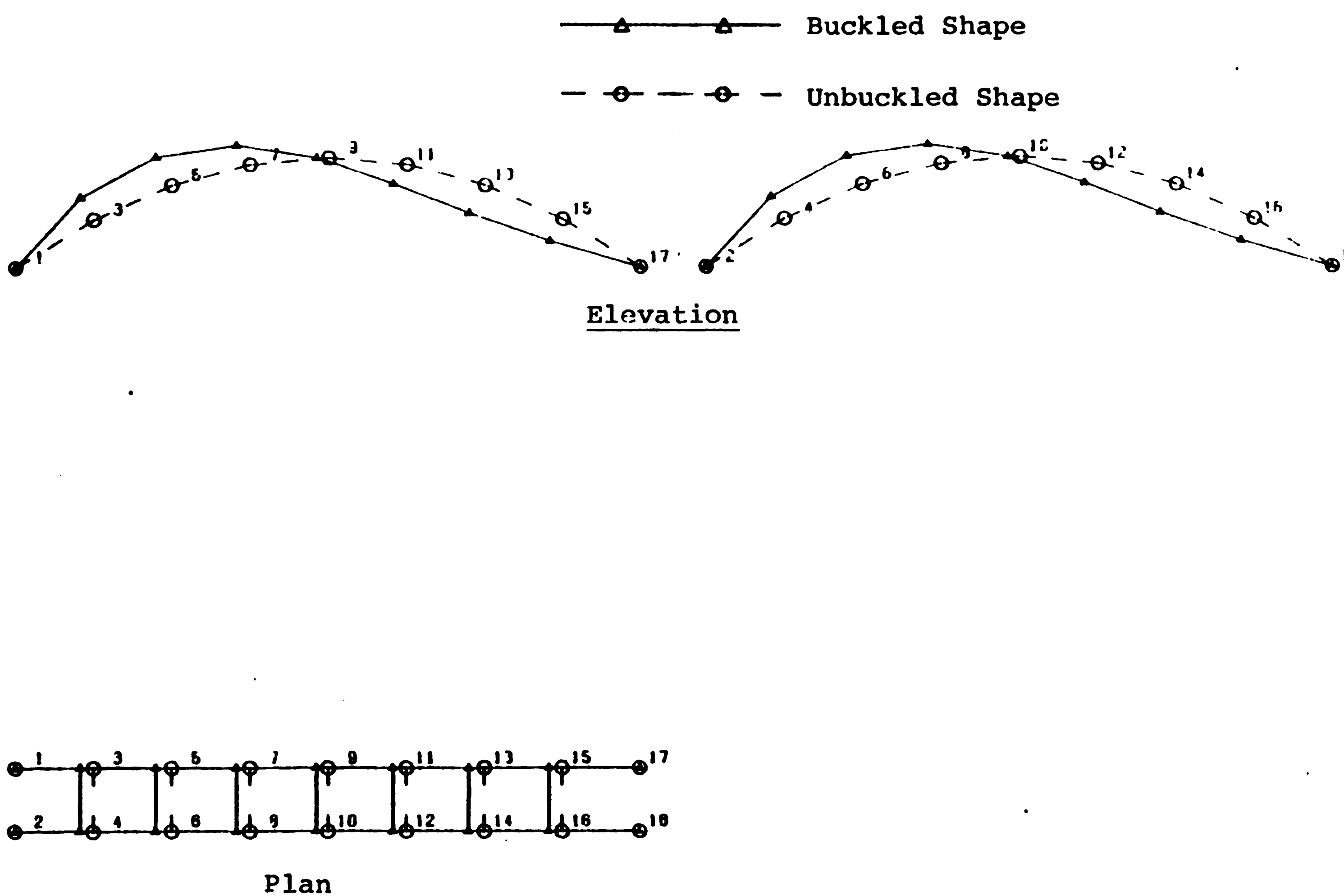


Figure 4-13. Buckled Shape of MCSCB Ribs, Transverse Bracing (Fifth Mode).

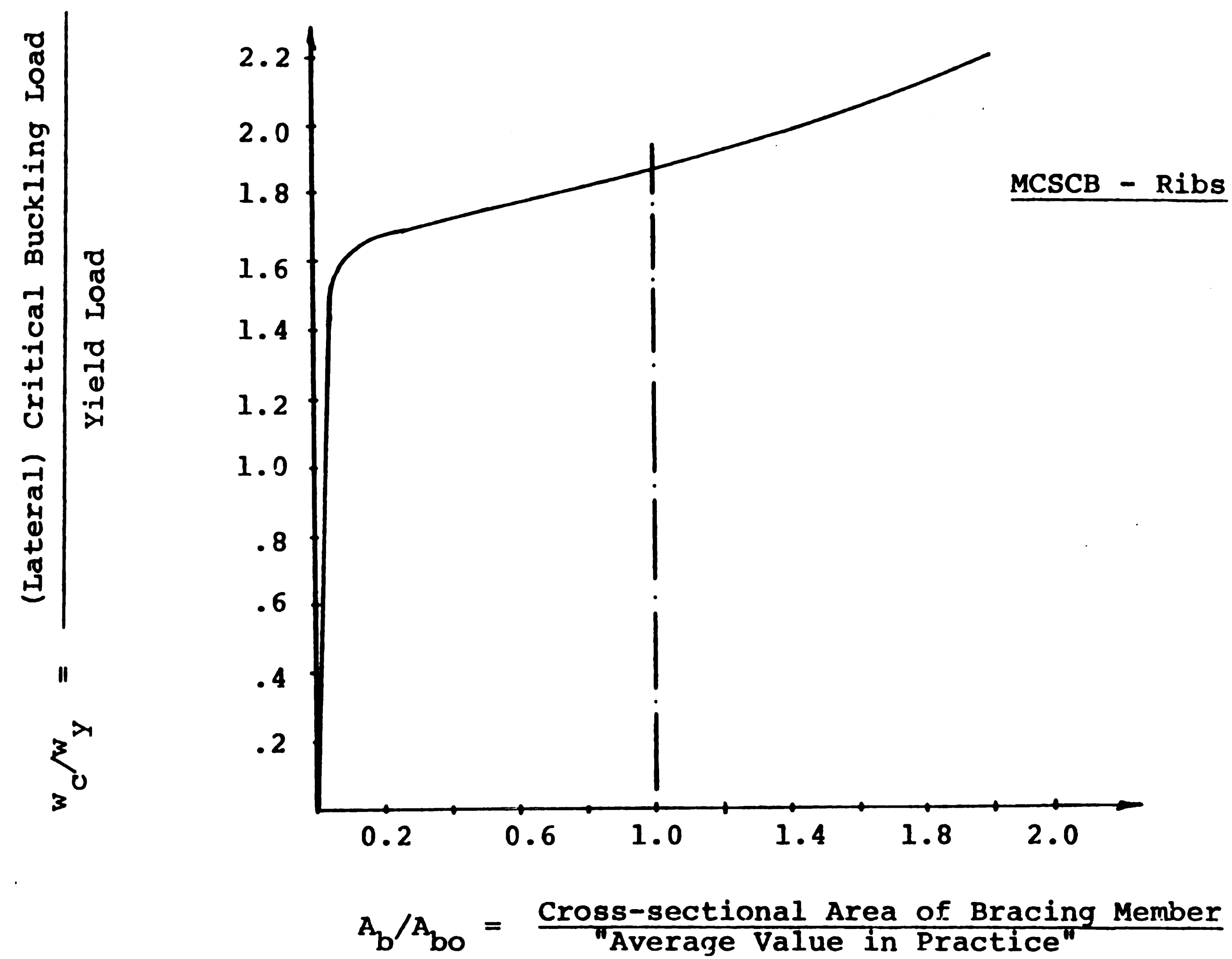


Figure 4-14. Effect of Amount of Bracing on Lateral Buckling Load (A_{bo} = 1/6 Rib Area).

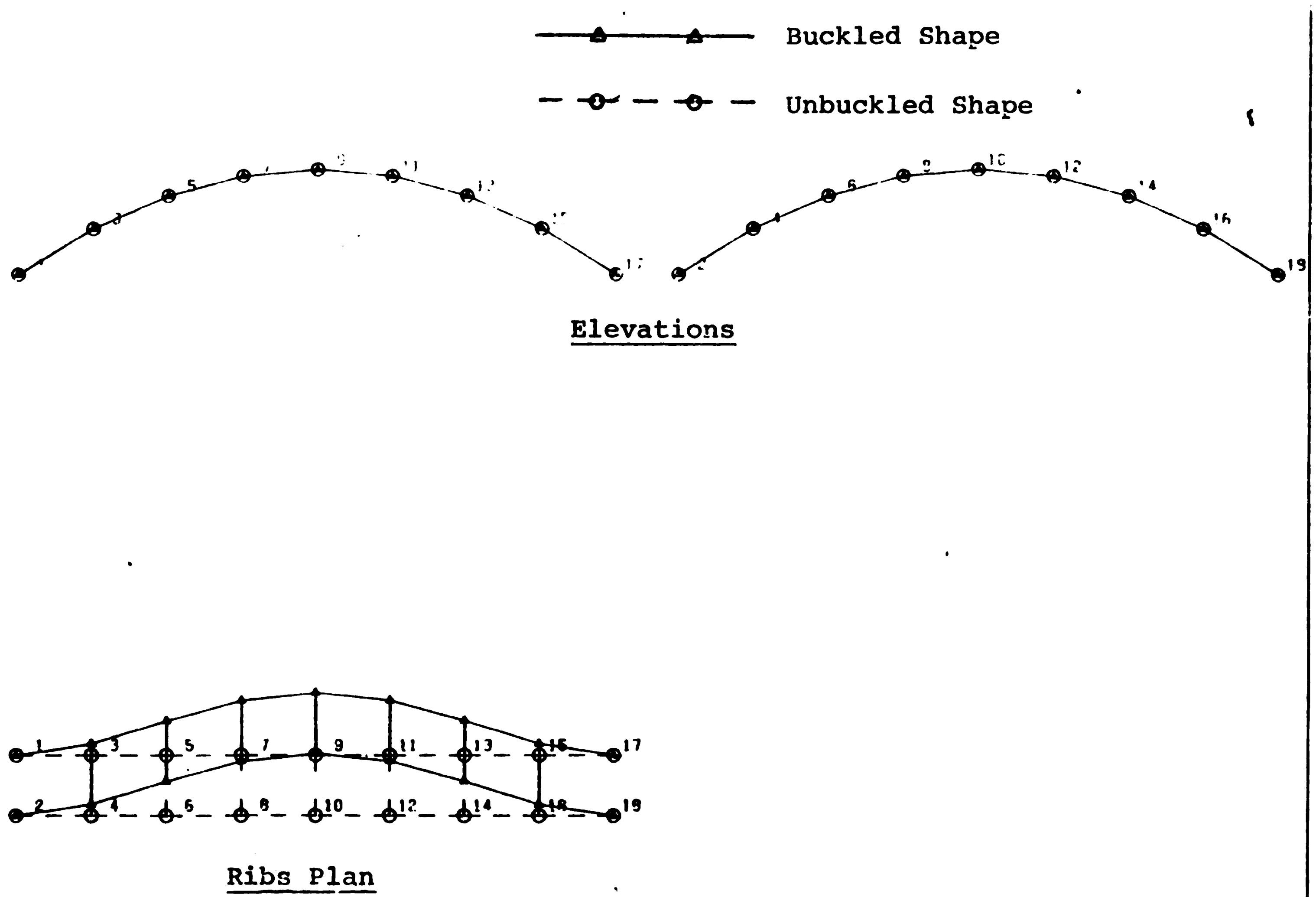


Figure 4-15. Buckled Shape of the MCSCB, 8 Panels, No Deck. $A_b/A_{bo} = 0.001$.

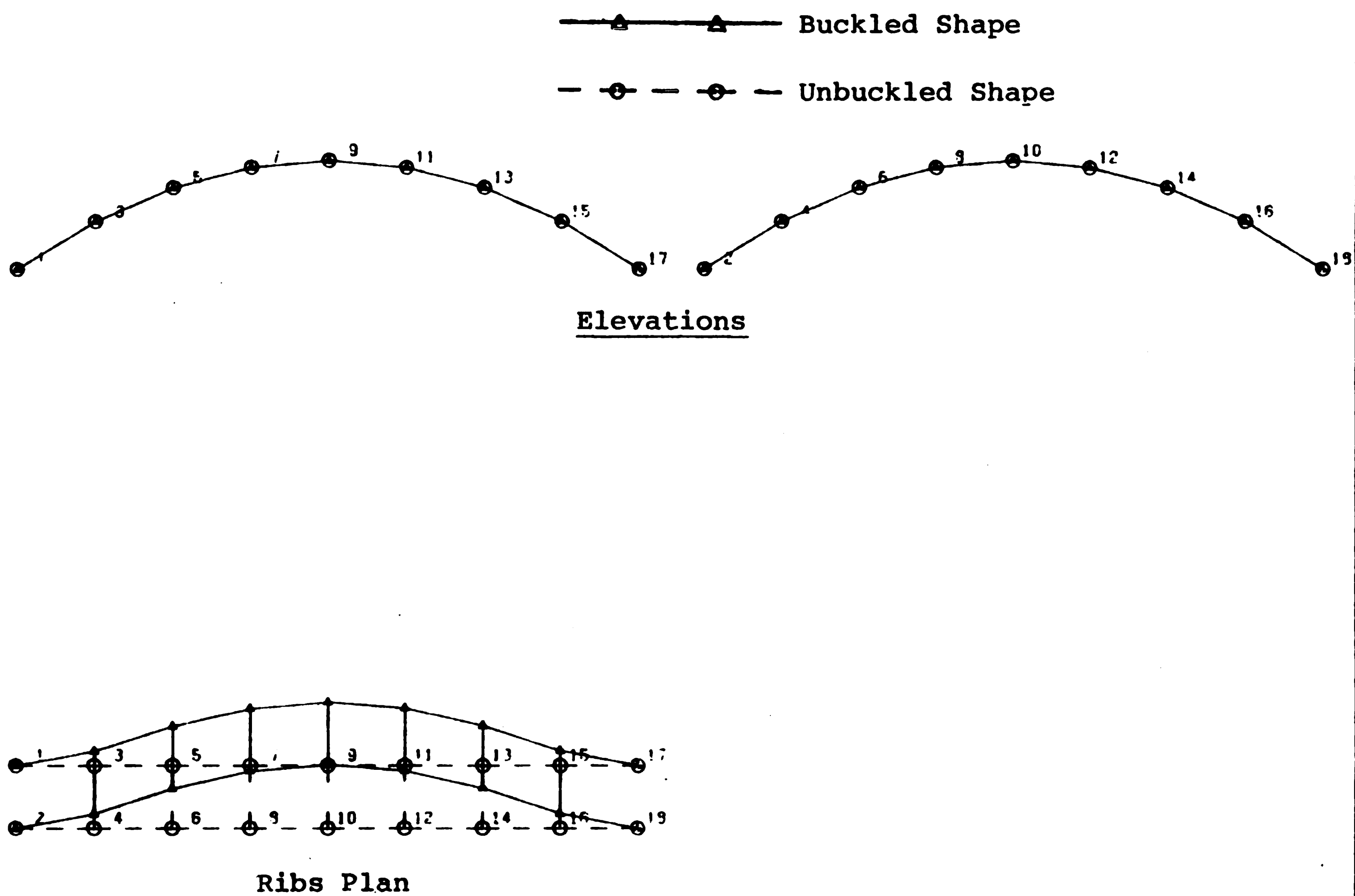


Figure 4-16. Buckled Shape of the MCSCB, 8 panels, No Deck. $A_b/A_{bo} = 0.01$.

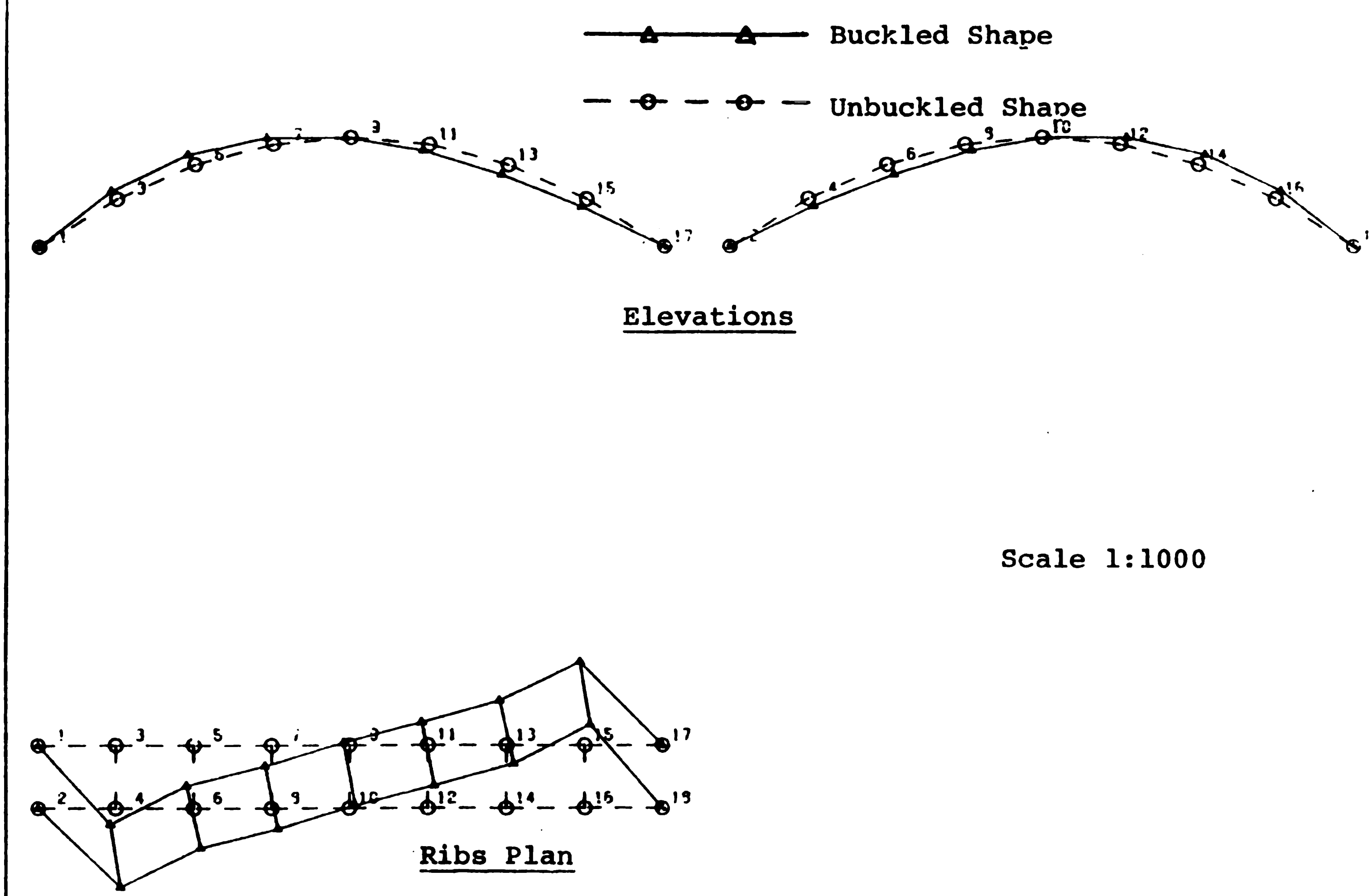


Figure 4-17. Buckled Shape of the MCSCB, 8 Panels, No Deck. $A_b/A_{bo} = 0.25$.

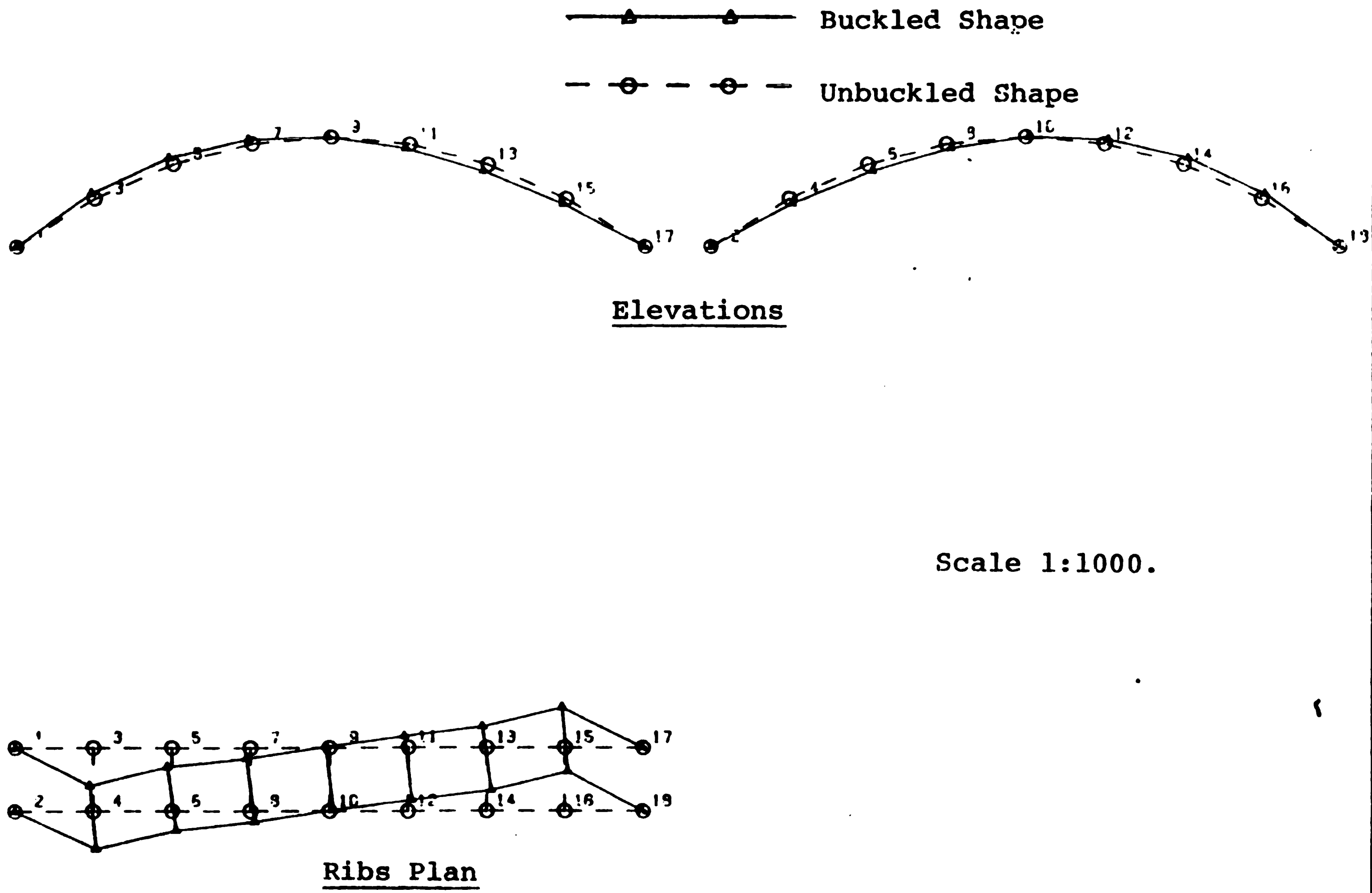


Figure 4-18. Buckled Shape of the MCSCB, 8-Panels, No Deck. $A_b/A_{bo} = 0.5$.

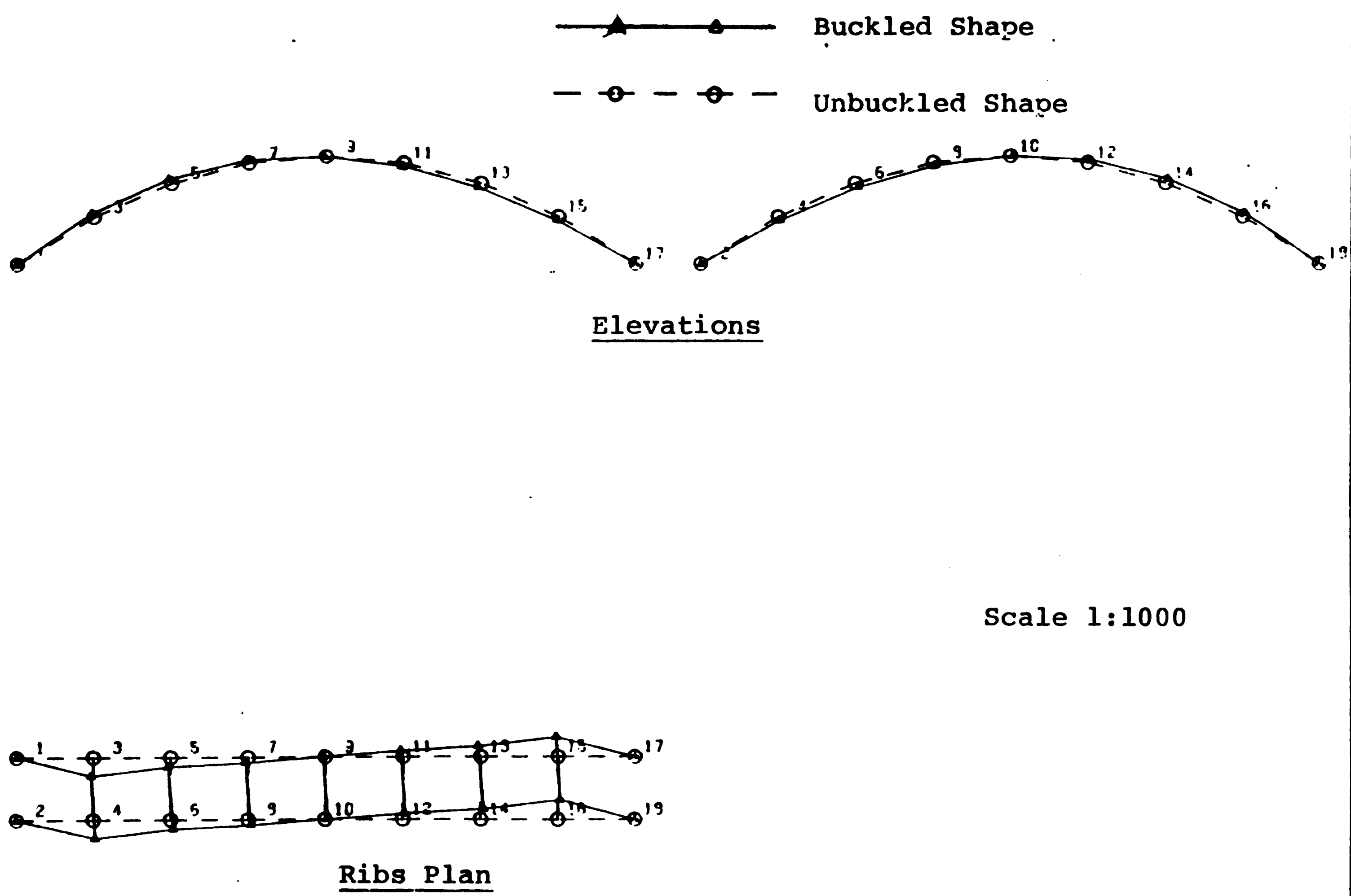


Figure 4-19. Buckled Shape of the MCSCB, 8-Panels, No Deck. $A_b/A_{bo} = 1.0$.

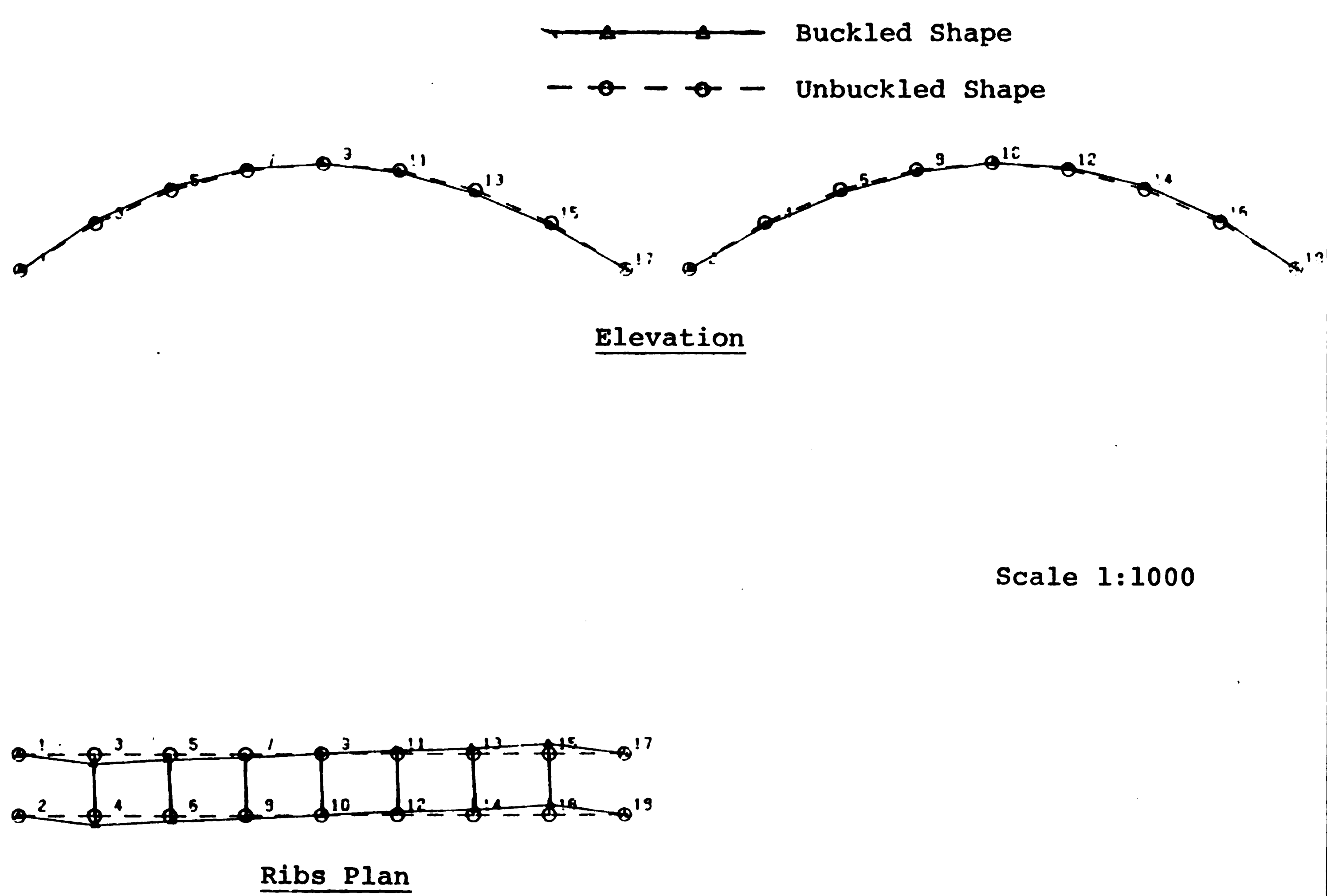


Figure 4-20. Buckled Shape of the MCSCB, 8-Panels, No Deck. $A_b/A_{bo} = 2.0$.

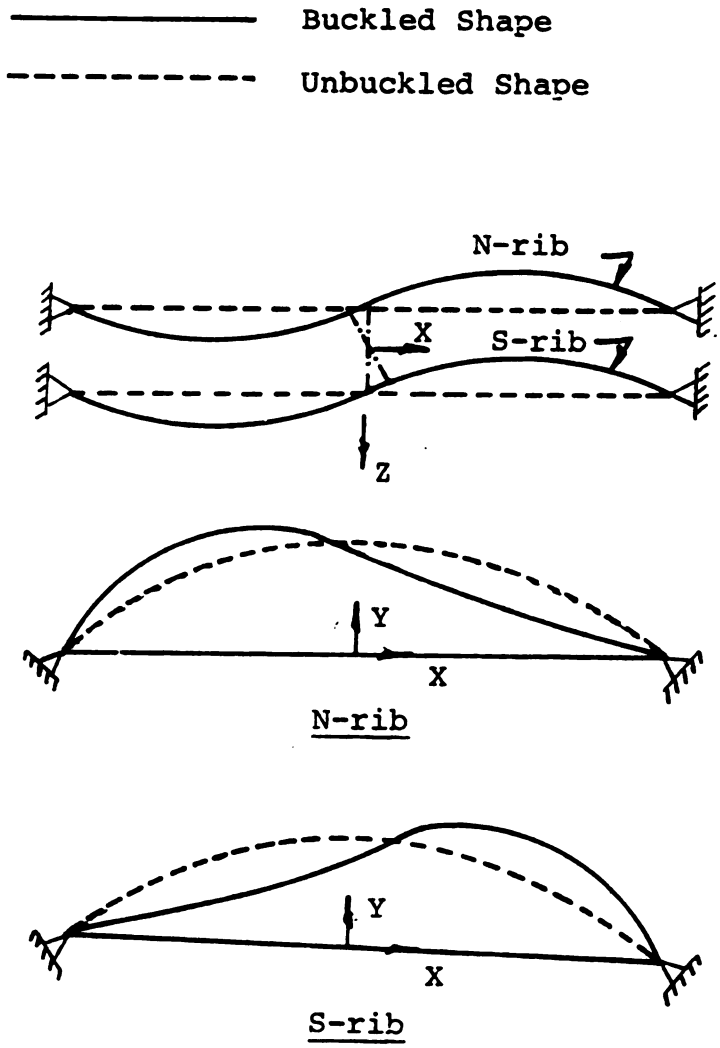
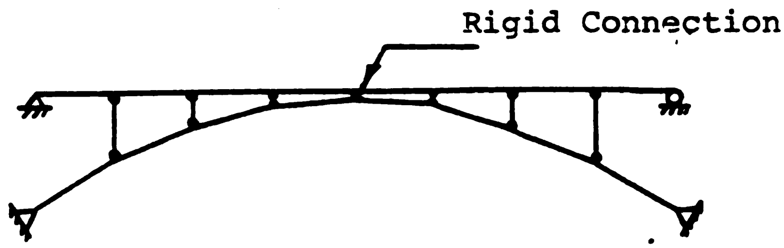
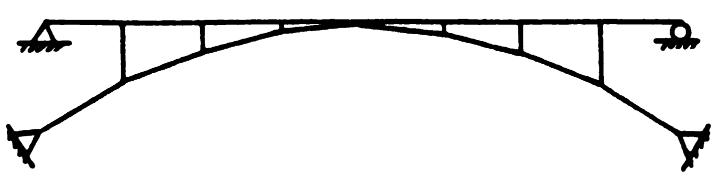


Figure 4-21. The Lowest Anti-Symmetric Out-of-Plane Mode for X-truss Braced Bridge.



(a) Rigidly Connected Column at Crown Only.

(b) Spandrel Bracing of One Middle Panel.



(c) Deck and Ribs Rigidly Connected.

Figure 4-22. Different Types of Column Connections Between Deck and Arch Ribs.

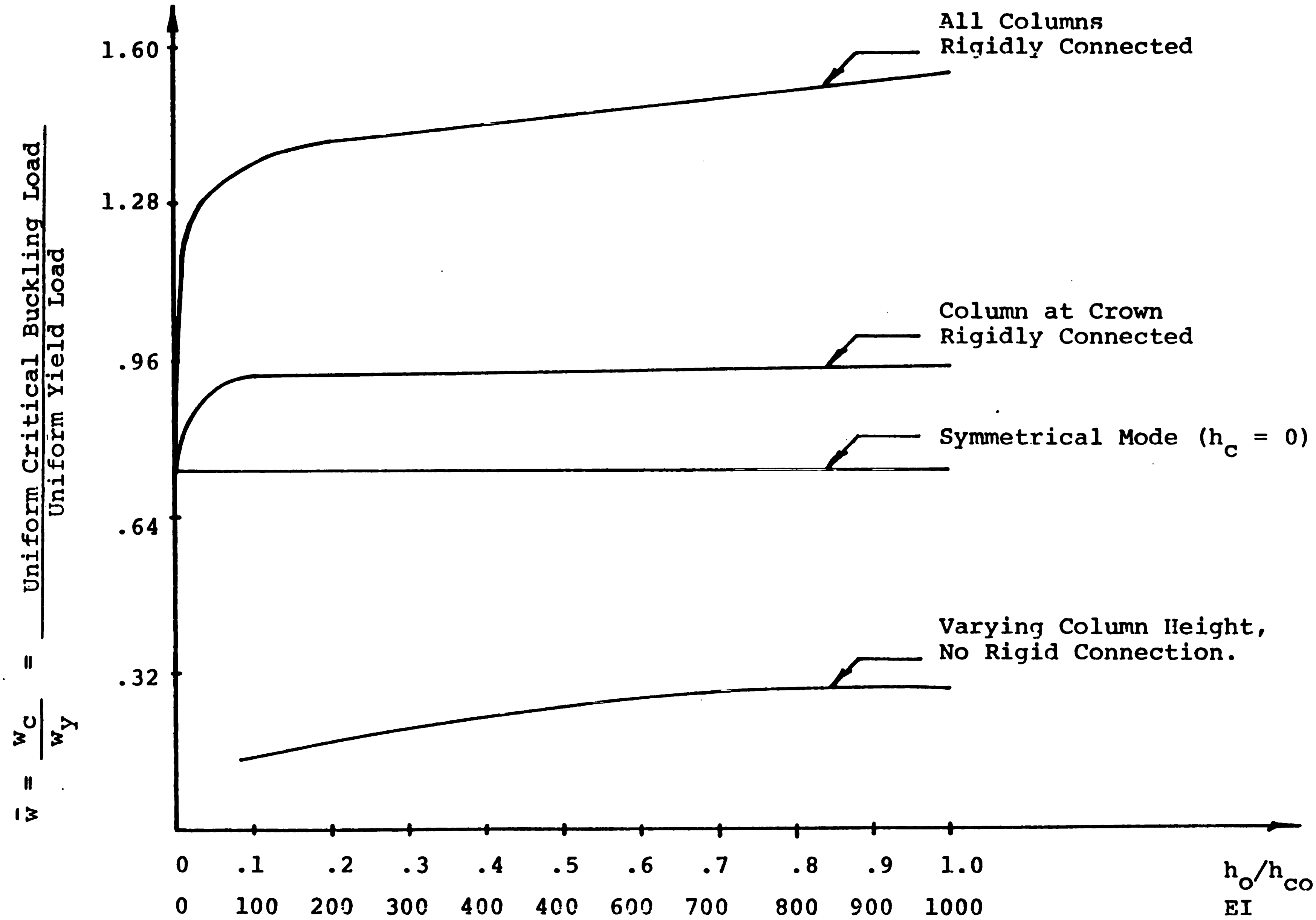


Figure 4-23. Effect of Deck Elevation and Rigidly Connected Columns Flexural Stiffness.

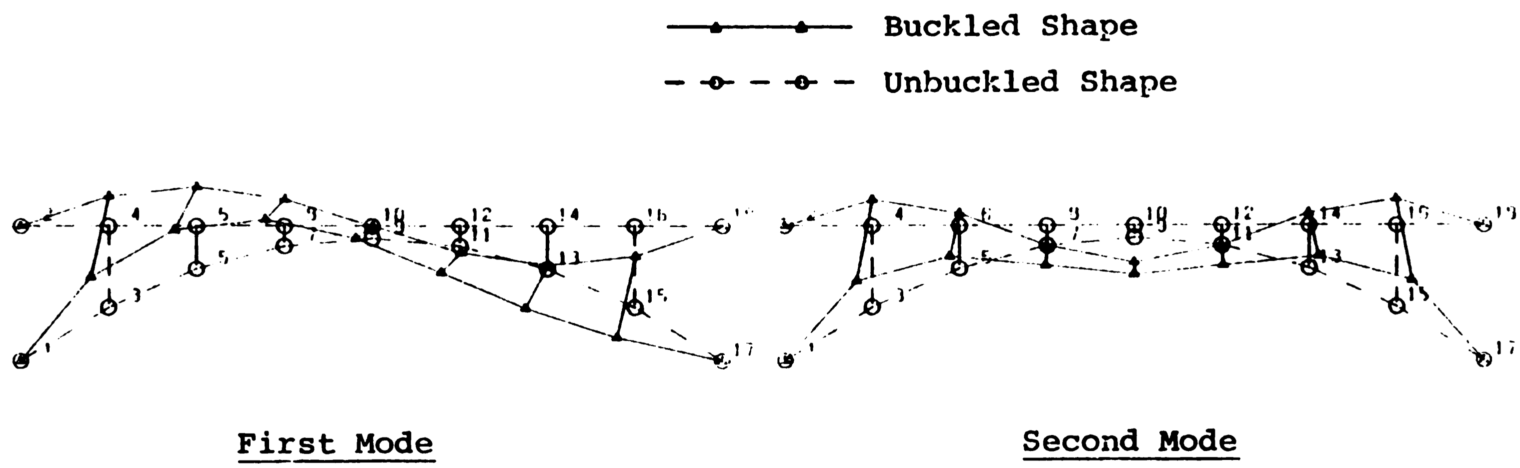


Figure 4-24. Buckled Shape of MCSCB, Single Rib with Deck, Pinned Columns.

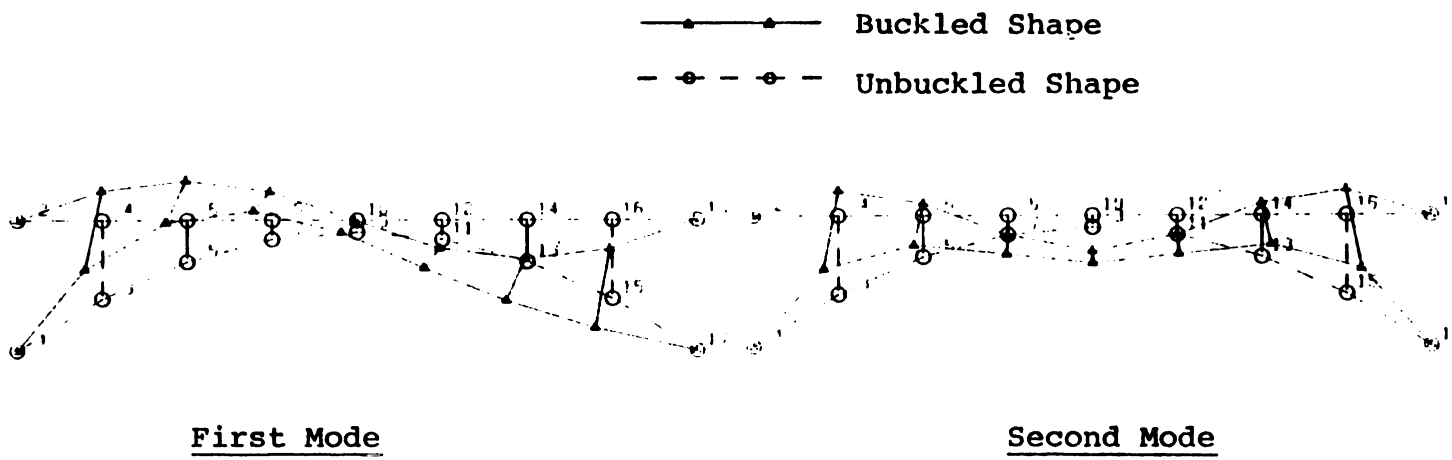


Figure 4-25. Buckled Shape of MCSCB, single rib with deck. Column at crown is rigidly connected ($EI = EI_o$) all other columns are pinned.

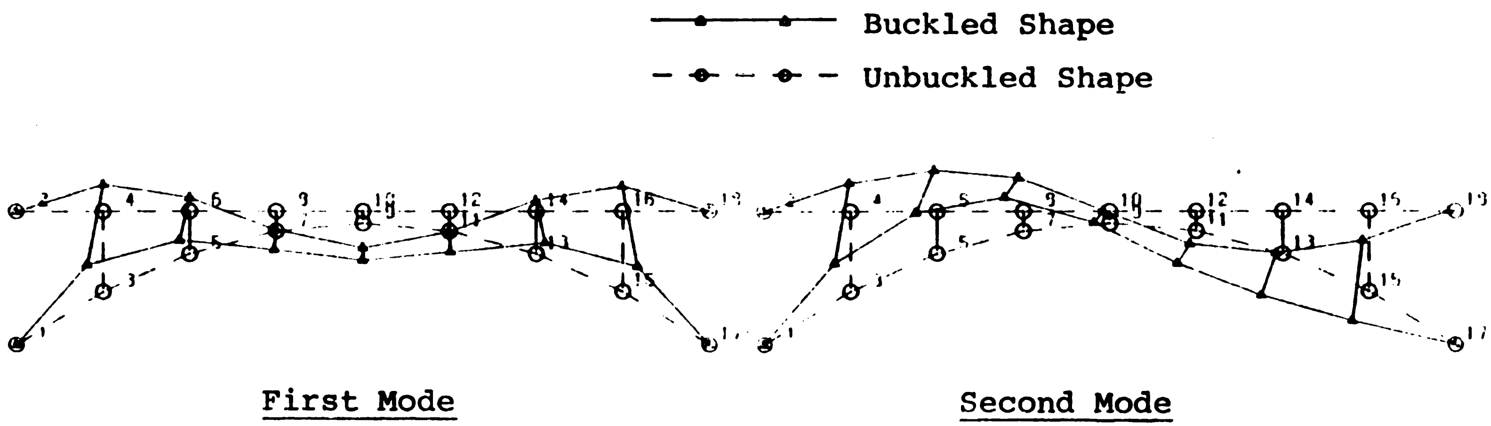


Figure 4-26. Buckled Shape of MCSCB, single rib with deck. Column at crown is rigidly connected ($EI = 10EI_o$). All other columns are pinned.

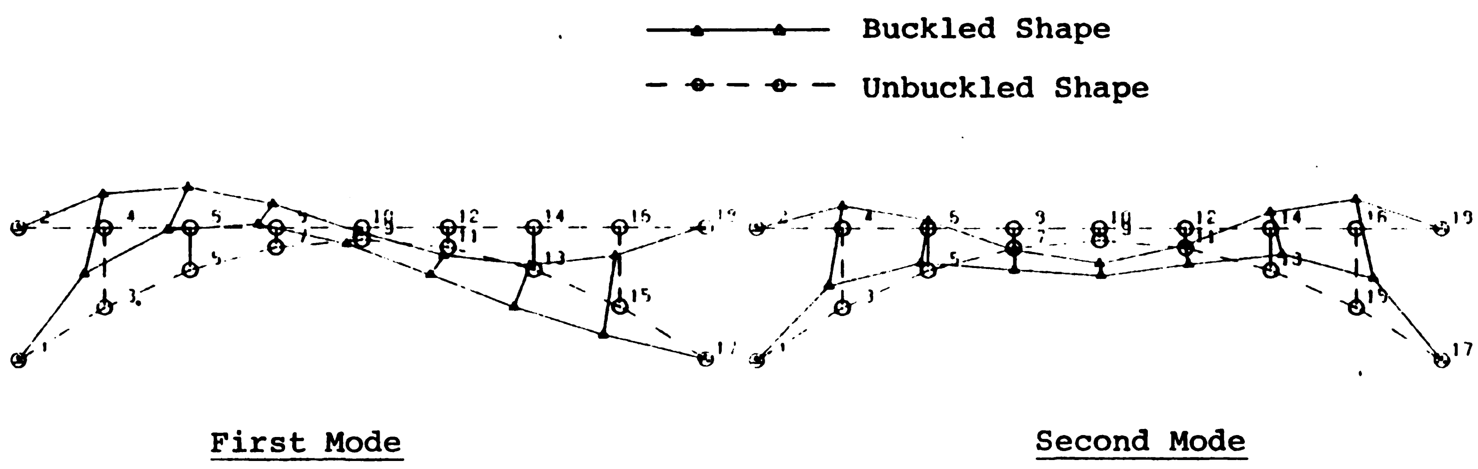


Figure 4-27. Buckled Shape of MCSCB, Single Rib with Deck, All Columns are Rigidly Connected. $EI = EI_o$

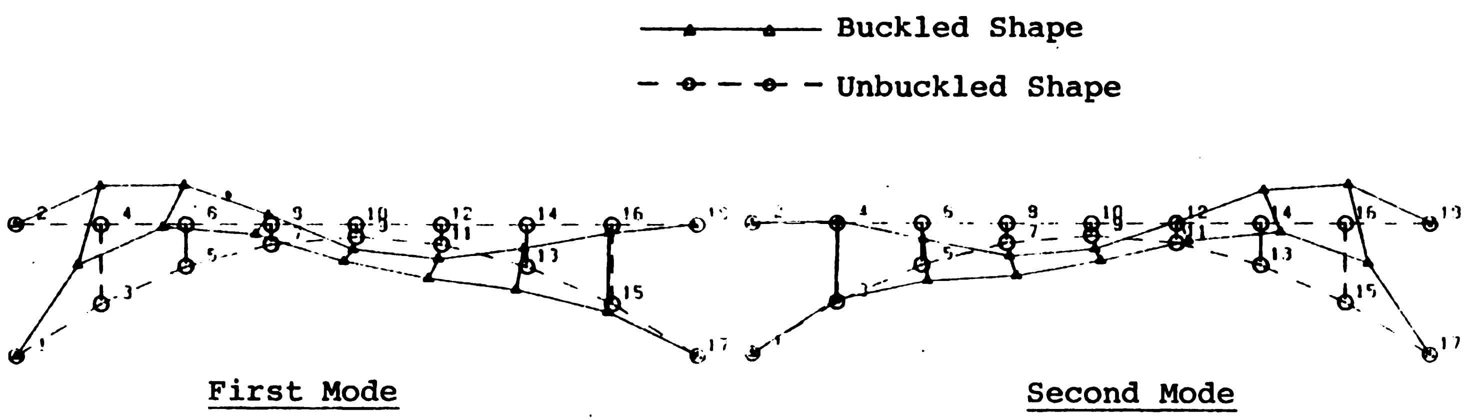


Figure 4-28. Buckled Shape of MCSCB, Single Rib with Deck with Spandrel Bracing in Panel 7-8-9-10, Spandrel Bracing 1-5/8 in. Steel Rope

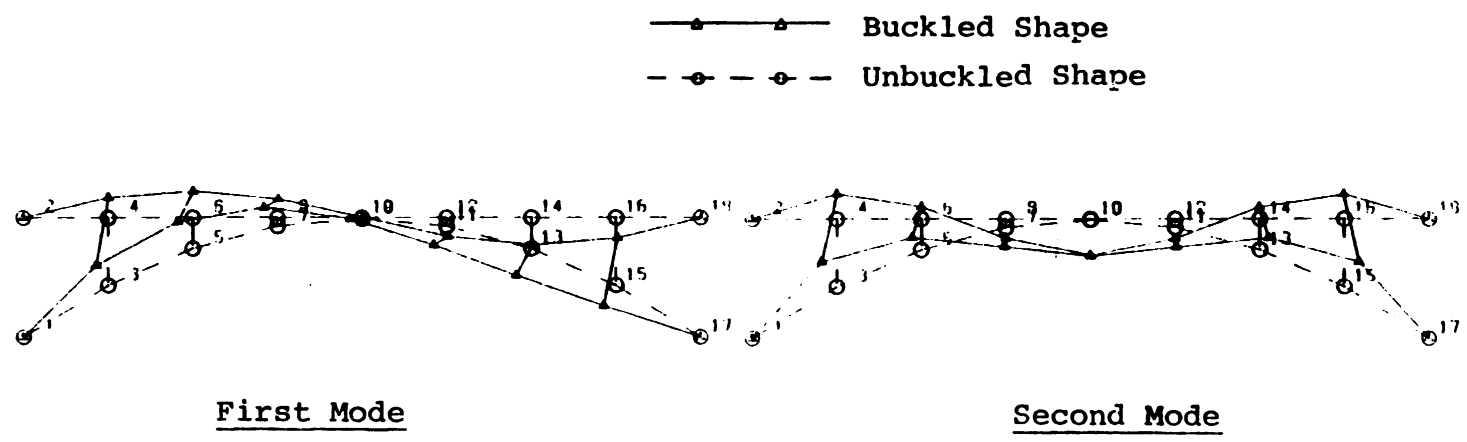


Figure 4-29. Buckled Shape of MCSCB, Single Rib with Deck with Short Pinned Column at Crown. Column Height = 0.08 h_o

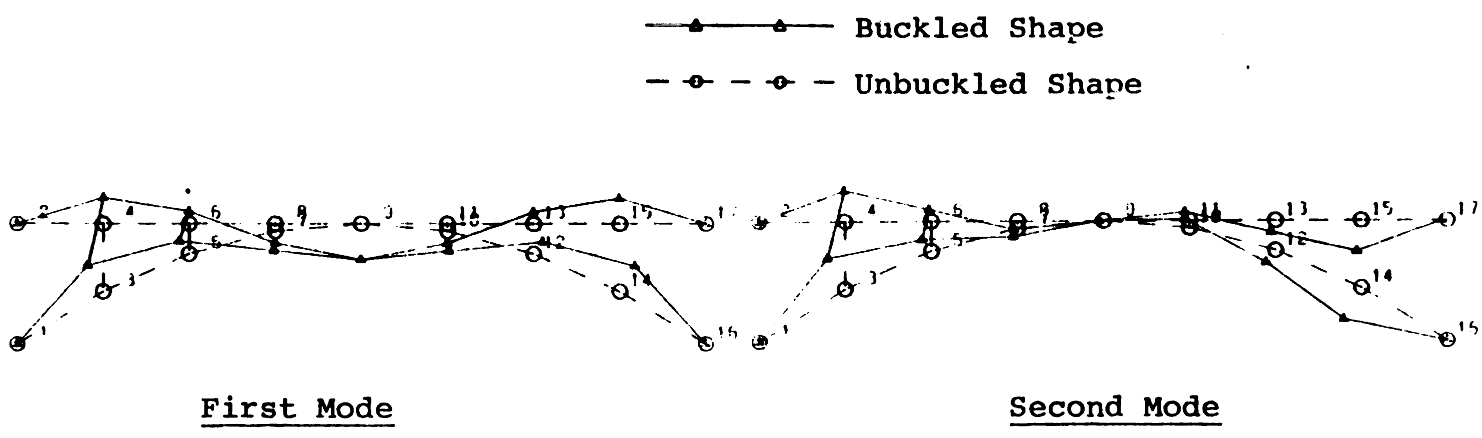


Figure 4-30. Buckled Shape of MCSCB, Single Rib with Deck, Ribs and Deck Rigidly Connected.

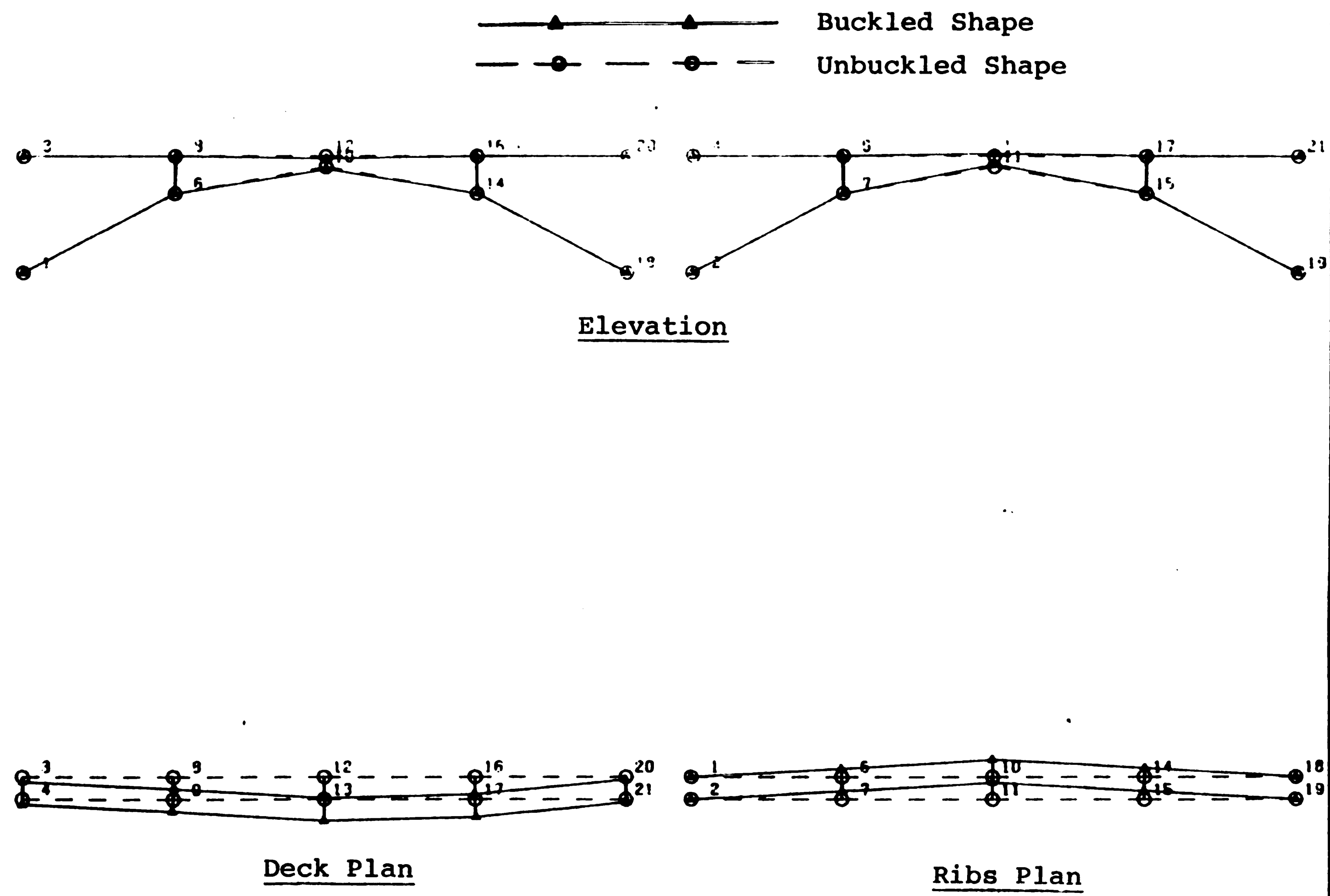


Figure 4-31. Buckled Shape of MCSCB bridge, 4 Panels, No Transverse Bracing. (First Mode)

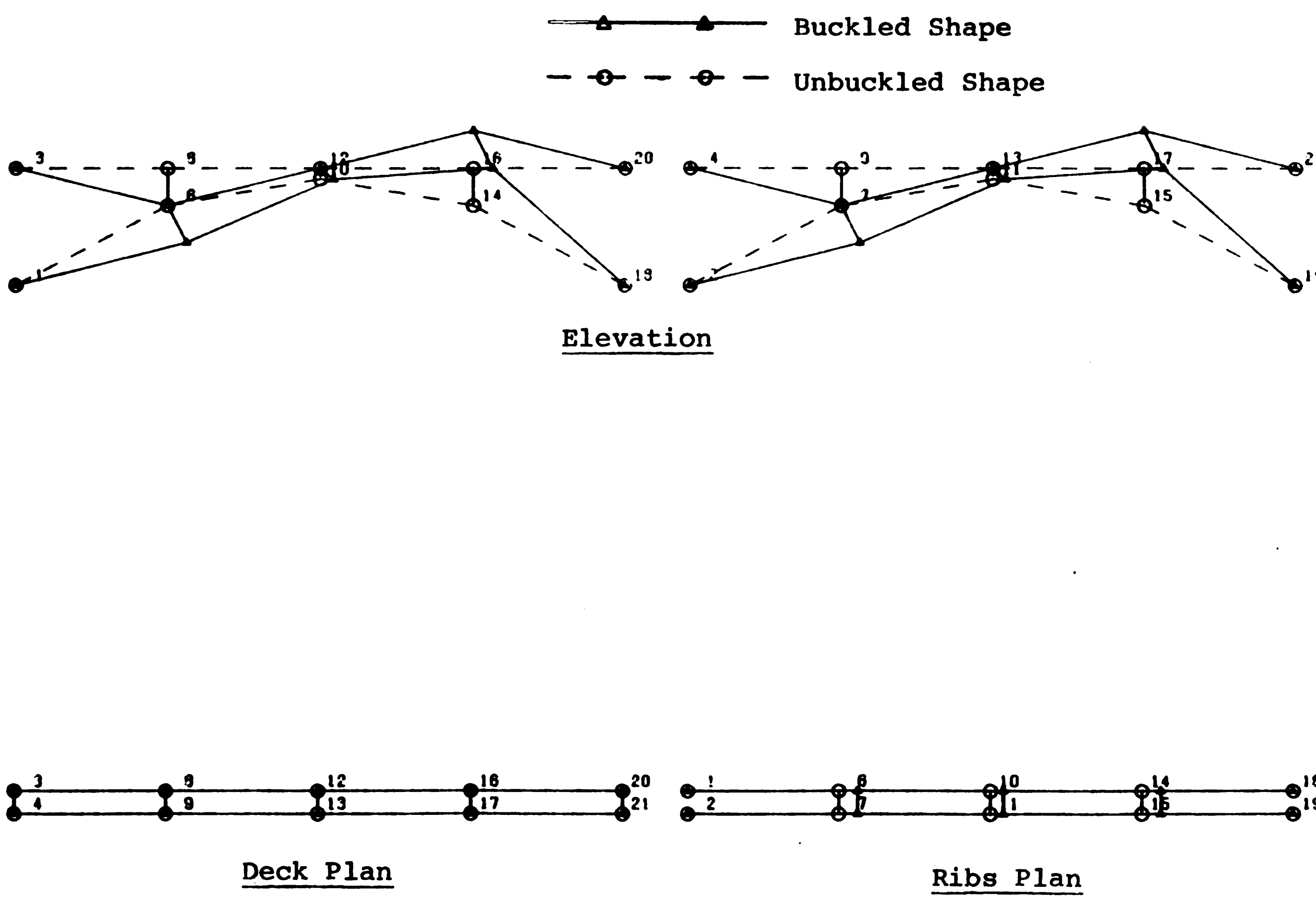


Figure 4-32. Buckled Shape of MCSCB, 4 Panels, No Transverse Bracings. (Second Mode)

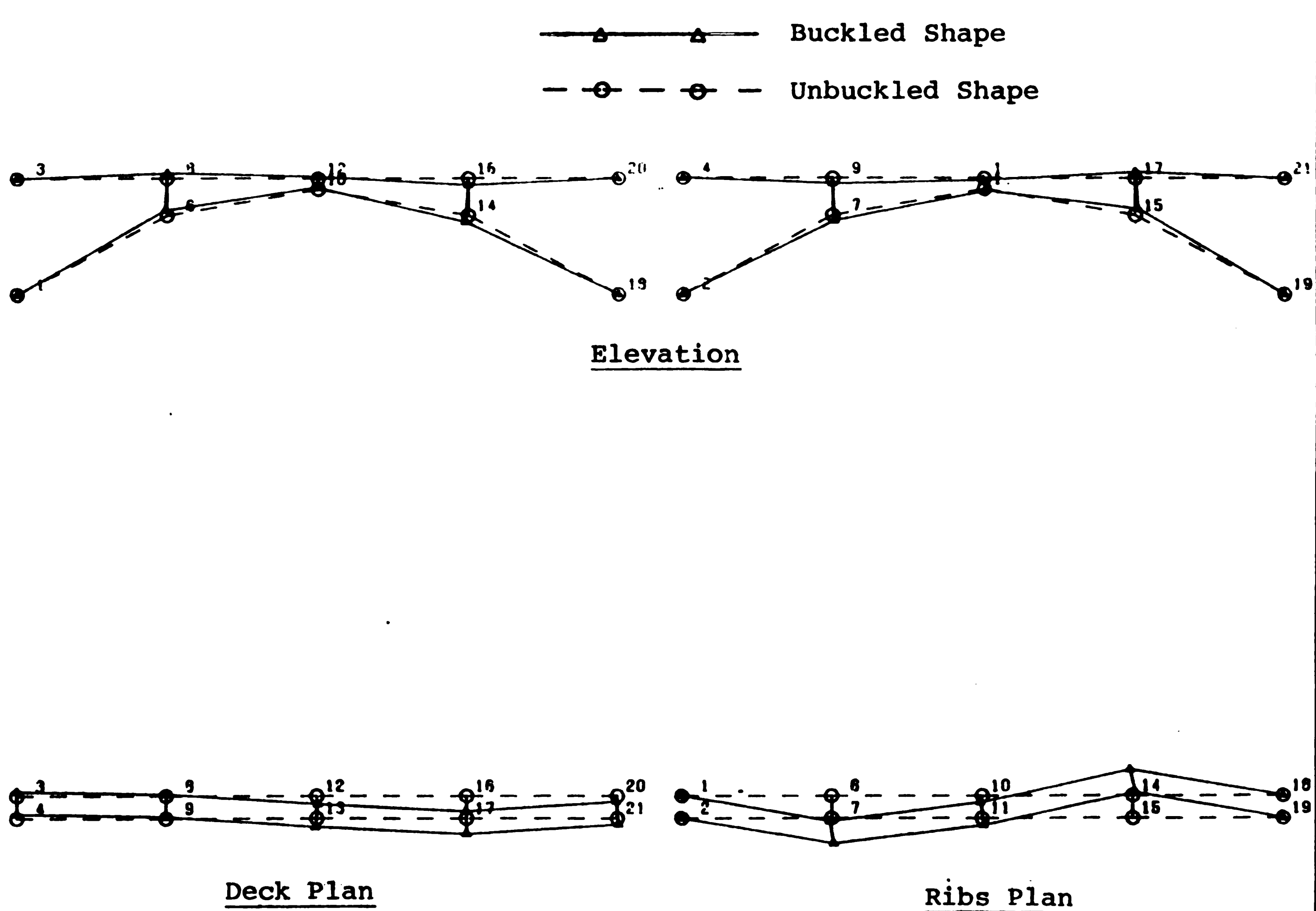


Figure 4-33. Buckled Shape of MCSCB, 4 Panels, No Transverse Bracing. (Fourth Mode)

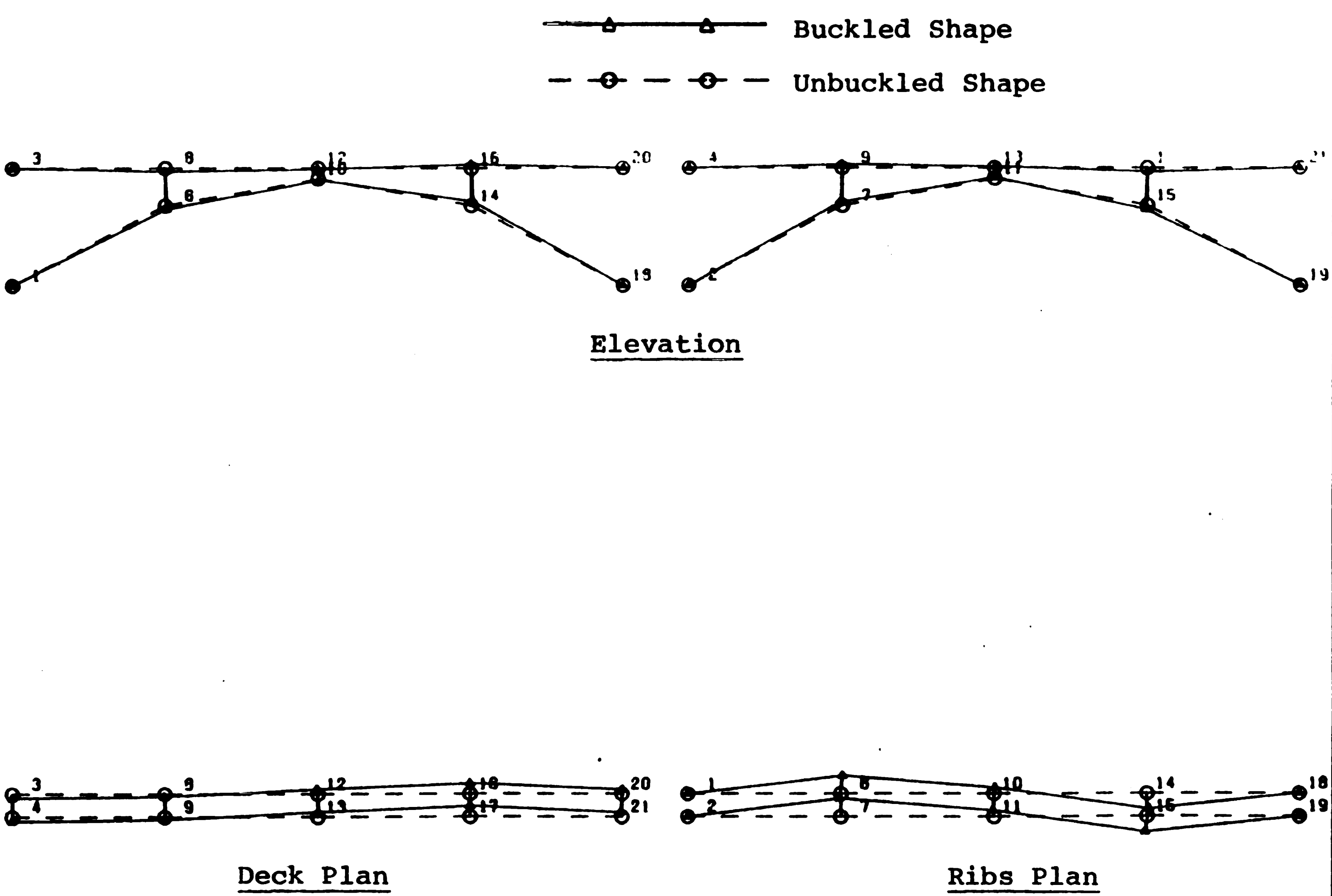


Figure 4-34. Buckled Shape of MCSCB, 4 Panels, No Transverse Bracing. (Fifth Mode)

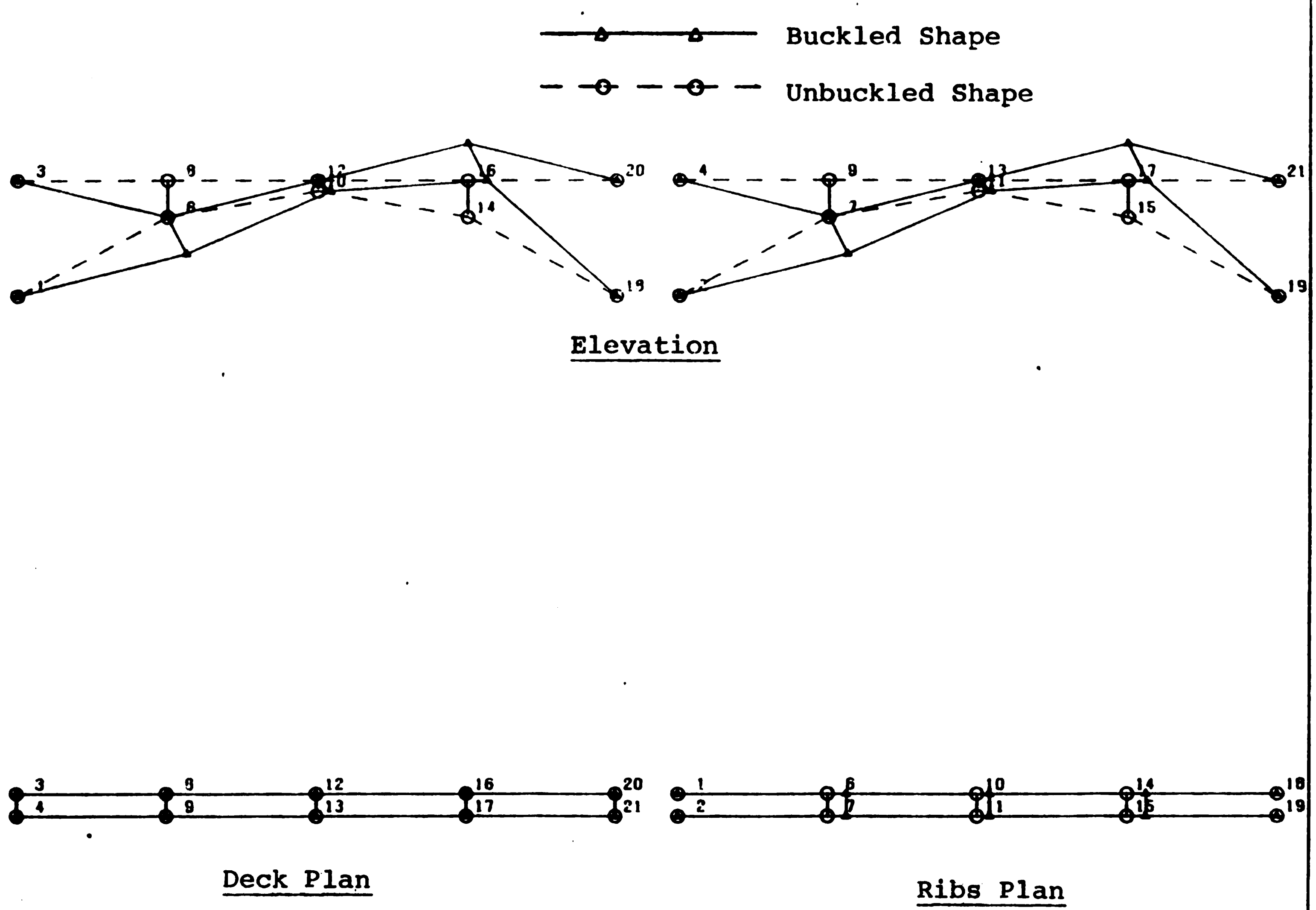


Figure 4-35. Buckled Shape of MCSCB, 4 Panels, Transverse Bracing at Crown. (First Mode)

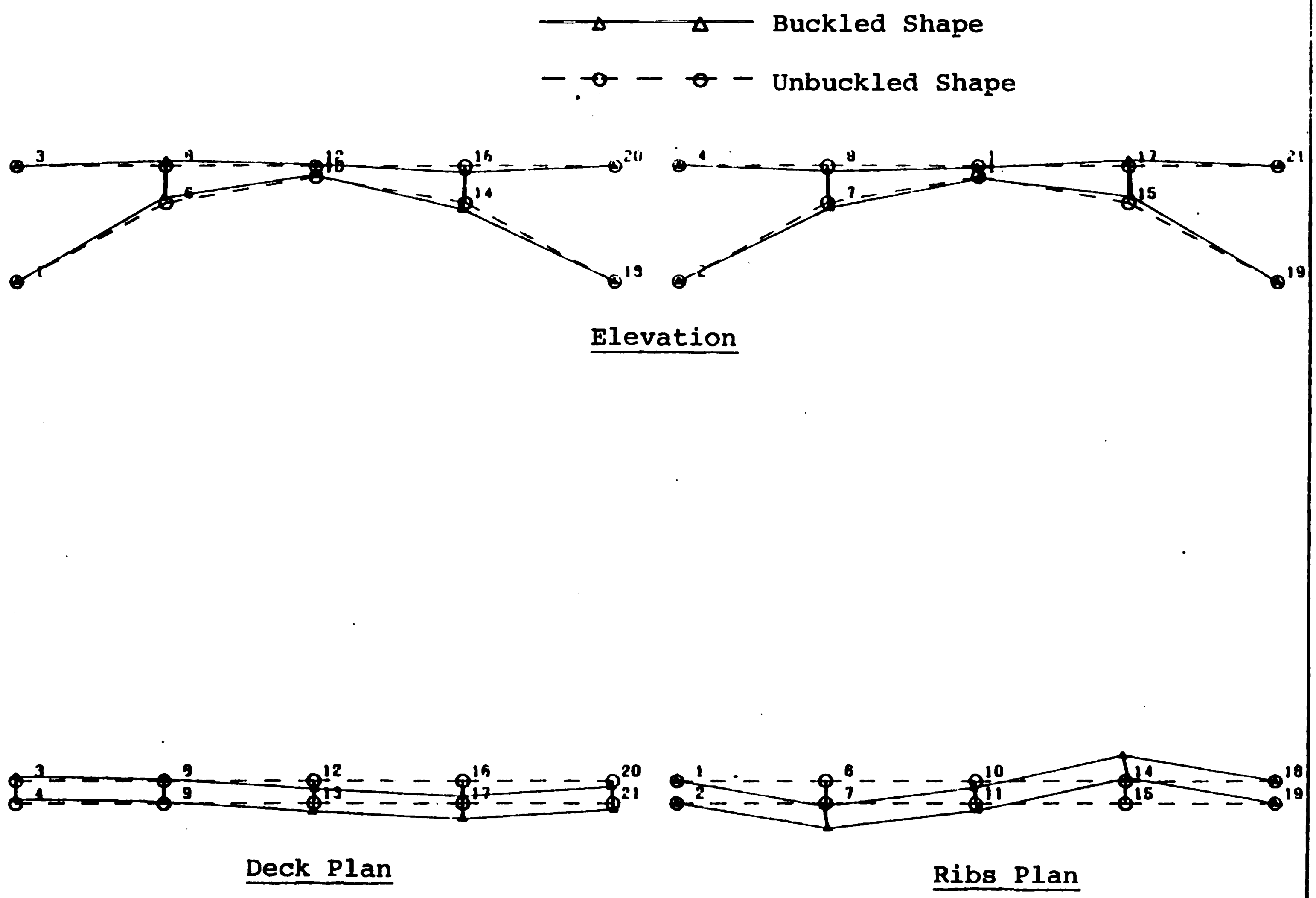


Figure 4-36. Buckled Shape of MCSCB, 4 Panels, Transverse Bracing at Crown. (Third Mode)

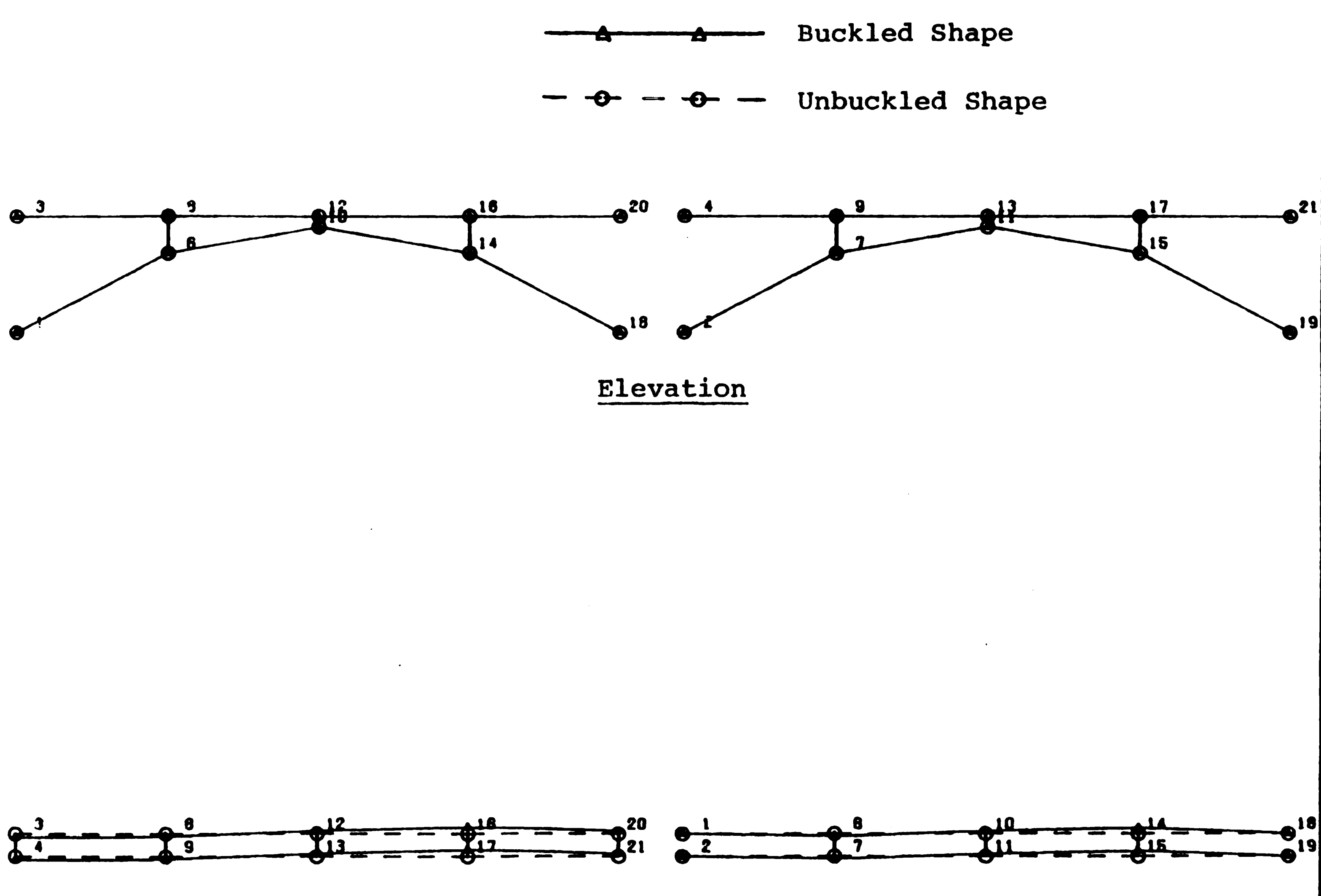


Figure 4-37. Buckled Shape of MCSCB, 4 Panels, Uniformly Transversely Braced. (Third Mode)

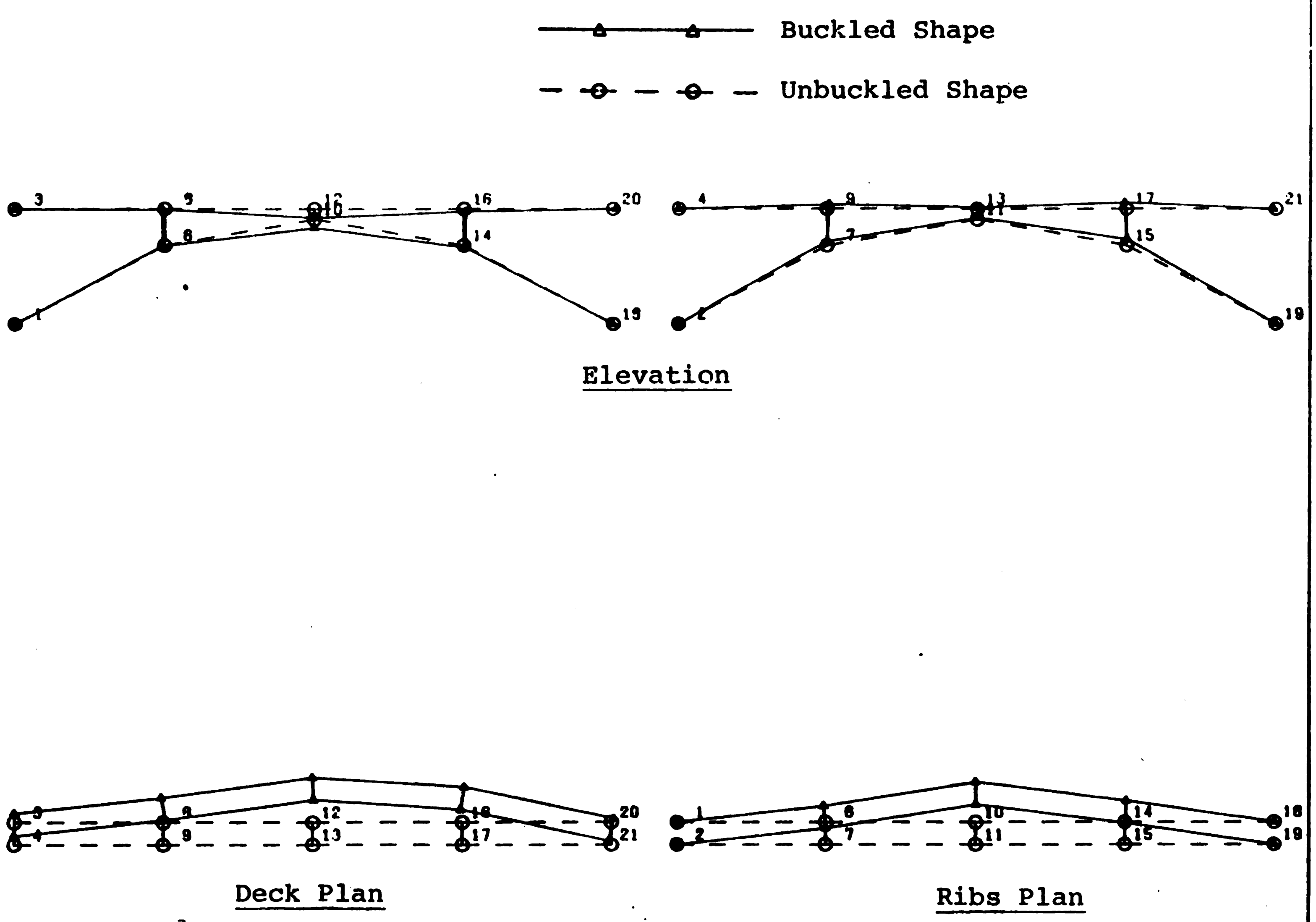


Figure 4-38. Buckled Shape of MCSCB, 4 Panels, Uniformly Transversely Braced. (Fifth Mode)

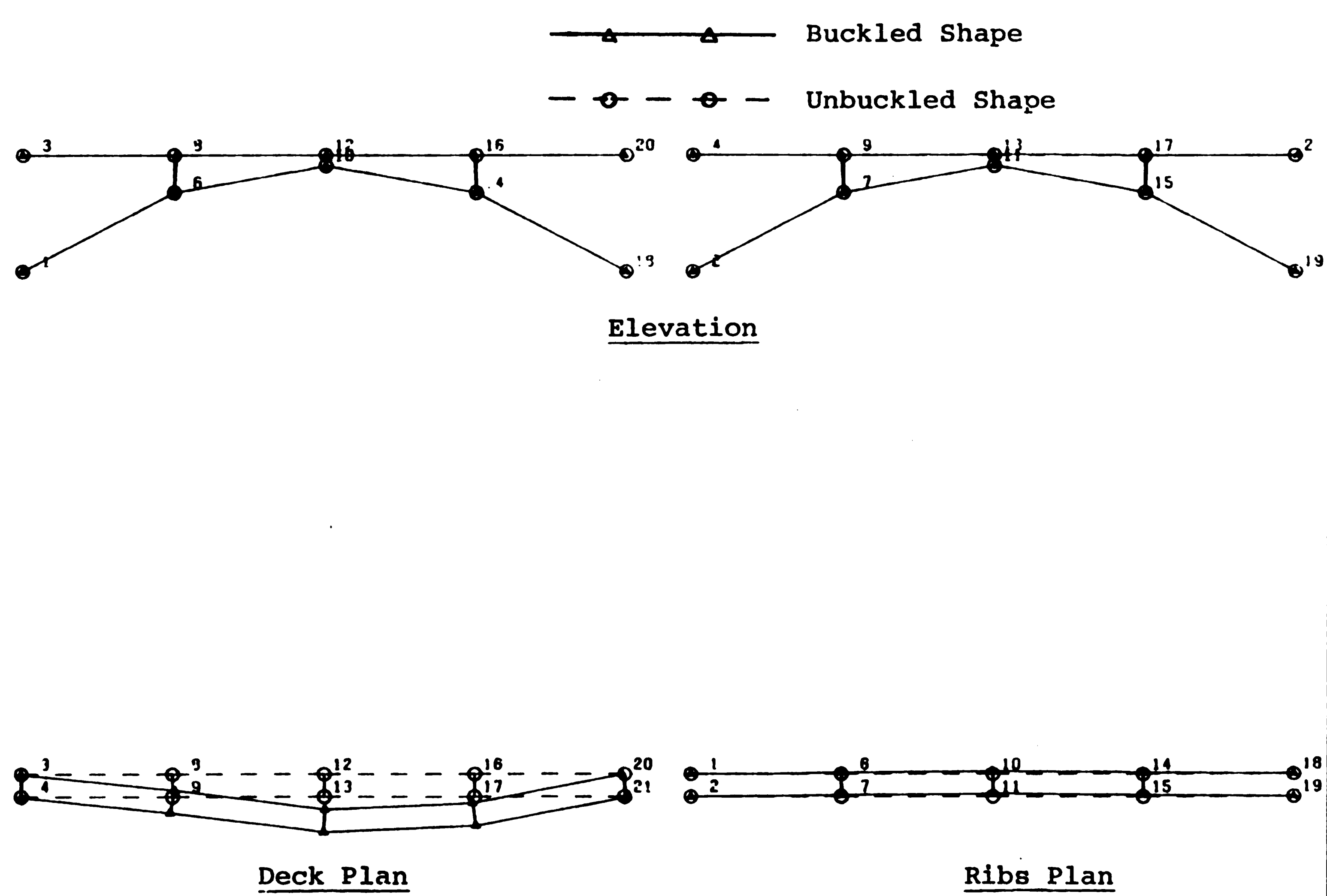


Figure 4-39. Buckled Shape of MCSCB, 4 Panels, No Transverse Bracing. "Deck Lateral Stiffness Reduced By 10." (First Mode)

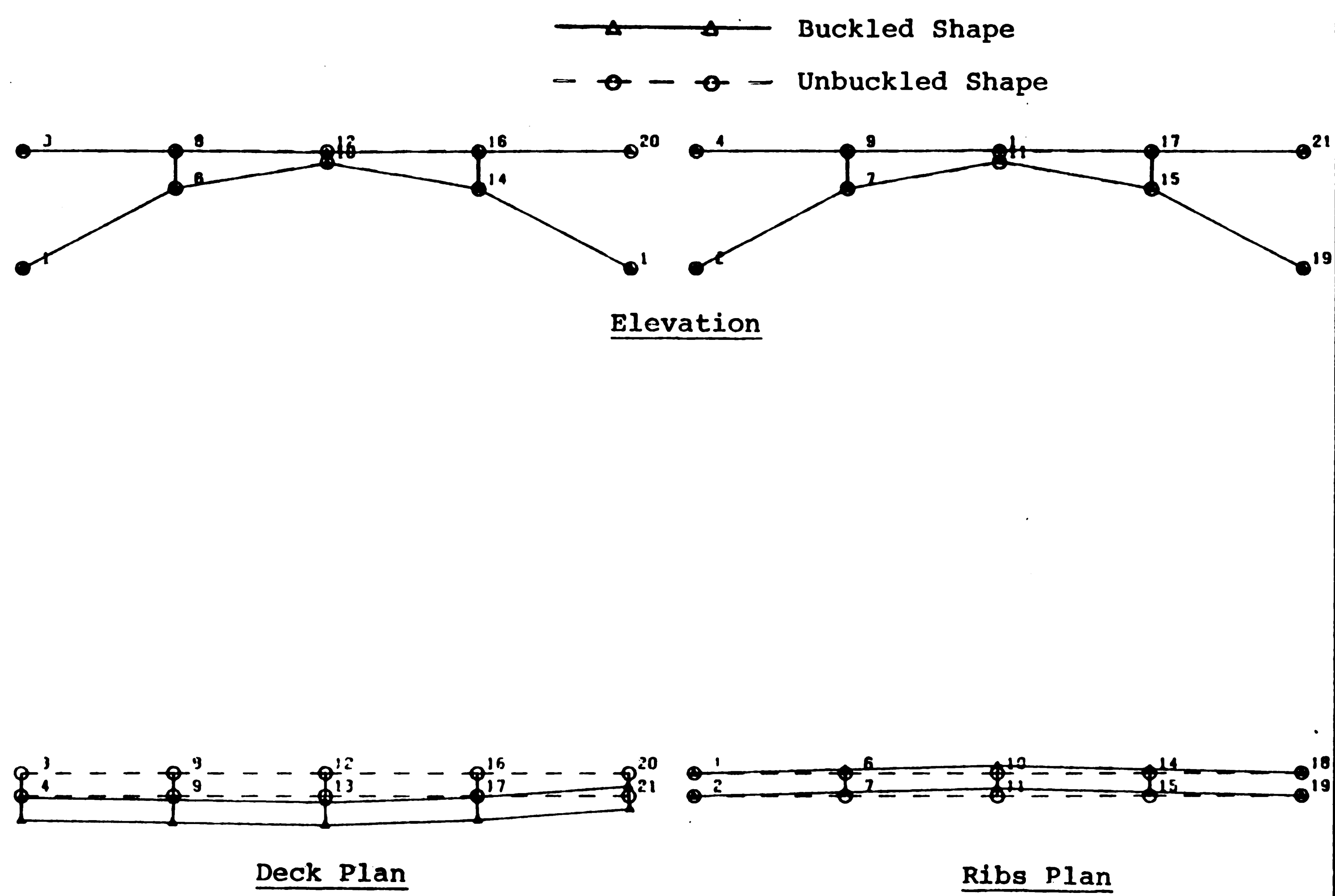
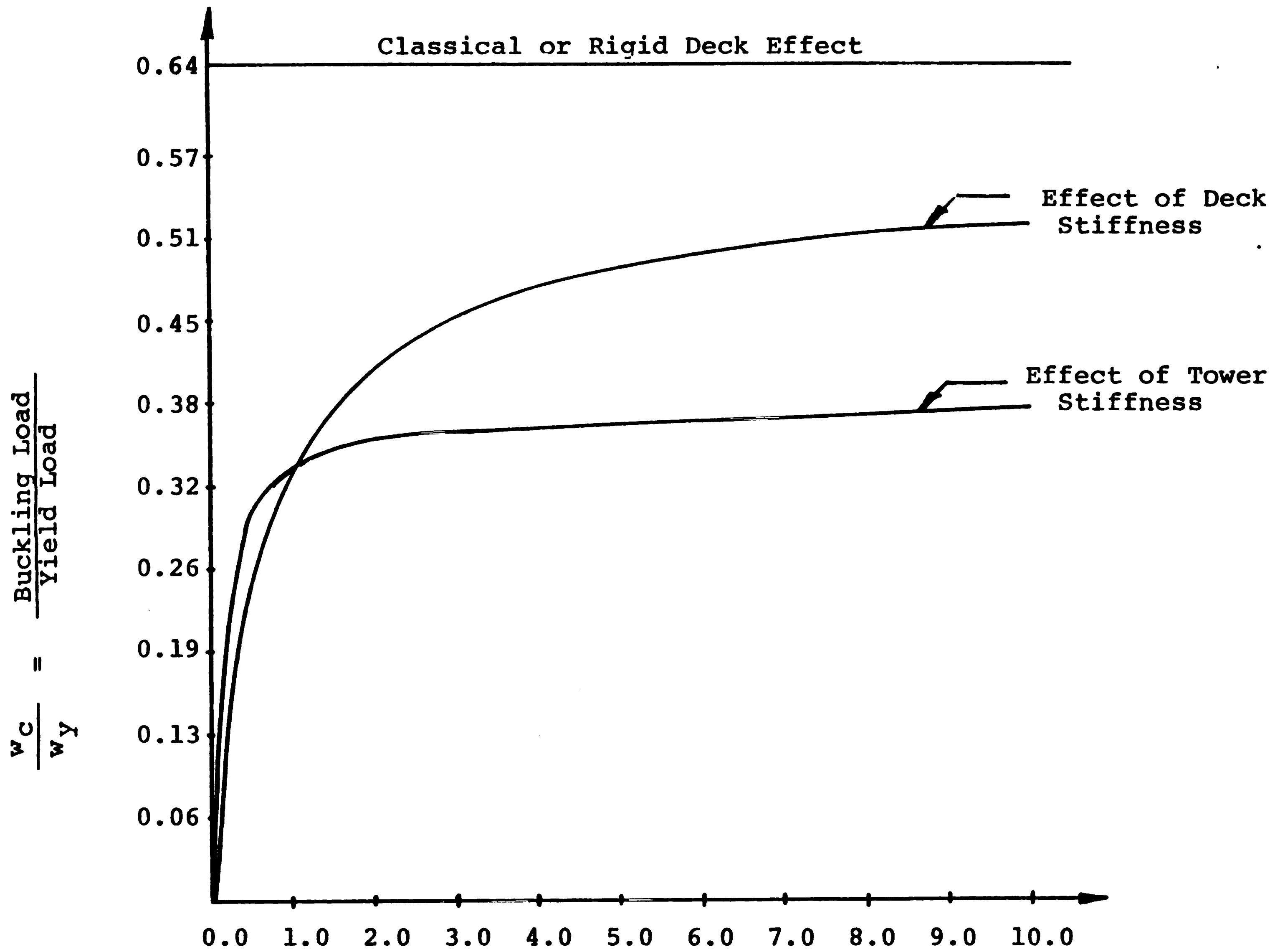


Figure 4-40. Buckled Shape of MCSCB, 4 Panels, No Transverse Bracing Reduced Tower Stiffness by 10. (First Mode)



$$\frac{I_{yy}}{\bar{I}}, \frac{K_T}{\bar{K}_T} = \frac{\text{Deck and Tower Stiffness}}{\text{Practical Value of Deck and Tower Stiffness}}$$

Figure 4-41. Effect of Deck and Tower on Out-of-Plane Buckling.

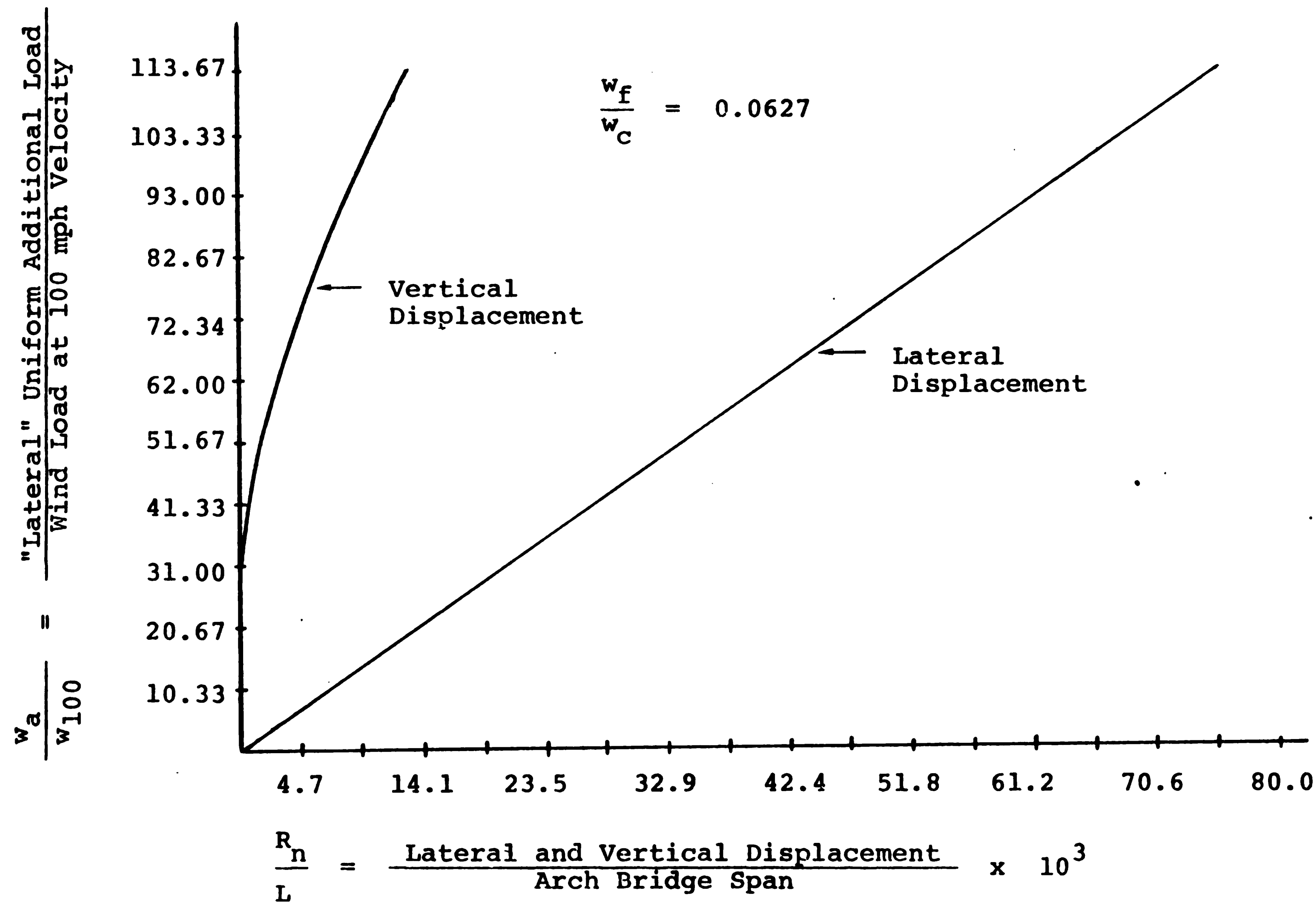


Figure 5-1. Equilibrium Solution of Vertical and Lateral Displacement at Crown.

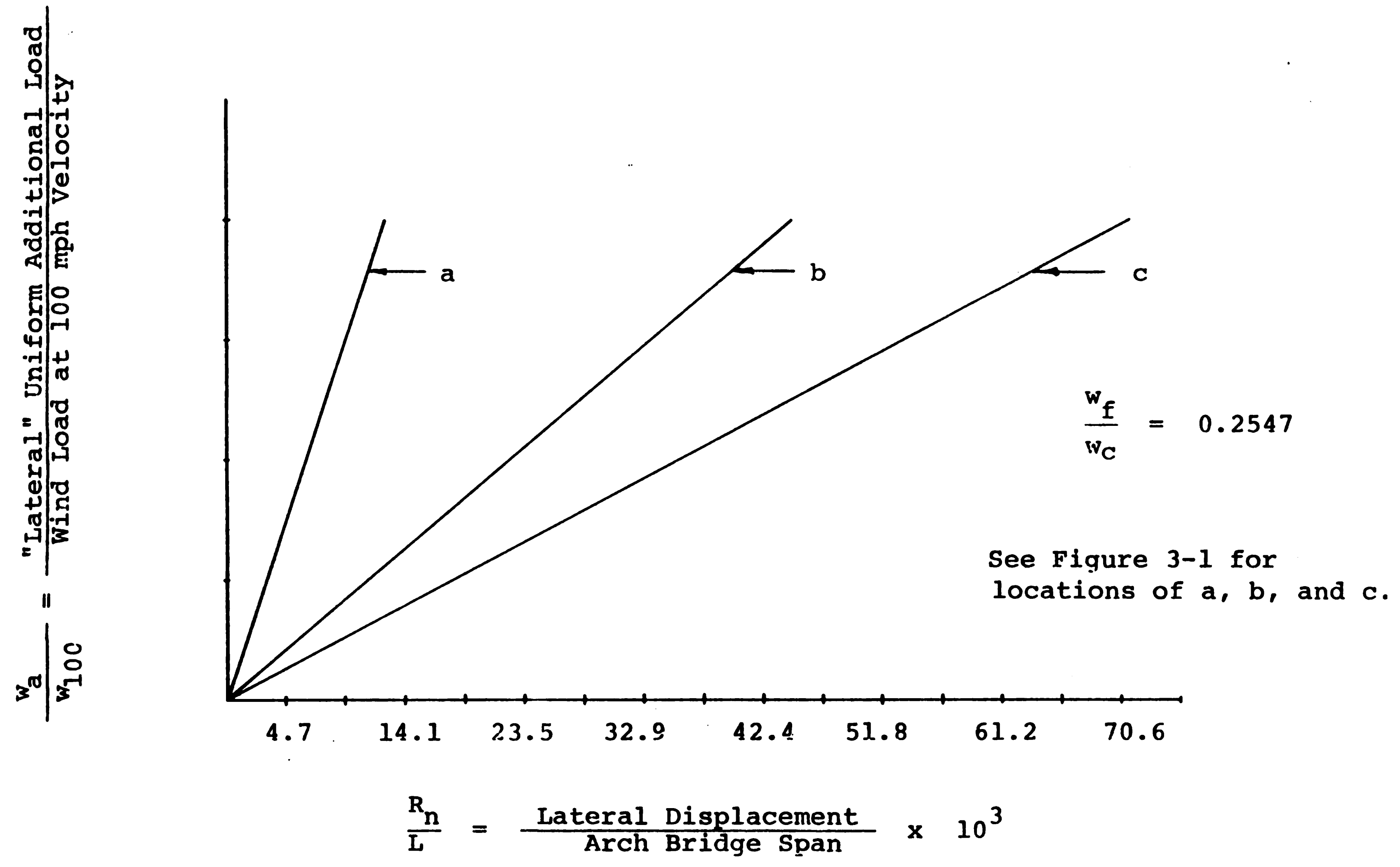
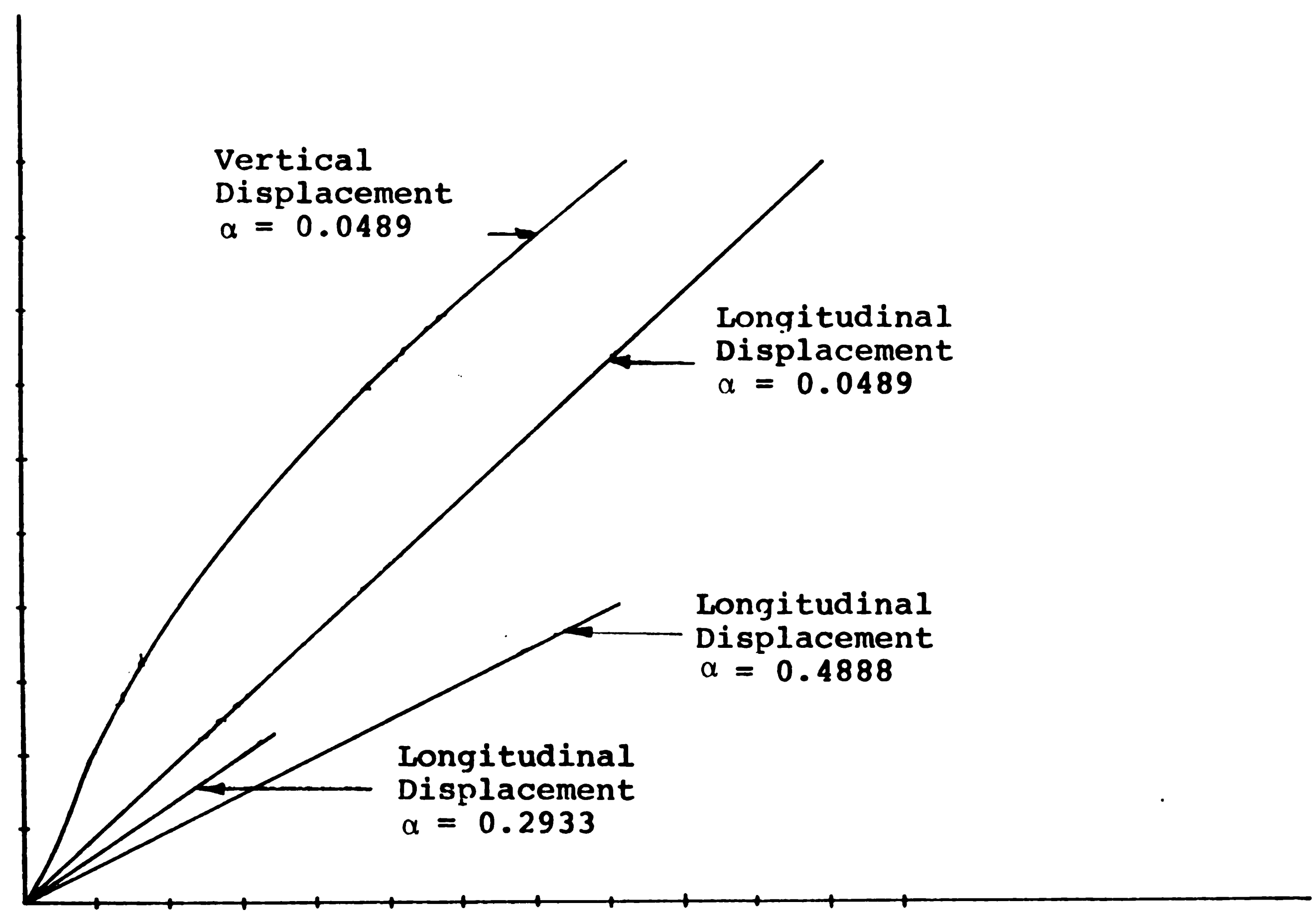


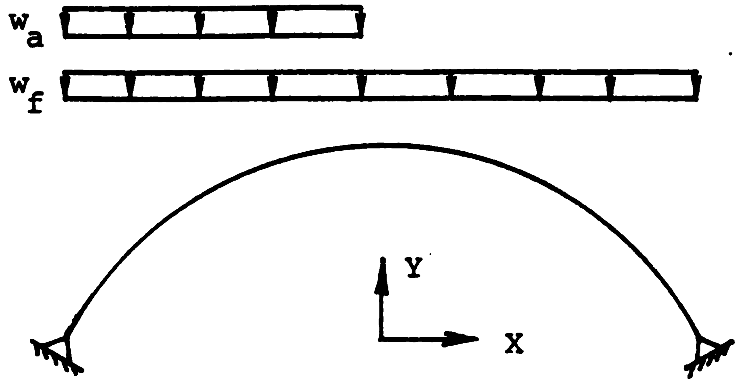
Figure 5-2. Equilibrium Solution of Lateral Displacement for Three Points on Bridge.



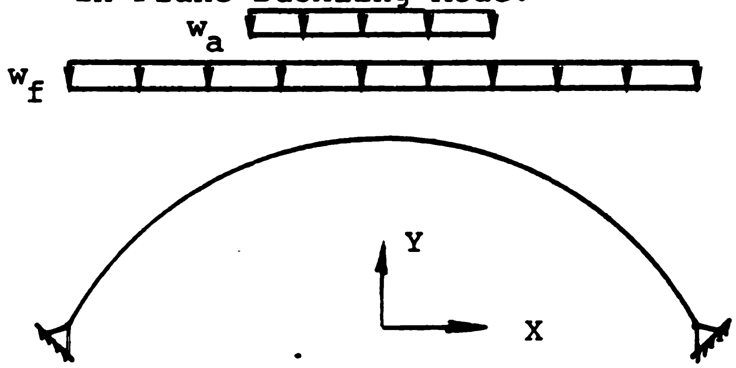
$$\frac{w_a}{w_{100}} = \frac{\text{"Longitudinal" Uniform Additional Load}}{\text{Wind Load at 100 mph Velocity}}$$

$$\frac{R_n}{L} = \frac{\text{Longitudinal and Vertical Displacement}}{\text{Arch Bridge Span}} \times 10^3$$

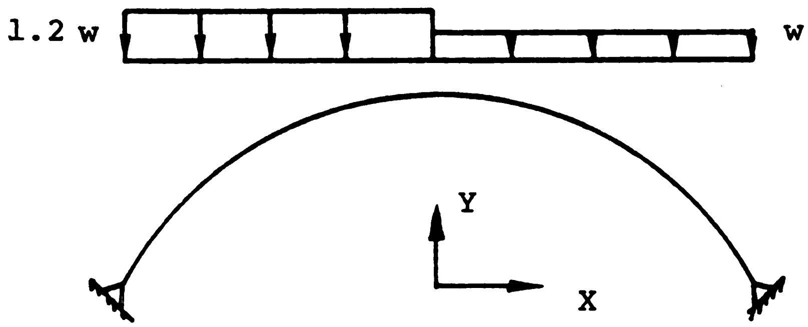
Figure 5-3. Equilibrium Solution of Vertical and Longitudinal Displacement at Crown.



(a) Loading Condition Corresponding to Anti-Symmetric In-Plane Buckling Mode.



(b) Loading Condition Corresponding to Symmetric In-Plane Buckling Mode.



(c) Loading Condition Corresponding to Anti-Symmetric In-Plane Buckling Mode, Non-Symmetric loading.

Figure 5-4. Loading Conditions Corresponding to Symmetric and Anti-Symmetric In-Plane Buckling Modes.

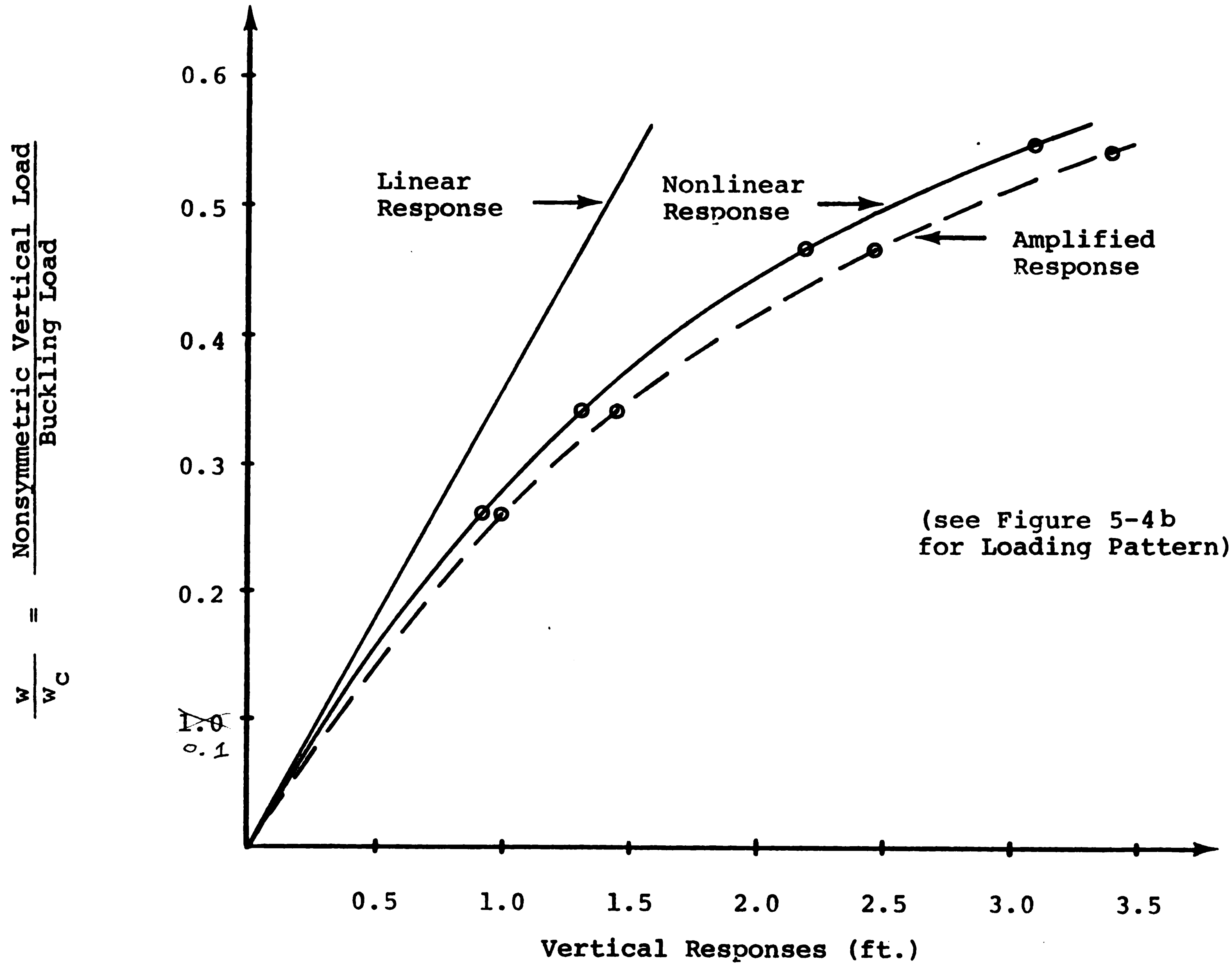


Figure 5-5. Comparison of Vertical Responses for Nonuniform Loading.

REFERENCES

1. Ojalvo, M., and Newman, M., "Buckling of Naturally Curved and Twisted Beams," Journal of the Engineering Mechanics Division, ASCE, Vol. 94, EMS, 1968, pp. 1067-1087.

2. Ojalvo, M., Demuts, E., and Tokarz, F. J., "Out-of-plane Buckling of Curved Members," Journal of the Structural Division, ASCE, Vol. 95, ST10, 1969, pp. 2305-2316.

3. Wen, R. K., and Lange, J., "Curved Beam Element for Arch Buckling Analysis," Journal of the Structural Division, ASCE, Vol. 107, ST11, 1981, pp. 2053-2069.

4. Tokarz, F. J., "Experimental Study of Lateral Buckling of Arches," Journal of the Structural Division, ASCE, Vol. 97, ST2, 1971, pp. 545-559.

5. Tokarz, F. J., and Sandhu, R. S., "Lateral Torsional Buckling of Parabolic Arches," Journal of the Structural Division, ASCE, Vol. 98, ST5, 1972, pp. 1161-1179.

6. Donald, P. T. A., and Godden, W. G., "A Numerical Solution to the Curved Beam Problem," The Structural Engineer, Vol. 41, No. 6, 1963.

7. Godden, W. G., "The Lateral Buckling of Tied Arch," Proceedings of the Institution of Civil Engineers, Vol. 93, No. 3, 1954, pp. 496-514.

8. Godden, W. G., and Thompson, J. G., "An Experimental Study of a Model Tied-Arch Bridge," Proceedings of the Institution of Civil Engineers, Vol. 14, 1959, pp. 383-394.

9. Shukla, S. N., and Ojalvo, M., "Lateral Buckling of Parabolic Arches with Tilting Loads," Journal of the Structural Division, ASCE, Vol. 97, ST6, 1971, pp. 1763-1773.

10. Johnston, B. G. (editor), Guide to Stability Design Criteria for Metal Structures, Structural Stability Research Council, Third Edition, John Wiley and Sons, N.Y., 1976.

11. Nettleton, D. A., and Torkelson, J. S., "Arch Bridges," Bridge Division, Office of Engineering, Federal Highway Administration, U.S. Department of Transportation, Washington, D.C.

12. Bleich, F., "Buckling Strength of Metal Strucutres," McGraw-Hill Book Co., New York, 1955.

13. Ostlund, S., "Lateral Stability of Bridge Arches Braced with Transverse Bars," Transactions of Royal Institute of Technology, Stockholm, Sweden, No. 84, 1954.

14. Wastlund, G., "Stability Problems of Compressed Steel Members and Arch Bridges," Journal of the Structural Division, ASCE, Vol. 86, ST6, 1960, pp. 47-71.

15. Almeida, N. F., "Lateral Buckling of Twin Arch Ribs with Transverse Bars," Ph.D. Dissertation, Ohio State University, 1970.

16. Sakimoto, T., and Namita, Y., "Out-of-plane Buckling of Solid Rib Arches Braced with Transverse Bars," Proceedings of the Japan Society of Civil Engineers, No. 191, 1971, pp. 109-116.

17. Sakimoto, T., and Komatsu, S., "Ultimate Strength of Arches with Bracing Systems," Journal of the Structural Division, ASCE, ST5, May 1982, pp. 1064-1076.

18. Sakimoto, T., and Komatsu, S., "Ultimate Strength Formula for Steel Arches," Journal of the Structural Division, ASCE, Mar. 1983, pp. 613-627.

19. Yabuki, T., and Vinnakota, S., "Stability of Steel Arch Bridges in State-of-the-Art Report," Unpublished Paper.

20. Merritt, F. S., "Structural Steel Designers' Handbook," Chapter 13, McGraw-Hill Book Co., New York, 1972.

21. Lange, J. S., "Elastic Buckling of Arches by Finite Element Method," Ph.D. Thesis, Michigan State University, 1980.

22. Rahimzadeh-Hanachi, J., "Nonlinear Elastic Frame Analysis by Finite Element," Ph.D. Thesis, Michigan State University, 1981.

23. Wen, R. K., and Rahimzadeh-Hanchi, J., "Nonlinear Elastic Frame Analysis by Finite Element," Journal of the Structural Division, ASCE, August, 1983.

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

25. Mallet, R. H., and Marcal, P. V., "Finite Element Analysis of Nonlinear Structures," Journal of the Structural Division, ASCE, Vol. 94, No. ST9, Proc. Paper 6115, Sept. 1968, pp. 2081-2105.

26. Cook, R. D., "Concepts and Applications of Finite Element Analysis," John Wiley and Sons, Inc., New York, N.Y., 1974.

27. Timoshenko, S. P., and Gere, J. M., "Theory of Elastic Stability," McGraw-Hill Book Co., New York, N.Y., 1961.

28. Gere, J. M., and Weaver, W. J., "Analysis of Framed Structures," D. Van Nostrand Co., 1965.

APPENDICES

APPENDIX A

LINEAR AND FIRST ORDER NONLINEAR STIFFNESS MATRICES OF A TRUSS ELEMENT

For the linear and first order nonlinear stiffness matrices of a truss element, the twelve degrees of freedom defined previously for the beam element (see Figures 2-5) are used here also. Note that only non-zero elements are given below.

Linear Stiffness Matrix

$$k(1,1) = k(7,7) = \frac{EA}{\ell}.$$

$$k(7,1) = k(1,7) = \frac{-EA}{\ell}.$$

First Order Nonlinear Stiffness Matrix

$$n_1\ (2,2) = n_1\ (3,3) = n_1\ (8,8) = n_1\ (9,9) = V$$

$$n_1\ (2,8) = n_1\ (8,2) = N_1\ (3,9) = n_1\ (9,3) = -V$$

in which

$$V = \frac{EA}{\ell^2}\ (u_2' - u_1')$$

and u_2', u_1' are axial displacements of the ends in local coordinates.

APPENDIX B

PROGRAM NEAMAH

B.1 Description of Subroutines.

Program NEAMAH, has been described in Chapter III, and a listing of the program is given at the end of the section. The input data is explained with enough comment statements as they enter into the program. Other comment statements are used as needed. In the following, a brief description of the subroutines is given.

The main program (EIGEQDK) directs the flow of execution by calling the appropriate subroutines for each step of the solution procedure. Subroutine NODDATA reads the data of the structure geometry which includes mainly the coordinates and degrees of freedom of the nodes. Subroutine ELEMENT calls the appropriate subroutine (BEAM or TRUSS) to read the elements properties data. Subroutine BAND computes the semibandwidth, MBAND, of the structural stiffness matrix.

Subroutine BEAM and TRUSS evaluate the linear stiffness of the beam and truss elements, respectively. Subroutine TRANSFORM and INVTRNS are used for geometric transformation from local coordinates to global coordinates and vice versa. Subroutine SBEAM1, SBEAM2, and KEPSIO1, respectively, evaluate the non-zero entries of $[n_1]$, $[n_2]$, and $[K_{\epsilon o}]$. The assembly of $[k]$, $[n_1]$, $[n_2]$, and $[K_{\epsilon o}]$ into the appropriate global stiffness is accomplished with subroutine ASEMBLE. Subroutine LINSOLN solves the system linear equation by Gauss elimination. Subroutine STCONDN condenses the structural linear stiffness matrix and load vector into the degrees of freedom which have been established in subroutine NODDATA. Subroutine RECOVER recovers the internal degrees of freedom of the structure after using subroutine LINSOLN. Subroutine IDENT identifies the displacements obtained from LINSOLN with the nodal displacements similar to those found in the recovery process.

Subrutine DECK was developed for the inclusion of the rigid deck effect or the "tilted load effect." Subroutine LINDECK and NONDECK included approximate deck effect "tilted load effect," assuming that the deck is flexible. The approximate effect of a tower was included in subroutine TOWER.

The linear eigenvalue and the multi-eigenvalue solutions are obtained using program EIGENVL. The nonlinear eigenvalue solution is obtained using NLEIGNP. For the solution of the quadratic problem, subroutine NLEIGNP uses the modified regula falsi method of iteration by calling subroutine MRGFLS and the function subprogram DET. Function subprogram DET1 evaluates the determinant of the structural tangent stiffness matrix. Subroutines ENDFORC evaluates the element end forces.

B.2 Variables Used in the Computer Program.

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

Program NEAMAH.

A(M) = The cross-sectional area of element M;

A701D(M), A7TOT(M) = Parameters related to element M for evaluation of the initial strain stiffness matrix;

BOL(M,J), BTO(M,J), BE(J) = Intermediate parameters for the evaluation of initial strain stiffness matrix;

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

DTOT(I), DACTUAL(I) = The same as D(I) but for total displacement measured with reference to the beginning of each load increment or initial geometry, respectively;

DETER, DETERMNT = Determinant of the structural secant or tangent stiffness matrices;

DETCHK = Constant always zero, so the program will stop if the detrminant increases as an indication of instability;

DN(I,1) = End forces in global coordinates for each element. I varies from 1 to 6;

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

ES(I,M) = End forces in local coordinates for element M. I varies from 1 to 3;

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

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 of the D.O.F. if initially IB(N,I) = -1);

ICAL1, ICAL2, ICAL3 = Variables controlling print-out (more details are indicated by "comment statement" in the listing of programs);

ICAL4, ICAL5, ICAL6 = Similar to above, use for approximate in-plane and out-of-plane deck and tower effect only.

ICHECK = Parameter used for Newton-Raphson approach in Lagrangian coordinates to control the type of computation needed in each load increment;

IDET = Parameter used for evaluation of the determinant of the secant or tangent stiffness matrices either before or after Gauss elimination process;

IGOPTIN = Paramether used to specify type of the geometry for plane frames (i.e., circular, parabolic arch or arbitrary geometry);

IPAR = Variable identifying appropriate "Tape" for storage of different structural stiffness matrices (i.e., [K], [K_o], [N_1], [N_2]);

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

IXX(M) = Moment of inertia about the x-axis of the cross section of element M;

IZZ(M) = Moment of inertia about the z-axis of the cross section of element M;

KT(M) = Torsion constant of element M;

L(N,K) = Variable identifying the Kth element in the element group N;

LE(M) = Length of element M;

LODPON1 = The degree of freedom at which the load to be increased (automatically generated by the program with proper use of LOADDIR. If the load to be incremented is not at the first node, then LODPON1 is to be inputted manualy by changing ITETO value to 0);

LNODE1 = The node at which the load to be incremented, see previous note;

LDOF1 = The local degree of freedom of the incremented load at the previously indicated node;

MBAND = Semibandwidth of structural stiffness matrix;

NCOND = Total number of degrees of freedom to be condensed out;

NCOUNT = The order of load increment in incremental approaches;

NE = Total number of elements in the structure;

NEQ = Total number of equations;

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 = Total number of degrees of freedom, condensed and free, of the system. (NSIZE = NEQ + NCOND);

NUMEG = Total number of element groups;

NUMEL(I) = Total number of elements in element group I (in NEAMAH);

NUMITER = Number of iterations at each stage of computation;

NUMNP = Total number of nodal points;

PINT(N,I) = Initial load applied at node N, in the Ith direction.

PINC(N,I) = Load increment at node N, in the Ith direction.

PTOT(N,I) = Total load at node N, in the Ith direction.

PACTUAL(I) = Applied load related to the Ith D.O.F. in the structural load vector at each stage;

PSAVE(I) = Initial reference load in the Ith direction, initially equal to zero;

PSTART(I) = Initial load in the Ith direction;

PTEMP(I) = Resistance in the Ith direction, with reference to current updated state;

R(I) = Load vector of the system;

ROT(I,J), ROTRAN(I,J) = Rotation and inverse rotation matrix for each element (I = 1, 6, J = 1, 6), respectively;

S(I,J) = Tangent stiffness matrix of the system;

SCALE = Scale factor in the evaluation of the determinant of the structural stiffness matrix;

SE(I,J), SEI(I,J), SE2(I,J) = Element stiffness matrices (i.e., [k], [n_1], [n_2], respectively);

ULOC(M,I) = Identifies local displacement in the Ith direciton of element M (I varies from 1 to 12 for three dimensional case and from 1 to 6 for two dimensional);

W(I,J), WCHK(I,J) = Incremental recovered displacements (used In iterative process) related to node I in the Jth direction;

WTOT(I,J) = The same as W(I,J) but for total displacements;

X(N), Y(N), Z(N) = Global X, Y, Z-coordinates of node N;

ZPGM(M) = Rotation of local major principal axis.

SUBROUTINE TRANSFM

Rcol(I) = Identifies the entries of rotation matrix for three dimensional beam element. I varies from 1 to 9;

SUBROUTINE INVTRNS

V(NP,I) = Identifies the element local displacements for nodal point NP and Ith direction (I varies from 1 to 6);

SUBROUTINE STCNDN

RC(I) = Condensed structural load vector (I = 1, NEQ);

SC(I,J) = Condensed structure linear tangent stiffness matrix;

SUBROUTINE EIGENVL (EIGEN, IDATA)

EIGEN = Eigenvalue;

EIGNVTR = Eigenvector corresponding to EIGEN;

EPSI = Tolerance;

MAX = Maximum number of iterations allowed;

RHO = Rayleigh quotient;

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

SUBROUTINE ENDFORC

DN(I) = Stress resultants on the nodes of each element;

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 $S = K + L*N_1 + L*L*N_2$ at the converged value of the eigenvalue;

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

L = Converged value of the eigenvalue;

$L = (A + B)/2$;

NTOL = Maximum number of iterations allowed;

XTOL = Tolerance;

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,B, are on the same side of the root, hence method of iteration cannot 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 values 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 $S = K + L*N_1 + L*L*N_2$ at a particular value of L;

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

L = Load parameter;

N_1(I,J) = Part of element S(I,J) corresponding to matrix N_1(I,J);

N_2(I,J) = Part of element S(I,J) corresponding to matrix N_2(I,J).

```
T      PROGRAM NEAMAH (INPUT,OUTPUT=65,TAPE60=INPUT,TAPE61=OUTPUT,
     +TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6,TAPE7,TAPE8,TAPE9,TAPE10,
     +TAPE11,TAPE12,TAPE13,TAPE14,TAPE15,TAPE16,TAPE21)
C
C
***********************************************************************
*                                                                     *
*             PROGRAM NEAMAH                                          *
*         ---------------------                                       *
*                                                                     *
*          THIS PROGRAM WAS PREPARED FOR THE SOLUTION OF LINEAR AND   *
*         NONLINEAR EQUILIBRIUM AND CLASSICAL BUCKLING PROBLEMS OF    *
*         ARCH BRIDGE SYSTEMS USING THE FINITE ELEMENT METHOD WITH    *
*         NONLINEAR ELASTIC STRAIGHT BEAM AND TRUSS ELEMENTS IN THREE *
*         DIMENSIONAL SPACE.                                          *
*                                                                     *
*                                                                     *
*          THE PROGRAM CAN ALSO BE USED FOR SOLUTIONS OF OTHER        *
*         STRUCTURES MADE UP OF THESE ELEMENTS IN THREE DIMENSIONAL   *
*         SPACE.                                                      *
*                                                                     *
***********************************************************************
C
      REAL IXX,IYY,IZZ,KT,II,JJ,LE,N1STTOT

      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
      ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54),
     +L(3,54),IZZ(54),ZPGM(54)
      COMMON/8/PINT(30,6),R(168)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      COMMON/10/D(168),G1(168),G2(168),G3(168),G4(168,10),RC(168),
      SC(168,36),IGAUS
      COMMON/11/DN(12),W(30,6),V(30,6)
      COMMON/12/ULOC(54,12),U(12),RCOL(9),MSUOPTN,N1GOPTN
      COMMON/16/PRIOPTN
      COMMON/17/A7TOT(54),A7OLD(54),BOL(54,5),BTO(54,5),BE(5)
      COMMON/SHEAR/ ISHEAR

      DIMENSION  DTEMP(168),PTEMP(168),PSTART(168),DTOT(168)
      DIMENSION  PACTUAL(168),STEMP(168,36)
      DIMENSION  PSAVE(168),DACTUAL(168),N1STTOT(168,36),PINC(30,6)
      DIMENSION  SOLD(168,36),SRK(168,36),SRN1(168,36),PTOT(30,6)
      DIMENSION  REFSTRT(30,6),REFPTMP(30,6),SRN2(168,36)

      INTEGER  PROTYPE,EIGVALU,PRIOPTN,DETOPTN
      INTEGER  DOF,TLDOF
***********************************************************************
*               DATA FILES                                            *
*             ---------------                                         *
*         TAPES 8,7,4   FOR K BEAM ELEMENT                            *
*         TAPES 10,11,4   FOR K TRUSS ELEMENT                         *
*         TAPE  9   FOR (K+N1) STRUCTURAL                             *
*         TAPE  13 FOR (-K)                                           *
*         TAPES 1,2,6 FOR N1                                          *
*         TAPES 3,5,12  FOR N2                                        *
*         TAPES 14,15,16  FOR KEPSIO                                  *
*                                                                     *
***********************************************************************


***********************************************************************
*                       RECORD NO. ONE                                *
*                     ------------------                              *
*         TITLE1,2,3: PRGRAM TITLE                                    *
*         NE        :NUMBER OF ELEMENTS                               *
*         NUMNP: NUMBER OF NODAL POINTS                               *
*         IDATA: 0 TO CHECK THE DATA INPUT ,1 PROGRAM EXECUTION       *
*         ICAL1:0 FOR LOAD VECTOR AND STRUCTURAL LINEAR STIFFNESS     *
*               MATRIX TO BE PRINTED(ICAL1=1 SKIP)                    *
*         ICAL2:0 FOR DISPLACEMENT VECTOR TO BE PRINTED               *
*               (ICAL2=1 SKIP)                                        *
*         ICAL3:0 FOR LINEAR STIFFNESS MATRIX IN LOCAL OR GLOBAL      *
*               COORDINATE TO BE PRINTED,ALSO FOR DETAILS OF EIGENVALUE *
*               SOLUTION(ICAL3=1 SKIP)                                *
*         ICAL4:1 INTERMEDIAT RESULTS IN SUBROUTINE DECK PRINTED      *
*         ICAL5:1 INTERMEDIAT RESULTS IN SUBROUTINE LINDECK PRINTED   *
*         ICAL6:1 INTERMEDIAT RESULTS IN SUBROUTINE NONDECK PRINTED   *
*                                                                     *
***********************************************************************
```

```
      READ(60,1010) TITLE1,TITLE2,TITLE3,NE,NUMNP,NUMEG,IDATA,ICAL1,
     + ICAL2,ICAL3,ICAL4,ICAL5,ICAL6

*****************************************************************************
*                            RECORD NO. TWO                                 *
*                            --------------                                 *
*        ENTEIS IN THIS RECORD IS 1 FOR AFFERMATIVE RESPONSE                *
*              ELSE INPUT 0 UNLESS OTHERWISE INSTUCTED.                     *
*        ----------------------------------------------------------         *
*        PRIOPTN:ITERMEDIAT CALCULATION TO BE PRINTED.                      *
*        N2OPTIN:FOR N2 TO BE USED                                          *
*        N1OPTIN:FOR N1 TO BE USED                                          *
*        ITERCHK:FOR ITERATION APPROACH.                                    *
*        MSUOPTN:1 CONSTANT AVERAGE STRAIN FOR EACH ELEMENT.                *
*        MSUOPTN:2 STRAIN IS QUADRATIC FUNCTION OF SLOPE AT EACH            *
*                 ELEMENT.                                                  *
*        N1GOPTN:0 FOR LINEAR EIGENVALUE SOLUTION N1 IS USED                *
*               :1 FOR LINEAR EIGENVALUE SOLUTION N1STAR IS USED            *
*        IFIX   :0 FOR UPDATED LAGRANGIAN APPROCH.                          *
*               :1 FOR FIXED LAGRANGIAN APPROCH.                            *
*        JUSTK  :1 ONLY UPDATED LINEAR STIFFNESS MATRIX USED ELSE=0         *
*        TOLER  :TOLERANCE FOR SUCCESSIVE ITERATION CONVERGENCE ONLY        *
*        DETOPTN:0 NO CONTROL ON THE DETERMINANT OF THE T.STIFFNESS M.      *
*        DETOPTN:1 EXECUTION WOULD BE TERMINATED IF THE DETERMINANT IS      *
*                 EQUAL TO ZERO OR THE VALUE OF THE DET. INCREASED.         *
*****************************************************************************

      READ(60,6971)PRIOPTN,N2OPTIN,N1OPTIN,ITERCHK,
     +MSUOPTN,N1GOPTN,IFIX,JUSTK,TOLER,DETOPTN
 6971 FORMAT(8I5,F10.4,I5)

*****************************************************************************
*                            RECORD NO. THREE                               *
*                            ----------------                               *
*        DELTA1=ALLOWABLE TOLERANCE FOR FORCE COMPONENTS OF                 *
*        UNBALANCED FORCE VECTOR                                            *
*        DELTA2=ALLOWABLE TOLERANCE FOR MOMENT COMPONENTS OF                *
*        UNBALANCED FORCE VECTOR                                            *
*****************************************************************************

      IF(ITERCHK.EQ.1)READ(60,6948)  DELTA1,DELTA2
 6948 FORMAT(2E21.15)

*****************************************************************************
*                            RECORD NO. FOUR                                *
*                            ---------------                                *
*                                                                           *
*        PROTYPE:1 IS FOR INCREMENTAL LOADING IN FIXED COORDINATES          *
*                 SECANT STIFFNESS APPROACH(SUCCESIVE ITERATIONS)           *
*        PROTYPE:2 IS  FOR EIGENVALUE PROBLEM                               *
*        PROTYPE:3 FOR INCREMENTAL LOADING IN UPDATED-COORDINATES           *
*        PROTYPE:3 FOR NEWTON-RAPHSON FIXED COORDINATE APPROACH             *
*        PROTYPE:3 ,ITERCHK:0 & JUSTK:1 FOR THE UPDATED-COORDINATE          *
*                 APPROACH BY UPDATING ONLY LINEAR STIFFNESS MATRIX         *
*                 WITH NO ITERATION.                                        *
*        EIGVALU:1 IS FOR LINEAR EIGENVALUE PROBLEM                         *
*        EIGVALU:2 IS FOR QUADRATIC EIGENVALUE PROBLEM                      *
*        EIGVALU:3 IS FOR INCREMENTAL LOADING IN FIXED COORDINATES          *
*        EIGVALU:4 FOR INCREMENTAL LOADING IN UPDATED-COORDINATES(OR        *
*                 FIXED COORDINATES)                                        *
*        ISTRESS:1 ELEMENT END FORCES SHOULD BE EVALUATED.                  *
*        ISTRESS:0 ELEMENT END FORCES SHOULDN,T BE EVALUATED                *
*                 (EFFICIENT CODING)                                        *
*        IPART  :1 IF CONV. OF DISPLACEMENTS  AND INTERMEDIAT VALUES        *
*                 OF DISPLACEMENTS IS TO BE PRINTED ELSE IPART=0            *
*                 (EQUIL.SOL.ONLY)                                          *
*        LOADDIR:-1 IF LOAD IS TO BE ICREMENTED IN THE X DIR.               *
*        LOADDIR:0  IF LOAD IS TO BE INCREMENTED IN THE Y DIR.              *
*        LOADDIR:1  IF LOAD IS TO BE INCREMENTED IN THE Z DIR.              *
*        IDECK  :0 NO DECK INCLUDED.                                        *
*        IDECK  :1 RIGED DECK.                                              *
*        IDECK  :2 INPLANE EFFECT OF DECK.                                  *
*        IDECK  :3 OUT OF PLANE EFFECT OF DECK.                             *
*        IDECK  :4 EFFECT OF DECK AND TOWER.                                *
*        IDECK  :5 COMBINED OUT-OFF AND IN PLANE PLANE EFFECT OF DECK.      *
*        ISHEAR :1 SHEAR EFFECT IS INCLUDED ELSE 0                          *
*                                                                           *
*****************************************************************************
```

```
      READ(60,1) PROTYPE,EIGVALU,ISTRESS,IPART,LOADDIR,IDECK,ISHEAR

      WRITE(61,2010)TITLE1,TITLE2,TITLE3,NE,NUMNP,NUMEG,IDATA,ICAL1,
     + ICAL2,ICAL3,ICAL4,ICAL5,ICAL6

      WRITE(61,6972)PRIOPTN,N2OPTIN,N1OPTIN,ITERCHK,
     +MSUOPTN,N1GOPTN,IFIX,JUSTK,DETOPTN,TOLER

6972  FORMAT(* *,//10X,8HPRIOPTN=,I2/,10X,8HN2OPTIN=,I2/,
     +10X,*N1OPTIN=*,I2/10X,*ITERCHK=*,I2/,10X,*MSUOPTN=*,I2/,10X,
     +*N1GOPTN=*,I2/,10X,*IFIX=*,I2/,10X,*JUSTK=*,I2/,
     +10X,*DETOPTN=* I2/,10X,*TOLER=*,F10.5)
C
      IF(ITERCHK.EQ.1) WRITE(61,6931) DELTA1,DELTA2
6931  FORMAT(10X,6HEPSI1=,E21.15/10X,6HEPSI2=,E21.15,/)
C
      WRITE(61,8765) PROTYPE,EIGVALU,ISTRESS,IPART,LOADDIR,IDECK
8765  FORMAT(///12X,*PROTYPE=*,I2/,12X,*EIGVALU=*,I2/,
     +12X,*ISTRESS=*,I2/,12X,*IPART   =*,I2/,12X,*LOADDIR=*,I2/,
     +12X,*IDECK   =*,I2/,12X,*ISHEAR =*,I2/)


*************************************************************************
*                                RECORD FIVE                            *
*                                -------------------                    *
*         IGOPTIN:1 FOR CIRCULAR ARCH,IGOPTIN=2 FOR PARABOLIC           *
*                 AND IGOPTIN=0 FOR OTHER GEOMETRIES                    *
*                                                                       *
*          "NOTE THIS PROGRAM NOW ONLY WORKS WITH IGOPTIN=0."           *
*************************************************************************

      READ(60,10)       IGOPTIN
      WRITE(61,5287)    IGOPTIN
5287  FORMAT(//,10X,*IGOPTIN=*,I2,/)
C
*************************************************************************
*                      RECORD SIX       (  LOADINGS  )                  *
*                      ---------------------------------                *
*     MAXITER          MAX. NUMBER OF ITERATIONS FOR EACH LOAD INCREMENT. *
*     TLDOF            NUMBER OF TOTAL LOADS APPLIED ON THE STRUCTURE.  *
*     ..................................................................*
*     PINT(N,I)        INTIAL LOADING                                   *
*     PINC(N,I)        INCREMENTAL LOADING                              *
*     PTOT(N,I)        TOTAL LOADING                                    *
*                                                                       *
*     N NODE NUMBER     FROM 1 TO NUMNP,                                *
*     I   D.O.F.    1 TO 6 (1,2 AND 3  FOR X,Y AND Z CONCENTRATED FORCES *
*              (4,5 AND 6  FOR X,Y AND Z MOMENTS                        *
*                                                                       *
*************************************************************************
C
C
      READ(60,400)MAXITER,TLDOF
      WRITE(61,4801) MAXITER,TLDOF
4801  FORMAT(* *,10X,*MAXITER =*,I2,10X,*NUMBER OF LOADS =*,I2)
      WRITE(61,409)
      DO 407 N=1,NUMNP
      DO 407 I=1,6
407   PINT(N,I)=PINC(N,I)=PTOT(N,I)=0.0
      I=0
406   CONTINUE
      I=I+1
      READ(60,405)N,DOF,PINT(N,DOF),PINC(N,DOF),PTOT(N,DOF)
      WRITE(61,410)N,DOF,PINT(N,DOF),PINC(N,DOF),PTOT(N,DOF)
      IF(I.LT.TLDOF) GO TO 406
400   FORMAT(2I5)
401   FORMAT(* *,////5X,*MAXITER =*,I5)
405   FORMAT(2I5,3F10.4)
409   FORMAT(*1*,10X,* LOADING CONDITIONS : *//,6X,*NODE*,7X,*DOF*,16X,
     +*PINT*,16X,*PINC*,16X,*PTOT*//)
410   FORMAT(*0*,3X,I5,4X,I5,11X,F10.4,10X,F10.4,10X,F10.4)
C
*************************************************************************
*         READ NODAL POINT DATA                                         *
*************************************************************************
```

```
C
      CALL NODDATA(IGOPTIN)
      IF(PROTYPE.NE.3) GO TO 3021
C
      SCALE=1.E+10
      IF(LOADDIR.EQ.-1)SCALE=10000.0
      DETCHK=0.0
      DO 5001 I=1,NEQ
5001  PSAVE(I)=DACTUAL(I)=DTOT(I)=0.0
*      AUTOMATIC GENERATION OF LODPON1 AND CORRESPONDING D.O.F.     *
*******************************************************************
      IF(LOADDIR)1655,1665,1675
1655  IHORZ=1
      IVERT=0
      ILAT=0
      GO TO 1685
1665  IHORZ=0
      IVERT=1
      ILAT=0
      GO TO 1685
1675  IHORZ=0
      IVERT=0
      ILAT=1
1685  CONTINUE
*      WRITE(61,1686)IHORZ,IVERT,ILAT
* 1686  FORMAT(* *,10X,*IHORZ=*,I3,5X,*IVERT=*,I3,5X,*ILAT=*,I3//)
*
*       IF LOAD WANTED FOR SPESIFIC LOAD LET ITETO=1
*
*       CHOOSE THE APPROPRIATE VALUES OF LODPON1,LNODE1,AND LDOF1
*
      ITETO=0
      IF(ITETO.EQ.0) GO TO 9152
      LODPON1=20
      LNODE1=8
      LDOF1=2
      IF(ITETO.NE.0)GO TO 700
9152  DO 200 N=1,NUMNP
      DO 300 I=1,6
      IF(IA(N,I).EQ.0) GO TO 300
      IF(PINT(N,I).EQ.0)GO TO 300
      IF(I.EQ.1.AND.IHORZ.EQ.0)GO TO 300
      IF(IHORZ.EQ.0)GO TO 909
      LODPON1=IA(N,I)
      LNODE1=N
      LDOF1=I
      GO TO 700
909   IF(IVERT.EQ.0.AND.I.EQ.2)GO TO 300
      IF(IVERT.EQ.0)GO TO 499
      LODPON1=IA(N,I)
      LNODE1=N
      LDOF1=I
      GO TO 700
499   IF(ILAT.EQ.0.AND.I.EQ.3)GO TO 1093
      IF(ILAT.EQ.0)GO TO 1093
      LODPON1=IA(N,I)
      LNODE1=N
      LDOF1=I
      GO TO 700
1093  PRINT *,"      PROGRAM CAN NOT CALCULATE THE VALUE OF LODPON1"
      PRINT *,"       HELP WANTED, PROGRAM STOPPED AT APP. LINE 266"
      GO TO 900
300   CONTINUE
200   CONTINUE
700   CONTINUE
      WRITE(61,2900)LODPON1,LNODE1,LDOF1
2900  FORMAT(*0*,///11X,*THE D.O.F. IN WHICH LOAD HAS BEEN INCREASED=*,
     +I3,//10X,*AT NODE=*,I3,5X,*WITH D.O.F=*,I3)
      DO 3010 I=1,NUMNP
      DO 3010 J=1,6
3010  W(I,J)=0.0
      ICHECK=1
C
C     UPDATE ORIENTATION OF PRINCIPAL AXES AND NODE COORDINATES
C
1001  IF (ICHECK.EQ.1) GO TO 3021
      IF (IFIX.EQ.1) GO TO 3021
      DO 3020 M=1,NE
      I=NODEI(M)
```

```
      J=NODEJ(M)
      XI=X(I)
      YI=Y(I)
      ZI=Z(I)
      XJ=X(J)
      YJ=Y(J)
      ZJ=Z(J)
      EL=SQRT((XJ-XI)**2+(YJ-YI)**2+(ZJ-ZI)**2)
      CX=(XJ-XI)/EL
      CY=(YJ-YI)/EL
      CZ=(ZJ-ZI)/EL
      DALFA=0.5*((W(I,4)+W(J,4))*CX+(W(I,5)+W(J,5))*CY+(W(I,6)+W(J,6))*
     +CZ)
3020  ZPGM(M)=ZPGM(M)+DALFA
C
      DO 3025 I=1,NUMNP
      X(I)=X(I)+W(I,1)
      Y(I)=Y(I)+W(I,2)
3025  Z(I)=Z(I)+W(I,3)
      IF (PRIOPTN.EQ.0)   GO TO 3021
      WRITE(61,4995)
4995  FORMAT(* *,//,10X,*NODE*,10X,*X(I)*,10X,*Y(I)*,10X,*Z(I)*,/)
      DO 4996 I=1,NUMNP
      WRITE (61,4997) I,X(I),Y(I),Z(I)
4997  FORMAT(* *,/,10X,I5,3F15.8)
4996  CONTINUE
      WRITE(61,4998)
4998  FORMAT(* *,/,10X,*MEMBER*,10X,*ALPHA*,/)
      DO 4999 I=1,NE
      WRITE(61,5998)I,ZPGM(I)
5998  FORMAT(* *,/,10X,I5,F15.8)
4999  CONTINUE
3021  CONTINUE

*****************************************************************
*                 READ AND STORE ELEMENT DATA                   *
*                 ---------------------------                   *
*     ALL ELEMENT PROPERTIES WILL BE READ AT THIS STAGE AS FOLLOWS:  *
*          1.BEAM ELEMENT      2.TRUSS ELEMENT                  *
*       IF ONLY TRUSS ELEMENTS ARE USED LET NUMEL(N)=0 FOR BEAM ELEM. *
*                                                               *
*****************************************************************

      IPAR=NUMITER=1
      N=0
5927  N=N+1
      IF(PROTYPE.NE.3)  GO TO 2222
      IF(ICHECK.NE.1)   GO TO 4444
2222  READ(60,1020) NUMEL(N)
4444  IF(NUMEL(N).EQ.0)  GO TO 5927
      IF(PROTYPE.NE.3)  ICHECK=0
      CALL ELEMENT (N,ICHECK,PROTYPE)
100   CONTINUE
      IF(N.EQ.NUMEG)  GO TO 5928
      GO TO 5927
5928  CONTINUE
      IF(IDATA.EQ.1) GO TO 900
C
      IF(PROTYPE.NE.3) GO TO 3335
      IF(ICHECK.NE.1)  GO TO 3333
3335  CONTINUE
C        COMPUTE SEMIBANDWIDTH OF STRUCTURE STIFFNESS MATRIX
C     *****************************************************************
      DO 3916 N=1,NUMEG
      IF(NUMEL(N).EQ.0) GO TO 3916
      CALL BAND(N)
3916  CONTINUE
      WRITE(61,3917)MBAND
3917  FORMAT(*1*,///10X,*SEMIBANDWIDTH =*,I3)
      IF(JUSTK.EQ.1)    GO TO 2110
      IF(PROTYPE.NE.3)  GO TO 2110
      DO 5745  NN=1,NUMEG
      IF(NUMEL(NN).EQ.0)    GO TO 5745
      NAME=NUMEL(NN)
      DO 5341  K=1,NAME
      M=L(NN,K)
      IF(MSUOPTN.EQ.1)  A7OLD(M)=0.0
      DO 5341 I=1,5
      IF(MSUOPTN.EQ.2)  BOL(M,I)=0.0
5341  CONTINUE
```

```
5745  CONTINUE
2110  CONTINUE
C
C
C
C        ASSEMBLE INITIAL LOADS AND NODAL LOADS INTO LOAD VECTOR
C        SET ARRAYS -S- AND -R- EQUAL TO ZERO
C     *****************************************************************
C
C
      CALL ASEMBLE (M)
3333  CONTINUE
      IF(PROTYPE.NE.3)  GO TO 3336
      IF(ICHECK.EQ.1)   GO TO 2111
      GO TO 2112
2111  DO 5003  I=1,NEQ
5003  PSTART(I)=0.0
      J=0
      DO 415 N=1,NUMNP
      DO 420 I=1,6
      IF(IA(N,I).EQ.0) GO TO 420
      J=J+1
      PSTART(J)=PINT(N,I)
420   CONTINUE
415   CONTINUE
      IF(PRIOPTN.EQ.0) GO TO 423
      WRITE(61,421)
421   FORMAT(*0*,10X,*PSTART :*)
      WRITE(61,422) (PSTART(I),I=1,NEQ)
422   FORMAT(* *,8X,F10.5)
423   CONTINUE
C
2112  IF(JUSTK.EQ.1)  ICHECK=2
      IF(JUSTK.EQ.1)  GO TO 3336
      IF(ICHECK.EQ.1) GO TO 3336
      IPAR=2
      DO 2113  I=1,NSIZE
      DO 2113  J=1,MBAND
2113  S(I,J)=0.0
      CALL ELEMENT(N,ICHECK,PROTYPE)
1901  IF(ITERCHK.NE.0) CALL INVTRNS
      IF(ICHECK.EQ.3)  GO TO 4988
      IF(N1OPTIN.EQ.0) GO TO 4987
      DO 1801 I=1,NSIZE
      DO 1801 J=1,MBAND
1801  S(I,J)=0.0
      IPAR=3
      DO 2101 N=1,NUMEG
      IF(NUMEL(N).EQ.0) GO TO 2101
      CALL SBEAME1(N,PROTYPE)
2101  CONTINUE
      REWIND 6
4987  CONTINUE
      IF(ICHECK.EQ.3)  GO TO 4988
      IF(N2OPTIN.EQ.0) GO TO 4988
      DO 7691 I=1,NSIZE
      DO 7691 J=1,MBAND
7691  S(I,J)=0.0
      IPAR=4
      DO 7692 N=1,NUMEG
      IF(NUMEL(N).EQ.0) GO TO 7692
      CALL SBEAME2(N)
7692  CONTINUE
4988  CONTINUE
      IF(ICHECK.EQ.2.OR.PSAVE(LODPON1).EQ.0.) GO TO 5010
      DO 9436 I=1,NSIZE
      DO 9436 J=1,MBAND
9436  S(I,J)=0.0
      IPAR=7
      DO  9437 N=1,NUMEG
      IF(NUMEL(N).EQ.0) GO TO 9437
      CALL KEPSI01(N)
9437  CONTINUE
      IF(ICHECK.EQ.2)   GO TO 5010
      REWIND 4
      DO 3071 I=1,NEQ
      DO 3081 J=1,MBAND
      READ(4,11) RK
      SRK(I,J)=RK
      READ(16,11) RN1STAR
      SRN1(I,J)=RN1STAR
```

```
      N1STTOT(I,J)=RN1STAR
      IF(IFIX.EQ.1)  S(I,J)=SOLD(I,J)=RK
      IF(IFIX.EQ.0)  S(I,J)=SOLD(I,J)=RK+N1STTOT(I,J)
3081  CONTINUE
3071  CONTINUE
      REWIND 4
      REWIND 16
      IF(PRIOPTN.EQ.0) GO TO 7233
      WRITE(61,8005)
8005  FORMAT(///,10X,*KEPSI0 MATRIX*,/)
      WRITE(61,8002)  ((SRN1(I,J),J=1,MBAND),I=1,NEQ)
      WRITE(61,8008)
8008  FORMAT(///,10X,*K LINEAR STIFFNESS MATRIX*,/)
      WRITE(61,8002)  ((SRK(I,J),J=1,MBAND),I=1,NEQ)
      WRITE(61,8009)
8009  FORMAT(///,10X,*S(I,J)    MATRIX*,/)
      WRITE(61,8002)  ((S(I,J),J=1,MBAND),I=1,NEQ)
7233  CONTINUE
      GO TO 5011
5010  DO 4071 I=1,NEQ
      DO 4081 J=1,MBAND
      IF(N1OPTIN.EQ.1)READ(6,11) RN1
      IF(PSAVE(LODPON1).EQ.0.)    READ(4,11)  RK
      IF(N2OPTIN.EQ.1) READ(12,11) RN2
      IF(N1OPTIN.EQ.1)SRN1(I,J)=RN1
      IF(PSAVE(LODPON1).EQ.0..AND.N2OPTIN.EQ.1)SP(I,J)=RK+.5*RN1+RN2/3.
      IF(PSAVE(LODPON1).NE.0..AND.N2OPTIN.EQ.1)
     +SP(I,J)=SOLD(I,J)+.5*RN1+RN2/3.
      IF(PSAVE(LODPON1).EQ.0..AND.N1OPTIN.EQ.0)SP(I,J)=S(I,J)=RK
      IF(PSAVE(LODPON1).EQ.0..AND.N1OPTIN.EQ.1.AND.N2OPTIN.EQ.0)SP(I,J)=
     +RK+.5*RN1
      IF(PSAVE(LODPON1).NE.0..AND.N1OPTIN.EQ.0)SP(I,J)=S(I,J)=SOLD(I,J)
      IF(PSAVE(LODPON1).NE.0..AND.N1OPTIN.EQ.1.AND.N2OPTIN.EQ.0)SP(I,J)=
     +SOLD(I,J)+.5*RN1
      IF(PSAVE(LODPON1).EQ.0..AND.N2OPTIN.EQ.1)S(I,J)=RK+RN1+RN2
      IF(PSAVE(LODPON1).NE.0..AND.N2OPTIN.EQ.1)S(I,J)=SOLD(I,J)+RN1+RN2
      IF(PSAVE(LODPON1).EQ.0..AND.N1OPTIN.EQ.1.AND.N2OPTIN.EQ.0)
     +S(I,J)=RK+RN1
      IF(PSAVE(LODPON1).NE.0..AND.N1OPTIN.EQ.1.AND.N2OPTIN.EQ.0)
     +S(I,J)=SOLD(I,J)+RN1
4081  CONTINUE
4071  CONTINUE
      IF(N1OPTIN.EQ.1) REWIND 6
      IF(N2OPTIN.EQ.1) REWIND 12
      IF(PSAVE(LODPON1).EQ.0.)   REWIND 4
      IF(PRIOPTN.EQ.0) GO TO 5011
      IF(N2OPTIN.EQ.0) GO TO 4989
      WRITE(61,7693)
7693  FORMAT(///,10X,11HN2    MATRIX,/)
      DO 7694 I=1,NEQ
      DO 7695 J=1,MBAND
      READ(12,11) RN2
      SRN2(I,J)=RN2
7695  CONTINUE
7694  CONTINUE
4989  CONTINUE
      IF(N2OPTIN.EQ.1) REWIND 12
      IF(N2OPTIN.EQ.1) WRITE(61,8002)((SRN2(I,J),J=1,MBAND),I=1,NEQ)
      IF(N1OPTIN.EQ.1) WRITE(61,8004)
8004  FORMAT(//,10X,*N1 NONLINEAR STIFFNESS MATRIX*,/)
      IF(N1OPTIN.EQ.1) WRITE(61,8002)   ((SRN1(I,J),J=1,MBAND),I=1,NEQ)
      IF(PSAVE(LODPON1).NE.0.) WRITE(61,8010)
8010  FORMAT(///,10X,*SOLD(I,J)  MATRIX*,/)
      IF(PSAVE(LODPON1).NE.0.) WRITE(61,8002)
     +((SOLD(I,J),J=1,MBAND),I=1,NEQ)
      WRITE(61,8018)
8018  FORMAT(///,10X,*SP(I,J) MATRIX*,/)
      WRITE(61,8002)  ((SP(I,J),J=1,MBAND),I=1,NEQ)
      WRITE(61,8009)
      WRITE(61,8002)   ((S(I,J),J=1,MBAND),I=1,NEQ)
5011  IF(ICHECK.NE.3)  GO TO 7001
      GO TO 6001
7001  ICHECK=3
      GO TO 3339
3336  CONTINUE
C
C        COMPUTE ELEMENT LINEAR STIFFNESS AND ASSEMBLE INTO STRUCTURE
C        LINEAR STIFFNESS
C     ******************************************************************
C
C
```

```
C
      DO 2114  I=1,NSIZE
      DO 2114  J=1,MBAND
2114  S(I,J)=0.0
      IF(NUMEG.EQ.1)GO TO 5326
      IF(NUMEL(1).EQ.0.)GO TO 5326
        DO 5327 I=1,NSIZE
        DO 5327 J=1,MBAND
5327      STEMP(I,J)=0.0
5326  CONTINUE
      IPAR=2
      DO 110 N=1,NUMEG
      IF(NUMEL(N).EQ.0.) GO TO 110
      CALL ELEMENT (N,ICHECK,PROTYPE)
110   CONTINUE
        REWIND 4
      IF(NUMEG.EQ.1) GO TO 5356
      IF(NUMEL(1).EQ.0.)GO TO 5356
        READ(4,5333)((S(I,J),J=1,MBAND),I=1,NSIZE)
        READ(4,5333)((STEMP(I,J),J=1,MBAND),I=1,NSIZE)
      IF(PRIOPTN.EQ.0) GO TO 3911
*     WRITE(61,3903)
*     DO 3904 I=1,NEQ
*     WRITE(61,3902) I,S(I,1),STEMP(I,1)
*3904  CONTINUE

*       PRINT OUT OF THE BEAM STIFFNESS MATRIX.......*
*****************************************************
      WRITE(61,1301)
      JSTART=1
1300  JEND=JSTART+5
      IF(JEND.GT.MBAND)JEND=MBAND
      WRITE(61,1302)JSTART,JEND
      DO 1310 I2=1,NEQ
      WRITE(61,1303)(S(I2,J),J=JSTART,JEND)
1310  CONTINUE
      IF(JEND.EQ.MBAND)GO TO 1350
      JSTART=JEND+1
      GO TO 1300
1350  CONTINUE
1301  FORMAT(*1*,25X,*PRINT OUT OF THE BEAM STIFFNESS MATRIX*//)
1302  FORMAT(* *,15X,*JSTART=*,I2,5X,*JEND=*,I2)
1303  FORMAT(* *,2X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,
     +E20.13)

*       PRINT OUT OF THE TRUSS STIFFNESS MATRIX.......*
******************************************************
      WRITE(61,1304)
      JSTART=1
1309  JEND=JSTART+5
      IF(JEND.GT.MBAND)JEND=MBAND
      WRITE(61,1305)JSTART,JEND
      DO 1311 I2=1,NEQ
      WRITE(61,1306)(STEMP(I2,J),J=JSTART,JEND)
1311  CONTINUE
      IF(JEND.EQ.MBAND)GO TO 1351
      JSTART=JEND+1
      GO TO 1309
1351  CONTINUE
1304  FORMAT(*1*,25X,*PRINT OUT OF THE TRUSS STIFFNESS MATRIX*//)
1305  FORMAT(* *,15X,*JSTART=*,I2,5X,*JEND=*,I2)
1306  FORMAT(* *,2X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,
     +E20.13)

3911  CONTINUE
*3903  FORMAT(*1*,12X,*I*,17X,*BEAM S.*,23X,*TRUSS S.*,//)
*3902  FORMAT(* *,10X,I3,10X,E21.14,10X,E21.14)
        REWIND 4
      DO 5310 I=1,NSIZE
      DO 5310 J=1,MBAND
5310    S(I,J)=S(I,J)+STEMP(I,J)
      WRITE(4,5333)((S(I,J),J=1,MBAND),I=1,NSIZE)
      REWIND 4
      IF(PRIOPTN.EQ.0) GO TO 3912
*     WRITE(61,3914)
*     DO 3913 I=1,NEQ
*     WRITE(61,3915) I,S(I,1)
*3913  CONTINUE

*       PRINT OUT OF THE TOTAL STIFFNESS MATRIX.......*
*******************************************************
```

```
      WRITE(61,1361)
      JSTART=1
1360  JEND=JSTART+5
      IF(JEND.GT.MBAND)JEND=MBAND
      WRITE(61,1362)JSTART,JEND
      DO 1364 I2=1,NEQ
      WRITE(61,1363)(S(I2,J),J=JSTART,JEND)
1364  CONTINUE
      IF(JEND.EQ.MBAND)GO TO 1365
      JSTART=JEND+1
      GO TO 1360
1365  CONTINUE
1361  FORMAT(*1*,25X,*PRINT OUT OF THE TOTAL STIFFNESS MATRIX*//)
1362  FORMAT(* *,15X,*JSTART=*,I2,5X,*JEND=*,I2)
1363  FORMAT(* *,2X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,
     +E20.13)

3912  CONTINUE
*3914  FORMAT(*1*,12X,*I*,17X,*TOTAL STIFF.*,//)
*3915  FORMAT(* *,10X,I3,10X,E21.14)
5356  CONTINUE
C
      IF(PROTYPE.NE.3)  GO TO 3337
      DO 1071 I=1,NEQ
      DO 1081 J=1,MBAND
      READ(4,11) RK
      S(I,J)=SP(I,J)=RK

1081  CONTINUE
1071  CONTINUE
      REWIND 4
      IF(NCOND.EQ.0) GO TO 1809
      CALL STCONDN
1809  CONTINUE
      IF(PRIOPTN.EQ.0) GO TO 9431
      IF(ICHECK.EQ.1) WRITE(61,8008)
      IF(ICHECK.EQ.1) WRITE(61,8002)((S(I,J),J=1,MBAND),I=1,NEQ)
9431  CONTINUE
6001  IDET=1
8002  FORMAT(1X,6(2X,E19.13),/)
      CALL LINSOLN
      DETRMNT=DET1(SCALE)
      DO 5005 I=1,NEQ
      DTOT(I)=DTOT(I)+D(I)
      DACTUAL(I)=DACTUAL(I)+D(I)
      IF(IFIX.EQ.0)  D(I)=DTOT(I)
      IF(IFIX.EQ.1)  D(I)=DACTUAL(I)
5005  CONTINUE
      IF(NCOND.NE.0)  CALL RECOVER
      CALL IDENT
      IF(ITERCHK.EQ.0) CALL INVTRNS
      IF(JUSTK.EQ.1)  GO TO 3339
      IF(ITERCHK.NE.0) GO TO 8537
      DO 5763  NN=1,NUMEG
      IF(NUMEL(NN).EQ.0.)  GO TO 5763
      NAME=NUMEL(NN)
      DO 5002  K=1,NAME
      M=L(NN,K)
5002  A7TOT(M)=A7OLD(M)+ULOC(M,7)-ULOC(M,1)
5763  CONTINUE
8537  CONTINUE
      ICHECK=2
      IF(ITERCHK.NE.0) GO TO 2120
      DO 2121 I=1,NEQ
      R(I)=0.0
2121  PACTUAL(I)=PSAVE(I)+PSTART(I)
      GO TO 3342
2120  CONTINUE
      GO TO 1901
3339  DO 2001 I=1,NEQ
      PTEMP(I)=0.0
      IM=I+1
      IF(IM.GT.NEQ) GO TO 2001
      DO 3901 J=2,MBAND
      IF(SP(I,J).EQ.0.)  GO TO 1804
      PTEMP(I)=PTEMP(I)+SP(I,J)*D(IM)
1804  IM=IM+1
      IF(IM.GT.NEQ)  GO TO 2001
3901  CONTINUE
2001  CONTINUE
      DO 2301 I=1,NEQ
```

```
      IM=I
      JM=1
2108  IF(SP(IM,JM).EQ.0.) GO TO 2201
      PTEMP(I)=PTEMP(I)+SP(IM,JM)*D(IM)
2201  IM=IM-1
      JM=JM+1
      IF(IM.EQ.0)       GO TO 2301
      IF(JM.GT.MBAND)   GO TO 2301
      GO TO 2108
2301  CONTINUE
      DO 5006 I=1,NEQ
      IF(IFIX.EQ.0)   PACTUAL(I)=PTEMP(I)+PSAVE(I)
      IF(IFIX.EQ.1)   PACTUAL(I)=PTEMP(I)
5006  CONTINUE
      IF(ITERCHK.EQ.0) GO TO 6975
      IF(PRIOPTN.EQ.1) WRITE(61,8011)
8011  FORMAT(15X,*I*,5X,*PACTUAL(I)*,10X,*PTEMP(I)*,10X,*PSAVE(I)*,//)
      IF(PRIOPTN.EQ.0) GO TO 4990
      DO 8012 I=1,NEQ
      WRITE(61,8013)  I,PACTUAL(I),PTEMP(I),PSAVE(I)
8012  CONTINUE
4990  CONTINUE
8013  FORMAT(10X,I5,5X,E21.15,10X,E21.15,10X,E21.15,/)
      IF(IPART.EQ.1)WRITE(61,8547)
8547  FORMAT(/,10X,*DTOT(I)*,/)
      IF(IPART.EQ.0) GO TO 1003
      DO 8541 MM=1,NEQ
8541  WRITE(61,8542) DACTUAL(MM)
8542  FORMAT(10X,E21.15)
1003  CONTINUE
      IF(IHORZ.EQ.1)LODPON2=LODPON1+1
      IF(IHORZ.EQ.1)LODPON3=LODPON1+2
      IF(IVERT.EQ.1)LODPON1=LODPON1-1
      IF(IVERT.EQ.1)LODPON2=LODPON1+1
      IF(IVERT.EQ.1)LODPON3=LODPON1+2
      IF(ILAT.EQ.1) LODPON1=LODPON1-2
      IF(ILAT.EQ.1) LODPON2=LODPON1+1
      IF(ILAT.EQ.1) LODPON3=LODPON1+1
      WRITE(61,9001)PACTUAL(LODPON1),PACTUAL(LODPON2),PACTUAL(LODPON3),
     +DACTUAL(LODPON1),DACTUAL(LODPON2),DACTUAL(LODPON3)
9001  FORMAT(*0*,10X,11HPACTUAL(X)=,E21.15,10X,11HPACTUAL(Y)=,E21.15,
     +10X,11HPACTUAL(Z)=,E21.15//16HDISPLACEMENT(X)=,E21.15
     +,10X,16HDISPLACEMENT(Y)=,E21.15,10X,16HDISPLACEMENT(Z)=,E21.15//)
      IF(IVERT.EQ.1)LODPON1=LODPON1+1
      IF(ILAT.EQ.1)LODPON1=LODPON1+2
6975  CONTINUE
      DO 2115 I=1,NEQ
2115  R(I)=PSTART(I)-PTEMP(I)
      IF(PRIOPTN.EQ.0) GO TO 6976
      WRITE(61,9731)
9731  FORMAT(//,20X,*R(I)*,15X,*PSTART(I)*,15X,*PTEMP(I)*,/)
      DO 9732  IMM=1,3
9732  WRITE(61,9733)  R(IMM),PSTART(IMM),PTEMP(IMM)
9733  FORMAT(//,10X,E21.15,10X,E21.15,10X,E21.15,/)
6976  CONTINUE
      IF(ITERCHK.EQ.0)     GO TO 3342
      DO 2451  NN=1,NUMEG
      IF(NUMEL(NN).EQ.0)   GO TO 2451
      NAME=NUMEL(NN)
      DO 2351  K=1,NAME
      M=L(NN,K)
      NI=NODEI(M)
      NJ=NODEJ(M)
      DO 2351  K1=1,2
      IF(K1.EQ.1)   NP=NI
      IF(K1.EQ.2)   NP=NJ
      DO 2251   I=1,6
      IF(IA(NP,I))   1651,1551,1571
1571  NL=IA(NP,I)
      REFSTRT(NP,I)=PSTART(NL)
      REFPTMP(NP,I)=PTEMP(NL)
      GO TO 2251
1551  REFSTRT(NP,I)=0.0
      REFPTMP(NP,I)=0.0
      GO TO 2251
1651  IF(IB(NP,I).LT.0)  GO TO 1751
      NM=IB(NP,I)
      GO TO 1851
1751  NL=-IB(NP,I)+NEQ
      REFSTRT(NP,I)=PSTART(NL)
      REFPTMP(NP,I)=PTEMP(NL)
```

```
      GO TO 2251
1851  IF(IA(NM,I)) 1951,2051,2151
1951  NL=-IB(NM,I)+NEQ
      REFSTRT(NP,I)=PSTART(NL)
      REFPTMP(NP,I)=PTEMP(NL)
      GO TO 2251
2051  REFSTRT(NP,I)=0.0
      REFPTMP(NP,I)=0.0
      GO TO 2251
2151  NL=IA(NM,I)
      REFSTRT(NP,I)=PSTART(NL)
      REFPTMP(NP,I)=PTEMP(NL)
2251  CONTINUE
2351  CONTINUE
2451  CONTINUE
      DO 6949 NP=1,NUMNP
      DO 6949 J=1,3
      KJJ=J+3
      PART1=ABS(REFSTRT(NP,J)-REFPTMP(NP,J))
      PART2=ABS(REFSTRT(NP,KJJ)-REFPTMP(NP,KJJ))
      IF(PART1.GT.DELTA1.OR.PART2.GT.DELTA2)  GO TO 6950
      IF(IPART.EQ.1)WRITE(61,2116)  PART1,PART2
6949  CONTINUE
      GO TO 3342
6950  IF(IPART.EQ.1)WRITE(61,2116)  PART1,PART2
2116  FORMAT(10X,6HPART1=,E21.15,10X,6HPART2=,E21.15)
      NUMITER=NUMITER+1
      IF(NUMITER.LE.MAXITER)  GO TO  2117
      GO TO 900
2117  GO TO 6001
3342  CONTINUE
      IF(ITERCHK.NE.1)  GO TO 8945
      IF(MSUOPTN.EQ.2)  GO TO 8945
      DO 8538  NN=1,NUMEG
      IF(NUMEL(NN).EQ.0)  GO TO 8538
      NAME=NUMEL(NN)
      DO 8539 K=1,NAME
      M=L(NN,K)
      T0=(ULOC(M,8)-ULOC(M,2))/LE(M)
      SI0=(ULOC(M,3)-ULOC(M,9))/LE(M)
      TA=ULOC(M,6)-T0
      TB=ULOC(M,12)-T0
      SIA=ULOC(M,5)-SI0
      SIB=ULOC(M,11)-SI0
8539  A7TOT(M)=A7OLD(M)+ULOC(M,7)-ULOC(M,1)
     ++.5*(T0**2+SI0**2)*LE(M)
     ++LE(M)*(2.*TA**2-TA*TB+2.*TB**2)/30.
     ++LE(M)*(2.*SIA**2-SIA*SIB+2.*SIB**2)/30.
8538  CONTINUE
8945  IF(ITERCHK.NE.1)  GO TO 4993
      IF(MSUOPTN.EQ.1)  GO TO 4993
      DO 4992  NN=1,NUMEG
      IF(NUMEL(NN).EQ.0)   GO TO 4992
      NAME=NUMEL(NN)
      DO 5344  K=1,NAME
      M=L(NN,K)
      ALFA1=ULOC(M,6)
      ALFA2=2.*(-3.*ULOC(M,2)-2.*ULOC(M,6)*LE(M)+3.*ULOC(M,8)-
     +ULOC(M,12)*LE(M))/LE(M)
      ALFA3=3.*(2.*ULOC(M,2)+ULOC(M,6)*LE(M)-2.*ULOC(M,8)+ULOC(M,12)
     +*LE(M))/LE(M)
      BETA1=-ULOC(M,5)
      BETA2=2.*(-3.*ULOC(M,3)+2.*ULOC(M,5)*LE(M)+3.*ULOC(M,9)
     ++ULOC(M,11)*LE(M))/LE(M)
      BETA3=3.*(2.*ULOC(M,3)-ULOC(M,5)*LE(M)-2.*ULOC(M,9)-
     +ULOC(M,11)*LE(M))/LE(M)
      BE(1)=(-ULOC(M,1)+ULOC(M,7))/LE(M)+(ALFA1**2+BETA1**2)/2.
      BE(2)=ALFA1*ALFA2+BETA1*BETA2
      BE(3)=(ALFA2**2+BETA2**2)/2.+ALFA1*ALFA3+BETA1*BETA3
      BE(4)=ALFA2*ALFA3+BETA2*BETA3
      BE(5)=(ALFA3**2+BETA3**2)/2.
      DO 5343   I=1,5
5343  BTO(M,I)=BOL(M,I)+BE(I)
5344  CONTINUE
4992  CONTINUE
4993  CONTINUE
      WRITE(61,8649)
8649  FORMAT(/,10X,*DACTUAL(I)*,/)
      DO 8653   I=1,NEQ
8653  WRITE(61,8654)   DACTUAL(I)
8654  FORMAT(10X,E21.15)
```

```
      WRITE(61,399) PACTUAL(LODPON1),DETRMNT,NUMITER
      IF(DETRMNT.LE.0..AND.DETOPTN.EQ.1)       GO TO 900
      IF(DETCHK.EQ.0) GO TO 8888
      DETINC=DETRMNT-DETCHK
      IF(DETINC.GT.0) GO TO 900
8888  DETCHK=DETRMNT
      WRITE(61,11111)PACTUAL(LODPON1),PTOT(LNODE1,LDOF1),LODPON1,LNODE1
     +,LDOF1
11111 FORMAT(* *,10X,*PACTUAL=*,F10.4,10X,*PTOT=*,F10.4,//
     +11X,*LODPON1=*,I3,10X,*LNODE1=*,I3,10X,*LDOF1=*,I3)
      IF(ABS(PACTUAL(LODPON1)).GE.ABS(PTOT(LNODE1,LDOF1))) GO TO 900
      DO 5007 I=1,NEQ
      PSAVE(I)=PACTUAL(I)
      DTOT(I)=0.0
5007  CONTINUE
      IF(JUSTK.EQ.1)  GO TO 2118
      DO 5281  NN=1,NUMEG
      IF(NUMEL(NN).EQ.0)    GO TO 5281
      NAME=NUMEL(NN)
      DO 5342  K=1,NAME
      M=L(NN,K)
      IF(MSUOPTN.EQ.1)    A7OLD(M)=A7TOT(M)
      DO 5342  I=1,5
      IF(MSUOPTN.EQ.2)    BOL(M,I)=BTO(M,I)
5342  CONTINUE
5281  CONTINUE
2118  CONTINUE
      J=0
      DO 450 N=1,NUMNP
      DO 451 I=1,6
      IF(IA(N,I).EQ.0) GO TO 451
      J=J+1
      R(J)=R(J)+PINC(N,I)
451   CONTINUE
450   CONTINUE
      DO  2119  I=1,NEQ
      IF(IFIX.EQ.0)  PSTART(I)=R(I)
      IF(IFIX.EQ.1)  PSTART(I)=R(I)+PSAVE(I)
2119  CONTINUE
      IF(PRIOPTN.EQ.0) GO TO 6977
      WRITE(61,9735)
9735  FORMAT(///,10X,*R(I)*,/)
      DO  9736  IMM=1,NEQ
9736  WRITE(61,9737)  R(IMM)
9737  FORMAT(//,10X,E21.15,/)
6977  CONTINUE
      ICHECK=3
      GO TO 1001
3337  CONTINUE
C
C         CONDENSE LINEAR STIFFNESS AND LOAD VECTOR OF STRUCTURE
C     ********************************************************************
      IF(NCOND.EQ.0) GO TO 801
      CALL STCONDN
801   CONTINUE
C
C
C
C         SOLVE SYSTEM OF LINEAR EQUATIONS S*D=R
C     ********************************************************************
      IF(PROTYPE.NE.1) GO TO 601
      SCALE=1.E+10
      IF(IHORZ.EQ.1)SCALE=10000.0
299   J=0
      DO 4666 N=1,NUMNP
      DO 4888 I=1,6
      IF(IA(N,I).EQ.0) GO TO 4888
      J=J+1
      R(J)=R(J)+PINC(N,I)
4888   CONTINUE
4666   CONTINUE
      IDET=1
601   CALL LINSOLN
      IF(PROTYPE.EQ.2)  GO TO 1778
      IF(R(LODPON1).EQ.PINT(LNODE1,LDOF1)) CALL IDENT
      IF(R(LODPON1).EQ.PINT(LNODE1,LDOF1)) CALL INVTRNS
C     ********************************************************************
C         IN CASE WHICH WE WANT THE END FORCES DUE TO
C         THE LINEAR SOLUTION SUBROUTINE ENDFORC MAY BE CALLED
C         AT THIS STAGE(THE FIRST ITERATION OF THE FIRST
C         LOAD INCREMENT)
C     ********************************************************************
```

```
      IF(R(LODPON1).EQ.PINT(LNODE1,LDOF1).AND.ISTRESS.EQ.1) CALL ENDFORC
      IF(PROTYPE.NE.1) GO TO 1778
      DETER=DET1(SCALE)
      WRITE(61,399)  PINT(LODPON1),D(LODPON1),DETER,NUMITER
      GO TO 709
C
777   NUMITER=0
C        RECOVER INTERNAL D.O.F."S OF STRUCTURE
C     *****************************************************************
C
      DO 1555 I=1,NEQ
1555  DTEMP(I)=D(I)
      IF(PROTYPE.NE.1) GO TO 715
778   NUMITER=NUMITER+1
715   CONTINUE
      NN=MAXITER+1
      IF(NUMITER.EQ.NN)  GO TO 9999
1778  IF(NCOND.NE.0)  CALL RECOVER
C
C        IDENTIFY DISPLACEMENTS FOUND FROM SOLUTION OF S*D=R AND FROM
C        THE RECOVERY PROCESS
C     *****************************************************************
      CALL IDENT
C        TO HAVE NODAL DEGREES OF FREEDOM IN LOCAL COORDINATES
C     *****************************************************************
      CALL INVTRNS
C
C
C
      DO 180 I=1,NSIZE
      DO 180 J=1,MBAND
180   S(I,J)=0.0

      IF(NUMEG.EQ.1)GO TO 5376
      IF(NUMEL(1).EQ.0)GO TO 5376
       DO 5370 I=1,NSIZE
       DO 5370 J=1,MBAND
5370     STEMP(I,J)=0.
5376  CONTINUE

      IPAR=3
      DO 210 N=1,NUMEG
      IF(NUMEL(N).EQ.0) GO TO 210
      CALL SBEAME1(N,PROTYPE)
210   CONTINUE
       REWIND 6

       IF(NUMEG.EQ.1)GO TO 5371
       IF(NUMEL(1).EQ.0) GO TO 5371
          READ(6,5333)((S(I,J),J=1,MBAND),I=1,NSIZE)
          READ(6,5333)((STEMP(I,J),J=1,MBAND),I=1,NSIZE)
            REWIND 6
        DO 5372 I=1,NSIZE
        DO 5372 J=1,MBAND
5372      S(I,J)=S(I,J)+STEMP(I,J)
       WRITE(6,5333)((S(I,J),J=1,MBAND),I=1,NSIZE)
         REWIND 6
5371  CONTINUE

      IF(NCOND.EQ.0) GO TO 802
      CALL STCONDN
802   CONTINUE
      IF(N2OPTIN.EQ.0)  GO TO 4991
      DO 190 I=1,NSIZE
      DO 190 J=1,MBAND
190   S(I,J)=0.0
      IPAR=4
      DO 310 N=1,NUMEG
      IF(NUMEL(N).EQ.0) GO TO 310
      CALL SBEAME2(N)
310   CONTINUE
4991  CONTINUE
      IF(NCOND.EQ.0) GO TO 899
      CALL STCONDN
899   CONTINUE
      IF(PROTYPE.NE.2) GO TO 222
      IF(EIGVALU.NE.1)GO TO 444
      IF(IDECK.EQ.0)CALL EIGENVL(EIGEN,IDATA)
      IF(IDECK.EQ.1)CALL DECK(EIGEN,IDATA,ICAL4)
      IF(IDECK.EQ.2.OR.IDECK.EQ.3.OR.IDECK.EQ.5)
     +CALL LINDECK(IDECK,ICAL5)
```

```
      IF(IDECK.EQ.2.OR.IDECK.EQ.3.OR.IDECK.EQ.5)
     +CALL NONDECK(IDECK,EIGEN,IDATA,ICAL6)
      IF(IDECK.EQ.4)CALL TOWER(IDECK,ICAL6,EIGEN,IDATA)
      GO TO 900
222   CONTINUE
      DO 107 I=1,NEQ
      DO 108 J=1,MBAND
      READ(4,11) RK
      READ(6,11) RN1
      IF(N2OPTIN.EQ.1) READ(12,11) RN2
      IF(N2OPTIN.EQ.0) S(I,J)=RK+.5*RN1
      IF(N2OPTIN.EQ.1) S(I,J)=RK+.5*RN1+RN2/3.
      IF(N2OPTIN.EQ.0) SP(I,J)=RK+RN1
      IF(N2OPTIN.EQ.1) SP(I,J)=RK+RN1+RN2
108   CONTINUE
107   CONTINUE
      REWIND 4
      REWIND 6
      IF(N2OPTIN.EQ.1) REWIND 12
      IF(NUMITER.EQ.1) GO TO 701
      GO TO 702
9999  J=0
      DO 16 N=1,NUMNP
      DO 17 I=1,6
      IF(IA(N,I).EQ.0) GO TO 17
      J=J+1
      R(J)=R(J)+.5*PINC(N,I)
17    CONTINUE
16    CONTINUE
      DO 333 I=1,NEQ
333   D(I)=DTEMP(I)
      WRITE(61,1899)  PINT(LODPON1),PINC(LNODE1,LDOF1)
1899  FORMAT(15X,23HLOADINCREMENT IS HALVED/
     +15X,6HPINT=,F10.5/15X,5HPINC=,F10.5)
      GO TO 1777
701   UOLD=D(LODPON1)
      GO TO 778
702   IF(ABS((UOLD-D(LODPON1))/D(LODPON1)).LE.TOLER)  GO TO 708
      UOLD=D(LODPON1)
      GO TO 778
708   DO 2555 I=1,NEQ
2555  DTEMP(I)=D(I)
      IDET=3
      WRITE(61,1399) PINT(LODPON1),D(LODPON1),NUMITER
1399  FORMAT(//,10X,5HLOAD=,F10.5/10X,7HDEFLEC=,F15.10/
     +10X,11HITERATIONS=,I5)
      DETER=DET1(SCALE)
      WRITE(61,399)  PINT(LODPON1),D(LODPON1),DETER,NUMITER
      IF(DETER.LE.0..AND.DETOPTN.EQ.1)     GO TO 900
709   J=0
      DO 161 N=1,NUMNP
      DO 171 I=1,6
      IF(IA(N,I).EQ.0) GO TO 171
      J=J+1
      R(J)=R(J)+PINC(N,I)
171   CONTINUE
161   CONTINUE
1777  IF(ABS(PINT(LODPON1)).GT.ABS(PTOT(LNODE1,LDOF1))) GO TO 900
      GO TO 299
C
444   CONTINUE
C     ****************************************************************
C        TO HAVE EIGENVALUE SOLUTION USING DETERMINANT SEARCH METHOD
C        INTHE CASE OF IEIGEN=1  (N1) STIFFNESS MATRIX WOULD
C        BE CONSIDERED IN SUBROUTINE NLEIGNP FOR NONLINEAR
C        EIGENVALUE PROBLEM.
C        FOR IEIGEN=2  (N1+K)  WOULD BE CONSIDERED
C     ****************************************************************
      IEIGEN=1
      SCALE=1.E+10
      IF(IHORZ.EQ.1)SCALE=10000.0
      CALL NLEIGNP(SCALE)
900   IF(ISTRESS.EQ.1)CALL ENDFORC
C
1     FORMAT(7I5)
10    FORMAT(I5)
11    FORMAT(E21.15)
399   FORMAT(///,10X,5HLOAD=,F15.9//
     +10X,12HDETERMINANT=,E25.15/10X,11HITERATIONS=,I5)
699   FORMAT(6F10.6,I5)
799   FORMAT(10X,*PINT(LNODE1,LDOF1)=*,F15.8,10X,*PINC(LNODE1,LDOF1)=*,F
```

```
     +15.8/
     +10X,*PTOT(LNODE1,LDOF1)=*,F15.8,10X,*PINIT2=*,F15.8,10X,*PINC2=*,F
     +15.8/
     +10X,*PTOT2=*,F15.8,10X,*MAXITER=*,I5)
1010  FORMAT(A10,A10,A10,10I5)
1015  FORMAT(I5,6F10.1)
1020  FORMAT(I5)
2010  FORMAT(*1*,A10,A10,A10//8X,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)
2015  FORMAT(*1*,15H  INITIAL LOADS//7H   NODE,27X,14HLOAD DIRECTION//
     +         7H NUMBER,13X,1HX,9X,1HY,9X,1HZ,9X,2HTX,8X,2HTY,8X,2HTZ//)
2020  FORMAT(I5,6X,6F10.5)
5333  FORMAT(E21.14)
      STOP

      END
C
C
C
C
C
C
      SUBROUTINE NODDATA(IGOPTIN)
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
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
C
C     ******************************************************************
C        READ NODAL POINT DATA
C        EXPRESSIONS FOR X(N) AND Y(N) SHOULD BE CHANGED
C        ACCORDING TO CURVE DEFINING THE ARCH.
C     ******************************************************************
C
      WRITE(61,2000)
      WRITE(61,2010)
      WRITE(61,2015)
      IF(IGOPTIN.EQ.0) GO TO 100
      IF(IGOPTIN.EQ.2) GO TO 202
      READ(60,10)  ALFZERO,RADIUS
      WRITE(61,20) ALFZERO,RADIUS
      ALFINC=ALFZERO/NE
      DO 201 I=1,NUMNP
      Z(I)=0.0
      X(I)=RADIUS*SIN((-ALFZERO/2)+(I-1)*ALFINC)
      Y(I)=SQRT(RADIUS**2-X(I)**2)
201   CONTINUE
      GO TO 204
202   READ(60,10) RISE,SPAN
      WRITE(61,30) RISE,SPAN
      DO 203 I=1,NUMNP
      Z(I)=0.0
      X(I)=-SPAN/2+(I-1)*SPAN/NE
      Y(I)=RISE-4*RISE*X(I)**2/(SPAN**2)
203   CONTINUE
204   CONTINUE
90    READ(60,1001)  N,(IA(N,I),I=1,6),(IB(N,I),I=1,6)
      WRITE(61,2020) N,(IA(N,I),I=1,6),(IB(N,I),I=1,6),X(N),Y(N),Z(N)
      IF(N.NE.NUMNP) GO TO 90
      GO TO 101
100   READ(60,1000)  N,(IA(N,I),I=1,6),(IB(N,I),I=1,6),X(N),Y(N),Z(N)
      WRITE(61,2020) N,(IA(N,I),I=1,6),(IB(N,I),I=1,6),X(N),Y(N),Z(N)
      IF(N.NE.NUMNP) GO TO 100
C
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     ******************************************************************
C
101   NEQ=0
```

```
      NCOND=0
      DO 125 N=1,NUMNP
      DO 120 I=1,6
      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     ******************************************************************
C
      WRITE(61,2030)
      WRITE(61,2040)
      WRITE(61,2050)(N,(IA(N,I),I=1,6),(IB(N,I),I=1,6),N=1,
     +NUMNP)
      WRITE(61,2060) NSIZE,NEQ,NCOND
      RETURN
C
10    FORMAT(2F15.10)
20    FORMAT(//,10X,5HALFA=,F15.10,10X,7HRADIUS=,F15.10,//)
30    FORMAT(//,10X,5HRISE=,F15.10,10X,5HSPAN=,F15.10,//)
1000  FORMAT(I5,12I3,3F10.5)
1001  FORMAT(I5,12I3)
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,1HX,4X,1HY,4X,1HZ,4X,2HTX,3X,2HTY,3X,
     +2HTZ,10X),9X,4HX(N),8X,4HY(N),8X,4HZ(N))
2020  FORMAT(I5,6X,6I5,11X,6I5,14X,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,1HX,4X,1HY,4X,1HZ,4X,2HTX,3X,2HTY,3X,2HT
     +Z,10X))
2050  FORMAT(I5,6X,6I5,11X,6I5)
2060  FORMAT(*-*,6HNSIZE=,I3,3X,4HNEQ=,I3,3X,6HNCOND=,I3)
C
      END
C
C
C
C
C
C
C
      SUBROUTINE ELEMENT (N,ICHECK,PROTYPE)
C
C     ******************************************************************
C        TO CALL THE APPROPRIATE ELEMENT SUBROUTINE
C     ******************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      INTEGER PROTYPE
C
      IF(N.EQ.1)  CALL BEAM(N,ICHECK,PROTYPE)
      IF(N.EQ.2)  CALL TRUSS(N,ICHECK,PROTYPE)
      RETURN
C
      END
C
C
C
      SUBROUTINE BAND
C
C     ******************************************************************
C        TO COMPUTE SEMIBANDWIDTH OF STRUCTURE STIFFNESS MATRIX
C        DONE BY FINDING THE MAXIMUM DIFFERENCE BETWEEN THE
C        EQUATION NUMBERS ASSOTIATED WITH THE NODES OF A
C        PARTICULAR ELEMENT
C     ******************************************************************
```

```
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54),
     +L(3,54),IZZ(54),ZPGM(54)
      DIMENSION II(12)
C
      DO 100 N=1,NUMEG
      IF(NUMEL(N).EQ.0) GO TO 100
      NAME=NUMEL(N)
*     WRITE(61,101)N,NUMEG,NUMEL(N),NAME
      DO 900 K=1,NAME
       M=L(N,K)
       NI=NODEI(M)
       NJ=NODEJ(M)
       MIN =1000
       MAX =0

        IF(N.EQ.1) IN=6
        IF(N.EQ.2) IN=3
*     WRITE(61,102)K,NI,NJ,IN,L(N,K),M

         DO 800 I=1,IN
          II(I)=IA(NI,I)
          II(I+IN)=IA(NJ,I)
800      CONTINUE
      NI=IN*2
      DO 300 I=1,NI
*     WRITE(61,103)I,II(I)
300   CONTINUE

         DO 700 I=1,NI
          IF(II(I).NE.0.AND.II(I).GE.MAX) MAX=II(I)
          IF(II(I).NE.0.AND.II(I).LE.MIN) MIN=II(I)
*     WRITE(61,104)I,II(I),MAX,MIN
700      CONTINUE

        MMBAND=MAX-MIN+1

        IF(MMBAND.GE.MBAND)MBAND=MMBAND
*     WRITE(61,105)MAX,MIN,MMBAND,MBAND
900   CONTINUE
100   CONTINUE
      RETURN
*101   FORMAT(*1*,10X,*N=*,I5,5X,*NUMEG=*,I5,5X,*NUMEL(N)=*,I5,5X,*NAME*
*    +,I5//)*
*102   FORMAT(* *,10X,*K=*,I5,5X,*NI=*,I5,5X,*NJ=*,I5,5X,*IN=*,I5,5X,
*    +*L(N,K)=*,I5,5X,*M=*,I5/)
*103   FORMAT(* *,10X,*I=*,I5,5X,*II(I)=*,I5)
*104   FORMAT(* *,10X,*I=*,I5,5X,*II(I)=*,I5,5X,*MAX=*,I5,5X,*MIN=*,I5)
*105  FORMAT(* *,10X,*MAX=*,I5,5X,*MIN=*,I5,5X,*MMBAND=*,I5,5X,*MBAND=*
*    +,I5)
C
      END
C
C
C
      SUBROUTINE BEAM(N,ICHECK,PROTYPE)
C
C     *****************************************************************
C        BEAM ELEMENT SUBROUTINE
C        ONLY STRAIGHT BEAM ELEMENTS ARE CONSIDERED
C     *****************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54)
     +,L(3,54),IZZ(54),ZPGM(54)
      COMMON/8/PINT(30,6),R(168)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      COMMON/10/D(168),G1(168),G2(168),G3(168),G4(168,10),RC(168),
     +SC(168,36),IGAUS
      COMMON/SHEAR/ISHEAR

      DIMENSION AX(54),AZ(54)
```

```
      REAL IXX,IYY,IZZ,KT,LE
      INTEGER PROTYPE
      IF(IPAR.EQ.2)  GO TO 200
C
      IF(PROTYPE.NE.3)   GO TO 1001
      IF(ICHECK.EQ.1)    GO TO 1001
      DO 5555 M=1,NE
      XI=X(NODEI(M))
      YI=Y(NODEI(M))
      ZI=Z(NODEI(M))
      XJ=X(NODEJ(M))
      YJ=Y(NODEJ(M))
      ZJ=Z(NODEJ(M))
      LE(M)=SQRT((XJ-XI)**2+(YJ-YI)**2+(ZJ-ZI)**2)
5555  CONTINUE
      GO TO 120
C
C        READ MATERIAL INFORMATION
C     ***************************************************************
1001  WRITE(61,2000) N
      READ(60,1010) E(N),G(N)
      WRITE(61,2020) NUMEL(N),E(N),G(N)
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),ZPGM(M),
     +IXX(M),IZZ(M),KT(M)
      WRITE(61,2022) M,NODEI(M),NODEJ(M),A(M),ZPGM(M),
     +IXX(M),IZZ(M),KT(M)
      K=K+1
      L(N,K)=M
      NI=NODEI(M)
      NJ=NODEJ(M)
      AA=(X(NJ)-X(NI))**2
      B=(Y(NJ)-Y(NI))**2
      C=(Z(NJ)-Z(NI))**2
      LE(M)=SQRT(AA+B+C)
      IF(K.NE.NUMEL(N)) GO TO 105
120   CONTINUE
      RETURN
C
200   CONTINUE
C
C        CALCULATE AND STORE LINEAR STIFFNESS MATRIX OF BEAM ELEMENTS
C     ***************************************************************
      IFLAG=0
      NAME=NUMEL(N)
      DO 220 K=1,NAME
      IFLAG=IFLAG+1
      M=L(N,K)
C        TERMS OF STIFFNESS MATRIX OF BEAM ELEMENTS IN LOCAL COORDINATES
C     ***************************************************************
      DO 400 I=1,12
      DO 500 J=1,I
      SE(I,J)=0.0
500   CONTINUE
400   CONTINUE
      IF(ISHEAR.EQ.1)GO TO 300
      SE(1,1)=SE(7,7)=E(N)*A(M)/LE(M)
      SE(7,1)=-SE(1,1)
      SE(2,2)=SE(8,8)=12.*IXX(M)*E(N)/LE(M)**3
      SE(8,2)=-SE(2,2)
      SE(3,3)=SE(9,9)=12.*IZZ(M)*E(N)/LE(M)**3
      SE(9,3)=-SE(3,3)
      SE(10,10)=SE(4,4)=G(N)*KT(M)/LE(M)
      SE(10,4)=-SE(10,10)
      SE(6,6)=SE(12,12)=4.*IXX(M)*E(N)/LE(M)
      SE(5,5)=SE(11,11)=4.*IZZ(M)*E(N)/LE(M)
      SE(12,2)=SE(6,2)=6.*IXX(M)*E(N)/LE(M)**2
      SE(8,6)=SE(12,8)=-SE(12,2)
      SE(11,3)=SE(5,3)=-6.*IZZ(M)*E(N)/LE(M)**2
      SE(9,5)=SE(11,9)=-SE(11,3)
      SE(12,6)=2.*IXX(M)*E(N)/LE(M)
      SE(11,5)=2.*IZZ(M)*E(N)/LE(M)
300   CONTINUE
      IF(ISHEAR.NE.1)GO TO 310
      READ(60,320) M,AX(M),AZ(M)
      WRITE(61,340)
      WRITE(61,330)M,AX(M),AZ(M)
```

```
      GX=6*IXX(M)*E(N)/(G(N)*AZ(M)*LE(M)*LE(M))
      GZ=6*IZZ(M)*E(N)/(G(N)*AX(M)*LE(M)*LE(M))

      GGX=1+2*GX
      GGZ=1+2*GZ

      SE(1,1)=SE(7,7)=E(N)*A(M)/LE(M)
      SE(7,1)=-SE(1,1)
      SE(2,2)=SE(8,8)=GGX*(12.*IXX(M)*E(N)/LE(M)**3)
      SE(8,2)=-SE(2,2)
      SE(3,3)=SE(9,9)=GGZ*(12.*IZZ(M)*E(N)/LE(M)**3)
      SE(9,3)=-SE(3,3)
      SE(10,10)=SE(4,4)=G(N)*KT(M)/LE(M)
      SE(10,4)=-SE(10,10)
      SE(6,6)=SE(12,12)=((2+GX)/(2*GGX))*(4.*IXX(M)*E(N)/LE(M))
      SE(5,5)=SE(11,11)=((2+GZ)/(2*GGZ))*(4.*IZZ(M)*E(N)/LE(M))
      SE(12,2)=SE(6,2)=GGX*(6.*IXX(M)*E(N)/LE(M)**2)
      SE(8,6)=SE(12,8)=-SE(12,2)
      SE(11,3)=SE(5,3)=GGZ*(-6.*IZZ(M)*E(N)/LE(M)**2)
      SE(9,5)=SE(11,9)=-SE(11,3)
      SE(12,6)=.5*SE(12,12)
      SE(11,5)=.5*SE(11,11)
310   CONTINUE
C        ...FILL-IN UPPER HALF OF MATRIX BY SYMMETRY
C     ****************************************************************
      DO 210 I=1,12
      DO 210 J=I,12
210   SE(I,J)=SE(J,I)

      IF(ICAL3.EQ.0) WRITE(7,10) ((SE(I,J),J=1,12),I=1,12)
      IF(ISTRESS.EQ.0)  GO TO 601
      WRITE(7,10) ((SE(I,J),J=1,12),I=1,12)
601   CONTINUE
C
C
C        ...TO TRANSFORM SE(12,12) FROM LOCAL TO GLOBAL COORDINATE
C     ****************************************************************
C        SE(12,12) IS THE  STIFFNESS  MATRIX IN GLOBAL
      CALL TRANSFM(M)
      IF(ICAL3.EQ.0) WRITE(8,10) ((SE(I,J),J=1,12),I=1,12)
      IF(ISTRESS.EQ.0)  GO TO 602
      WRITE(8,10) ((SE(I,J),J=1,12),I=1,12)
602   CONTINUE
C
C        ...ASSEMBLE STIFFNESS OF EACH ELEMENT INTO STRUCTURAL
C     ****************************************************************
C        LINEAR STIFFNESS
      CALL ASEMBLE(M)
220   CONTINUE
      IF(ICAL3.EQ.0) REWIND 7
      IF(ICAL3.EQ.0) REWIND 8
      IF(ISTRESS.EQ.1) REWIND 7
      IF(ISTRESS.EQ.1) REWIND 8
      WRITE(4,10) ((S(I,J),J=1,MBAND),I=1,NSIZE)

      IF (ICAL3.EQ.1)GO TO 991

*       PRINT OUT OF THE DIAGONAL STIFFNESS MATRIX........*
***********************************************************
      DO 930 I=1,NEQ
      WRITE(61,931) I,S(I,1)
930   CONTINUE
931   FORMAT(* *,10X,*I=*,I3,*S(I,1)=*,E21.14)

*       PRINT OUT OF THE STRUCTURAL STIFFNESS MATRIX.....*
***********************************************************
      WRITE(61,801)
      JSTART=1
800   JEND=JSTART+5
      IF(JEND.GT.MBAND)JEND=MBAND
      WRITE(61,802)JSTART,JEND
      DO 810  I2=1,NSIZE
      WRITE(61,803)(S(I2,J),J=JSTART,JEND)
810   CONTINUE
      IF(JEND.EQ.MBAND)GO TO 850
      JSTART=JEND+1
      GO TO 800
850   CONTINUE
```

```
320    FORMAT(I5,2F10.4)
340    FORMAT(*0*,13X,*SHEAR AREAS ARE : *//
     +        13X,*M*,15X,*AX(M)*,16X,*AZ(M)*)
330    FORMAT(* *,10X,I5,10X,F10.4,10X,F10.4)
801    FORMAT(*1*,10X,* BEAM STRUCTURAL STIFFNESS MATRIX*//)
802    FORMAT(* *,25X,*JSTART=*,I2,5X,*JEND=*,I2)
803    FORMAT(* *,2X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,
     +E20.13)

991    CONTINUE

       IF (ICAL3.EQ.1)GO TO 903
       WRITE(61,3001)
       DO 3003 I=1,NSIZE
3003   WRITE(61,3004)I,S(I,1)
903    CONTINUE
C         TO PRINT ENTRIES OF EACH ELEMENT
C         STIFFNESS MATRIX IN LOCAL
C      ****************************************************************
       IF(ICAL3.NE.0) GO TO 251
       WRITE(61,2054)
       NAME=NUMEL(N)
       DO 250 M=1,NAME
       WRITE(61,2053) M
       READ(7,10)  ((SE(I,J),J=1,12),I=1,12)
       WRITE(61,2030)
       WRITE(61,2032)  ((SE(I,J),J=1,6),I=1,6)
       WRITE(61,2034)
       WRITE(61,2032)  ((SE(I,J),J=7,12),I=1,6)
       WRITE(61,2036)
       WRITE(61,2032)  ((SE(I,J),J=1,6),I=7,12)
       WRITE(61,2038)
       WRITE(61,2032)  ((SE(I,J),J=7,12),I=7,12)
250    CONTINUE
251    CONTINUE
       REWIND 7
C         TO PRINT ENTRIES OF EACH ELEMENT
C         STIFFNESS MATRIX IN GLOBAL
C      ****************************************************************
       IF(ICAL3.NE.0) GO TO 230
       WRITE(61,2052)
       NAME=NUMEL(N)
       DO 242 M=1,NAME
       WRITE(61,2053) M
       READ(8,10)      ((SE(I,J),J=1,12),I=1,12)
       WRITE(61,2030)
       WRITE(61,2032)  ((SE(I,J),J=1,6),I=1,6)
       WRITE(61,2034)
       WRITE(61,2032)  ((SE(I,J),J=7,12),I=1,6)
       WRITE(61,2036)
       WRITE(61,2032)  ((SE(I,J),J=1,6),I=7,12)
       WRITE(61,2038)
       WRITE(61,2032)  ((SE(I,J),J=7,12),I=7,12)
242    CONTINUE
       REWIND 8
230    CONTINUE
       RETURN
C
10     FORMAT(E21.15)
1010   FORMAT(2E10.2)
1020   FORMAT(3I5,2F10.6,3F10.4)
2000   FORMAT(*1*,23HG R O U P   N U M B E R ,I2//2X,6HNUMBER,6X,7HMODULUS
     +       ,11X,5HSHEAR/4X,2HOF,11X,2HOF,12X,7HMODULUS/1X
     +       ,8HELEMENTS,4X,10HELASTICITY)
2020   FORMAT(I6,2E17.6)
2021   FORMAT(//8H ELEMENT,3X,8HNODEI(M),3X,8HNODEJ(M),12X,4HA(M),10X,
     +7HZPGM(M),7X,6HIXX(M),8X,6HIZZ(M)
     +,10X,5HKT(M))
2022   FORMAT(I6,5X,I5,6X,I5,6X,2F15.6,3E15.4)
2030   FORMAT(*1*,33HSTIFFNESS MATRIX OF BEAM ELEMENTS///9H BLOCK II)
2032   FORMAT(//1X,6F20.6)
2034   FORMAT(///9H BLOCK IJ)
2036   FORMAT(*1*//9H BLOCK JI)
2038   FORMAT(///9H BLOCK JJ)
2052   FORMAT(///,5X,*ENTRIES OF EACH ELEMENT
     + STIFFNESS MATRIX IN GLOBAL*)
2053   FORMAT(///,10X,15HELEMENT NUMBER ,I2)
2054   FORMAT(///,5X,*ENTRIES OF EACH ELEMENT STIFFNESS
     +MATRIX IN LOCAL *)
3001   FORMAT(* *,13X,*DIAGONAL OF LINEAR MATRIX OF BEAM ELEMENTS*)
3004   FORMAT(* *,10X,*I=*,I3,5X,*S(I,1)=*,E21.14)
```

```
      END
C
C
C
C
C
C
      SUBROUTINE TRANSFM(M)
C     *****************************************************************
C
C     ...TO TRANSFER FROM LOCAL TO GLOBAL COORDINATES
C     *****************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54)
     +,L(3,54),IZZ(54),ZPGM(54)
      COMMON/12/ULOC(54,12),U(12),RCOL(9),MSUOPTN,N1GOPTN
      DIMENSION SENEW(12,12)
      REAL  LE,LAMBDA
C
C
C     ...TO EVALUATE ROTATION MATRIX FOR ELEMENT(M)
C     *****************************************************************
      XI=X(NODEI(M))
      YI=Y(NODEI(M))
      ZI=Z(NODEI(M))
      XJ=X(NODEJ(M))
      YJ=Y(NODEJ(M))
      ZJ=Z(NODEJ(M))
      LE(M)=SQRT((XJ-XI)**2+(YJ-YI)**2+(ZJ-ZI)**2)
      RCOL(1)=CX=(XJ-XI)/LE(M)
      RCOL(2)=CY=(YJ-YI)/LE(M)
      RCOL(3)=CZ=(ZJ-ZI)/LE(M)
      SALF=SIN(ZPGM(M))
      CALF=COS(ZPGM(M))
      IF(CX.EQ.0..AND.CZ.EQ.0.)  GO TO 699
      LAMBDA=SQRT(CX**2+CZ**2)
      RCOL(4)=-(CX*CY*CALF+CZ*SALF)/LAMBDA
      RCOL(5)=CALF*LAMBDA
      RCOL(6)=(-CY*CZ*CALF+CX*SALF)/LAMBDA
      RCOL(7)=(CX*CY*SALF-CZ*CALF)/LAMBDA
      RCOL(8)=-LAMBDA*SALF
      RCOL(9)=(CY*CZ*SALF+CX*CALF)/LAMBDA
      GO TO 499
699   IF(CY.EQ.1.) GO TO 799
      RCOL(1)=RCOL(3)=RCOL(5)=RCOL(8)=0.0
      RCOL(4)=RCOL(9)=CALF
      RCOL(6)=SALF
      RCOL(7)=-RCOL(6)
      RCOL(2)=-1.
      GO TO 499
799   RCOL(1)=RCOL(3)=RCOL(5)=RCOL(8)=0.0
      RCOL(4)=-CALF
      RCOL(9)=-RCOL(4)
      RCOL(6)=RCOL(7)=SALF
      RCOL(2)=1.
499   DO 101 MM=1,4
      NAME1=3*MM-2
      NAME2=3*MM
      DO 102 JJ=NAME1,NAME2
      JK=JJ-(MM-1)*3
      DO 103 I=1,12
      SENEW(I,JJ)=0.0
      DO 104 LM=1,3
      LL=LM+(MM-1)*3
      SENEW(I,JJ)=SENEW(I,JJ)+SE(I,LL)*RCOL((LM-1)*3+JK)
104   CONTINUE
103   CONTINUE
102   CONTINUE
101   CONTINUE
      DO 201 MM=1,4
      NAME1=3*MM-2
      NAME2=3*MM
      DO 202 II=NAME1,NAME2
      IK=II-(MM-1)*3
      DO 203 J=1,12
```

```
      SE(II,J)=0.0
      DO 204 LM=1,3
      LL=LM+(MM-1)*3
      SE(II,J)=SE(II,J)+SENEW(LL,J)*RCOL((LM-1)*3+IK)
204   CONTINUE
203   CONTINUE
202   CONTINUE
201   CONTINUE
      RETURN
      END
C
C
C
C
C
C
C
      SUBROUTINE INVTRNS
C
C     *******************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54)
     +,L(3,54),IZZ(54),ZPGM(54)
      COMMON/11/DN(12),W(30,6),V(30,6)
      COMMON/12/ULOC(54,12),U(12),RCOL(9),MSUOPTN,N1GOPTN
      COMMON/16/PRIOPTN
      INTEGER PRIOPTN
      REAL  LE,LAMBDA
C        TO STORE DISPLACEMENTS OF EACH MEMBER IN U(12) ARRAY
C     *******************************************************************
      DO 100 M=1,NE
      DO 210 I=1,12
      NI=NODEI(M)
      NJ=NODEJ(M)
      IF(I.LE.6) GO TO 250
      IF(I.GT.6) GO TO 300
250   NP=NI
      GO TO 350
300   NP=NJ
      GO TO 400
350   U(I)=W(NP,I)
      GO TO 210
400   U(I)=W(NP,I-6)
210   CONTINUE
      XI=X(NODEI(M))
      YI=Y(NODEI(M))
      ZI=Z(NODEI(M))
      XJ=X(NODEJ(M))
      YJ=Y(NODEJ(M))
      ZJ=Z(NODEJ(M))
      LE(M)=SQRT((XJ-XI)**2+(YJ-YI)**2+(ZJ-ZI)**2)
      RCOL(1)=CX=(XJ-XI)/LE(M)
      RCOL(2)=CY=(YJ-YI)/LE(M)
      RCOL(3)=CZ=(ZJ-ZI)/LE(M)
      SALF=SIN(ZPGM(M))
      CALF=COS(ZPGM(M))
      IF(CX.EQ.0..AND.CZ.EQ.0.)  GO TO 699
      LAMBDA=SQRT(CX**2+CZ**2)
      RCOL(4)=-(CX*CY*CALF+CZ*SALF)/LAMBDA
      RCOL(5)=CALF*LAMBDA
      RCOL(6)=(-CY*CZ*CALF+CX*SALF)/LAMBDA
      RCOL(7)=(CX*CY*SALF-CZ*CALF)/LAMBDA
      RCOL(8)=-LAMBDA*SALF
      RCOL(9)=(CY*CZ*SALF+CX*CALF)/LAMBDA
      GO TO 499
699   IF(CY.EQ.1.) GO TO 799
      RCOL(1)=RCOL(3)=RCOL(5)=RCOL(8)=0.0
      RCOL(4)=RCOL(9)=CALF
      RCOL(6)=SALF
      RCOL(7)=-RCOL(6)
      RCOL(2)=-1.
      GO TO 499
799   RCOL(1)=RCOL(3)=RCOL(5)=RCOL(8)=0.0
      RCOL(4)=-CALF
      RCOL(9)=-RCOL(4)
      RCOL(6)=RCOL(7)=SALF
      RCOL(2)=1.
499   DO 101 MM=1,4
```

```
      NAME1=3*MM-2
      NAME2=3*MM
      DO 102 JJ=NAME1,NAME2
      JK=JJ-(MM-1)*3
      ULOC(M,JJ)=0.0
      DO 103 KL=1,3
      LL=KL+(MM-1)*3
      ULOC(M,JJ)=ULOC(M,JJ)+U(LL)*RCOL(3*(JK-1)+KL)
103   CONTINUE
102   CONTINUE
101   CONTINUE
100   CONTINUE
C        TO HAVE PRINT OUT OF NODAL DISPLACEMENTS IN LOCAL COORDINATES
C     ****************************************************************
      IF(PRIOPTN.EQ.0) GO TO 299
      WRITE(61,104)
104   FORMAT(//,10X,*NODAL DISPLACEMENTS ON EACH ELEMENT IN
     +LOCAL COORDINATES*,//)
299   CONTINUE
      DO 105 NN=1,NUMEG
      IF(NUMEL(NN).EQ.0) GO TO 105
      NAME=NUMEL(NN)
      DO 106 K=1,NAME
      M=L(NN,K)
      IF(PRIOPTN.EQ.0) GO TO 399
      WRITE(61,117) M
399   CONTINUE
      NI=NODEI(M)
      NJ=NODEJ(M)
      DO 106 K1=1,2
      IF(K1.EQ.1) NP=NI
      IF(K1.EQ.2) NP=NJ
      NAME1=6*(K1-1)+1
      NAME2=6*K1
      DO 106 I=NAME1,NAME2
      IF(K1.EQ.1) GO TO 91
      J=I-6
      V(NP,J)=ULOC(M,I)
      IF(PRIOPTN.EQ.0) GO TO 99
      WRITE(61,109)  M,NP,J,V(NP,J)
99    CONTINUE
      GO TO 106
91    V(NP,I)=ULOC(M,I)
      IF(PRIOPTN.EQ.0) GO TO 199
      WRITE(61,109) M,NP,I,V(NP,I)
199   CONTINUE
106   CONTINUE
105   CONTINUE
117   FORMAT(*-*,7HELEMENT,I3//)
109   FORMAT(* *,5X,2HV(,I2,1H,,I2,1H,,I1,2H)=,E25.15)
      RETURN
      END
C
C
C
C
C
C
      SUBROUTINE TRUSS(N,ICHECK,PROTYPE)
C
C     ****************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54)
     +,L(3,54),IZZ(54),ZPGM(54)
      COMMON/8/PINT(30,6),R(168)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      COMMON/10/D(168),G1(168),G2(168),G3(168),G4(168,10),RC(168),
     +SC(168,36),IGAUS
      REAL IXX,IYY,IZZ,KT,LE
      INTEGER PROTYPE
      IF(IPAR.EQ.2)  GO TO 200
      IF(PROTYPE.NE.3)  GO TO 1001
      IF(ICHECK.EQ.1)   GO TO 1001
      DO 5555 M=1,NE
      XI=X(NODEI(M))
```

```
      YI=Y(NODEI(M))
      ZI=Z(NODEI(M))
      XJ=X(NODEJ(M))
      YJ=Y(NODEJ(M))
      ZJ=Z(NODEJ(M))
      LE(M)=SQRT((XJ-XI)**2+(YJ-YI)**2+(ZJ-ZI)**2)
5555  CONTINUE
      GO TO 120
C        READ MATERIAL INFORMATION
C     ******************************************************************
1001  WRITE(61,2000)N
      READ(60,1010) E(N),G(N)
      WRITE(61,2020) NUMEL(N),E(N),G(N)
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),ZPGM(M)
      WRITE(61,2022) M,NODEI(M),NODEJ(M),A(M),ZPGM(M)
      K=K+1
      L(N,K)=M
      NI=NODEI(M)
      NJ=NODEJ(M)
      AA=(X(NJ)-X(NI))**2
      B=(Y(NJ)-Y(NI))**2
      C=(Z(NJ)-Z(NI))**2
      LE(M)=SQRT(AA+B+C)
      IF(K.NE.NUMEL(N)) GO TO 105
120   CONTINUE
      RETURN
C
200   CONTINUE
C
C
C        CALCULATE AND STORE LINEAR STIFFNESS MATRIX OF TRUSS ELEMENTS
C     ******************************************************************
      IFLAG=0
      NAME=NUMEL(N)
      DO 220 K=1,NAME
      IFLAG=IFLAG+1
      M=L(N,K)
C
C        TERMS OF STIFFNESS MATRIX OF TRUSS ELEMENTS IN LOCAL COORDINATE
C     ******************************************************************
      DO 400 I=1,12
      DO 500 J=1,I
      SE(I,J)=0.0
500   CONTINUE
400   CONTINUE
      SE(1,1)=SE(7,7)=E(N)*A(M)/LE(M)
      SE(7,1)=-SE(1,1)
C
C        FILL IN UPPER HALF OF MATRIX BY SYMMETRY
C     ******************************************************************
      DO 210 I=1,12
      DO 210 J=1,12
      SE(I,J)=SE(J,I)
210   CONTINUE
      IF(ISTRESS.EQ.0)  GO TO 601
      WRITE(10,10) ((SE(I,J),J=1,12),I=1,12)
601   CONTINUE
C
C
C        TO TRANSFORM SE(12,12) FROM LOCAL TO GLOBAL COORDINATE
C        SE(12,12) IS THE  STIFFNESS  MATRIX IN GLOBAL
C     ******************************************************************
      CALL TRANSFM(M)
      IF(ISTRESS.EQ.0)  GO TO 602
      WRITE(11,10)  ((SE(I,J),J=1,12),I=1,12)
602   CONTINUE
C        ASSEMBLE STIFFNESS OF EACH TRUSS ELEMENT INTO
C        STRUCTURAL LINEAR STIFFNESS
C     ******************************************************************
      CALL ASEMBLE(M)
220   CONTINUE
      IF(ISTRESS.EQ.1) REWIND 10
      IF(ISTRESS.EQ.1) REWIND 11
      WRITE(4,10) ((S(I,J),J=1,MBAND),I=1,NSIZE)

      IF(ICAL3.NE.0) GO TO 991
```

```
*       PRINT OF THE DIAGONAL STIFFNESS MATRIX           *
*****************************************************
      DO 935 I=1,NEQ
      WRITE(61,936)I,S(I,1)
935   CONTINUE
936   FORMAT(* *,10X,*I=*,I3,10X,*S(I,1)=*,E21.14)

*       PRINT OUT OF THE STRUCTURAL STIFFNESS MATRIX           *
***************************************************************
      WRITE(61,701)
      JSTART=1
700   JEND=JSTART+5
      IF(JEND.GT.MBAND)JEND=MBAND
      WRITE(61,702)JSTART,JEND
      DO 710 I2=1,NSIZE
      WRITE(61,703)(S(I2,J),J=JSTART,JEND)
710   CONTINUE
      IF(JEND.EQ.MBAND)GO TO 750
      JSTART=JEND+1
      GO TO 700
750   CONTINUE
701   FORMAT(*1*,20X,*   TRUSS STRUCTURAL STIFFNESS MATRIX*//)
702   FORMAT(* *,25X,*JSTART=*,I2,10X,*JEND=*,I2)
703   FORMAT(* *,2X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,E20.13,3X,
     +E20.13)
991   CONTINUE

C         TO PRINT ENTRIES OF EACH ELEMENT
C         STIFFNESS MATRIX IN LOCAL
C     ******************************************************************
      IF(ICAL3.NE.0) GO TO 251
      WRITE(61,2054)
      NAME=NUMEL(N)
      DO 250 M=1,NAME
      WRITE(61,2053) M
      READ(10,10)  ((SE(I,J),J=1,12),I=1,12)
      WRITE(61,2030)
      WRITE(61,2032)  ((SE(I,J),J=1,6),I=1,6)
      WRITE(61,2034)
      WRITE(61,2032)  ((SE(I,J),J=7,12),I=1,6)
      WRITE(61,2036)
      WRITE(61,2032)  ((SE(I,J),J=1,6),I=7,12)
      WRITE(61,2038)
      WRITE(61,2032)  ((SE(I,J),J=7,12),I=7,12)
250   CONTINUE
251   CONTINUE
      REWIND 10
C         TO PRINT ENTRIES OF EACH ELEMENT
C         STIFFNESS MATRIX IN GLOBAL
C     ******************************************************************
      IF(ICAL3.NE.0) GO TO 230
      WRITE(61,2052)
      NAME=NUMEL(N)
      DO 242 M=1,NAME
      WRITE(61,2053) M
      READ(11,10)  ((SE(I,J),J=1,12),I=1,12)
      WRITE(61,2030)
      WRITE(61,2032)  ((SE(I,J),J=1,6),I=1,6)
      WRITE(61,2034)
      WRITE(61,2032)  ((SE(I,J),J=7,12),I=1,6)
      WRITE(61,2036)
      WRITE(61,2032)  ((SE(I,J),J=1,6),I=7,12)
      WRITE(61,2038)
      WRITE(61,2032)  ((SE(I,J),J=7,12),I=7,12)
242   CONTINUE
      REWIND 11
230   CONTINUE
      RETURN
C
10    FORMAT(E21.15)
1010  FORMAT(2E10.2)
1020  FORMAT(3I5,2F10.6)
2000  FORMAT(*1* 23HG R O U P   N U M B E R ,I2//2X,6HNUMBER,6X,7HMODULUS
     +,11X,5HSHEAR/4X,2HOF,11X,2HOF,12X,7HMODULUS/1X
     +,8HELEMENTS,4X,10HELASTICITY)
2020  FORMAT(I6,2E17.6)
2021  FORMAT(//8H ELEMENT,3X,8HNODEI(M),3X,8HNODEJ(M),12X,4HA(M),9X,
     +7HZPGM(M)/)
2022  FORMAT(I6,5X,I5,6X,I5,6X,F15.6,6X,F15.6)
2030  FORMAT(*1*33HSTIFFNESS MATRIX OF TRUSS ELEMENT///9H BLOCK II)
```

```
2032  FORMAT(//11X,6F20.6)
2034  FORMAT(///9H BLOCK IJ)
2036  FORMAT(*1*//9H BLOCK JI)
2038  FORMAT(///9H BLOCK JJ)
2052  FORMAT(///,5X,*ENTRIES OF EACH ELEMENT
     + STIFFNESS MATRIX IN GLOBAL*)
2053  FORMAT(///,10X,15HELEMENT NUMBER ,I2)
2054  FORMAT(///,5X,*ENTRIES OF EACH ELEMENT STIFFNESS
     +MATRIX IN LOCAL *)
C
      END
C
C
C
C
C
C
C
      SUBROUTINE SBEAME1(N,PROTYPE)
C
C     ****************************************************************
C        EVALUATION OF ENTRIES OF N1 USING DISPLACEMENTS AND
C        ROTATIONS IN LOCAL COORDINATE SYSTEM
C     ****************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54),
     +L(3,54),IZZ(54),ZPGM(54)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      COMMON/12/ULOC(54,12),U(12),RCOL(9),MSUOPTN,N1GOPTN
      REAL LE
      NAME=NUMEL(N)
      IFLAG=0
      DO 220 K=1,NAME
      IFLAG=IFLAG+1
      M=L(N,K)
      DO 105  I=1,12
      DO 105  J=I,12
105   SE(I,J)=0.0
       IF(N.NE.2)GO TO 9
        SE(2,2)=SE(8,8)=(ULOC(M,7)-ULOC(M,1))*E(2)*A(M)/(LE(M)*LE(M))
        SE(3,3)=SE(9,9)=SE(2,2)
        SE(2,8)=SE(3,9)=-SE(2,2)
        GO TO 199
9      CONTINUE
      IF(N1GOPTN.EQ.0)   GO TO 99
      SE(2,2)=SE(3,3)=SE(8,8)=SE(9,9)=6.*(ULOC(M,7)-ULOC(M,1))
     +*E(1)*A(M)/(5.*LE(M)**2)
      SE(2,8)=SE(3,9)=-SE(2,2)
      SE(5,5)=SE(6,6)=SE(11,11)=SE(12,12)=
     +2.*(ULOC(M,7)-ULOC(M,1))*E(1)*A(M)/15.
      SE(2,6)=SE(2,12)=SE(5,9)=SE(9,11)=(ULOC(M,7)-ULOC(M,1))*
     +E(1)*A(M)/(10.*LE(M))
      SE(3,5)=SE(3,11)=SE(8,12)=SE(6,8)=-SE(2,6)
      SE(5,11)=SE(6,12)=-(ULOC(M,7)-ULOC(M,1))*E(1)*A(M)/30.
99    IF(N1GOPTN.EQ.1)  GO TO 199
      T0=(ULOC(M,8)-ULOC(M,2))/LE(M)
      SI0=(ULOC(M,3)-ULOC(M,9))/LE(M)
      F4=ULOC(M,6)+ULOC(M,12)-12.*T0
      G4=ULOC(M,5)+ULOC(M,11)-12.*SI0
      F51=4.*ULOC(M,6)-ULOC(M,12)-3.*T0
      G51=4.*ULOC(M,5)-ULOC(M,11)-3.*SI0
      F52=4.*ULOC(M,12)-ULOC(M,6)-3.*T0
      G52=4.*ULOC(M,11)-ULOC(M,5)-3.*SI0
      SE(1,2)=SE(7,8)=(-F4/10.)*E(1)*A(M)/LE(M)
      SE(1,8)=SE(2,7)=-SE(1,2)
      SE(2,2)=SE(3,3)=SE(8,8)=SE(9,9)=6.*(ULOC(M,7)-ULOC(M,1))
     +*E(1)*A(M)/(5.*LE(M)**2)
      SE(2,8)=SE(3,9)=-SE(2,2)
      SE(5,5)=SE(6,6)=SE(11,11)=SE(12,12)=
     +2.*(ULOC(M,7)-ULOC(M,1))*E(1)*A(M)/15.
      SE(1,3)=SE(7,9)=(G4/10.)*E(1)*A(M)/LE(M)
      SE(1,9)=SE(3,7)=-SE(1,3)
```

```
      SE(1,6)=-F51*E(1)*A(M)/30.
      SE(6,7)=-SE(1,6)
      SE(1,5)=-G51*E(1)*A(M)/30.
      SE(5,7)=-SE(1,5)
      SE(1,12)=-F52*E(1)*A(M)/30.
      SE(7,12)=-SE(1,12)
      SE(1,11)=-G52*E(1)*A(M)/30.
      SE(7,11)=-SE(1,11)
      SE(2,6)=SE(2,12)=SE(5,9)=SE(9,11)=(ULOC(M,7)-ULOC(M,1))*
     +E(1)*A(M)/(10.*LE(M))
      SE(3,5)=SE(3,11)=SE(8,12)=SE(6,8)=-SE(2,6)
      SE(5,11)=SE(6,12)=-(ULOC(M,7)-ULOC(M,1))*E(1)*A(M)/30.
199   CONTINUE
C
C        FILL IN LOWER HALF OF MATRIX BY SYMMETRY
C     ****************************************************************
      DO 106  I=1,12
      DO 106  J=1,I
106   SE(I,J)=SE(J,I)

      IF(ISTRESS.EQ.0)  GO TO 601
      WRITE(1,10)  ((SE(I,J),J=1,12),I=1,12)
601   CONTINUE
C
C
C
C        TO TRANSFER SE(12*12) FROM LOCAL TO GLOBAL COORDINATE
C        SE(12*12) IS THE (N1) STIFFNESS MATRIX IN GLOBAL
C     ****************************************************************
      CALL TRANSFM(M)
      IF(ISTRESS.EQ.0)  GO TO 602
      WRITE(2,10)   ((SE(I,J),J=1,12),I=1,12)
602   CONTINUE
C
C        ASSEMBLE STIFFNESS OF EACH ELEMENT INTO STRUCTURAL
C        STIFFNESS
C     ****************************************************************
      CALL ASEMBLE(M)
220   CONTINUE
      IF(ISTRESS.EQ.1) REWIND 1
      IF(ISTRESS.EQ.1) REWIND 2
       WRITE(6,10)  ((S(I,J),J=1,MBAND),I=1,NSIZE)
C
C
      RETURN
10    FORMAT(E21.15)
      END
C
C
C
C
C
C
C
      SUBROUTINE KEPSI01(N)
C
C     ****************************************************************
C         TO HAVE THE INITIAL STRAIN STIFFNESS MATRIX
C     ****************************************************************
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54),
     +L(3,54),IZZ(54),ZPGM(54)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      COMMON/12/ULOC(54,12),U(12),RCOL(9),MSUOPTN,N1GOPTN
      COMMON/17/A7TOT(54),A7OLD(54),BOL(54,5),BTO(54,5),BE(5)
      REAL LE
      INTEGER PROTYPE
      NAME=NUMEL(N)
      IFLAG=0
      DO 220 K=1,NAME
      IFLAG=IFLAG+1
      M=L(N,K)
      IF(MSUOPTN.EQ.1)  A7=A7TOT(M)
      DO 104 I=1,12
      DO 105 J=1,I
      SE(I,J)=0.0
105   CONTINUE
104   CONTINUE
```

```
      IF(MSUOPTN.EQ.1)  GO TO 99
      RO1=BTO(M,1)+BTO(M,2)/2.+BTO(M,3)/3.+BTO(M,4)/4.+BTO(M,5)/5.
      RO2=BTO(M,1)/2.+BTO(M,2)/3.+BTO(M,3)/4.+BTO(M,4)/5.+BTO(M,5)/6.
      RO3=BTO(M,1)/3.+BTO(M,2)/4.+BTO(M,3)/5.+BTO(M,4)/6.+BTO(M,5)/7.
      RO4=BTO(M,1)/4.+BTO(M,2)/5.+BTO(M,3)/6.+BTO(M,4)/7.+BTO(M,5)/8.
      RO5=BTO(M,1)/5.+BTO(M,2)/6.+BTO(M,3)/7.+BTO(M,4)/8.+BTO(M,5)/9.
      SE(2,2)=SE(3,3)=SE(8,8)=SE(9,9)=(RO3-2.*RO4+RO5)*36.*A(M)*
     +E(1)/LE(M)
      SE(8,2)=SE(9,3)=-SE(2,2)
      SE(5,3)=SE(8,6)=(RO2-5.*RO3+7.*RO4-3.*RO5)*6.*E(1)*A(M)
      SE(6,2)=SE(9,5)=-SE(5,3)
      SE(6,6)=SE(5,5)=(RO1-8.*RO2+22.*RO3-24.*RO4+9.*RO5)*
     +E(1)*A(M)*LE(M)
      SE(11,11)=SE(12,12)=(4.*RO3-12.*RO4+9.*RO5)*E(1)*A(M)*LE(M)
      SE(11,9)=SE(12,2)=(2.*RO3-5.*RO4+3.*RO5)*6.*E(1)*A(M)
      SE(11,3)=SE(12,8)=-SE(11,9)
      SE(11,5)=SE(12,6)=(-2.*RO2+11.*RO3-18.*RO4+9.*RO5)*E(1)*A(M)*LE(M)
99    IF(MSUOPTN.EQ.2)  GO TO 199
      SE(2,2)=SE(3,3)=SE(8,8)=SE(9,9)=6.*A7*A(M)*E(1)/(5.*LE(M)**2)
      SE(9,3)=SE(8,2)=-SE(2,2)
      SE(5,5)=SE(6,6)=SE(11,11)=SE(12,12)=A7*2.*A(M)*E(1)/15.
      SE(6,2)=SE(9,5)=SE(11,9)=A(M)*A7*E(1)/(10.*LE(M))
      SE(5,3)=SE(8,6)=SE(12,8)=-SE(6,2)
      SE(12,2)=A7*A(M)*E(1)/(10.*LE(M))
      SE(11,3)=-SE(12,2)
      SE(12,6)=SE(11,5)=-A7*A(M)*E(1)/(30.)
199   CONTINUE
C
C        FILL IN UPPER HALF OF MATRIX BY SYMMETRY
C     *******************************************************************
      DO 106 I=1,12
      DO 107 J=I,12
      SE(I,J)=SE(J,I)
107   CONTINUE
106   CONTINUE
      IF(ISTRESS.EQ.0)  GO TO 601
      WRITE(14,10)  ((SE(I,J),J=1,12),I=1,12)
601   CONTINUE
C
C
C        TO TRANSFER SE(12*12) FROM LOCAL TO GLOBAL COORDINATE
C        SE(12*12) IS THE (N1) STIFFNESS MATRIX IN GLOBAL
C     *******************************************************************
      CALL TRANSFM(M)
      IF(ISTRESS.EQ.0)  GO TO 602
      WRITE(15,10)  ((SE(I,J),J=1,12),I=1,12)
602   CONTINUE
C
C        ASSEMBLE STIFFNESS OF EACH ELEMENT INTO STRUCTURAL
C        STIFFNESS
C     *******************************************************************
      CALL ASEMBLE(M)
220   CONTINUE
      WRITE(16,10) ((S(I,J),J=1,MBAND),I=1,NSIZE)
      IF(ISTRESS.EQ.1) REWIND 14
      IF(ISTRESS.EQ.1) REWIND 15
      REWIND 16
C
C
      RETURN
10    FORMAT(E21.15)
      END
C
C
C
C
C
C
C
C
C
C
      SUBROUTINE SBEAME2(N)
C
C     *******************************************************************
C        EVALUATION OF ENTRIES OF N2 USING DISPLACEMENTS
C        AND ROTATIONS IN LOCAL COORDINATES
C     *******************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
```

```
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54),
     +L(3,54),IZZ(54),ZPGM(54)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      COMMON/12/ULOC(54,12),U(12),RCOL(9),MSUOPTN,N1GOPTN
      REAL LE
      NAME=NUMEL(N)
      IFLAG=0
      DO 220 K=1,NAME
      IFLAG=IFLAG+1
      M=L(N,K)
      DO 105  I=1,12
      DO 105  J=I,12
105   SE(I,J)=0.0
      T0=(ULOC(M,8)-ULOC(M,2))/LE(M)
      SI0=(ULOC(M,3)-ULOC(M,9))/LE(M)
      T1=ULOC(M,6)
      T2=ULOC(M,12)
      SI1=ULOC(M,5)
      SI2=ULOC(M,11)
      T00=T0*T0
      T10=T1*T0
      T20=T2*T0
      T11=T1*T1
      T21=T2*T1
      T22=T2*T2
      SI00=SI0*SI0
      SI10=SI1*SI0
      SI20=SI2*SI0
      SI11=SI1*SI1
      SI21=SI2*SI1
      SI22=-SI2*SI2
      TS00=T0*SI0
      TS10=T1*SI0
      TS20=T2*SI0
      TS01=T0*SI1
      TS11=T1*SI1
      TS21=T2*SI1
      TS02=T0*SI2
      TS12=T1*SI2
      TS22=T2*SI2
      CON1=A(M)*E(1)
      CON2=CON1*LE(M)
      IF(MSUOPTN.EQ.1)  GO TO 99
      B1=.4*(T11+T22)+18.*T00/5.-3.*T0*(T1+T2)/5.
     +-T21/5.-68.*SI11/15.+2.*SI22/15.+6.*SI00/5.
     ++39.*SI10/5.-SI20/5.-17.*SI1*SI2/5.
      B2=.2*T11+.6*T22+3.6*T00-1.2*T10-.2*T21
     +-20.8*SI11+.2*SI22+1.2*SI00+11.6*SI10-15.2*SI1*SI2
      B3=(6.*T11+27.*T22+108.*T00-45.*T10+18.*T20
     +-9.*T21-292.*SI11+9.*SI22+36.*SI00+489.*SI10
     ++6.*SI20-213.*SI1*SI2)/35.
      B4=33.*T11/140.+39.*T22/28.-1107.*T00/7.-1.8*T10
     ++1.8*T20-33.*T21/70.-1949.*SI11/140.+13.*SI22/28.
     ++9.*SI00/7.+117.*SI10/5.+339.*SI20/7.-711.*SI1*SI2/70.
      B5=9.*T11/35.+27.*T22/14.+27.*T00/7.-27.*T10/14.
     ++27.*T20/14.-9.*T21/14.-627.*SI11/35.+9.*SI22/14.
     ++9.*SI00/7.+423.*SI10/14.+9.*SI20/14.-183.*SI1*SI2/14.
      B6=.4*(SI11+SI22)+18.*SI00/5.-3.*SI0*(SI1+SI2)/5.
     +-SI1*SI2/5.-68.*T11/15.+2.*T22/15.+6.*T00/5.
     ++39.*T10/5.-T20/5.-17.*T21/5.
      B7=.2*SI11+.6*SI22+3.6*SI00-1.2*SI10-.2*SI1*SI2
     +-20.8*T11+.2*T22+1.2*T00+11.6*T10-15.2*T21
      B8=(6.*SI11+27.*SI22+108.*SI00-45.*SI10+18.*SI20
     +-9.*SI1*SI2-292.*T11+9.*T22+36.*T00+489.*T10
     ++6.*T20-213.*T21)/35.
      B9=33.*SI11/140.+39.*SI22/28.-1107.*SI00/7.-1.8*SI10
     ++1.8*SI20-33.*SI1*SI2/70.-1949.*T11/140.+13.*T22/28.
     ++9.*T00/7.+117.*T10/5.+339.*T20/7.-711.*T21/70.
      B10=9.*SI11/35.+27.*SI22/14.+27.*SI00/7.-27.*SI10/14.
     ++27.*SI20/14.-9.*SI1*SI2/14.-627.*T11/35.+9.*T22/14.
     ++9.*T00/7.+423.*T10/14.+9.*T20/14.-183.*T21/14.
      B11=-4.*TS11/15.+TS10/5.+TS12/15.+TS01/5.+TS21/5.
     +-12.*TS00/5.+TS02/5.+TS20/5.-4.*TS22/15.
      B12=-2.*TS11/15.+2.*TS10/5.+TS12/15.+2.*TS01/5.
     ++TS21/15.-12.*TS00/5.-2.*TS22/5.
      B13=-4.*TS11/35.+3.*TS10/7.+3.*TS12/35.+3.*TS01/7.
     ++3.*TS21/35.-72.*TS00/35.-6.*TS02/35.-6.*TS20/35.
     +-18.*TS22/35.
      B14=-11.*TS11/70.+3.*TS10/5.+11.*TS12/70.+
     +3.*TS01/5.+11.*TS21/70.
```

```
+-18.*TS00/7.-3.*TS02/7.-3.*TS20/7.-13.*TS22/14.
 B15=-6.*TS11/35.+9.*TS10/14.+3.*TS12/14.+9.*TS01/14.
++3.*TS21/14.-18.*TS00/7.-9.*TS02/14.
+-9.*TS20/14.-9.*TS22/7.
 SE(2,2)=SE(8,8)=CON1*(6.*B3-6.*B4+2.*B5)/LE(M)
 SE(2,8)=-SE(2,2)
 SE(3,3)=SE(9,9)=CON1*(6.*B8-6.*B9+2.*B10)/LE(M)
 SE(3,9)=-SE(3,3)
 SE(2,3)=CON1*(6.*B13-6.*B14+2.*B15)/LE(M)
 SE(2,9)=-SE(2,3)
 SE(2,6)=(-3.*B2+10.*B3-7.*B4+2.*B5)*CON1/2.
 SE(6,8)=-SE(2,6)
 SE(3,6)=SE(5,8)=(-3.*B12+10.*B13-7.*B14+2.*B15)*CON1/2.
 SE(6,9)=SE(2,5)=-SE(3,6)
 SE(6,6)=(B1-4.*B2+22.*B3/3.-4.*B4+B5)*CON2/2.
 SE(5,5)=(B6-4.*B7+22.*B8/3.-4.*B9+B10)*CON2/2.
 SE(12,12)=(4.*B3/3.-2.*B4+B5)*CON2/2.
 SE(11,11)=(4.*B8/3.-2.*B9+B10)*CON2/2.
 SE(3,5)=(3.*B7-10.*B8+7.*B9-2.*B10)*CON1/2.
 SE(5,9)=-SE(3,5)
 SE(5,6)=-(B11-4.*B12+22.*B13/3.-4.*B14+B15)*CON2/2.
 SE(3,8)=-(6.*B13-6.*B14+2.*B15)*CON1/LE(M)
 SE(8,9)=-SE(3,8)
 SE(2,12)=(4.*B3-5.*B4+2.*B5)*CON1/2.
 SE(8,12)=-SE(2,12)
 SE(3,12)=SE(8,11)=(4.*B13-5.*B14+2.*B15)*CON1/2.
 SE(9,12)=SE(2,11)=-SE(3,12)
 SE(3,11)=(-4.*B8+5.*B9-2.*B10)*CON1/2.
 SE(9,11)=-SE(3,11)
 SE(6,12)=(-B2+11.*B3/3.-3.*B4+B5)*CON2/2.
 SE(6,11)=SE(5,12)=(B12-11.*B13/3.+3.*B14-B15)*CON2/2.
 SE(5,11)=(-B7+11.*B8/3.-3.*B9+B10)*CON2/2.
 SE(11,12)=(-4.*B13/3.+2.*B14-B15)*CON2/2.
99 IF(MSUOPTN.EQ.2)  GO TO 199
 F4=ULOC(M,6)+ULOC(M,12)-12.*T0
 G4=ULOC(M,5)+ULOC(M,11)-12.*SI0
 F51=4.*ULOC(M,6)-ULOC(M,12)-3.*T0
 G51=4.*ULOC(M,5)-ULOC(M,11)-3.*SI0
 F52=4.*ULOC(M,12)-ULOC(M,6)-3.*T0
 G52=4.*ULOC(M,11)-ULOC(M,5)-3.*SI0
 F1=9.*(T11+T22)-2.*T21-36.*T0*(T1+T2)+216.*T00
 G1=9.*(SI11+SI22)-2.*SI1*SI2-36.*SI0*(SI1+SI2)+216.*SI00
 F2=2.*(T11+T22)-T21-3.*T0*(T1+T2)+18.*T00
 G2=2.*(SI11+SI22)-SI1*SI2-3.*SI0*(SI1+SI2)+18.*SI00
 F7=-2.*(T11+T22)+6.*T21-2.*T0*(T1+T2)-3.*T00
 G7=-2.*(SI11+SI22)+6.*SI1*SI2-2.*SI0*(SI1+SI2)-3.*SI00
 F31=6.*T11+T22+2.*T21-54.*T10+6.*T20+54.*T00
 F32=6.*T22+T11+2.*T21-54.*T20+6.*T10+54.*T00
 G31=6.*SI11+SI22+2.*SI1*SI2-54.*SI10+6.*SI20+54.*SI00
 G32=6.*SI22+SI11+2.*SI1*SI2-54.*SI20+6.*SI10+54.*SI00
 F61=8.*T11+3.*T22-4.*T21-12.*T10-2.*T20+27.*T00
 F62=8.*T22+3.*T11-4.*T21-12.*T20-2.*T10+27.*T00
 G61=8.*SI11+3.*SI22-4.*SI1*SI2-12.*SI10-2.*SI20+27.
+*SI00
 G62=8.*SI22+3.*SI11-4.*SI1*SI2-12.*SI20-2.*SI10+27.
+*SI00
 SE(2,2)=SE(8,8)=E(1)*A(M)*(F1/100.+G2/25.)/LE(M)
 SE(2,8)=-SE(2,2)
 SE(2,3)=SE(8,9)=-E(1)*A(M)*F4*G4/(100.*LE(M))
 SE(2,9)=SE(3,8)=-SE(2,3)
 SE(3,3)=SE(9,9)=E(1)*A(M)*(G1/100.+F2/25.)/LE(M)
 SE(3,9)=-SE(3,3)
 SE(2,6)=E(1)*A(M)*(F31+G2)/300.
 SE(6,8)=-SE(2,6)
 SE(2,5)=E(1)*A(M)*F4*G51/300.
 SE(3,6)=-E(1)*A(M)*G4*F51/300.
 SE(6,9)=-SE(3,6)
 SE(3,5)=-E(1)*A(M)*(G31+F2)/300.
 SE(5,9)=-SE(3,5)
 SE(6,6)=E(1)*A(M)*LE(M)*(F61/300.+G2/225.)
 SE(5,6)=E(1)*A(M)*LE(M)*F51*G51/900.
 SE(5,5)=E(1)*A(M)*LE(M)*(G61/300.+F2/225.)
 SE(5,8)=-E(1)*A(M)*F4*G51/300.
 SE(2,12)=E(1)*A(M)*(F32+G2)/300.
 SE(8,12)=-SE(2,12)
 SE(2,11)=E(1)*A(M)*F4*G52/300.
 SE(8,11)=-SE(2,11)
 SE(3,12)=-E(1)*A(M)*G4*F52/300.
 SE(9,12)=-SE(3,12)
 SE(3,11)=-E(1)*A(M)*(G32+F2)/300.
 SE(9,11)=-SE(3,11)
```

```
      SE(6,12)=E(1)*A(M)*LE(M)*(F7-G2/3.)/300.
      SE(6,11)=E(1)*A(M)*LE(M)*F51*G52/900.
      SE(5,12)=E(1)*A(M)*LE(M)*F52*G51/900.
      SE(5,11)=E(1)*A(M)*LE(M)*(G7-F2/3.)/300.
      SE(12,12)=E(1)*A(M)*LE(M)*(F62/300.+G2/225.)
      SE(11,12)=E(1)*A(M)*LE(M)*F52*G52/900.
      SE(11,11)=E(1)*A(M)*LE(M)*(G62/300.+F2/225.)
199   CONTINUE
C
C        FILL IN LOWER HALF OF MATRIX BY SYMMETRY
C     ******************************************************************
      DO 111  I=1,12
      DO 111  J=1,I
111   SE(I,J)=SE(J,I)
      IF(ISTRESS.EQ.0)  GO TO 601
      WRITE(3,10)  ((SE(I,J),J=1,12),I=1,12)
601   CONTINUE
C
C
C        TO TRANSFER SE(12*12) FROM LOCAL TO GLOBAL COORDINATE
C        SE(12*12) IS THE (N2) STIFFNESS MATRIX IN GLOBAL
C     ******************************************************************
      CALL TRANSFM(M)
      IF(ISTRESS.EQ.0)  GO TO 602
      WRITE(5,10)  ((SE(I,J),J=1,12),I=1,12)
602   CONTINUE
C
C        ASSEMBLE STIFFNESS OF EACH ELEMENT INTO STRUCTURAL
C        STIFFNESS
C     ******************************************************************
      CALL ASEMBLE(M)
220   CONTINUE
      IF(ISTRESS.EQ.1) REWIND 3
      IF(ISTRESS.EQ.1) REWIND 5
      WRITE(12,10)   ((S(I,J),J=1,MBAND),I=1,NSIZE)
      REWIND 12
C
C
C
      RETURN
10    FORMAT(E21.15)
      END
C
C
C
C
C
C
C
      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,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54),
     +L(3,54),IZZ(54),ZPGM(54)
      COMMON/8/PINT(30,6),R(168)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
C
C        SET STRUCTRUE STIFFNESS ARRAY AND LOAD VECTOR
C        ARRAY EQUAL TO ZERO
C     ******************************************************************
      IF(IPAR.NE.1) GO TO 90
      DO 5 I=1,NSIZE
      R(I)=0.0
      DO 5 J=1,MBAND
      S(I,J)=0.0
5     CONTINUE
C
C        PROCESSING OF INITIAL LOADS AND NODAL LOADS
C     ******************************************************************
      IF(ICAL1.EQ.0) WRITE(61,2000)
      DO 80 N=1,NUMNP
      DO 70 I=1,6
```

```
      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)=PINT(N,I)
      IF(ICAL1.EQ.0) WRITE(61,2010) II,N,I,R(II)
70    CONTINUE
80    CONTINUE
      RETURN
C
C        ASSEMBLE ELEMENT STIFFNESS AND NODAL LOAD VECTORS
C     ****************************************************************
90    CONTINUE

      IF(IFLAG.NE.1)GO TO 900
      DO 901 I1=1,NSIZE
      DO 902 I2=1,MBAND
       S(I1,I2)=0.0
902   CONTINUE
901   CONTINUE
900   CONTINUE

      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=1,6
      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=1,6
      IF(IA(ND,J)) 125,150,120
120   JJ=IA(ND,J)
*     WRITE(61,931)NI,NJ,K1,NP,I,II,K2,ND,J,JJ
*931  FORMAT(* *,5X,*NI=*,I3,5X,*NJ=*,I3,5X,*K1=*,I3,5X,*NP=*,I3,5X,*I=*
*    +,I3,5X,*II=*,I3/,5X,*K2=*,I3,5X,*ND=*,I5,5X,*J=*,I3,5X,*JJ=*,I3/)
      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 LOWER TRIANGLE INCLUDING MAIN DIAGONAL
C     ****************************************************************
      IF(JJ.LT.II) GO TO 150
      IF(K1.EQ.1) IE=I
      IF(K1.EQ.2) IE=I+6
      IF(K2.EQ.1) JE=J
      IF(K2.EQ.2) JE=J+6
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
       IF(JJ.GT.MBAND)GO TO 150
*      WRITE(61,932)II,JJ,IE,JE,S(II,JJ),S(IE,JE)
*932    FORMAT(* *,5X,*II=*,I5,5X,*JJ=*,I5,5X,*IE=*,I5,5X,*JE=*,I5,5X/
*     +10X,*S(II,JJ)=*,E21.14,10X,*S(IE,JE)=*,E21.14/)
       S(II,JJ)=S(II,JJ)+SE(IE,JE)
150    CONTINUE
155    CONTINUE
160    CONTINUE
165    CONTINUE
C
       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
C
C
C
C
C
C
C
       SUBROUTINE STCONDN
C
C      ******************************************************************
C         TO CONDENSE STRUCTURE STIFFNESS ACCORDING TO D. O.F."S IN
C         ARRAYS IA AND IB, ALSO TO CONDENSE LOAD VECTOR OF STRUCTURE.
C      ******************************************************************
C
       COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
      +ISTRESS
       COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
       COMMON/8/PINT(30,6),R(168)
       COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
       COMMON/10/D(168),G1(168),G2(168),G3(168),G4(168,10),RC(168),
      +SC(168,36),IGAUS
       IF(ICAL1.EQ.0) WRITE(61,2050)
       IF(ICAL1.EQ.0) WRITE(61,2060) (I,R(I),I=1,NSIZE)
C
C         WRITE UNCONDENSED STRUCTURE LINEAR STIFFNESS
C      ******************************************************************
       IF(ICAL1.NE.0) GO TO 90
       IF(IPAR.EQ.2) WRITE(61,2030)
       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
C
       IF(NCOND.EQ.0) GO TO 115
       DO 112 K=1,NCOND
       LL=NSIZE-K
       KK=LL+1
       DO 110 L=1,LL
       J=L-KK+MBAND
       IF(J.LE.0) GO TO 110
       IF(S(KK,J).EQ.0) GO TO 110
       DUM=S(KK,J)/S(KK,MBAND)
       DO 100 MM=1,L
       JJ=MM-L+MBAND
       IF(JJ.LE.0) GO TO 100
```

```
      II=MM-KK+MBAND
      IF(II.LE.0) GO TO 100
      S(L,JJ)=S(L,JJ)-S(KK,II)*DUM
100   CONTINUE
110   CONTINUE
112   CONTINUE
115   CONTINUE
C
C        STORE CONDENSATION DATA
C     ****************************************************************
      IF(NCOND.EQ.0) GO TO 150
      IF(IPAR.NE.2) GO TO 150
      DO 140 I=1,NCOND
      K=NSIZE-NCOND+I
      DO 130 J=1,MBAND
130   SC(I,J)=S(K,J)
140   RC(I)=R(K)
150   CONTINUE
C
C        CHECK DATA GENERATION
C     ****************************************************************
      IF(ICAL1.EQ.0) WRITE(61,2070)
      IF(ICAL1.EQ.0) WRITE(61,2080) (I,R(I),I=1,NEQ)
      IF(ICAL1.NE.0)GO TO 185
      IF(IPAR.EQ.2) WRITE(61,2000)
      K1=1
      K2=8
      K3=MBAND-K1
      IF(K3.LE.7) GO TO 180
160   WRITE(61,2015) K1,K2
      WRITE(61,2020) ((S(I,J),J=K1,K2),I=1,NEQ)
      K1=K1+8
      K2=K2+8
      K3=MBAND-K1
      IF(K3.LE.7) GO TO 180
      GO TO 160
180   WRITE(61,2015)K1,MBAND
      IF(K3.EQ.0) WRITE(61,2027) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF(K3.EQ.1) WRITE(61,2021) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF(K3.EQ.2) WRITE(61,2022) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF(K3.EQ.3) WRITE(61,2023) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF(K3.EQ.4) WRITE(61,2024) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF(K3.EQ.5) WRITE(61,2025) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF(K3.EQ.6) WRITE(61,2026) ((S(I,J),J=K1,MBAND),I=1,NEQ)
      IF(K3.EQ.7) WRITE(61,2020) ((S(I,J),J=K1,MBAND),I=1,NEQ)
185   CONTINUE
C
C        STORE CONDENSED LINEAR STIFFNESS OF STRUCTURE
C     ****************************************************************
      WRITE(4,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 4
C
      RETURN
C
10    FORMAT(E21.15)
2000  FORMAT('1',43HCONDENSED LINEAR STIFFNESS OF STRUCTURE (S))
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',6E21.6)
2026  FORMAT('0',7E16.5)
2027  FORMAT('0',E16.5)
2030  FORMAT('1',45HUNCONDENSED LINEAR STIFFNESS OF STRUCTURE (S))
2050  FORMAT('1',28HUNCONDENSED LOAD VECTOR R(I)//)
2060  FORMAT(' ',2HR(,I2,2H)=,F16.6)
2070  FORMAT('1',26HCONDENSED LOAD VECTOR R(I)//)
2080  FORMAT(' ',2HR(,I2,2H)=,F16.6)
C
      END
C
C
C
C
C
C
      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         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/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
       COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
       COMMON/8/PINT(30,6),R(168)
       COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
       COMMON/10/D(168),G1(168),G2(168),G3(168),G4(168,10),RC(168),
     +SC(168,36),IGAUS
       COMMON/16/PRIOPTN
       INTEGER PRIOPTN
C
       IF(IGAUS.EQ.1) GO TO 99
C         FILL-IN ARRAY D(I) WITH VALUES OF LOAD VECTOR R(I)
C         AFTER SOLUTION D(I) WILL CONTAIN THE DISPLACEMENT VALUES
C      ******************************************************************
C
       DO 110 I=1,NEQ
110    D(I)=R(I)
C
C         CHECK DATA GENERATION FOR SOLUTION OF EQUATIONS
C      ******************************************************************
C
       IF(PRIOPTN.EQ.1) ICAL2=0
       IF (ICAL2.EQ.0) WRITE(61,2020)
       IF (ICAL2.EQ.0) WRITE(61,2010) (I,D(I),I=1,NEQ)
C
C
C         SOLVE SYSTEM OF -NEQ- LINEAR EQUATIONS
C
C
C         FORWARD REDUCTION OF MATRIX (GAUSS ELIMINATION)
C      ******************************************************************
C
99     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      ******************************************************************
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      ******************************************************************
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
       IF(IGAUS.EQ.1) GO TO 140
C
C         CHECK DATA GENERATION
```

```
C       *******************************************************************
C
        IF(PRIOPTN.EQ.1) ICAL2=0
        IF (ICAL2.NE.0) GO TO 140
        WRITE(61,2000)
        WRITE(61,2010) (I,D(I),I=1,NEQ)
        ICAL2=1
140     RETURN
C
2000    FORMAT(* *,34HDISPLACEMENTS FROM LINEAR SOLUTION//)
2010    FORMAT(* *,2HD(,I3,2H)=,E25.15)
2020    FORMAT(* *,31HLOAD VECTOR FOR LINEAR SOLUTION//)
C
        END
C
C
C
C
C
C
C
        SUBROUTINE RECOVER
C
C       *******************************************************************
C          TO RECOVER THE INTERNAL D.O.F."S OF THE STRUCTURE AFTER
C          SOLVING THE SYSTEM OF EQUATINS
C       *******************************************************************
C
        COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
       +ISTRESS
        COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
        COMMON/10/D(168),G1(168),G2(168),G3(168),G4(168,10),RC(168),
       +SC(168,36),IGAUS
        COMMON/16/PRIOPTN
        INTEGER PRIOPTN
C
        IF(PRIOPTN.EQ.1) ICAL2=0
        IF(NCOND.EQ.0) GO TO 120
        DO 110 J=1,NCOND
        JJ=NSIZE-NCOND+J
        DUM=0.0
        K=JJ-1
        DO 100 I=1,K
        II=I-JJ+MBAND
        IF(II.LE.0) GO TO 100
        DUM=DUM+SC(J,II)*D(I)
100     CONTINUE
110     D(JJ)=(RC(J)-DUM)/SC(J,MBAND)
C
        IF(ICAL2.NE.0) GO TO 120
        WRITE(61,2000)
        N=NEQ+1
        WRITE(61,2010) (I,D(I),I=N,NSIZE)
        ICAL2=1
120     RETURN
C
2000    FORMAT(*1*,15HINTERNAL D.O.F.)
2010    FORMAT(* *,2HD(,I3,2H)=,E25.15)
C
        END
C
C
C
C
C
C
C
        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/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
       +ISTRESS
        COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
        COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
        COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54),
       +L(3,54),IZZ(54),ZPGM(54)
        COMMON/10/D(168),G1(168),G2(168),G3(168),G4(168,10),RC(168),
```

```
     +SC(168,36),IGAUS
      COMMON/11/DN(12),W(30,6),V(30,6)
      COMMON/16/PRIOPTN
      INTEGER PRIOPTN
C
C        IDENTIFICATION OF DISPLACEMENTS
C     ******************************************************************
C
      IF(PRIOPTN.EQ.1) ICAL2=0
      IF(ICAL2.EQ.0) WRITE(61,2000)
      DO 240 NN=1,NUMEG
      IF(NUMEL(NN).EQ.0) GO TO 240
      NAME=NUMEL(NN)
      DO 230 K=1,NAME
      M=L(NN,K)
      IF(ICAL2.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=1,6
      IF(IA(NP,I)) 160,155,150
150   NL=IA(NP,I)
      W(NP,I)=D(NL)
      IF(ICAL2.EQ.0) WRITE(61,2020)    M,NP,I,W(NP,I)
      GO TO 220
155   W(NP,I)=0.0
      IF(ICAL2.EQ.0) WRITE(61,2020)    M,NP,I,W(NP,I)
      GO TO 220
160   IF (IB(NP,I).LT.0) GO TO 170
      NM=IB(NP,I)
      GO TO 180
170   NL=-IB(NP,I)+NEQ
      W(NP,I)=D(NL)
      IF(ICAL2.EQ.0) WRITE(61,2020)    M,NP,I,W(NP,I)
      GO TO 220
180   IF (IA(NM,I)) 190,200,210
190   NL=-IB(NM,I)+NEQ
      W(NP,I)=D(NL)
      IF(ICAL2.EQ.0) WRITE(61,2020)    M,NP,I,W(NP,I)
      GO TO 220
200   W(NP,I)=0.0
      IF(ICAL2.EQ.0) WRITE(61,2020)    M,NP,I,W(NP,I)
      GO TO 220
210   NL=IA(NM,I)
      W(NP,I)=D(NL)
      IF(ICAL2.EQ.0) WRITE(61,2020)    M,NP,I,W(NP,I)
220   CONTINUE
230   CONTINUE
240   CONTINUE
      ICAL2=1
      RETURN
C
2000  FORMAT(*1*,35HNODAL DISPLACEMENTS ON EACH ELEMENT)
2010  FORMAT(*-*,7HELEMENT,I3//)
2020  FORMAT(* *,5X,2HU(,I2,1H,,I2,1H,,I1,2H)=,E25.15)
C
      END
C
C
      SUBROUTINE DECK(EIGEN,IDATA,ICAL4)
C
C**********************************************************************
*        SUBROUTINE DECK INCLUDE THE EFFECT OF DECK ON THE ARCH       *
*      BUCKLING PROBLEM .THE EFFECT IS IN THE STIFFNES MATRIX ,       *
*      S(I,J)=K+N1 ,WHICH BECOME :S(I,J)=K+N1+P/H(N)        *
***********************************************************************
C
C**********************************************************************
*     HD  :ELEVATION OF THE DECK. Y(N):THE Y COORDINATE OF THE ARCH RIB*
*     H(N):HEIGHT OF COLUMNS SUPPORTING THE DECK CONNECTED TO THE RIB  *
*          1 FOR THROUGH BRIDGE                *
*     IDECK:2 FOR HALF THROUGH BRDGE           *
*          3 FOR DECK TYPE BRIDGE              *
***********************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
```

```
      COMMON/8/PINT(30,6),R(168)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      DIMENSION H(30)
      INTEGER TYPARCH
************************************************************************
      READ(60,100)HD,TYPARCH
      WRITE(61,300)
      WRITE(61,400)HD,TYPARCH
      IF(TYPARCH.EQ.1)WRITE(61,500)
      IF(TYPARCH.EQ.2)WRITE(61,600)
      IF(TYPARCH.EQ.3)WRITE(61,700)
      READ(6,200)((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 6
      DO 30 N=1,NUMNP
      IF(HD.GT.Y(N))GO TO 10
      H(N)=Y(N)
      GO TO 20
10    H(N)=Y(N)-HD
20    CONTINUE
      WRITE(61,800)N,H(N)
      IF(H(N).EQ.0) GO TO 30
      IF(IA(N,3).LE.0)GO TO 30
      I=IA(N,3)
      WRITE(61,900)I,S(I,1)
      IF(ICAL4.EQ.0)WRITE(61,9999)HD,IA(N,3),H(N),Y(N),I,N
9999  FORMAT(1X,5X,*HD=*,F10.4,3X,*IA=*,I3,3X,*H=*,F10.4,3X,*Y=*,F10.4,
     +3X,*I=*,I3,3X,*N=*,I3)
      S(I,1)=S(I,1)-PINT(N,2)/H(N)
      WRITE(61,900)I,S(I,1)
30    CONTINUE
      WRITE(6,200)((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 6
      CALL EIGENVL(EIGEN,IDATA)
      RETURN
100   FORMAT(F10.4,I5)
200   FORMAT(E21.15)
300   FORMAT(*1*,//20X,*DECK EFFECT ON ARCH BUCKLING IS INCLUDED*//)
400   FORMAT(* *,10X,*DECK ELEVATION =*,F10.4,//,10X,*TYPAARCH=*,I5//)
500   FORMAT(* *,10X,*ANALYSIS OF THROUGH ARCH BRIDGE TYPE*//)
600   FORMAT(* *,10X,*ANALYSIS OF HALF THROUGH ARCH BRIDGE*//)
700   FORMAT(* *,10X,*ANALYSIS OF DECK ARCH BRIDGE TYPE*//)
800   FORMAT(* *,10X,*N=*,I2,10X,*H(N)=*,F10.4//)
900   FORMAT(* *,10X,*I=*,I2,10X,*S(I,1)=*,E21.15//)
      END
C
C
C
C
      SUBROUTINE LINDECK(IDECK,ICAL5)
C
************************************************************************
*         THIS SUBROUTINE IS TO INCLUDE THE LINEAR EFFECT OF BRIDGE    *
*      DECK ON ARCH BUCKLING.INPLANE AND OUT OF PLANE BUCKLING IS      *
*      INCLUDED.                    *
*          FOR IN-PLANE BUCKLING ASSUME THE FOLLOWING : *
*       1.PANELS ARE EQUAL.         *
*       2.HINGE AT LEFT SIDE,ROLLER AT RIGHT SIDE.        *
*       3.INPUT NODES FROM LEFT TO RIGHT    *
************************************************************************
C
************************************************************************
*         LIST OF PARMETERS IN SUB. LINDECK.*
*         ------------------------------------                         *
*      NPANELS:NUMBER OF PANELS     WD     :DECK WIDTH      *
*      PLENGTH:PANEL LENGTH         STRAREA:STRINGERS AREA*
*      A PANEL IS A HORIZONTAL PLANE NORMAL TO THE ARCH AT EACH NODE  *
************************************************************************
C///
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),ACH(54),IXX(54),KT(54),
     +LCH(3,54),IZZ(54),ZPGM(54)
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      DIMENSION H(30),ND(15),F(15,15)
************************************************************************
      READ(60,101)NPANELS,WD,PLENGTH,STRAREA
      WRITE(61,102)NPANELS,WD,PLENGTH,STRAREA
      READ(4,104)((S(I,J),J=1,MBAND),I=1,NSIZE)
      REWIND 4
      NPA=NPANELS-2
      READ(60,103) (ND(I),I=1,NPA)
```

```
      WRITE(61,106)(ND(I),I=1,NPA)
      WRITE(61,105)
C///    MODIFICATION OF LINEAR STIFFNESS FOR IN-PLANE BUCKLING...........
************************************************************************
      IF(IDECK.EQ.3)GO TO 200
      L=0
      DO 100  I=1,NPA
      K=ND(I)
      IF(IA(K,1).LE.0)GO TO 100
      J=IA(K,1)
      L=L+1
C      EQUIVELENT CONCRETE AREA PLUS STRINGERS..............................
************************************************************************
      A=7.*WD/96.+STRAREA
      STIFF=A*E(I)/(L*PLENGTH)
      PRINT 10199,NPANELS,IDECK,ICAL5,L,K,J,A,STIFF,S(J,1)
      IF(ICAL5.EQ.0)WRITE(61,205)K,J,A,STIFF,S(J,1)
      S(J,1)=S(J,1)+STIFF
      IF(ICAL5.EQ.0)WRITE(61,206)S(J,1)
100   CONTINUE
200   CONTINUE
C//     MODIFICATION TO THE LINEAR STIFFNESS MATRIX
C//        OUT-OFF PLANE BUCKLING..............................................
************************************************************************
      IF(IDECK.EQ.2) GO TO 300
      NN=NPANELS-1
      MM=NPANELS-2
C       EQUIVELENT MOMENT OF INERTIA.........................................
************************************************************************
      XID=7./1152.*WD**3
      EL=(PLENGTH**3)/(6.*NN*E(1)*XID)
      PRINT 9999,XID,EL
C       GENERATE THE FLEXEBILITY MATRIX......................................
************************************************************************
      DO 1000 IN=1,MM
      DO 1100 JN=IN,MM
      F(IN,JN)=EL*IN*(NN-JN)*(NN*NN-IN*IN-(NN-JN)*(NN-JN))
1100  CONTINUE
1000  CONTINUE
C      COMPLETE LOWER HALF OF THE STIFFNESS MATRIX OF THE DECK...............
************************************************************************
      DO 111 IN=1,MM
      DO 111 JN=1,IN
111   F(IN,JN)=F(JN,IN)
C       GET THE STIFFNESS MATRIX BY INVERTING THE FLEXIBILITY MATRIX....
************************************************************************
      CALL INVERSE(F,MM,ICAL5)
C       ASSEMBELE THE DECK STIFFNESS MATRIX INTO THE STRUCTURAL S.M......
************************************************************************
      DO 1200 I=1,NPA
      L=ND(I)
      K=IA(L,3)
      IF(K.LE.0) GO TO 1200
      DO 1300 J=1,NPA
      LL=ND(J)
      KK=IA(LL,3)
      IF(KK.LE.0.OR.KK.LT.K)GO TO 1300
      KK=KK-K+1
      IF(KK.GT.MBAND) GO TO 1300
      IF(ICAL5.EQ.0) WRITE(61,210)K,KK,I,J,S(K,KK),F(I,J)
      S(K,KK)=S(K,KK)+F(I,J)
      IF(ICAL5.EQ.0) WRITE(61,220)S(K,KK)
1300  CONTINUE
1200  CONTINUE
300   CONTINUE
      WRITE(4,104) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 4
      READ(4,104) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 4
      DO 1 I=1,NEQ
      WRITE(61,2) I,S(I,1)
1     CONTINUE
2     FORMAT(* *,5X,*I=*,I2,10X,E20.13/)
      RETURN
101   FORMAT(I5,3F10.4)
102   FORMAT(*1*,25X,*EFFECT OF DECK ON IN-PLANE BUCKLING.*///
     +10X,*NUMBER OF PANELS=*,I5,//
     +10X,*DECK WIDTH       =*,F10.4//
     +10X,*PANEL LENGTH     =*,F10.4//
     +10X,*STRINGER AREA    =*,F10.4//)
103   FORMAT(16I5)
```

```
104    FORMAT(E21.15)
105    FORMAT(* *,10X,*DECK NODE INPUTED FROM LEFT TO RIGHT*)
106    FORMAT(* *,10X,I5,5X,I5,5X,I5,5X,I5,5X,I5//)
205    FORMAT(*0*,10X,*K=*,I5,5X,*J=*,I5,5X,*A=*,F10.4,5X,*STIFF=*,E21.15
      +10X,*S(I,J)=*,E21.15)
206    FORMAT(* *,10X,*S(I,J)=*,E21.15//)
210    FORMAT(* *,10X,*K=*,I3,*KK=*,I3,*II=*,I3,*JJ=*,I3//
      +*S(K,KK)=*,E21.15,*F(I,J)=*,E21.15//)
220    FORMAT(*0--*,10X,*S(I,J)=*,E21.15//)
10199  FORMAT(*0--*,1X,*NPANELS*,I5,*IDECK*,I5,*ICAL5*,I2,*L*,I3,*K*,I5,
      +*J*,I5,*A*,F10.4,*STIFF*,E21.15,*S(J,1)*,E21.15)
9999   FORMAT(*0--*,1X,*XID=*,F10.4,*EL=*,F10.4)
       END
C
C
C///
       SUBROUTINE INVERSE(F,MM,ICAL5)
C
*************************************************************************
*          THIS SUBROUTINE CALCULATE THE INVERSE OF A MATRIX           *
*        IT IS USED FOR INVERTING THE DECK FLEXEBILITY MATRIX          *
*        TO THE STIFFNESS MATRIX S(I,J)=F(I,J)**-1.          *
*************************************************************************
C
       DIMENSION F(15,15)
       DO 900 II=1,MM
900    IF(ICAL5.EQ.0)WRITE(61,101)MM,(F(II,J),J=1,MM)
       DO 800 K=1,MM
       B=F(K,K)
       DO 600 I=1,MM
       IF(I.EQ.K)GO TO 600
       Z=F(I,K)/B
       IF(Z.EQ.0)GO TO 600
       DO 500 J=1,MM
500    IF(J.NE.K)F(I,J)=F(I,J)-Z*F(K,J)
       F(I,K)=-Z
600    CONTINUE
       DO 700 J=1,MM
700    IF(J.NE.K)F(K,J)=F(K,J)/B
800    F(K,K)=1./B
       DO 901 JJ=1,MM
901    IF(ICAL5.EQ.0)WRITE(61,101)MM,(F(JJ,J),J=1,MM)
       RETURN
102    FORMAT(*1*,10X,*INVERTION OF THE FLEXIBILITY MATRIX*//)
101    FORMAT(* *,=E21.15)
       END
C
C///
C
C///
       SUBROUTINE NONDECK(IDECK,EIGEN,IDATA,ICAL6)
*************************************************************************
*          THIS PROGRAM TO CACULATED THE MODIFICATION TO THE NONLINEAR  *
*        STIFFNESS MATRIX DUE TO THE DECK.          *
*************************************************************************
C
*************************************************************************
*              PARAMETERS USED IN THE PROGRAM.   *
*              ---------------------------------            *
*         ND:NODE NUMBER ON THE DECK,ONLY ONE NODE PER PANEL.          *
*      NI,NJ:NODE I,J ON RIB(BOTH NODES AT THE SAME PANEL OF ND.)      *
*         WD:DECK WIDTH  HD:DECK ELEVATION  PLENGTH:PANEL LENGTH       *
*   NPANELS:NUMBER OF PANELS.          *
*************************************************************************
C///
       COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
       COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
       COMMON/8/PINT(30,6),R(168)
       COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
       DIMENSION H(30)
*************************************************************************
C
       READ(60,301)NPANELS,WD,PLENGTH,HD
       WRITE(61,302)
       WRITE(61,303)NPANELS,WD,PLENGTH,HD
       READ(6,304)((S(I,J),J=1,MBAND),I=1,NEQ)
       REWIND 6
       NPA=NPANELS-2
C          MODIFICATION OF THE NONLINEAR STIFFNESS MATRIX
C             IN-PLANE EFFECT.
*************************************************************************
```

```
      IF(IDECK.EQ.3)GO TO 300
      WRITE(61,211)
211   FORMAT(1X,10X,*TEST TEST TEST TEST ----*)
      DO 500 IH=1,NPA
      READ(60,305)ND,NI,NJ
      WRITE(61,221)ND,NI,NJ
221   FORMAT(1X,10X,*ND*,I5,5X,*NI*,I5,5X,*NJ*,I5)
      WRITE(61,241)Y(NI),Y(NJ)
241   FORMAT(11X,*Y(NI)*,F10.4,10X,*Y(NJ)*,F10.4)
      IF(HD.EQ.Y(NI).OR.HD.EQ.Y(NJ))GO TO 500
      I=IA(NI,1)
      J=IA(NJ,1)
      K=IA(ND,1)
      H(NI)=HD-Y(NI)
      H(NJ)=HD-Y(NJ)
      IF(K.LE.0)GO TO 500
      IF(I.LE.0.OR.J.LE.0) GO TO 500
      IF(ICAL6.EQ.0)WRITE(61,101)I,J,K,S(I,1),S(J,1),S(K,1)
600   S(K,1)=S(K,1)+PINT(NI,2)/H(NI)+PINT(NJ,2)/H(NJ)
      S(I,1)=S(I,1)+PINT(NI,2)/H(NI)
      S(J,1)=S(J,1)+PINT(NJ,2)/H(NJ)
      IF(ICAL6.EQ.0)WRITE(61,101)I,J,K,S(I,1),S(J,1),S(K,1)
700   IF(K.GT.MBAND)GO TO 500
      IF(I.GT.K)GO TO 125
      IF(ICAL6.EQ.0)WRITE(61,103)I,K,S(I,K)
      S(I,K)=S(I,K)-PINT(NI,2)/H(NI)
      IF(ICAL6.EQ.0)WRITE(61,103)I,K,S(I,K)
125   CONTINUE
      IF(J.GT.K)GO TO 500
      IF(ICAL6.EQ.0)WRITE(61,103)I,K,S(I,K)
      S(J,K)=S(J,K)-PINT(NJ,2)/H(NJ)
      IF(ICAL6.EQ.0)WRITE(61,103)I,K,S(I,K)
500   CONTINUE
300   CONTINUE
      WRITE(61,231)
231   FORMAT(1X,10X,*TEST2 TEST2 TEST2 TEST2*)
C          MODIFICATION TO THE NONLINEAR STIFFNESS MATRIX
C             OUT-OFF PLANE EFFECT.
*************************************************************************
      IF(IDECK.EQ.2)GO TO 200
      WRITE(61,271)IDECK,ICAL6,NPA
271   FORMAT(* *,10X,*TEST3              TEST3          TEST3*
     +11X,*IDECK=*,I5,*ICAL6=*,I5,*NPA=*,I5)
      DO 800 IH=1,NPA
      READ(60,305)ND,NI,NJ
      WRITE(61,281)IH,ND,NI,NJ,HD,Y(NI),Y(NJ)
281   FORMAT(* *,10X,*IH=*,I5,*ND=*,I5,*NI=*,I5,*NJ=*,I5//
     +11X,*HD=*,F10.4,*Y(NI)=*,F10.4,*Y(NJ)=*,F10.4)
      IF(HD.EQ.Y(NI).OR.HD.EQ.Y(NJ))GO TO 800
      I=IA(NI,3)
      J=IA(NJ,3)
      K=IA(ND,3)
      H(NI)=HD-Y(NI)
      H(NJ)=HD-Y(NJ)
      PRINT 1111,IDECK,IH,NPANELS,ND,NI,NJ,HD,Y(NI),Y(NJ),I,J,K,H(NI)
     +,H(NJ),ICAL6
1111  FORMAT(*0--*,2X,*IDECK*,I3,*IH*,I5,*NPANELS*,I5,*ND*,I5,*NI*,I5,
     +*NJ*,I5,//2X,*HD*,F10.4,*Y(NI)*,F10.4,*Y(NJ)*,F10.4,5X,*I*,I5,*J*
     +,I5,*K*,I5//2X,*H(NI)*,F10.4,*H(NJ)*,F10.4,5X,*ICAL6*,I5//)
      IF(K.EQ.0)GO TO 800
      IF(I.LE.0.OR.J.LE.0)GO TO 800
      IF(ICAL6.EQ.0)WRITE(61,101)I,J,K,S(I,1),S(J,1),S(K,1)
      PRINT 2111,S(I,1),S(J,1),S(K,1)
2111  FORMAT(*0-*,5X,*S(I,1)*,E21.15,*S(J,1)*,E21.15,*S(K,1)*,E21.15//)
900   S(K,1)=S(K,1)+PINT(NI,2)/H(NI)+PINT(NJ,2)/H(NJ)
      S(I,1)=S(I,1)+PINT(NI,2)/H(NI)
      S(J,1)=S(J,1)+PINT(NJ,2)/H(NJ)
      PRINT 3111,S(K,1),S(I,1),S(J,1),PINT(NI,2),PINT(NJ,2),H(NI),H(NJ)
3111  FORMAT(*0*,2X,*S(K,1)*,E21.15,*S(I,1)*,E21.15,*S(J,1)*,E21.15//
     +2X,*PINT(NI,2)*,F10.4,*PINT(NJ,2)*,F10.4,*H(NI)*,F10.4,*H(NJ)*,
     +F10.4//)
      IF(ICAL6.EQ.0)WRITE(61,101)I,J,K,S(I,1),S(J,1),S(K,1)
909   IF(K.GT.MBAND)GO TO 800
      IF(I.GT.K)GO TO 825
      IF(ICAL6.EQ.0)WRITE(61,103)I,K,S(I,K)
      S(I,K)=S(I,K)-PINT(NI,2)/H(NI)
      PRINT 4111,I,K,S(I,K)
4111  FORMAT(*0---*,5X,*I*,I5,*K*,I5,*S(I,K)*,E21.15)
      IF(ICAL6.EQ.0)WRITE(61,103)I,K,S(I,K)
825   CONTINUE
      IF(J.GT.K)GO TO 800
```

```
      IF(ICAL6.EQ.0)WRITE(61,103)I,K,S(I,K)
      S(J,K)=S(J,K)-PINT(NJ,2)/H(NJ)
      PRINT 5111,J,K,S(J,K)
5111  FORMAT(*0--*,5X,*J*,I5,*K*,I5,*S(J,K)*,E21.15)
      IF(ICAL6.EQ.0)WRITE(61,103)I,K,S(I,K)
800   CONTINUE
200   CONTINUE
      WRITE(6,304)((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 6
      CALL EIGENVL(EIGEN,IDATA)
      RETURN
305   FORMAT(3I5)
101   FORMAT(* *,//10X,*I=*,I2,5X,*J=*,I2,5X,*K=*,I2/
     +10X,*S(I,1)=*,E21.15,5X,*S(J,1)=*,E21.15,5X,*S(K,1)=*,E21.15//)
103   FORMAT(* *,10X,*I=*,I2,5X,*K=*,I2,5X,*S(I,K)=*,E21.15//)
301   FORMAT(I5,3F10.4)
302   FORMAT(* *,///,20X,*DATA OF THE DECK :*//)
303   FORMAT(* *,10X,*NUMBER OF PANELS=*,I2/
     +         1X,10X,*DECK WIDTH       =*,F10.4/
     +         1X,10X,*PANEL LENGTH     =*,F10.4/
     +         1X,10X,*DECK ELEVATION   =*,F10.4//)
304   FORMAT(E21.15)
      END
C
C
C///
      SUBROUTINE TOWER(IDECK,ICAL6,EIGEN,IDATA)
**********************************************************************
*        THIS SUBROUTINE IS TO CONSEDER THE EFFECT OF DEFORMABEL     *
*       TOWER AND DECK ON THE OUT OFF PLANE BUCKLING OF ARCH BRIDGES *
*              ................................  *
*           LIST OF VARIABELS                    *
*          ---------------------                 *
*     NET:NUMBER OF ELEMENT FOR BRIDGE AND TOWER.          *
*     ET :MODULUS OF ELASTICITY FOR TOWER        *
*     ED :MODULUS OF ELASTICITY FOR DECK         *
*     AT :CROSS SECTIONAL AREA FOR TOWER         *
*     AD :CROSS SECTIONAL AREA FOR DECK          *
*     YIT:MOMENT OF INERTIA FOR TOWER            *
*     YID:MOMENT OF INERTIA FOR DECK             *
*     WD :WIDTH OF DECK                          *
*     PL :PANEL LENGTH                           *
*     HL :HIGHT OF TOWER                         *
*           ---------------------------          *
*        NOTE: THE NODE NUMBERING OF THE SYSTEM SHOULD MATCH THAT OF *
*       THE ARCH SYSTEM,USE AN INCREMENT OF 3 FOR NODE NUMBERS OF THE *
*       DECK AND 1,2 FOR THE TOWERS BASES.       *
**********************************************************************
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),DUMMY(30,6),X(30),Y(30),Z(30)
      COMMON/9/S(168,36),SP(168,36),IDENT,IFLAG
      DIMENSION SSE(6,6),SS(54,54)
      DIMENSION IAT(17,6),NN(20)
      READ(4,101)((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 4
      DO 3 J=1,NEQ
      IF(ICAL6.EQ.0) WRITE(61,1)J,S(J,1)
3     CONTINUE
1     FORMAT(* *,5X,*J=*,I2,,10X,E20.13)
      WRITE(61,2000)
      READ(60,202)NUMNPT
      DO 100 K=1,NUMNPT
      READ(60,1000)  N,(IAT(N,I),I=1,6)
      WRITE(61,2020) (IAT(N,I),I=1,6)
100   CONTINUE
      READ(60,199) NET
      READ(60,1010) ET,ED,AT,YIT,WD,PL,HT
      WRITE(61,2010)ET,ED,AT,YIT,WD,PL,HT
C
C     ****************************************************************
C         PROCESS ARRAYS -IAT- TO FIND EQUATION NUMBERS AND
C        STORE NEQ"S FOR THE TOWER IN ARRAYS IAT.

C     ****************************************************************
      DO 5001 I=1,6
5001  IAT(1,I)=IAT(2,I)=0
C
      INUM=3*(NUMNPT-2)
      NEQT=0
      DO 125 N=3,INUM,3
```

```
      DO 120 I=1,6
      IF(IAT(N,I).NE.1) GO TO 401
      IAT(N,I)=0
      GO TO 120
401   NEQT=NEQT+1
      IAT(N,I)=NEQT
120   CONTINUE
125   CONTINUE
      WRITE(61,911)
      DO 208 N=3,INUM,3
      WRITE(61,2021)N,(IAT(N,I),I=1,6)
208   CONTINUE
      DO 5005 NH=1,2
5005  WRITE(61,2021)NH,(IAT(NH,I),I=1,6)
      WRITE(61,209)NEQT
209   FORMAT(*0*,1X,*NEQT=*,I5//)
      DO 225 K=1,NEQT
      DO 225 KK=1,NEQT
225   SS(K,KK)=0.0
      DO 325 KM=1,6
      DO 325 MK=1,6
325   SSE(KM,MK)=0.0
C
************************************************************************
C.......ELEMENTS OF ELEMENT STIFFNESS MATRIX(PLANE FRAME)..................
************************************************************************
      WRITE(61,2100)
      SEAT=ET*AT/HT
      SEL3T=12.*ET*YIT/(HT**3)
      SEL2T=6.*ET*YIT/(HT*HT)
      SELT=4.*ET*YIT/HT
      AD=7./96.*WD
      YID=7./1152.*(WD**3)
      SEAD=ED*AD/PL
      SEL3D=12.*ED*YID/(PL**3)
      SEL2D=6.*ED*YID/(PL*PL)
      SELD=4.*ED*YID/PL
      IF(ICAL6.EQ.0)WRITE(61,777)ET,AT,YIT,PL,WD,AD,YID,ED
      IF(ICAL6.EQ.0)WRITE(61,707)SEAT,SEL3T,SEL2T,SELT
      IF(ICAL6.EQ.0)WRITE(61,707)SEAD,SEL3D,SEL2D,SELD
777   FORMAT(* *,5X,3HET=,E21.15,3HAT=,F10.4,4HYIT=,F10.4,3HPL=,F10.4,
     +3HWD=,F10.4,3HAD=,F10.4,4HYID=,F10.4,3HED=,E21.15/)
707   FORMAT(* *,10X,*SEAT=*,E21.15,*SEL3T=*,E21.15,*SEL2T=*,E21.15,
     +*SELT=*,E21.15/)
      DO 30 IM=1,NET
      READ(60,1100) M,NI,NJ
      WRITE(61,2200)M,NI,NJ
      IF(IM.NE.1.AND.IM.NE.NET) GO TO 400
      SSE(1,1)=SSE(4,4)=SEAT
      SSE(1,4)=SSE(4,1)=-SEAT
      SSE(2,2)=SSE(5,5)=SEL3T
      SSE(2,5)=SSE(5,2)=-SEL3T
      SSE(2,3)=SSE(3,2)=SSE(2,6)=SSE(6,2)=SEL2T
      SSE(3,5)=SSE(5,3)=SSE(5,6)=SSE(6,5)=-SEL2T
      SSE(3,3)=SSE(6,6)=SELT
      SSE(3,6)=SSE(6,3)=SELT/2
400   IF(IM.EQ.1.OR.IM.EQ.NET) GO TO 500
      SSE(1,1)=SSE(4,4)=SEAD
      SSE(1,4)=SSE(4,1)=-SEAD
      SSE(2,2)=SSE(5,5)=SEL3D
      SSE(2,5)=SSE(5,2)=-SEL3D
      SSE(2,3)=SSE(3,2)=SSE(2,6)=SSE(6,2)=SEL2D
      SSE(3,5)=SSE(5,3)=SSE(5,6)=SSE(6,5)=-SEL2D
      SSE(3,3)=SSE(6,6)=SELD
      SSE(3,6)=SSE(6,3)=SELD/2
500   CONTINUE
      IF(ICAL6.EQ.0)WRITE(61,1001)((SSE(IS,JS),JS=1,6),IS=1,6)
1001..FORMAT(3X,6E21.15)
****************************************************************
C.......ASSEMBEL STIFFNESS MATRIX INTO STRUCTURAL STIFF.................
*******************************************************************
      KI=KJ=0
      DO 165 K1=1,2
      IF(K1.EQ.1) NP=NI
      IF(K1.EQ.2) NP=NJ
      DO 160 I=1,6
      IF(IAT(NP,I).LE.0)GO TO 160
      KI=KI+1
      II=IAT(NP,I)
      DO 155 K2=1,2
      IF(K2.EQ.1) ND=NI
```

```
      IF(K2.EQ.2) ND=NJ
      DO 150 J=1,6
      IF(IAT(ND,J).LE.0) GO TO 150
      KJ=KJ+1
      JJ=IAT(ND,J)
C
C        FILL IN STRUCTURAL STIFFNESS MATRIX
C     ******************************************************************
      IF(K1.EQ.1) IE=KI
      IF(K1.EQ.2) IE=KI+3
      IF(K2.EQ.1) JE=KJ
      IF(K2.EQ.2) JE=KJ+3
C
C        CHANGE -JJ-SUBSCRIPT OF FULL MATRIX TO -JJ- SUBSCRIPT
C        OF BANDED FORMAT, LOOP OVER TERMS OUTSIDE OF BAND
C     ******************************************************************
      IF(ICAL6.EQ.0)WRITE(61,1005)I,J,NI,NJ,NP,ND,IAT(NP,I),IAT(ND,J)
1005  FORMAT(*0*,3X,*I=*,I2,*J=*,I2,*NI=*,I2,*NJ=*,I2,*NP=*,I2,*ND=*,I2/
     +10X,*IAT(NP,I)=*,I5,*IAT(ND,J)=*,I5)
      IF(ICAL6.EQ.0)WRITE(61,1002)II,JJ,SS(II,JJ),IE,JE,SSE(IE,JE)
      SS(II,JJ)=SS(II,JJ)+SSE(IE,JE)
150   CONTINUE
      KJ=0
155   CONTINUE
160   CONTINUE
      KI=0
165   CONTINUE
30    CONTINUE
C
      DO 6 J=1,NEQ
      IF(ICAL6.EQ.0)WRITE(61,8)J,S(J,1)
6     CONTINUE
8     FORMAT(* *,5X,*J=*,I2,,10X,E20.13)
      DO 901 M=3,INUM,3
      DO 909 J=1,6
      IF(IA(M,J).LE.0)GO TO 909
      I=IA(M,J)
      IB=IAT(M,J)
      DO 801 MM=3,INUM,3
      DO 809 JJ=1,6
      IF(IA(MM,JJ).LE.0)GO TO 809
      II=IA(MM,JJ)
      IIB=IAT(MM,JJ)
      IF(II.LT.I) GO TO 809
      II=II-I+1
      IF(II.GT.MBAND)GO TO 809
      IF(ICAL6.EQ.0)WRITE(61,1007)M,J,MM,JJ,IA(M,J),IA(MM,JJ)
      IF(ICAL6.EQ.0)WRITE(61,805)S(I,II)
805   FORMAT(*0-*,10X,*S(I,II)=*,E21.14/)
      S(I,II)=S(I,II)+SS(IB,IIB)
      IF(ICAL6.EQ.0)WRITE(61,1003)I,II,S(I,II),SS(IB,IIB)
809   CONTINUE
801   CONTINUE
909   CONTINUE
901   CONTINUE
1003  FORMAT(*0-*,*I=*,I5,*II=*,I5,*S(I,II)=*,E21.14,*SS(I,II)=*,E21.14)
1007  FORMAT(*0-*,5X,*M=*,I2,*J=*,I2,*MM=*,I2,*JJ=*,I2,*IA(M,J)=*,I5,
     +*IA(MM,JJ)=*,I5/)
      WRITE(4,101)((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 4
      IDECK=3
      CALL NONDECK(IDECK,EIGEN,IDATA,ICAL6)
      RETURN
199   FORMAT(I5)
202   FORMAT(I5)
101   FORMAT(E21.15)
911   FORMAT(* *,///17X,*GENERATING NUMBER OF EQUATIONS*,//)
1000  FORMAT(I5,6I3)
1010  FORMAT(2E15.8,5F10.4)
1100  FORMAT(3I5)
2000  FORMAT(*1*,10X,*ARCH BUCKLING INCLUDING FLEXIBEL DECK AND TOWER*//
     +           14X,*  N                   IA(N,I)*,7X//)
2020  FORMAT(* *,10X,6I3/)
2021  FORMAT(* *,15X,I3,6I3/)
1002  FORMAT(5X,*II=*,I5,*JJ=*,I5,*SS(II,JJ)=*,E21.14/
     +4X,*IE=*,I5,*JE=*,I5,*SSE(IE,JE)=*,E21.14)
2010  FORMAT(*0*,9X,*NUMBER OF ELEMENTS              =*,I5//
     +           10X,*TOWER MODULUS OF ELASTICITY=*,E15.8//
     +           10X,*DECK MODULUS OF ELASTICITY =*,E15.8//
     +           10X,*TOWER CROSS SECTIONAL AREA =*,F10.4//
     +           10X,*TOWER SECOND MOMENT OF AREA=*,F10.4//
```

```
     +              10X,*DECK WIDTH     =*,F10.4//
     +              10X,*PANEL LENGTH   =*,F10.4//
     +              10X,*TOWER HIGHT    =*,F10.4////)
2100  FORMAT(* *,12X,*  M   *,2X,*NI*,3X,*NJ*//)
2200  FORMAT(* *,10X,3I5)
      END
C
C
C///
      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/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      COMMON/10/XB(168),YB(168),X(168),Y(168),EIGNVTR(168,10)
     +,RC(168),SC(168,36),IGAUS
      DIMENSION Z(168),XO(168)
      INTEGER WICHEIG
C
C        ASSUME STARTING EHIFT ,STARTING VECTOR,AND
C        MAXIMUM NUMBER OF ITERATIONS ALLOWED.
C     ****************************************************************
      WRITE(61,2010)
      READ(60,1000)   MAX,EPSI,RHO,WICHEIG
      WRITE(61,2000) MAX,EPSI,RHO,WICHEIG
      IB=0
      DO 100 I=1,NEQ
100   X(I)=XO(I)=1.0
700   IB=IB+1
C
C        OBTAIN VECTOR Y(I) FROM Y(I)=S1(I,J)*X(I)
C        FIRST CHANGE SIGN OF MATRIX S1
C     ****************************************************************
C
      READ(6,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 6
      IF (ICAL3.EQ.1) GO TO 201
      WRITE(61,1004)
1004  FORMAT(*1*,10X,*PRINT OUT OF THE DIAGONAL MATRIX*//)
       DO 1009 ID=1,NEQ
1009    WRITE(61,1002)ID,S(ID,1)
1002    FORMAT(* *,5X,*ID=*,I5,5X,*S(ID,1)=*,E21.14/)
201   CONTINUE
      DO 107 I=1,NEQ
      DO 105 J=1,MBAND
      S(I,J)=-S(I,J)
105   CONTINUE
107   CONTINUE
      WRITE(13,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 13
C
C        HORIZONTAL SWEEP OF S1(I,J)*X(I),DIAGONAL NOT INCLUDED
C     ****************************************************************
C
      DO 130 I=1,NEQ
      Y(I)=0.0
      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     ****************************************************************
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
       IF(IB.NE.1)GO TO 971
       DO 695 I=1,NEQ
695    Z(I)=Y(I)
971    CONTINUE
C
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      ***************************************************************
C
       DO 300 K=1,MAX
       DO 165 I=1,NEQ
165    XB(I)=Y(I)
C         IDATA=0 AND TAPE 4 FOR K ONLY
C         IDATA=1 AND TAPE 9 FOR (K+N1)
C      ***************************************************************
C
       IF(IDATA.EQ.0) READ(4,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
       IF(IDATA.EQ.1) READ(9,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
       REWIND 4
       REWIND 9
       IF(ICAL3.NE.0) GO TO 176
C
C         PRINT DATA SENT TO SUBROUTINE GAUSSOL
C      ***************************************************************
C
       IF(K.NE.1) GO TO 178
       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)
178    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      ***************************************************************
C
       IGAUS=1
       CALL LINSOLN
       IF(ICAL3.EQ.0) WRITE(61,2030)
C
C         OBTAIN VECTOR YB(I) FROM YB(I)=S1(I,J)*XB(I)
C      ***************************************************************
C
       READ(13,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
       REWIND 13
C
C         HORIZONTAL SWEEP OF S1(I,J)*XB(I),DIAGONAL NOT INCLUDED
C      ***************************************************************
C
       DO 200 I=1,NEQ
       YB(I)=0.0
       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     ******************************************************************
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     CLEAN XB(I) FOR (C1,Q1),(C2,Q2),-----
C     ******************************************************************
      IF(IB.EQ.1) GO TO 500
      JB=IB-1
      LB=0
540   LB=LB+1
      ALPHA=0.0
      DO 510 I=1,NEQ
510   ALPHA=ALPHA+EIGNVTR(I,LB)*YB(I)
      DO 530 I=1,NEQ
530   XB(I)=XB(I)-ALPHA*EIGNVTR(I,LB)
      IF(LB.EQ.JB) GO TO 501
      GO TO 540
501   CONTINUE
C
C      OBTIAN A CLEAN YB(I) FROM XB(I)
C     ******************************************************************
C
      READ(13,10) ((S(I,J),J=1,MBAND),I=1,NEQ)
      REWIND 13
C
C       HORIZONTAL SWEEP OF S1(I,J)*XB(I),DIAGONAL NOT INCLUDED
C
      DO 800 I=1,NEQ
      YB(I)=0.0
      II=I+1
      IF (II.GT.NEQ) GO TO 800
      DO 890 J=2,MBAND
      IF (S(I,J).EQ.0) GO TO 880
      YB(I)=YB(I)+S(I,J)*XB(II)
880   II=II+1
      IF (II.GT.NEQ) GO TO 800
890   CONTINUE
800   CONTINUE
C
C       DIAGONAL SWEEP OF S1(I,J)*XB(I)
C
      DO 830 I=1,NEQ
      II=I
      JJ=1
810   IF(S(II,JJ).EQ.0.) GO TO 820
      YB(I)=YB(I)+S(II,JJ)*XB(II)
820   II=II-1
      JJ=JJ+1
      IF(II.EQ.0) GO TO 830
      IF(JJ.GT.MBAND) GO TO 830
      GO TO 810
830   CONTINUE
500   CONTINUE
C
C       COMPUTE RAYLEIGH QUOTIENT
C     ******************************************************************
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)/(ABS(Q2)**.5)
C
C          CHECK CONVERGENCE TO DESIRED EIGENVALUE
C       ******************************************************************
C
        CHECK=ABS((RHO-RQ)/RHO)
        IF(CHECK.LE.EPSI) GO TO 310
        EIGEN=RHO
        DO 260 I=1,NEQ
260     EIGNVTR(I,IB)=XB(I)/(ABS(Q2)**.5)
        IF(ICAL3.EQ.0) WRITE(61,2035) K,EIGEN
        IF(ICAL3.NE.0) GO TO 300
        WRITE(61,2040) K,RHO,CHECK,EIGEN
        WRITE(61,2050)(XB(I),YB(I),Y(I),EIGNVTR(I,IB),I=1,NEQ)
300     CONTINUE
C
C          OBTAIN EIGENVALUE AND CORRESPONDING EIGENVECTOR
C       ******************************************************************
C
310     CONTINUE
        EIGEN=RHO
        DO 320 I=1,NEQ
320     EIGNVTR(I,IB)=XB(I)/(ABS(Q2)**.5)
        ILAST=K
        WRITE(61,9999)IB
        WRITE(61,2070)ILAST
        WRITE(61,2080)EIGEN
        WRITE(61,2090) (EIGNVTR(I,IB),I=1,NEQ)
C
C         CLEAN ORIGINAL VECTOR FOR HIGHER EIGEN VALUE
C       ******************************************************************
C
        IF(IB.GE.WICHEIG)GO TO 720
        LB=0
640     LB=LB+1
        CNN=0.0
        DO 610 I=1,NEQ
610     CNN=CNN+EIGNVTR(I,LB)*Z(I)
        DO 630 I=1,NEQ
630     X(I)=XO(I)-CNN*EIGNVTR(I,LB)
        IF(LB.EQ.IB) GO TO 700
        GO TO 640

720     CONTINUE
        CALL GRAPH(LB)
        RETURN
C
10      FORMAT(E21.15)
1000    FORMAT(I5,2F20.15,I5)
2000    FORMAT(*-*,5X,8HMAX     =,I3//5X,8HEPSI    =,F20.15//5X,8HRHO     =,
       +F10.6//5X,8HWICHEIG=,I3)
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(*0*,5X,10HEIGENVALUE,9X,11HEIGENVECTOR,5X,6HILAST=,I3)
9999    FORMAT(*1*,5X,*EIGEN MODE NO*,I2/)
2080    FORMAT(*-*,E21.14)
2090    FORMAT(* *,20X,E15.8)
C
        END
C
C
C
C
C
```

```
C
C
      SUBROUTINE ENDFORC
C
C     ******************************************************************
C          TO COMPUTE ELEMENT END FORCES
C     ******************************************************************
C
      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/4/SE(12,12)
      COMMON/5/E(3),G(3),NODEI(54),NODEJ(54),A(54),IXX(54),KT(54),
     +L(3,54),IZZ(54),ZPGM(54)
      COMMON/8/PINT(30,6),R(168)
      COMMON/11/DN(12),W(30,6),V(30,6)
      COMMON/12/ULOC(54,12),U(12),RCOL(9),MSUOPTN,N1GOPTN
C
      WRITE(61,2000)
      DO 100 LN=1,NUMNP
      DO 100 I=1,6
100   PINT(LN,I)=0.
C
C     ...PROCESS EVERY ELEMENT OF EACH ELEMENT GROUP...
C     ******************************************************************
C
      DO 200 K=1,NUMEG
      IF(NUMEL(K) .EQ.0) GO TO 200
      NAME=NUMEL(K)
      DO 190 KK=1,NAME
      M=L(K,KK)
      NI=NODEI(M)
      NJ=NODEJ(M)
      IF(K.EQ.2) GO TO 150
      READ(7,10)   ((SE(I,J),J=1,12),I=1,12)
      GO TO 151
150   READ(10,10)  ((SE(I,J),J=1,12),I=1,12)
151   CONTINUE
C
C
C     ...OBTAIN RESULTANT LOADS...
C     ******************************************************************
C
      LN=NI
      DO 145 I=1,6
      DN(I)=0.
      DO 140 J=1,12
140   DN(I)=DN(I)+SE(I,J)*ULOC(M,J)
145   PINT(LN,I)=PINT(LN,I)+DN(I)
      LN=NJ
      DO 160 I=7,12
      DN(I)=0.
      DO 155 J=1,12
155   DN(I)=DN(I)+SE(I,J)*ULOC(M,J)
160   CONTINUE
C
C     ...WRITE RESULTANT LOADS OF THE NODES OF EACH ELEMENT...
C     ******************************************************************
C
      WRITE(61,2010) M,(DN(I),I=1,6),(DN(I),I=7,12)
C
C
190   CONTINUE
200   CONTINUE
C
      REWIND 7
      REWIND 10
      RETURN
C
10    FORMAT(E21.15)
2000  FORMAT(10X,31HRESULTANT LOADS ON EACH ELEMENT///)
2010  FORMAT(*-*,8H ELEMENT,I3//4X,6HNODE-I,6E15.9//
     +4X,6HNODE-J,6E15.9)
C
      END
C
C
C
C
C
```

```
C
      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(2F10.7,I10,F10.7)
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=,F10.7///
     +   6H FTOL=,F10.7///6H NTOL=,I3///7H DINCR=,F10.7)
2020  FORMAT(*-*,E25.15,5X,E25.15//)
2030  FORMAT(////13X,6HLAMBDA,17X,11HDETERMINANT//)
C
      END
C
C
C
C
C
C
C
      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 ROOTS 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     ******************************************************************
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     ******************************************************************
C
      IF(ABS(B-A)/2..LE.XTOL)  RETURN
C
C        CHECK FOR SUFFICIENTLY SMALL DETERMINANT VALUE
C PROTYPE=3   FOR INCREMENTAL LOADING IN MOVING COORDINATES
```

```
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     *****************************************************************
C
      NM1=N-1
      WRITE(61,2020) NM1,A,W,B,FA,FW,FB
C
C        CHANGE TO NEW INTERVAL
C     *****************************************************************
C
      IF(SI GNFA*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
C
C
C
C
C
C
C
      FUNCTION DET1(SCALE)
C
C     *****************************************************************
C        THIS FUNCTION COMPUTES THE VALUE OF THE DETERMINANT OF
C        THE MATRIX S=K+N1+N2
C     *****************************************************************
C
      COMMON/2/ NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/9/ S(168,36),SP(168,36),IDET,IFLAG
C
      IF(IDET.EQ.1) GO TO 250
      IF(IDET.EQ.2) GO TO 450
      DO 490 I=1,NEQ
      DO 490 J=1,MBAND
490   S(I,J)=SP(I,J)
C        FORWARD REDUCTION OF MATRIX(GAUSS ELEMINATION)
C     *****************************************************************
450   DO 390 LN=1,NEQ
      DO 380 LL=2,MBAND
      IF(S(LN,LL).EQ.0.) GO TO 380
      I=LN+LL-1
      C=S(LN,LL)/S(LN,1)
      J=0
      DO 350 KK=LL,MBAND
      J=J+1
350   S(I,J)=S(I,J)-C*S(LN,KK)
      S(LN,LL)=C
380   CONTINUE
390   CONTINUE
250   CONTINUE
C
C        COMPUTE DETERMINANT OF MATRIX S
C        SCALE DOWN"DET1" BY A "SCALE" VALUE AFTER EACH STEP
C     *****************************************************************
C
      DT=1.
```

```
      DO 400 I=1,NEQ
      DT=DT*S(I,1)/SCALE
400   CONTINUE
      DET1=DT
      RETURN
C
      END
C
C
C
C
C
C
C
      FUNCTION DET(L,SCALE)
C
C     ******************************************************************
C
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/9/S(168,36),SP(168,36),IDET,IFLAG
      REAL K,L,N1,N2
      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(6,10)  N1
      IF(IEIGEN.EQ.2)  READ(9,10)  N1
      READ(12,10)  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  6
      REWIND  9
      REWIND  12
      DO 390  LN=1,NEQ
      DO 380  LL=2,MBAND
      IF(S(LN,LL).EQ.0.)  GO TO 380
      I=LN+LL-1
      C=S(LN,LL)/S(LN,1)
      J=0
      DO 350  KK=LL,MBAND
      J=J+1
350   S(I,J)=S(I,J)-C*S(LN,KK)
      S(LN,LL)=C
380   CONTINUE
390   CONTINUE
      DT=1.
      DO 400  I=1,NEQ
      DT=DT*S(I,1)/SCALE
400   CONTINUE
      DET=DT
      RETURN
10    FORMAT(E21.15)
      END
C
C
C///
      SUBROUTINE GRAPH(LB)

      INTEGER VIEW,WICHEIG
      LOGICAL FIRST

      COMMON/1/NE,NUMNP,NUMEG,LE(54),NUMEL(3),IPAR,ICAL1,ICAL2,ICAL3,
     +ISTRESS
      COMMON/2/NSIZE,NEQ,NCOND,MBAND,IEIGEN
      COMMON/3/IA(30,6),IB(30,6),X(30),Y(30),Z(30)
      COMMON/10/D(168),G1(168),G2(168),G3(168),EIGNVTR(168,10),RC(168),
     +SC(168,36),IGAUS

      DATA FIRST /.TRUE./

      REWIND 21

      IF (.NOT.FIRST) GOTO 100
        READ (60,10) VIEW
        READ (60,10) WICHEIG
        FIRST=.FALSE.
100   CONTINUE
```

```
      WRITE (21,10) VIEW
      WRITE (21,10) NUMNP
      WRITE (21,10) NEQ
      WRITE (21,10) WICHEIG
10    FORMAT (I5)
      WRITE (21,20) ((IA(I,J),J=1,6),I=1,NUMNP)
20    FORMAT (6I3)
      WRITE (21,30) (X(I),Y(I),Z(I),I=1,NUMNP)
30    FORMAT (3E21.14)
      WRITE (21,40) ((EIGNVTR(I,J),I=1,NEQ),J=1,WICHEIG)
40    FORMAT (E21.14)
      RETURN
      END
```