AFFDL-TR-68-38

LARGE DEFLECTION AND STABILITY ANALYSIS OF TWO-DIMENSIONAL TRUSS AND FRAME STRUCTURES

J. S. PRZEMIENIECKI

D. M. PURDY, 1st LT., USAF

FOREWORD

This report was prepared jointly by the Air Force Institute of Technology (AFIT) and the Air Force Flight Dynamics Laboratory (AFFDL), Wright-Patterson Air Force Base, Ohio, under AFIT Project No. 67-35, and DOL Project No. 1467, Task No. 146701, "Stress-Strain Analysis Methods for Structures Exposed to Creep Environments". The theoretical part of this report is based on a paper by Dr. J. S. Przemieniecki, AFIT, entitled, "Large Deflection Analysis of Frame Structures" which was presented at the 7th International Symposium on Space Technology and Science, 15-20 May 1967, Tokyo, Japan.

This report summarizes work performed during the period 1 August 1967 to 20 October 1967 and it includes description of the computer program. Further information regarding this program can be obtained by contacting AFFDL (Lt. D. M. Purdy), Wright-Patterson Air Force Base, Ohio 45433 (Tel: 513-255-5689).

This technical report has been reviewed and is approved.

FRANCIS J. JANIK, JR.
Chief, Theoretical Mechanics Branch
Structures Division
AF Flight Dynamics Laboratory

ABSTRACT

Frame structures with either pinned or fixed joints are frequently used in aerospace applications as the primary structure supporting light secondary panels or other structural assemblies. In the analysis of such structural designs it is very often necessary to consider loading conditions for which the deflections are large enough to cause significant changes in the geometry of the structure so that the equations of equilibrium must be formulated for the deformed configuration. In the present paper the general analysis for large deflections of frame structures is presented using the concept of discrete element idealization. The solution for deflections and stresses is presented as a step-by-step matrix method based on load increments and is particularly suitable for computer programming. As a bi-product of the large deflection analysis the eigenvalue equations for structural stability are also formulated. The theoretical results of the nonlinear, large deflection matrix solution are compared with the exact analytical results for a square frame. In addition, the results for deflections of a six-bay truss and buckling of columns with either constant axial load or gravity loading are also presented. The computer program listing and instructions for the preparation of input data are included.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

In the analysis of aerospace structures it is often necessary to consider loading conditions for which the deflections are large enough to cause significant changes in the geometry of the structure so that the equations of equilibrium must be formulated for the deformed structure. The differential equations describing the large deflections are nonlinear and therefore in order to determine deflections of complex structures we must use numerical methods. The matrix displacement method of analysis, which is a numerical method based on the concept of discrete element idealization, can conveniently be used for such applications. In this method all computational steps are expressed in matrix algebra, from the initial input information to the final output representing deflections and forces or stresses.

The matrix displacement method of analysis has initially been developed as a linear theory for small deflections; however, the basic method can also be extended to large deflections using a step-by-step solution with loading incrementation. Because of the presence of large deflections, the strain-displacement equations contain nonlinear terms which must be included in the calculation of stiffness properties of individual discrete elements used to represent the actual structure. This leads to the element stiffness matrix consisting of the sum of elastic and geometrical stiffnesses. By contrast, in the small deflection analysis only the elastic stiffness is present. The large deflection analysis is also used to determine structural stability for a given set of externally applied loads.

The basic concept of geometrical stiffnesses was first used by Turner (Reference 1) and his associates for the analysis of structures idealized into pin-jointed bars and triangular plates carrying membrane stresses. The method was essentially based on the strain energy formulation for large deflections. Similar approaches were used by several authors for the analysis of structures made up from bars and beam elements (References 2-10), triangular plates (References 9, 10), rectangular plates (References 11, 12), and axisymmetrical shell elements (Reference 13). In a different approach purely geometrical consideration was used to derive geometrical stiffnesses for bars, beams, triangular plates with membrane stresses, and tetrahedron elements (References 14-16). An excellent review of papers in this field was given by Martin (Reference 10) in a paper presented at the Conference on Matrix Methods in Structural Mechanics held at the Air Force Institute of Technology in October 1965.

In this report the basic principles of the large deflection analysis by the matrix displacement method are discussed and a general method of determining geometrical stiffnesses for arbitrary structural elements is derived. The proposed method is illustrated for the calculation of stiffnesses of typical frame elements such as, pin-jointed bars, fixed-pinned beams, and beam elements; however, for simplicity of presentation only two-dimensional elements are considered. For general types of frame structures, three-dimensional structural elements would have to be used. These elements could be used, for example, for the analysis of a three-dimensional truss structure of a lifting reentry body (Figure 1). This structure is made up from an assembly of pin-jointed bars and beams with either pinned or rigid connections and is used to support secondary panels and other structural assemblies such as nose cone and so forth.

The matrix displacement method of analysis is applied to a square frame loaded by a diagonally opposite pair of loads, either in tension or in compression, and the results are compared with experimentally obtained deflections. As a further check on the accuracy of the matrix solution an exact nonlinear solution for deflections in terms of elliptic integrals is also used. Both the theoretical and experimental results indicated clearly the deficiencies of the small deflection linear theories applied to highly flexible frame structures.

SECTION II

LARGE DEFLECTION MATRIX DISPLACEMENT ANALYSIS

In order to determine large deflections by the matrix displacement method, we must consider the nonlinear strain-displacement equations for an elastic continuum. These equations in Cartesian coordinates are:

$$
\begin{aligned}
e_{xx} &= \frac{\partial u_x}{\partial x} + \frac{1}{2}\left[\left(\frac{\partial u_x}{\partial x}\right)^2 + \left(\frac{\partial u_y}{\partial x}\right)^2 + \left(\frac{\partial u_z}{\partial x}\right)^2\right] \\
e_{yy} &= \frac{\partial u_y}{\partial y} + \frac{1}{2}\left[\left(\frac{\partial u_x}{\partial y}\right)^2 + \left(\frac{\partial u_y}{\partial y}\right)^2 + \left(\frac{\partial u_z}{\partial y}\right)^2\right] \\
e_{zz} &= \frac{\partial u_z}{\partial z} + \frac{1}{2}\left[\left(\frac{\partial u_x}{\partial z}\right)^2 + \left(\frac{\partial u_y}{\partial z}\right)^2 + \left(\frac{\partial u_z}{\partial z}\right)^2\right] \\
e_{xy} &= \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} + \frac{\partial u_x}{\partial x}\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial x}\frac{\partial u_z}{\partial y} \\
e_{yz} &= \frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z} + \frac{\partial u_x}{\partial y}\frac{\partial u_x}{\partial z} + \frac{\partial u_y}{\partial y}\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\frac{\partial u_z}{\partial z} \\
e_{zx} &= \frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z}\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial z}\frac{\partial u_y}{\partial x} + \frac{\partial u_z}{\partial z}\frac{\partial u_z}{\partial x}
\end{aligned}
\tag{1}
$$

where u_x, u_y, and u_z are the displacements in the x, y, and z directions, respectively. The above strains will be denoted collectively by a column matrix*

$$\underset{\sim}{e} = \left\{e_{xx}\ e_{yy}\ e_{zz}\ e_{xy}\ e_{yz}\ e_{zx}\right\} \tag{2}$$

which may be written as the sum of two matrices such that

$$\underset{\sim}{e} = \underset{\sim}{\hat{e}} + \underset{\sim}{e}' \tag{3}$$

where $\underset{\sim}{\hat{e}}$ represents the linear strains proportional to displacements, while $\underset{\sim}{e}'$ represents the nonlinear strains proportional to the squares of displacements.

* Wavy underscore denotes matrix symbols.

The displacements u_x, u_y, and u_z will be assumed to be expressible by the equation

$$u_i = \underset{\sim}{a}_i \underset{\sim}{u} \quad ; \quad i = x, y, z \tag{4}$$

where

$$\underset{\sim}{u} = \left\{ u_1 \; u_2 \cdots u_n \right\} \tag{5}$$

denotes a column matrix of the discrete displacements (including discrete rotations), and $\underset{\sim}{a}_i$ is a (lxn) rectangular matrix whose coefficients are, in general, functions of x, y, and z. For some structural elements $\underset{\sim}{a}_i$ may be determined exactly while for others assumed displacement distributions must be used to determine $\underset{\sim}{a}_i$. Introducing a new matrix

$$\underset{\sim}{b}_{i,j} = \frac{\partial \underset{\sim}{a}_i}{\partial j} \quad ; \quad i, j = x, y, z \tag{6}$$

it is clear that the nonlinear strain-displacement equations (Equation 1) can be expressed in matrix form as illustrated by Equation 7 where asterisks are introduced to indicate element-by-element matrix multiplication.

$$\underset{\sim}{e} = \begin{bmatrix} \underset{\sim}{b}_{x,x} \\ \underset{\sim}{b}_{y,y} \\ \underset{\sim}{b}_{z,z} \\ \underset{\sim}{b}_{y,x}+\underset{\sim}{b}_{x,y} \\ \underset{\sim}{b}_{z,y}+\underset{\sim}{b}_{y,z} \\ \underset{\sim}{b}_{x,z}+\underset{\sim}{b}_{z,x} \end{bmatrix} \underset{\sim}{u} + \begin{bmatrix} \underset{\sim}{b}_{x,x}/\sqrt{2} \\ \underset{\sim}{b}_{x,y}/\sqrt{2} \\ \underset{\sim}{b}_{x,z}/\sqrt{2} \\ \underset{\sim}{b}_{x,x} \\ \underset{\sim}{b}_{x,y} \\ \underset{\sim}{b}_{x,z} \end{bmatrix} \underset{\sim}{u} * \begin{bmatrix} \underset{\sim}{b}_{x,x}/\sqrt{2} \\ \underset{\sim}{b}_{x,y}/\sqrt{2} \\ \underset{\sim}{b}_{x,z}/\sqrt{2} \\ \underset{\sim}{b}_{x,y} \\ \underset{\sim}{b}_{x,z} \\ \underset{\sim}{b}_{x,x} \end{bmatrix} \underset{\sim}{u} + \begin{bmatrix} \underset{\sim}{b}_{y,x}/\sqrt{2} \\ \underset{\sim}{b}_{y,y}/\sqrt{2} \\ \underset{\sim}{b}_{y,z}/\sqrt{2} \\ \underset{\sim}{b}_{y,x} \\ \underset{\sim}{b}_{y,y} \\ \underset{\sim}{b}_{y,z} \end{bmatrix} \underset{\sim}{u} * \begin{bmatrix} \underset{\sim}{b}_{y,x}/\sqrt{2} \\ \underset{\sim}{b}_{y,y}/\sqrt{2} \\ \underset{\sim}{b}_{y,z}/\sqrt{2} \\ \underset{\sim}{b}_{y,y} \\ \underset{\sim}{b}_{y,z} \\ \underset{\sim}{b}_{y,x} \end{bmatrix} \underset{\sim}{u} + \begin{bmatrix} \underset{\sim}{b}_{z,x}/\sqrt{2} \\ \underset{\sim}{b}_{z,y}/\sqrt{2} \\ \underset{\sim}{b}_{z,z}/\sqrt{2} \\ \underset{\sim}{b}_{z,x} \\ \underset{\sim}{b}_{z,y} \\ \underset{\sim}{b}_{z,z} \end{bmatrix} \underset{\sim}{u} * \begin{bmatrix} \underset{\sim}{b}_{z,x}/\sqrt{2} \\ \underset{\sim}{b}_{z,y}/\sqrt{2} \\ \underset{\sim}{b}_{z,z}/\sqrt{2} \\ \underset{\sim}{b}_{z,y} \\ \underset{\sim}{b}_{z,z} \\ \underset{\sim}{b}_{z,x} \end{bmatrix} \underset{\sim}{u} \tag{7}$$

Symbolically Equation 7 may be expressed as

$$\underset{\sim}{e} = \underset{\sim}{\hat{b}} \underset{\sim}{u} + \sum_i \underset{\sim}{b}_{i1} \, \underset{\sim}{u} * \underset{\sim}{b}_{i2} \underset{\sim}{u} \quad ; \quad i = x, y, z \tag{8}$$

where $\hat{b}$ represents matrix of linear strains due to unit displacements u while b_{i1} and b_{i2} are respectively the first and second column matrices appearing in the asterisk multiplication products of Equation 7. Hence from Equations 3 and 8 it follows that

$$\hat{e} = \hat{b}u \qquad (9)$$

and

$$e' = \sum_i b_{i1}\, u * b_{i2}\, u \; ; \; i = x, y, z \qquad (10)$$

The strains e are related to the stresses $\hat{\sigma}$ through the Hooke's law

$$\sigma = \kappa e \qquad (11)$$

where κ is a matrix of elastic constants. The subsequent analysis requires the determination of the strain energy υ from the expression

$$\upsilon = \frac{1}{2} \int_V e^T \sigma \, dV \qquad (12)$$

Using Equations 3, 11, and 12 it may be shown that neglecting the nonlinear strain product $(e')^T \kappa e'$

$$\upsilon = \frac{1}{2} \int_V (\hat{e}^T \kappa \hat{e} + 2\hat{\sigma}^T e')\, dV \qquad (13)$$

where a new stress matrix $\hat{\sigma}$ denoting linear stress has been introduced. This matrix is obtained from

$$\hat{\sigma} = \kappa \hat{e} \qquad (14)$$

Subsequent substitution of Equations 9 and 10 into the strain energy expression Equation 13 finally leads to

$$\upsilon = \frac{1}{2} \int_V (u^T \hat{b}^T \kappa b u + 2 \sum_{i=x,y,z} \hat{\sigma}^T b_{i1}\, u * b_{i2}\, u)\, dV \qquad (15)$$

From the Castigliano's theorem which is applicable to large deflections provided the strain energy is evaluated for the nonlinear strains, we obtain the force-displacement relation of the form

$$\frac{\partial \upsilon}{\partial \underset{\sim}{u}} = \underset{\sim}{k}\underset{\sim}{u} = \underset{\sim}{S} \tag{16}$$

where $\underset{\sim}{k}$ represents the stiffness matrix, while $\underset{\sim}{S}$ is a column matrix of forces corresponding with the displacements $\underset{\sim}{u}$. Performing matrix differentiation on Equation 15 it can be shown that

$$\underset{\sim}{S} = \frac{\partial \upsilon}{\partial \underset{\sim}{u}} = \int_V \hat{\underset{\sim}{b}}^T \underset{\sim}{\kappa} \hat{\underset{\sim}{b}}\, dV \underset{\sim}{u} + \int_V \sum_{i=x,y,z} (\underset{\sim}{b}_{i1}^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_{i2} + \underset{\sim}{b}_{i2}^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_{i1}) dV \underset{\sim}{u} \tag{17}$$

where $\hat{\underset{\sim}{\sigma}}^D$ represents the column matrix of stresses $\hat{\underset{\sim}{\sigma}}$ changed into a diagonal matrix. In performing the differentiation with respect to $\underset{\sim}{u}$ it has been assumed that the linear stresses $\hat{\underset{\sim}{\sigma}}$ remain constant. Rearranging terms in Equation 17 we obtain

$$\underset{\sim}{S} = (\underset{\sim}{k}_E + \underset{\sim}{k}_G)\underset{\sim}{u} = \underset{\sim}{k}\underset{\sim}{u} \tag{18}$$

where

$$\underset{\sim}{k}_E = \int_V \hat{\underset{\sim}{b}}^T \underset{\sim}{\kappa} \hat{\underset{\sim}{b}}\, dV \text{, elastic stiffness matrix} \tag{19}$$

$$\underset{\sim}{k}_G = \int_V \sum_{i=x,y,z} (\underset{\sim}{b}_{i1}^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_{i2} + \underset{\sim}{b}_{i2}^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_{i1})\, dV \text{, geometrical stiffness matrix} \tag{20}$$

$$\underset{\sim}{k} = \underset{\sim}{k}_E + \underset{\sim}{k}_G \text{, total stiffness matrix} \tag{21}$$

The geometrical stiffness matrix can be simplified if we introduce two new matrices

$$\underset{\sim}{b}_1 = \left\{ \underset{\sim}{b}_{x1}\, \underset{\sim}{b}_{y1}\, \underset{\sim}{b}_{z1} \right\} \tag{22}$$

$$\underset{\sim}{b}_2 = \left\{ \underset{\sim}{b}_{x2}\, \underset{\sim}{b}_{y2}\, \underset{\sim}{b}_{z2} \right\} \tag{23}$$

so that Equation 20 can be written as

$$\underset{\sim}{k}_G = \int_V \left(\underset{\sim}{b}_1^T \hat{\underset{\sim}{\sigma}}_3^D b_2 + \underset{\sim}{b}_2^T \hat{\underset{\sim}{\sigma}}_3^D b_1 \right) dV \tag{24}$$

where

$$\underset{\sim}{\sigma}_3^D = \left[\; \hat{\underset{\sim}{\sigma}}^D \;\; \hat{\underset{\sim}{\sigma}}^D \;\; \hat{\underset{\sim}{\sigma}}^D \;\right] \tag{25}$$

In most practical applications we normally include only one of the nonlinear terms in the strain-displacement equations, the remaining two nonlinear terms rejected as being of higher order of smallness. This is the case for plates and shells where only the out of plane rotations are considered contributing to the nonlinear terms significantly. This implies then that the matrices $\underset{\sim}{b}_1$ and $\underset{\sim}{b}_2$ consist of only one submatrix and $\hat{\underset{\sim}{\sigma}}_3^D$ becomes $\hat{\underset{\sim}{\sigma}}^D$. For these cases the geometrical stiffness matrix $\underset{\sim}{k}_G$ is simply

$$\underset{\sim}{k}_G = \int_V \left(\underset{\sim}{b}_1^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_2 + \underset{\sim}{b}_2^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_1 \right) dV$$

$$= \int_V \left(\underset{\sim}{b}_1^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_2 + \left(\underset{\sim}{b}_1^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_2 \right)^T \right) dV \tag{26}$$

It has been demonstrated that the geometrical stiffness matrix can easily be determined from an integral of simple matrix products evaluated over the volume of the structural element. This new approach avoids the time-consuming determination of the strain energy in terms of displacements and its subsequent differentiation with respect to the displacements, as used in previous methods of determining geometrical stiffness matrices. Furthermore, the present formulation of $\underset{\sim}{k}_G$ allows one to investigate the effects of the other nonlinear terms in the strain-displacement relations by including one, two, or three submatrices in $\underset{\sim}{b}_1$ and $\underset{\sim}{b}_2$. It should also be noted that the present method can be used for strains and displacements in other coordinate systems, e.g. in the analysis of axisymmetrical shells.

The geometrical stiffness matrix $\underset{\sim}{k}_G$ derives its name from the fact that it depends only on the geometry of the element and is independent of any elastic properties of the material. Other names given to this matrix are: incremental stiffness matrix and initial stress matrix; however, the name geometric stiffness appears to be more suitable.

The total stiffness, $\underset{\sim}{k}$ is determined for each structural element, first in local coordinate system and then in the datum system, using a matrix transformation involving the direction cosines. The assembled structure stiffness matrix

$$\underset{\sim}{K} = \underset{\sim}{K}_E + \underset{\sim}{K}_G \tag{27}$$

is obtained from the summation of individual stiffnesses in the datum system. The equations of equilibrium are then formulated as

$$(\underset{\sim}{K}_E + \underset{\sim}{K}_G)\underset{\sim}{U} = \underset{\sim}{P} \tag{28}$$

where $\underset{\sim}{U}$ is a column matrix of node displacements (at the element joints) corresponding to the external forces $\underset{\sim}{P}$. In the subsequent analysis it will be assumed that the rows and columns in Equation 28 corresponding to zero displacements have been eliminated so that the displacements $\underset{\sim}{U}$ can be calculated from

$$\underset{\sim}{U} = (\underset{\sim}{K}_E + \underset{\sim}{K}_G)^{-1}\underset{\sim}{P} \tag{29}$$

For large deflections the solution to Equation 29 is obtained by a step-by-step linear approximation with loading incrementation. A typical nonlinear plot of P versus U and its step-by-step approximation are shown in Figure 2. The incremental displacements $\Delta\underset{\sim}{U}$ and internal (element) forces due to load increment $\Delta\underset{\sim}{P}$ are calculated in the conventional manner from

$$\Delta\underset{\sim}{U} = (\underset{\sim}{K}_E + \underset{\sim}{K}_G)^{-1}\Delta\underset{\sim}{P} \tag{30}$$

where both $\underset{\sim}{K}_E$ and $\underset{\sim}{K}_G$ depend on the deformed geometry in a given step and in addition $\underset{\sim}{K}_G$ depends also on the internal force distribution $\underset{\sim}{S}$ of the previous step. The total displacements for the final values of the applied loading are then obtained by summing the incremental values. The incremental step procedure, as used in this application, is presented symbolically in Table I.

TABLE I

INCREMENTAL STEP PROCEDURE FOR THE LARGE DEFLECTION ANALYSIS

Step No.	Stiffness	Incremental Displacements	Element Forces
1	$\underset{\sim}{K}_E(O) + \underset{\sim}{K}_G(O)$	$\Delta \underset{\sim}{U}$	$\underset{\sim}{S}_1$
2	$\underset{\sim}{K}_E(U_1) + \underset{\sim}{K}_G(U_1)$	$\Delta \underset{\sim}{U}_2$	$\underset{\sim}{S}_2$
.	.	.	.
.	.	.	.
.	.	.	.
n	$\underset{\sim}{K}_E(U_{n-1}) + \underset{\sim}{K}_G(U_{n-1})$	$\Delta \underset{\sim}{U}_n$	$\underset{\sim}{S}_n$

Total displacement $\underset{\sim}{U}_n = \sum_{i=1}^{n} \Delta \underset{\sim}{U}_i$

In the first step $\underset{\sim}{K}_G(O) = 0$ since the geometrical stiffness matrix is proportional to the internal forces which are zero at the start of step 1.

Introducing

$$\underset{\sim}{K}_G = \lambda \underset{\sim}{\overline{K}}_G \tag{31}$$

where $\underset{\sim}{\overline{K}}_G$ is evaluated for some unit values of the external loading and λ is a multiplying factor we note that $\Delta \underset{\sim}{U} \rightarrow \infty$ in Equation 30 when the determinant

$$|\underset{\sim}{K}_E + \lambda \underset{\sim}{\overline{K}}_G| = 0 \tag{32}$$

Equation 32 represents the stability criterion for the displacement method of analysis and the lowest root λ in this equation gives the buckling loading.

SECTION III

ELASTIC AND GEOMETRICAL STIFFNESS MATRICES

1. PIN-JOINTED BAR ELEMENTS

The approximate strain-displacement relationship for a bar element placed in the xz - plane (Figure 3) may be taken as

$$e_{xx} = \frac{\partial u_x}{\partial x} + \frac{1}{2}\left(\frac{\partial u_z}{\partial x}\right)^2$$

$$= \hat{\underset{\sim}{b}}\,\underset{\sim}{u} + \frac{1}{\sqrt{2}}\,\underset{\sim}{b}_{z,x}\,\underset{\sim}{u}^{*}\frac{1}{\sqrt{2}}\,\underset{\sim}{b}_{z,x}\,\underset{\sim}{u} \qquad (33)$$

The displacements u_x and u_z are expressed in terms of the discrete displacements by

$$u_x = \underset{\sim}{a}_x\,\underset{\sim}{u} \qquad (34)$$

$$u_z = \underset{\sim}{a}_z\,\underset{\sim}{u} \qquad (35)$$

where

$$\underset{\sim}{a}_x = \left[(1-\xi)\;0\;\xi\;0\right] \qquad (36)$$

$$\underset{\sim}{a}_z = \left[0\;(1-\xi)\;0\;\xi\right] \qquad (37)$$

$$\underset{\sim}{u} = \left\{u_1\;u_2\;u_3\;u_4\right\} \qquad (38)$$

and

$$\xi = x/d \qquad (39)$$

From Equations 33, 34, and 35 it follows that

$$\hat{\underset{\sim}{b}} = \frac{1}{d}\left[-1\;0\;1\;0\right] \qquad (40)$$

and

$$\underset{\sim}{b}_{z,x} = \frac{\partial a_z}{\partial x} = \frac{1}{d}\left[0\;-1\;0\;1\right] \qquad (41)$$

Hence

$$\underset{\sim}{b}_1 = \underset{\sim}{b}_2 = \frac{1}{\sqrt{2}}\,\underset{\sim}{b}_{z,x} = \frac{1}{\sqrt{2}\,d}\begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix} \tag{42}$$

Noting that

$$\underset{\sim}{\kappa} = E \tag{43}$$

it follows from Equations 19, 40, and 43 that the elastic stiffness matrix

$$\underset{\sim}{k}_E = \int_V \hat{\underset{\sim}{b}}^T \underset{\sim}{\kappa}\, \hat{\underset{\sim}{b}}\, dV = \frac{AE}{d}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{44}$$

where A is the cross-sectional area of the bar. Similarly, noting that

$$\hat{\underset{\sim}{\sigma}}^D = F/A \tag{45}$$

where F represents tensile force in the bar, it follows from Equations 26 and 42 that the geometric stiffness matrix is given by

$$\underset{\sim}{k}_G = \int_V \left(\underset{\sim}{b}_1^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_2 + (\underset{\sim}{b}_1^T \hat{\underset{\sim}{\sigma}}^D \underset{\sim}{b}_2)^T \right) dV$$

$$= \frac{F}{d}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{46}$$

To transform the stiffnesses from the local system with displacements $u_1 \ldots u_4$ into the datum system with displacements $\bar{u}_1 \ldots \bar{u}_4$ as shown in Figure 4 we use

$$\bar{\underset{\sim}{k}} = \underset{\sim}{k}_{datum} = \underset{\sim}{\lambda}^T \underset{\sim}{k}\, \underset{\sim}{\lambda} \tag{47}$$

where

$$\underset{\sim}{\lambda} = \begin{bmatrix} \ell & m & 0 & 0 \\ -m & \ell & 0 & 0 \\ 0 & 0 & \ell & m \\ 0 & 0 & -m & \ell \end{bmatrix} \tag{48}$$

where ℓ and m are the direction cosines for the direction pq with respect to the datum system $o\bar{x}$ and $o\bar{y}$. Hence using the congruent transformation relationship (Equation 47) we obtain elastic and geometric stiffness matrices in the datum system:

$$\underset{\sim}{\bar{k}}_E = \frac{AE}{d} \begin{bmatrix} \ell^2 & & \text{Symmetric} & \\ m\ell & m^2 & & \\ -\ell^2 & -\ell m & \ell^2 & \\ -m\ell & -m^2 & m\ell & m^2 \end{bmatrix} \tag{49}$$

and

$$\underset{\sim}{\bar{k}}_G = \frac{F}{d} \begin{bmatrix} m^2 & & \text{Symmetric} & \\ -\ell m & \ell^2 & & \\ -m^2 & m\ell & m^2 & \\ \ell m & -\ell^2 & -\ell m & \ell^2 \end{bmatrix} \tag{50}$$

2. PINNED-RIGID BEAM ELEMENT

The pinned-rigid beam element is illustrated in Figure 5. Neglecting the effects of shear deformations and retaining only one nonlinear team in the strain-displacement relations we can show that in this case

$$\underset{\sim}{\hat{b}} = \left[\frac{-1}{d} \quad \frac{-3z}{d^2\xi} \quad \frac{1}{d} \quad \frac{3z}{d^2\xi} \quad \frac{3z}{d\xi} \right] \tag{51}$$

$$\underset{\sim}{b}_1 = \underset{\sim}{b}_2 = \frac{1}{\sqrt{2}} \underset{\sim}{b}_{2,x} = \frac{-1}{2\sqrt{2}\,d} \left[0 \quad -3(1-\xi^2) \quad 0 \quad 3(1-\xi^2) \quad (-1+3\xi^2)d \right] \tag{52}$$

Hence from Equations 19 and 20 we obtain

$$\underset{\sim}{k}_E = \frac{EI}{d^3}\begin{bmatrix} c & & \text{Symmetric} & & \\ 0 & 3 & & & \\ -c & 0 & c & & \\ 0 & -3 & 0 & 3 & \\ 0 & 3d & 0 & -3d & 3d^2 \end{bmatrix} \qquad (53)$$

where

$$c = Ad^2/I \qquad (54)$$

and

$$\underset{\sim}{k}_G = \frac{F}{5d}\begin{bmatrix} 0 & & \text{Symmetric} & & \\ 0 & 6 & & & \\ 0 & 0 & 0 & & \\ 0 & -6 & 0 & 6 & \\ 0 & +d & 0 & -d & d^2 \end{bmatrix} \qquad (55)$$

where, as before, F represents the axial load (constant) in the element.

The transformation matrix $\underset{\sim}{\lambda}$ is given by

$$\underset{\sim}{\lambda} = \begin{bmatrix} \ell & m & 0 & 0 & 0 \\ -m & \ell & 0 & 0 & 0 \\ 0 & 0 & \ell & m & 0 \\ 0 & 0 & -m & \ell & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (56)$$

Using this matrix to derive stiffness properties in the datum system we obtain

$$\overline{\underset{\sim}{k}}_E = \frac{EI}{d^3}\begin{bmatrix} c\ell^2-3m^2 & & \text{Symmetric} & & \\ (c-3)\ell m & cm^2+3\ell^2 & & & \\ -c\ell^2-3m^2 & -(c-3)\ell m & c\ell^2+3m^2 & & \\ -(c-3)\ell m & -cm^2-3\ell^2 & (c-3)\ell m & cm^2+3\ell^2 & \\ 3dm & +3d\ell & -3dm & -3d\ell & +3d^2 \end{bmatrix} \qquad (57)$$

$$\bar{\underset{\sim}{k}}_G = \frac{F}{5d}\begin{bmatrix} 6m^2 & & & & \\ -6\ell m & 6\ell^2 & & \text{Symmetric} & \\ -6m^2 & 6\ell m & 6m^2 & & \\ 6\ell m & 6\ell^2 & -6\ell m & 6\ell^2 & \\ dm & d\ell & -dm & -d\ell & d^2 \end{bmatrix} \tag{58}$$

3. BEAM ELEMENT

The beam element is illustrated in Figure 6. For this case we use six element displacements and

$$\hat{\underset{\sim}{b}} = \left[\frac{-1}{d} \;\; \frac{6z}{d^2}(1-2\xi) \;\; \frac{2z}{d}(2-3\xi) \;\; \frac{1}{d} \;\; \frac{6z}{d^2}(-1+2\xi) \;\; \frac{2z}{d}(1-3\xi) \right] \tag{59}$$

$$\underset{\sim}{b}_1 = \underset{\sim}{b}_2 = \frac{1}{\sqrt{2}d}\left[0 \;\; -6(\xi-\xi^2) \;\; (1-4\xi+3\xi^2)d \;\; 0 \;\; -6(\xi-\xi^2) \;\; (-2\xi+3\xi^2)d \right] \tag{60}$$

When matrices (Equations 59 and 60) are used in Equations 19 and 20 we obtain

$$\underset{\sim}{k}_E = \frac{EI}{d^3}\begin{bmatrix} c & & & & & \\ 0 & 12 & & \text{Symmetric} & & \\ 0 & 6d & 4d^2 & & & \\ -c & 0 & 0 & c & & \\ 0 & -12 & -6d & 0 & 12 & \\ 0 & 6d & 2d^2 & 0 & -6d & 4d^2 \end{bmatrix} \tag{61}$$

and

$$\underset{\sim}{k}_G = \frac{F}{30d}\begin{bmatrix} 0 & & & & & \\ 0 & 36 & & \text{Symmetric} & & \\ 0 & 3d & 4d^2 & & & \\ 0 & 0 & 0 & 0 & & \\ 0 & -36 & -3d & 0 & 36 & \\ 0 & 3d & -d^2 & 0 & -3d & 4d^2 \end{bmatrix} \tag{62}$$

The transformation matrix $\underset{\sim}{\lambda}$ is simply

$$\underset{\sim}{\lambda} = \begin{bmatrix} \ell & m & 0 & 0 & 0 & 0 \\ -m & \ell & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ell & m & 0 \\ 0 & 0 & 0 & -m & \ell & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{63}$$

Hence from $\bar{\underset{\sim}{k}}_E = \underset{\sim}{\lambda}^T \underset{\sim}{k}_E \underset{\sim}{\lambda}$ and $\bar{\underset{\sim}{k}}_G = \underset{\sim}{\lambda}^T \underset{\sim}{k}_G \underset{\sim}{\lambda}$ it follows that

$$\bar{\underset{\sim}{k}}_E = \frac{EI}{d^3} \begin{bmatrix} c\ell^2 + 12m^2 & & & & & \\ (c-12)\ell m & cm^2 + 12\ell^2 & & & & \\ -6dm & 6d\ell & 4d^2 & & & \\ -c\ell^2 - 12m^2 & -(c-12)\ell m & 6dm & c\ell^2 + 12m^2 & & \\ -(c-12)\ell m & -cm^2 - 12\ell^2 & -6d\ell & (c-12)\ell m & cm^2 + 12\ell^2 & \\ -6dm & 6d\ell & 2d^2 & 6dm & -6d\ell & 4d^2 \end{bmatrix} \tag{64}$$

and

$$\bar{\underset{\sim}{k}}_G = \frac{F}{30d} \begin{bmatrix} 36m^2 & & & & & \\ -36\ell m & 36\ell^2 & & & & \\ -3dm & 3d\ell & 4d^2 & & & \\ -36m^2 & 36\ell m & 3dm & 36m^2 & & \\ 36\ell m & -36\ell^2 & -3d\ell & -36\ell m & 36\ell^2 & \\ -3dm & 3d\ell & -d^2 & 3dm & -3d\ell & 4d^2 \end{bmatrix} \tag{65}$$

SECTION IV

SAMPLE PROBLEMS

Numerical results are presented for sample problems selected to demonstrate the capabilities of the computer program and the accuracy of the technique. The solutions obtained for the sample problems are presented in the form of load-deflection curves.

Problems 1 and 2 introduce the basic capabilities of the program in the analysis of frame structures which undergo large displacements (small strains), while problems 3 and 4 demonstrate the use of the program in predicting the initial buckling load of frame structures. Several of the problems chosen had been solved previously using other techniques. Some of these previous solutions are presented to demonstrate the accuracy of the technique.

1. PROBLEM 1 (SQUARE FRAME)

The structure shown in Figure 7 is a flexible welded square steel frame. The cross-sectional dimensions of the members were 1.0 X 0.0625 inches and the joints were rigid. Because of the double symmetry, only one quadrant of the structure was required for the analysis. Several sets of solutions were obtained using different loading increments and a different number of elements. These solutions were then compared with an exact solution from Reference 17 and an approximate 1 element solution given in Reference 18.

Figures 8 and 9 demonstrate the solutions obtained using one element and loading increments of 10, 5, 2, and 1 pounds. The solutions were improved by increasing the number of loading cycles, however the use of a loading increment smaller than one pound had only a small effect on the solutions.

The fact that the solutions do not converge on the exact solution is explained by considering that the chosen deflection shapes may not be exact and that certain higher order terms have been neglected.

Solutions were also obtained using a loading increment of ten pounds and 1, 3, 5, and 10 elements. These solutions presented in Figures 10 and 11, show that the solutions were improved by increasing the number of elements.

2. PROBLEM 2 (SIX-BAY FRAME)

The program was designed to handle structures with up to fifty elements. An eighteen element six-bay steel frame, shown in Figure 12, was analyzed to demonstrate the ability of the program to solve more complex structures. The cross-sectional dimensions of the elements were again 1.0 X 0.0625 inches.

Figure 13 shows the vertical and horizontal deflections obtained at the free end of the frame when a vertical load was applied at the end. The deflections were carried well into the nonlinear range which began to appear at a load of about 40 or 50 pounds.

3. PROBLEM 3 (COLUMN BUCKLING)

The column of Figure 14 was analyzed using the eigenvalue option of the program. One element was used to represent the actual member. The program predicted a buckling load of

$$P_{crit} = 9.92 \frac{EI}{\ell^2}$$

This compares with the theoretical solution of

$$P_{crit} = 9.86 \frac{EI}{\ell^2}$$

4. PROBLEM 4 (COLUMN BUCKLING UNDER GRAVITY LOADING)

The buckling of the column shown in Figure 15 was studied using the eigenvalue option of the program. The column was subjected to a uniformly distributed axial load. Solutions were obtained by using several elements to represent the column and by applying the appropriate loads at the nodes. In this way the uniform load was replaced with a set of point loads distributed along the column.

When five elements were used the predicted buckling load was

$$q_{crit} = 7.59 \frac{EI}{\ell^3}$$

The number of elements was increased to ten and the predicted buckling load was

$$q_{crit} = 7.56 \frac{EI}{\ell^3}$$

These solutions compare quite favorably with the solution

$$q_{crit} = 7.53 \frac{EI}{\ell^3}$$

obtained from Reference 19.

REFERENCES

1. M. J. Turner, E. H. Dill, H. C. Martin, and R. J. Melosh, Large Deflections of Structures Subjected to Heating and External Loads, J. Aerospace Sciences, 27, 97-102 (1960).

2. W. P. Rodden, J. P. Jones, and P. G. Bhuta, A Matrix Formulation of the Transverse Structural Influence Coefficients of an Axially Loaded Timoshenko Beam, J. Am. Inst. Aero. Astro., 1, 225-227 (1963)

3. R. H. Gallagher and J. Padlog, Discrete Element Approach to Structural Instability Analysis, J. Am. Inst. Aero. Astro., 1, 1437-1439 (1963).

4. J. S. Archer, Consistent Matrix Formulations for Structural Analysis Using Finite-Element Techniques, J. Am. Inst. Aero. Astro., 3, 1910-1918 (1965).

5. H. C. Martin, Large Deflection and Stability Analysis by the Direct Stiffness Method, NASA Technical Report No. 32-931, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 1966.

6. S. J. McMinn, The Effect of Axial Loads on the Stiffness of Rigid-Jointed Plane Frames, Proc. Conf. on Matrix Methods in Structural Mechanics, 26-28 Oct 1965, Wright-Patterson Air Force Base, Ohio, AFFDL-TR-66-80, (1966).

7. J. T. Oden, Calculation of Geometric Stiffness Matrices for Complex Structures, J. Am. Inst. Aero. Astro., 4, 1480-1482 (1966).

8. J. S. Przemieniecki, "Theory of Matrix Structural Analysis," McGraw-Hill Book Company, Inc., 1968.

9. M. J. Turner, H. C. Martin, and R. C. Weikel, Further Development and Application of the Stiffness Method, AGARDograph 72, "Matrix Methods of Structural Analysis," 203-266, The Macmillan Company, New York, 1964.

10. H. C. Martin, On the Derivation of Stiffness Matrices for the Analysis of Large Deflection and Stability Problems, Proc. Conf. on Matrix Methods in Structural Mechanics, 26-28 Oct 1965, Wright-Patterson Air Force Base, Ohio, AFFDL-TR-66-80, (1966).

11. K. K. Kapur and B. J. Hartz, "Stability of Plates Using the Finite Element Method", J. Eng. Mech. Div. Am. Soc. Civil Engineers, 92, 177-195 (1966).

12. R. H. Gallagher, R. A. Gellatly, J. Padlog, and R. H. Mallet, A Discrete Element Procedure for Thin-Shell Instability Analysis, J. Am. Inst. Aero. Astro., 5, 138-145 (1967).

13. C. D. Wallace, "Matrix Analysis of Axisymmetric Shells Under General Loading," MS Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 1966.

REFERENCES (CONT)

14. J. H. Argyris, "Recent Advances in Matrix Methods of Structural Analysis," Progress in Aeronautical Sciences, 4, The Macmillan Company, New York, 1964, also in "Matrix Methods of Structural Analysis," edited by B. Fraeijs de Veubeke, AGARDograph 72, 1-164, The Macmillan Company, New York, 1964.

15. J. H. Argyris, Matrix Analysis of Three-Dimensional Elastic Media Small and Large Deflections, J. Am. Inst. Aero. Astro., 3, 45-51, (1965).

16. J. H. Argyris, Continua and Discontinua, Proc. Conf. on Matrix Methods in Structural Mechanics, 26-28 Oct 1965, Wright-Patterson Air Force Base, Ohio, AFFDL-TR-66-80 (1966).

17. J. A. Jenkins, T. B. Seitz, and J. S. Przemieniecki, Large Deflection Analysis of Diamond-Shaped Frames, International Journal of Solids and Structures, 2, 591-603, (1966).

18. R. H. Mallett and L. Berke, Automated Method for the Large Deflection and Instability Analysis of Three-Dimensional Truss and Frame Assemblies, Air Force Flight Dynamics Laboratory Report, AFFDL-TR-66-102, 1966.

19. J. S. Przemieniecki, Struts with Linearly Varying Axial Loading, The Aeronautical Quarterly, 11, 71-98 (1960).

APPENDIX I

INSTRUCTIONS FOR PREPARATION OF INPUT DATA

APPENDIX I

INSTRUCTIONS FOR PREPARATION OF INPUT DATA

The preparation of the data cards is discussed in the following paragraphs. There are four distinct types of cards required for the data deck. The first type introduces the problem, the second and third type give the nodal and element data, and the fourth type closes the problem.

a. CARD SET NO. 1: Problem Identification Cards (one card required)

I	NS	NR	NOELE	NLS		IGN	KFP		TITLE	
1	2	6	11	16	21	25	26	27	33	80

FORMAT (I1, I4, 3I5, 4X, 2I1, 6X, 8A6)

- I — Card type: Enter the integer 1.
- NS — Number of pinned nodes.
- NR — Number of rigid nodes.
- NOELE — Number of elements.
- NLS — Number of loading steps.
- IGN — Buckling option: Enter the integer 1 if a buckling solution is desired on the integer, 0 if it is not.
- KFP — Stiffness matrix option: Enter the integer 1 if a printout of the total stiffness matrix is desired or the integer 0 if it is not.
- TITLE — Alphermic page heading.

b. CARD SET NO. 2: Nodal Data (one card required for each node).

I	NN		I1	I2	I3	X	Y	XL	YL	ML
1	2	6	8	9	10	11	21	31	41	51

FORMAT (I1, I4, 2X, 3I1, 5D10.0)

I	Card type:	Enter the integer 2.
NN	Node number:	The pinned nodes must be numbered first.
I1	Nodel constraint:	Enter the integer 1 if the node is to be restrained from movement in the x_o direction.
I2	Nodal constraint:	Enter the integer 1 if the node is to be restrained from movement in the y_o direction.
I3	Nodal constraint:	Enter the integer 1 if the node is to be restrained from rotation.
X	Node location:	x-coordinate.
Y	Node location:	y-coordinate.
XL	Incremental load in the x_o direction.	
YL	Incremental load in the y_o direction.	
ML	Incremental moment.	

c. CARD SET NO. 3: Element Data (One card required for each element).

I	EN	P	Q	ELET	E	A	I
1	2	11	21	~~31~~ 40	41	51	61

FORMAT (I1, I4, 5X, 6D10.0)

I	Card type:	Enter the integer 3.
EN	Element number.	
P	Node number. (right justify)	
Q	Node number:	The node number P must be a smaller number than the node number Q. (right justify)
ELET	Element type:	Enter the integer 1 if both P and Q are pinned nodes; enter the integer 2 if P is a pinned node and Q is a rigid node; or enter the integer 3 if both P and Q are rigid nodes.

E Modulus of elasticity.

A Cross sectional area.

I Moment of inertia.

d. CARD SET NO. 4: Problem Closing Card (one card required).

I

1 2

FORMAT (I1)

I Card type: Enter the integer 9.

APPENDIX II

COMPUTER PROGRAM LISTING (FORTRAN IV)

```
$IBFTC
$IBFTC FRAME    DECK                                                    FRAM0000
C                                                                       FRAM0001
C        LARGE DEFLECTICN FRAME ANALYSIS FORTRAN IV COMPUTER PROGRAM    FRAM0002
C                                                                       FRAM0003
      DOUBLE PRECISION  KF, KE(21), KG(21), XY VECT(70), P VECT,        FRAM0004
     1 E ARRAY(50), A ARRAY(50), I ARRAY(50), L, L SQD, M, M SQD,       FRAM0005
     2 MOMENT, TS, DELTA U, U(70), XQ, XP, YQ, YP, E, A, D, F,          FRAM0006
     3 D SQD, C, DF, HF(50), HL(50), HM(50)                             FRAM0007
      INTEGER  P, P ARRAY(50), Q, Q ARRAY(50), EL TYPE, TYP ARR(50),    FRAM0008
     1 CN VECT(70), SIG DIG                                             FRAM0009
      COMMON  KF(2485), P VECT(140), DELTA U(140), NO ROWS, EVEN1,      FRAM0010
     1 TS(10), P, Q, XQ, XP, YQ, YP, E, A, MOMENT, NS, EL TYPE          FRAM0011
      DIMENSION  TITLE(8), ROUTE(3), STD RT(3), EGNV RT(3)              FRAM0012
      DATA  STD RT(1)/18H(STANDARD ROUTE)  /, EGNV RT(1)/18H(EIGENVALUE FRAM0013
     1ROUTE)/                                                           FRAM0014
   10 FORMAT  (I1, I4, 3I5, 4X, 2I1, 6X, 8A6)                           FRAM0015
   20 FORMAT  (I1, I4, 2X, 3I1, 6D10.0)                                 FRAM0016
   30 FORMAT  (1H8/ 1H8/ 31X66HDECKS OUT OF ORCER - THIS SHOULD HAVE BEEFRAM0017
     1N A TYPE 1 DECK.  (TYPE =, I2, 1H)/ 1H8/ 1H8)                     FRAM0018
   40 FORMAT  (1H8/ 1H8/ I45, 39H ROWS REQUESTED.  (MAX NUMBER OF ROWS =FRAM0019
     1, I4, 1H)/ 1H8/ 1H8)                                              FRAM0020
   50 FORMAT  (1H8/ 1H8/ I51, 28H ELEMENTS REQUESTED.  (MAX =,          FRAM0021
     1  I3, 1H)/ 1H8/ 1H8)                                              FRAM0022
   60 FORMAT  (1H8/ 1H8/ 53X25HTYPE 1 CARD OUT OF ORDER.)               FRAM0023
   70 FORMAT  (1H8/ 1H8/ 57X18HILLEGAL CARD TYPE.)                      FRAM0024
   80 FORMAT  (1HA, 23X12HBAD CARD = (, I1, I4, 2X, 3I1, 1P6D10.3,      FRAM0025
     1  1H)/ 1H8/ 1H8)                                                  FRAM0026
   90 FORMAT  (1H8/ 1H8/ I51, 27H IS AN ILLEGAL NODE NUMBER.)           FRAM0027
  100 FORMAT  (1H8/ 1H8/ I50, 30H IS AN ILLEGAL ELEMENT NUMBER.)        FRAM0028
  110 FORMAT  (1H8/ 1H8/ 47X37HILLEGAL NODE NUMBER ON A TYPE 3 CARD.)   FRAM0029
  120 FORMAT  (1H8/ 1H8/ 1H8/ 35X37H(DELTA U IS ACCURATE TO APPROXIMATELFRAM0030
     1Y, I3, 20H SIGNIFICANT DIGITS))                                   FRAM0031
  130 FORMAT  (1H1/ 15X33HSTRUCTURE DEFINITION - INPUT DATA, 20X,       FRAM0032
     1 8A6//// I3, 18H PIN-JOINTED NODES, I7, 12H RIGID NODES, I7,      FRAM0033
     2 9H ELEMENTS, I7, 14H LOADING STEPS, 10X3A6///                    FRAM0034
     3 52H ELEMENT      TYPE      NODE NUMBER      YOUNGS MODULUS, 11X  FRAM0035
     4 4HAREA, 14X9HMOMENT OF/ 7H NUMBER, 16X9H(P)    (Q), 11X3H(E),    FRAM0036
     5 36X7HINERTIA//)                                                  FRAM0037
  140 FORMAT  (I5, I11, I9, I6, 1P3D20.5)                               FRAM0038
  150 FORMAT  (1H1/ 15X33HSTRUCTURE DEFINITION - INPUT DATA, 20X,       FRAM0039
     1 8A6/// 6H  NODE, 13X1HX, 16X1HY, 14X4HP(X), 13X4HP(Y), 15X1HM,   FRAM0040
     2 10X11HCONSTRAINTS/ 7H NUMBER, 92X9HX     Y     Z//)              FRAM0041
  160 FORMAT  (I5, 3X, 1P4D17.5, 20X, 2(3X, I1))                        FRAM0042
  170 FORMAT  (1X, I4, 3X, 1P5D17.5, 3X, 3(3X, I1))                     FRAM0043
  180 FORMAT  (1H1/ 18X32H(KF MATRIX - BEFORE CONSTRAINTS), 20X, 8A6////FRAM0044
     1 / 6HO(   1), 5X1PD15.5/ 6HO(   2), 5X2D15.5/ 6HO(   3), 5X3D15.5/ FRAM0045
     2  6HO(   4), 5X4D15.5/ 6HO(   5), 5X5D15.5/ 6HO(   6), 5X6D15.5/  FRAM0046
     3  6HO(   7), 5X7D15.5)                                            FRAM0047
  190 FORMAT  (2HO(, I3, 1H), 5X, 1P8D15.5)                             FRAM0048
  200 FORMAT  (1H1/ 15X12HLOADING STEP, I3, 3H OF, I3, 7H STEPS.,       FRAM0049
     1 20X, 8A6/// 6H  NODE, 10X1HX, 14X1HY, 13X7HDELTA U, 8X7HDELTA U,FRAM0050
     2 8X7HDELTA U, 12X4HU(1), 11X4HU(2), 11X4HU(3)/ 7H NUMBER, 40X     FRAM0051
     3 3H(1), 12X3H(2), 12X3H(3)//)                                     FRAM0052
  210 FORMAT  (I5, 2X, 1P2D15.4, 2X, 2D15.4, 17X, 2D15.4)               FRAM0053
  220 FORMAT  (I5, 2X, 1P2D15.4, 2X, 3D15.4, 2X, 3D15.4)                FRAM0054
  230 FORMAT  (1H8/ 1H8/ 41X49H****AN OVERFLOW HAS OCCURRED DURING THIS FRAM0055
     1CASE****)                                                         FRAM0056
C                                                                       FRAM0057
C  FIRST EXECUTABLE STATEMENT.                                          FRAM0058
C                                                                       FRAM0059
      MAX ELE = 50                                                      FRAM0060
      MX ROWS = 70                                                      FRAM0061
  240 READ  (5,10)  I TYPE, NS, NR, NO ELE, NLS, IGN IND, KFP IND, TITLEFRAM0062
      IF  (I TYPE .EQ. 1)  GO TO 260                                    FRAM0063
C                                                                       FRAM0064
```

```
C  NOT A TYPE 1 DECK.
C
      WRITE  (6,30)   I TYPE
  250 CALL EXEM
  260 NO ROWS = 2*NS + 3*NR
      IF   (NO ROWS)   280,280,270
  270 IF   (NO ROWS .LE. MX ROWS)   GO TO 290
C
C  TOO MANY ROWS IN MATRIX.
C
  280 WRITE  (6,40)   NO ROWS, MX ROWS
      GO TO 250
  290 IF   (NO ELE)   310,310,300
  300 IF   (NO ELE .LE. MAX ELE)   GO TO 320
C
C  TOO MANY ELEMENTS.
C
  310 WRITE  (6,50)   NO ELE, MAX ELE
      GO TO 250
C
C  CLEAR THE U VECTOR.
C
  320 NO CELS = (NO ROWS*(NO ROWS+1))/ 2
      DO 330   I=1,NO ROWS
  330 U(I) = 0.0
C
C  CLEAR THE NODAL DATA.
C
      DO 340   I=1,NO ROWS
      XY VECT(I) = 0.0
      P  VECT(I) = 0.0
  340 CN VECT(I) = 0
C
C  CLEAR THE ELEMENT DATA.
C
      DO 350   I=1,NO ELE
      HL(I) = 0.0
      HM(I) = 0.0
      FF(I) = 0.0
      P ARRAY(I) = 0
      Q ARRAY(I) = 0
      TYP ARR(I) = 0
      L ARRAY(I) = 0.0
      A ARRAY(I) = 0.0
  350 I ARRAY(I) = 0.0
C
C  READ THE DATA.
C
  360 READ  (5,20)   I TYPE, I1, I2, I3, I4, (TS(I), I=1,6)
      GO TO (370,400,450,380,380,380,380,380,570), I TYPE
C  TYPE 1 CARD OUT OF ORDER.
C
C
  370 WRITE  (6,60)
      GO TO 390
C
C  ILLEGAL CARD TYPE.
C
  380 WRITE  (6,70)
C
C  PRINT THE OFFENDING CARD.
C
  390 WRITE  (6,80)   I TYPE, I1, I2, I3, I4, (TS(I), I=1,6)
      GO TO 250
C
C  TYPE 2 CARD  (NODAL DATA).
```

```
C
  400 NO NDS = NS + NR
      IF  (I1)  420,420,410
  410 IF  (I1 .LE. NO NDS)  GO TO 430
C
C  ILLEGAL NODE NUMBER.
C
  420 WRITE  (6,90)  I1
      GO TO 390
  430 IF  (I1 .GT. NS)  GO TO 440
C
C  PIN-JOINTED NODES.
C
      IND2 = 2*I1
      IND1 = IND2 - 1
      XY VECT(IND1) = TS(1)
      XY VECT(IND2) = TS(2)
      P VECT(IND1) = TS(3)
      P VECT(IND2) = TS(4)
      CN VECT(IND1) = I2
      CN VECT(IND2) = I3
      GO TO 360
C
C  RIGID NODES.
C
  440 I1 = 2*NS + 3*(I1 - NS)
      IND1 = I1 - 2
      IND2 = I1 - 1
      XY VECT(IND1) = TS(1)
      XY VECT(IND2) = TS(2)
      P VECT(IND1) = TS(3)
      P VECT(IND2) = TS(4)
      P VECT(I1)    = TS(5)
      CN VECT(IND1) = I2
      CN VECT(IND2) = I3
      CN VECT(I1)   = I4
      GO TO 360
C
C  TYPE 3 CARD  (ELEMENT DATA).
C
  450 IF  (I1)  470,470,460
  460 IF  (I1 .LE. NO ELE)  GO TO 480
C
C  ILLEGAL ELEMENT NUMBER.
C
  470 WRITE  (6,100)  I1
      GO TO 390
C
C  TEST P AND Q.
C
  480 P = TS(1)
      IF  (P .LE. 0)  GO TO 490
      IF  (P .GT. NO NDS)  GO TO 490
      Q = TS(2)
      IF  (Q .LE. 0)  GO TO 490
      IF  (Q .LE. NO NDS)  GO TO 500
C
C  ILLEGAL NODE NUMBER ON A TYPE 3 CARD.
C
  490 WRITE  (6,110)
      GO TO 390
  500 IF  (P - Q)  520,490,510
C
C  RE-ORDER NODE REFERENCES  (P MUST BE LESS THAN Q).
C
```

```
  510 P = TS(2)
      Q = TS(1)
  520 EL TYPE = TS(3)
C
C  TEST FOR LEGAL NODE REFERENCES.
C
      GO TO (530,540,550), EL TYPE
C
C  TYPE 1 ELEMENT.
C
  530 IF  (P .GT. NS)   GO TO 490
      IF  (Q .GT. NS)   GO TO 490
      GO TO 560
C
C  TYPE 2 ELEMENT.
C
  540 IF  (P .GT. NS)   GO TO 490
      IF  (Q .LE. NS)   GO TO 490
      GO TO 560
C
C  TYPE 3 ELEMENT.
C
  550 IF  (P .LE. NS)  GO TO 490
      IF  (Q .LE. NS)  GO TO 490
  560 P ARRAY(I1) = P
      Q ARRAY(I1) = Q
      TYP ARR(I1) = EL TYPE
      E ARRAY(I1) = TS(4)
      A ARRAY(I1) = TS(5)
      I ARRAY(I1) = TS(6)
      GO TO 360
C
C  BEGIN COMPUTATION.
C
  570 IF  (IGN IND .EQ. 0)  GO TO 590
C
C  SET TO PRINT EIGENVALUE ROUTE.
C
      DO 580  I=1,3
  580 ROUTE(I) = EGNV RT(I)
      NLS = 2
      GO TO 610
C
C  SET TO PRINT STANDARD ROUTE.
C
  590 DO 600  I=1,3
  600 ROUTE(I) = STD RT(I)
C
C  PRINT INITIAL CONDITIONS.
C
  610 WRITE  (6,130)  TITLE, NS, NR, NO ELE, NLS, ROUTE
      WRITE  (6,140)  (I, TYP ARR(I), P ARRAY(I), Q ARRAY(I),
     1   E ARRAY(I), A ARRAY(I), I ARRAY(I),  I=1,NO ELE)
      WRITE  (6,150)  TITLE
C
C  PRINT THE PIN-JOINTED NODES.
C
      IF(NS) 640,640,620
  620 DO 630 I=1,NS
      IND2 = 2*I
      IND1 = IND2 - 1
  630 WRITE  (6,160)  I, XY VECT(IND1), XY VECT(IND2), P VECT(IND1),
     1   P VECT(IND2), CN VECT(IND1), CN VECT(IND2)
C
C  PRINT THE RIGID NODES.
C
```

```
      IF(NR .EQ. 0) GO TO 660
      WRITE  (6,170)
  640 DO 650 I=1,NR
      I1 = NS + I
      I2 = 2*NS + 3*I
      IND1 = I2 - 2
      IND2 = I2 - 1
  650 WRITE  (6,170)  I1, XY VECT(IND1), XY VECT(IND2), P VECT(IND1),
     1   P VECT(IND2), P VECT(I2), CN VECT(IND1), CN VECT(IND2),
     2   CN VECT(I2)
  660 CONTINUE
C
C  MAIN LOOP.
C
      DO 1030  MAIN LP=1,NLS
C
C  CLEAR THE KF MATRIX.
C
      DO 670  I=1,NO CELS
  670 KF(I) = 0.0
C
C  PROCESS THE ELEMENTS.
C
      DO 870  I=1,NO ELE
      P = P ARRAY(I)
      Q = Q ARRAY(I)
      IF  (P .GT. NS)  GO TO 680
C
C  P SPECIFIES A PIN-JOINTED NODE.
C
      IND2 = 2*P
      GO TO 690
C
C  P SPECIFIES A RIGID NODE.
C
  680 I1 = 2*NS + 3*(P - NS)
      IND2 = I1 - 1
  690 IND1 = IND2 - 1
      XP = XY VECT(IND1)
      YP = XY VECT(IND2)
      IF  (Q .GT. NS)  GO TO 700
C
C  Q SPECIFIES A PIN-JOINTED NODE.
C
      IND2 = 2*Q
      GO TO 710
C
C  Q SPECIFIES A RIGID NODE.
C
  700 I1 = 2*NS + 3*(Q - NS)
      IND2 = I1 - 1
  710 IND1 = IND2 - 1
      XQ = XY VECT(IND1)
      YQ = XY VECT(IND2)
      EL TYPE = TYP ARR(I)
      E = E ARRAY(I)
      A = A ARRAY(I)
      MOMENT = I ARRAY(I)
      L=XP-XQ
      M=YP-YQ
      D = DSQRT(L**2 + M**2)
      L = L/ D
      M = M/ D
      L SQD = L**2
      M SQD = M**2
      TS(1) = L*M
```

```
      TS(2) = A*E/ D
      GO TC (720,730,740), EL TYPE
  720 IND1 = 2*Q
      IND2 = 2*P
      GO TC 750
  730 IND1 = 3*Q - NS - 1
      IND2 = 2*P
      GO TC 750
  740 IND1 = 3*Q - NS - 1
      IND2 = 3*P - NS - 1
  750 DF=TS(2)*(HL(I)*(DELTA U(IND2-1)-DELTA U(IND1-1))+HM(I)*(
     1DELTA U(IND2)-DELTA U(IND1)))
      HF(I)=HF(I)+DF
      F=HF(I)
      HL(I)=L
      HM(I)=M
      IF  (EL TYPE .NE. 1)  GO TC 760
C
C  TYPE 1 ELEMENT.
C
      KE(1) = TS(2)*L SQD
      KE(2) = TS(2)*TS(1)
      KE(3) = TS(2)*M SQD
      KE(4) = - KE(1)
      KE(5) = - KE(2)
      KE(6) = KE(1)
      KE(7) = KE(5)
      KE(8) = - KE(3)
      KE(9) = KE(2)
      KE(10) = KE(3)
      IF  (F .EQ. 0.0)  GO TO 790
      TS(3) = F/ D
      KG(1) = TS(3)*M SQD
      KG(2) = - TS(3)*TS(1)
      KG(3) = TS(3)*L SQD
      KG(4) = - KG(1)
      KG(5) = - KG(2)
      KG(6) = KG(1)
      KG(7) = KG(5)
      KG(8) = - KG(3)
      KG(9) = KG(2)
      KG(10) = KG(3)
      GO TC 810
  760 D SQD = D**2
      C = A*D SQD/ MOMENT
      TS(4) = 3.0*D
      TS(5) = TS(2)/ C
      I1 = EL TYPE - 1
      GO TO (770,780), I1
C
C  TYPE 2 ELEMENT.
C
  770 KE(1) = TS(5)*(C*L**2 + 3.0*M SQD)
      KE(2) = TS(5)*(C - 3.0)*TS(1)
      KE(3) = TS(5)*(C*M SQD + 3.0*L SQD)
      KE(4) = - KE(1)
      KE(5) = - KE(2)
      KE(6) = KE(1)
      KE(7) = KE(5)
      KE(8) = - KE(3)
      KE(9) = KE(2)
      KE(10) = KE(3)
      KE(11) = -TS(5)*TS(4)*M
      KE(12) = TS(5)*TS(4)*L
      KE(13) = - KE(11)
      KE(14) = - KE(12)
```

```
      KE(15) = TS(5)*3.0*D SQD
      IF  (F .EQ. 0.0)  GO TO 790
      TS(6) = F/ (10.0*D)
      TS(7) = 2.0*D
      KG(1) = TS(6)*12.0*M SQD
      KG(2) = - TS(6)*12.0*TS(1)
      KG(3) = TS(6)*12.0*L SQD
      KG(4) = - KG(1)
      KG(5) = - KG(2)
      KG(6) = KG(1)
      KG(7) = KG(5)
      KG(8) = - KG(3)
      KG(9) = KG(2)
      KG(10) = KG(3)
      KG(11) = TS(6)*TS(7)*M
      KG(12) = - TS(6)*TS(7)*L
      KG(13) = - KG(11)
      KG(14) = - KG(12)
      KG(15) = TS(6)*2.0*D SQD
      GO TO 810
C
C  TYPE 3 ELEMENT.
C
  780 TS(8) = 6.0*D
      TS(9) = 4.0*D SQD
      KE(1) = TS(5)*(C*L SQD + 12.0*M SQD)
      KE(2) = TS(5)*(C - 12.0)*TS(1)
      KE(3) = TS(5)*(C*M SQD + 12.0*L SQD)
      KE(4) = - TS(5)*TS(8)*M
      KE(5) = TS(5)*TS(8)*L
      KE(6) = TS(5)*TS(9)
      KE(7) = - KE(1)
      KE(8) = - KE(2)
      KE(9) = - KE(4)
      KE(10) = KE(1)
      KE(11) = KE(8)
      KE(12) = - KE(3)
      KE(13) = - KE(5)
      KE(14) = KE(2)
      KE(15) = KE(3)
      KE(16) = KE(4)
      KE(17) = KE(5)
      KE(18) = TS(5)*2.0*D SQD
      KE(19) = KE(9)
      KE(20) = KE(13)
      KE(21) = KE(6)
      IF  (F .EQ. 0.0)  GO TO 790
      TS(10) = F/ (30.0*D)
      KG(1) = TS(10)*36.0*M SQD
      KG(2) = - TS(10)*36.0*TS(1)
      KG(3) = TS(10)*36.0*L SQD
      KG(4) = - TS(10)*TS(4)*M
      KG(5) = TS(10)*TS(4)*L
      KG(6) = TS(10)*TS(9)
      KG(7) = - KG(1)
      KG(8) = - KG(2)
      KG(9) = - KG(4)
      KG(10) = KG(1)
      KG(11) = KG(8)
      KG(12) = - KG(3)
      KG(13) = - KG(5)
      KG(14) = KG(2)
      KG(15) = KG(3)
      KG(16) = KG(4)
      KG(17) = KG(5)
      KG(18) = - TS(10)*D SQD
```

```
      KG(19) = KG(9)                                                    FRAM0460
      KG(20) = KG(13)                                                   FRAM0461
      KG(21) = KG(6)                                                    FRAM0462
      GO TO 810                                                         FRAM0463
C                                                                       FRAM0464
C  F = 0.                                                               FRAM0465
C                                                                       FRAM0466
  790 DO 800  I1=1,21                                                   FRAM0467
  800 KG(I1) = 0.0                                                      FRAM0468
  810 CONTINUE                                                          FRAM0469
C                                                                       FRAM0470
C  BUILD KE BAR.                                                        FRAM0471
C                                                                       FRAM0472
      CALL BUILD(KE)                                                    FRAM0473
C                                                                       FRAM0474
C  KG = 0 FOR THE FIRST LOADING STEP.                                   FRAM0475
C                                                                       FRAM0476
      IF  (MAIN LP .EQ. 1)  GO TO 870                                   FRAM0477
      IF  (IGN IND .EQ. 0)  GO TO 860                                   FRAM0478
C                                                                       FRAM0479
C  EIGENVALUE ROUTE  - SAVE KG ON 15.                                   FRAM0480
C                                                                       FRAM0481
      GO TO (820,830,840), EL TYPE                                      FRAM0482
  820 IND1 = 10                                                         FRAM0483
      GO TO 850                                                         FRAM0484
  830 IND1 = 15                                                         FRAM0485
      GO TO 850                                                         FRAM0486
  840 IND1 = 21                                                         FRAM0487
  850 WRITE  (15)  (KG(IND2), IND2=1,IND1)                              FRAM0488
      GO TO 870                                                         FRAM0489
C                                                                       FRAM0490
C  STANDARD ROUTE - ADD IN THE KG MATRIX.                               FRAM0491
C                                                                       FRAM0492
  860 CALL BUILD(KG)                                                    FRAM0493
C                                                                       FRAM0494
C  END OF ELEMENT LOOP.                                                 FRAM0495
C                                                                       FRAM0496
  870 CONTINUE                                                          FRAM0497
      IF  (IGN IND .NE. 0)  GO TO 1040                                  FRAM0498
C                                                                       FRAM0499
C  STANDARD ROUTE.                                                      FRAM0500
C                                                                       FRAM0501
  880 IF  (KEP IND .EQ. 0)  GO TO 900                                   FRAM0502
C                                                                       FRAM0503
C  OPTIONAL PRINT OF KF MATRIX  (BAND).                                 FRAM0504
C                                                                       FRAM0505
      I2 = 28                                                           FRAM0506
      WRITE  (6,180)  TITLE, (KF(I3), I3=1,I2)                          FRAM0507
C                                                                       FRAM0508
C  I4 = ROW NUMBER                                                      FRAM0509
C                                                                       FRAM0510
      DO 890  I4=8,NO ROWS                                              FRAM0511
      I2 = I2 + I4                                                      FRAM0512
      I1 = I2 - 7                                                       FRAM0513
  890 WRITE  (6,190)  I4, (KF(I3), I3=I1,I2)                            FRAM0514
C                                                                       FRAM0515
C  INSERT THE CONSTRAINTS.                                              FRAM0516
C                                                                       FRAM0517
  900 DO 910  I=1,NO ROWS                                               FRAM0518
      IF  (CN VECT(I) .EQ. 0)  GO TO 910                                FRAM0519
      CALL SET(I)                                                       FRAM0520
  910 CONTINUE                                                          FRAM0521
      CALL MTXEQ(SIG DIG)                                               FRAM0522
      DO 920  I=1,NO ROWS                                               FRAM0523
  920 U(I) = U(I) + DELTA U(I)                                          FRAM0524
C                                                                       FRAM0525
```

```
C  UPDATE THE XY VECTOR.
C
      I1 = 2*NS
      IF(NS)  950,950,930
  930 DO 940 I=1,I1
  940 XY VECT(I) = XY VECT(I) + DELTA U(I)
  950 I1=I1-1
      IF(NR .EQ. 0) GO TO 970
      DO 960  I=1,NR
      I3 = I1 + 3*I
      I2 = I3 - 1
      XY VECT(I2) = XY VECT(I2) + DELTA U(I2)
  960 XY VECT(I3) = XY VECT(I3) + DELTA U(I3)
  970 CONTINUE
C
C  PRINT X, Y, DELTA U AND U.
C
      WRITE  (6,200)  MAIN LP, NLS, TITLE
      IF(NS)  1000,1000,980
  980 DO 990 I=1,NS
      IND2 = 2*I
      IND1 = IND2 - 1
  990 WRITE  (6,210)  I, XY VECT(IND1), XY VECT(IND2), DELTA U(IND1),
     1   DELTA U(IND2), U(IND1), U(IND2)
      WRITE  (6,170)
 1000 I1=2*NS
      IF(NR .EQ. 0) GO TO 1020
      DO 1010  I=1,NR
      I3 = NS + I
      I2 = I1 + 3*I
      IND1 = I2 - 2
      IND2 = I2 - 1
 1010 WRITE  (6,220)  I3, XY VECT(IND1), XY VECT(IND2), DELTA U(IND1),
     1   DELTA U(IND2), DELTA U(I2), U(IND1), U(IND2), U(I2)
 1020 CONTINUE
      WRITE  (6,120)  SIG DIG
C
C  END OF MAIN LOOP.
C
 1030 CONTINUE
      GO TO 1220
C
C  EIGENVALUE ROUTE.
C
 1040 END FILE 15
      REWIND 15
C
C  SAVE KE BAR ON 16.
C
      WRITE  (16)  (KE(I), I=1,NO CELS)
      END FILE 16
      REWIND 16
      GO TO (1070,1090), MAIN LP
C
C  COMPRESS THE MATRIX  (KE BAR OR KG BAR).
C
 1050 NW = NO ROWS
      ICTR = 0
      DO 1060  I=1,NO ROWS
      IF  (CN VECT(I) .EQ. 0)  GO TO 1060
      JR = I - ICTR
      CALL COMPRS(JR, NW)
C
C  COUNT THE ROWS DELETED.
C
      ICTR = ICTR + 1
```

```
 1060 CONTINUE                                                           FRAM0592
C                                                                        FRAM0593
C   WRITE THE MATRIX ON 17 FOR THE EIGENVALUE PROGRAM.                   FRAM0594
C                                                                        FRAM0595
      I1 = (NW*(NW+1))/2                                                 FRAM0596
      WRITE (17) NW,I1,(KF(I),I=1,I1)                                    FRAM0597
      GO TO (1080,1160), IND1                                            FRAM0598
C                                                                        FRAM0599
C   COMPRESS AND WRITE KE BAR.                                           FRAM0600
C                                                                        FRAM0601
 1070 IND1 = 1                                                           FRAM0602
      WRITE  (17)   TITLE                                                FRAM0603
      GO TO 1050                                                         FRAM0604
C                                                                        FRAM0605
C   RESET KE BAR.                                                        FRAM0606
C                                                                        FRAM0607
 1080 READ  (16)  (KF(I), I=1,NO CELS)                                   FRAM0608
      REWIND 16                                                          FRAM0609
      GO TO 880                                                          FRAM0610
C                                                                        FRAM0611
C   COMPUTE KG BAR.                                                      FRAM0612
C                                                                        FRAM0613
 1090 DO 1100  I=1,NO CELS                                               FRAM0614
 1100 KF(I) = 0.0                                                        FRAM0615
      DO 1150  I=1,NO ELE                                                FRAM0616
      P = P ARRAY(I)                                                     FRAM0617
      Q = Q ARRAY(I)                                                     FRAM0618
      EL TYPE = TYP ARR(I)                                               FRAM0619
      GO TO (1110,1120,1130), EL TYPE                                    FRAM0620
 1110 IND1 = 10                                                          FRAM0621
      GO TO 1140                                                         FRAM0622
 1120 IND1 = 15                                                          FRAM0623
      GO TO 1140                                                         FRAM0624
 1130 IND1 = 21                                                          FRAM0625
C                                                                        FRAM0626
C   READ KG AND BUILD KG BAR.                                            FRAM0627
C                                                                        FRAM0628
 1140 READ  (15)  (KG(IND2), IND2=1,IND1)                                FRAM0629
      CALL BUILD(KG)                                                     FRAM0630
 1150 CONTINUE                                                           FRAM0631
      REWIND 15                                                          FRAM0632
C                                                                        FRAM0633
C   COMPRESS AND WRITE KG BAR.                                           FRAM0634
C                                                                        FRAM0635
      IND1 = 2                                                           FRAM0636
      GO TO 1050                                                         FRAM0637
C                                                                        FRAM0638
C   RESET KE BAR.                                                        FRAM0639
C                                                                        FRAM0640
 1160 READ  (16)  (KF(I), I=1,NO CELS)                                   FRAM0641
      REWIND 16                                                          FRAM0642
C                                                                        FRAM0643
C   COMPUTE KF = KE BAR + KG BAR                                         FRAM0644
C                                                                        FRAM0645
      DO 1210  I=1,NO ELE                                                FRAM0646
      P = P ARRAY(I)                                                     FRAM0647
      Q = Q ARRAY(I)                                                     FRAM0648
      EL TYPE = TYP ARR(I)                                               FRAM0649
      GO TO (1170,1180,1190), EL TYPE                                    FRAM0650
 1170 IND1 = 10                                                          FRAM0651
      GO TO 1200                                                         FRAM0652
 1180 IND1 = 15                                                          FRAM0653
      GO TO 1200                                                         FRAM0654
 1190 IND1 = 21                                                          FRAM0655
C                                                                        FRAM0656
C   ADD KG BAR.                                                          FRAM0657
```

```
C
 1200 READ  (15)  (KG(IND2), IND2=1,IND1)
      CALL BUILD(KG)
 1210 CONTINUE
      REWIND 15
      GO TO 880
 1220 IF(IGN IND .EQ. 0) GO TO 1230
       CALL EIG
 1230 CALL OVERFL(I1)
      GO TO (1240,1250), I1
C
C  AN OVERFLOW HAS OCCURRED DURING THIS CASE.
C
 1240 WRITE  (6,230)
C
C  SET-UP THE NEXT CASE.
C
 1250 IF(I .EQ. 1) GO TO 240
      STOP
      END
$IBFTC BILD    DECK                 (05 JUL 67)
      SUBROUTINE BUILD(MT)
      COMMON  KF(2485), P VECT(140), DELTA U(140), NO ROWS, EVEN1,
     1  TS(10), P, Q, XQ, XP, YQ, YP, E, A, MOMENT, NS, EL TYPE
      INTEGER  P, Q, EL TYPE
      DOUBLE PRECISION  KF, MT(21), P VECT, DELTA U, TS, XQ, XP,
     1  YQ, YP, E, A, MOMENT
C
C
C     (PARAMETER USAGE)
C
C     MT - THE MATRIX TO BE ADDED.
C
C
      GO TO (10,20,30), EL TYPE
C
C  TYPE 1  (4X4).
C
   10 N1 = 2*(P - 1)
      L1 = 2
      N3 = 2*(Q - 1)
      I1 = 5
      I1 DEL = 2
      L2 = 2
      I2 = 3
      M1 = 2
      GO TO 40
C
C  TYPE 2  (5X5).
C
   20 N1 = 2*(P - 1)
      L1 = 2
      N3 = 2*NS + 3*(Q - NS - 1)
      I1 = 5
      I1 DEL = 2
      L2 = 3
      I2 = 3
      M1 = 2
      GO TO 40
C
C  TYPE 3  (6X6)
C
   30 N1 = 2*NS + 3*(P - NS - 1)
      L1 = 3
      N3 = 2*NS + 3*(Q - NS - 1)
      I1 = 9
```

```
      I1 DEL = 3                                                        BILD0046
      L2 = 3                                                            BILD0047
      I2 = 6                                                            BILD0048
      M1 = 3                                                            BILD0049
C                                                                       BILD0050
C   STORE THE UPPER TRIANGULAR  (P).                                    BILD0051
C                                                                       BILD0052
   40 N2 = (N1*(N1+3))/ 2                                               BILD0053
      I = 0                                                             BILD0054
      DO 60  L=1,L1                                                     BILD0055
      DO 50  M=1,L                                                      BILD0056
      I = I+1                                                           BILD0057
      N2 = N2+1                                                         BILD0058
   50 KF(N2) = KF(N2) + MT(I)                                           BILD0059
   60 N2 = N2+N1                                                        BILD0060
C                                                                       BILD0061
C   STORE THE LOWER TRIANGULAR  (Q).                                    BILD0062
C                                                                       BILD0063
      N2 = (N3*(N3+3))/ 2                                               BILD0064
      I = I1                                                            BILD0065
      DO 80  L=1,L2                                                     BILD0066
      DO 70  M=1,L                                                      BILD0067
      I = I+1                                                           BILD0068
      N2 = N2+1                                                         BILD0069
   70 KF(N2) = KF(N2) + MT(I)                                           BILD0070
      I = I + I1 DEL                                                    BILD0071
   80 N2 = N2+N3                                                        BILD0072
C                                                                       BILD0073
C   STORE THE LOWER LEFT MATRIX  (P, Q).                                BILD0074
C                                                                       BILD0075
      I = I2                                                            BILD0076
      DO 100  L=1,L2                                                    BILD0077
      N2 = (N3*(N3+1))/ 2 + N1                                          BILD0078
      DO 90  M=1,M1                                                     BILD0079
      I = I+1                                                           BILD0080
      N2 = N2+1                                                         BILD0081
   90 KF(N2) = KF(N2) + MT(I)                                           BILD0082
      I = I+L                                                           BILD0083
  100 N3 = N3+1                                                         BILD0084
      RETURN                                                            BILD0085
      END                                                               BILD0086
$IBFTC SET      DECK                  (05 JUL 67)                       SET 0000
      SUBROUTINE SET(JR)                                                SET 0001
      COMMON  KF(2485), P VECT(140), DELTA U(140), NO ROWS              SET 0002
      DOUBLE PRECISION  KF, P VECT, DELTA U                             SET 0003
   10 FORMAT  (1H8/ 1H8/ 45X42HROW NUMBER IN SUBROUTINE SET IS TOO LARGESET 0004
     1.// 1H0, 53X5H(JR =, I4, 9H,  MAX. =, I4, 1H)/ 1H8)               SET 0005
C                                                                       SET 0006
C   JR = ROW NUMBER                                                     SET 0007
C                                                                       SET 0008
      IF  (JR - NO ROWS)  30,30,20                                      SET 0009
C                                                                       SET 0010
C   JR IS ILLEGAL.                                                      SET 0011
C                                                                       SET 0012
   20 WRITE  (6,10)  JR, NO ROWS                                        SET 0013
      CALL FXEM                                                         SET 0014
C                                                                       SET 0015
C   SET P VECTOR CONSTRAINT.                                            SET 0016
C                                                                       SET 0017
   30 P VECT(JR) = 0                                                    SET 0018
      J1 = JR - 1                                                       SET 0019
      J2 = (JR*J1)/ 2                                                   SET 0020
C                                                                       SET 0021
C   TEST FOR ROW NUMBER 1.                                              SET 0022
C                                                                       SET 0023
```

```
      IF  (J1)  40,60,40
   40 DO 50  I=1,J1
      J2 = J2 + 1
   50 KF(J2) = 0
   60 J2 = J2 + 1
      KF(J2) = 1.0
C
C  TEST FOR LAST ROW.
C
      IF  (JR - NO ROWS)  70,90,70
   70 J3 = NO ROWS - 1
      DO 80  I=JR,J3
      J2 = J2 + I
   80 KF(J2) = 0
   90 CONTINUE
      RETURN
      END
$IBFTC CMPRS   DECK                    (05 JUL 67)
      SUBROUTINE  COMPRS(JR, NR)
      COMMON  KF(2485)
      DOUBLE PRECISION  KF
C
C
C     (PARAMETER USAGE)
C
C     JR - ROW AND COLUMN TO BE DELETED.
C     NR - CURRENT MATRIX DIMENSION.
C
C
   10 FORMAT  (1H8/ 1H8/ 27X73HYOU HAVE REQUESTED SUBROUTINE COMPRS TO D
     1ELETE A NON EXISTANT ROW.  (JR =, I3, 1H)/ 1H8)
      IF  (JR)  20,20,30
C
C  JR IS ILLEGAL.
C
   20 WRITE  (6,10)  JR
      CALL FXEM
C
C  TEST FOR LAST ROW.
C
   30 IF  (JR - NR)  40,80,20
C
C  DELETE THE COLUMN.
C
   40 N1 = NR - JR
      N2 = (JR*(JR+3))/ 2 - 1
      DO 60  L=1,N1
      DO 50  L1=1,L
      N2 = N2+1
   50 KF(N2) = KF(N2+1)
   60 N2 = N2 + JR
C
C  DELETE THE ROW.
C
      N1 = NR-1
      N2 = (JR*(JR-1))/ 2
      DO 70  L=JR,N1
      DO 70  L1=1,L
      N2 = N2+1
      N3 = N2+L
   70 KF(N2) = KF(N3)
C
C  REDUCE THE MATRIX DIMENSION.
C
   80 NR = NR-1
      RETURN
      END
```

```
$IBFTC MTXEQ    DECK                    (05 JUL 67)
      SUBROUTINE MTXEQ(ISD)
      DOUBLE PRECISION  C(70,72), PIV(72), ATPE, RM, A, B, X
      COMMON  A(2485), B(70,2), X(70,2), N
      DATA     NMAX, NKMAX/ 70, 72/, K/ 2/
C
C     MATRIX EQUATION SOLVER      (7094 FORTRAN IV)
C
C     USAGE...
C
C     TO SOLVE THE LINEAR SYSTEM      AX=B
C
C     CALL MTXEQ(ISD)
C
C        WHERE A IS A SYMMETRIC MATRIX.
C              X MUST BE DIMENSIONED N X K
C              B MUST BE DIMENSIONED N X K
C              N IS THE NO. OF EQUATIONS (ROWS IN A,X,B)
C              K IS THE NO. OF SOLUTION VECTORS (COLS. IN X,B)
C              ISD REPRESENTS THE ACCURACY OF THE SOLUTION
C                  IN SIGNIFICANT DIGITS.
C
C
C     NOTE...    TO CHANGE DIMENSIONS OF ARRAYS C AND PIV, ALSO
C            CHANGE VALUES OF NMAX AND NKMAX IN DATA STATEMENT.
C
C
C     TEST N AND K FOR CORRECT RANGE
C
      IF ( N .LE. 0  .OR.  N .GT. NMAX )     GO TO 220
      IF ( K .LE. 0  .OR.  (N+K) .GT. NKMAX )     GO TO 220
C
C     GET ARGUMENTS N AND K
C
      NP=N
      KP=K
C
C     MOVE ARRAYS A(I,J) AND B(I,J) INTO C(I,J)
C
      J1 = 0
      DO 20  I=1,NP
      DO 10  J=1,I
      J2 = J1 + J
      C(I,J) = A(J2)
      IF  (I .EQ. J)  GO TO 10
      C(J,I) = C(I,J)
   10 CONTINUE
   20 J1 = J1 + I
C
C  GENERATE ROW SUM FOR SIGNIFICANT DIGIT CHECK.
C
      DO 30  I=1,N
      B(I,2) = 0.0
      DO 30  J=1,N
   30 B(I,2) = B(I,2) + C(I,J)
      DO 40 J=1,KP
      NPJ=NP+J
      DO 40 I=1,NP
   40 C(I,NPJ)=B(I,J)
C
C     SET TO PERFORM N ELIMINATION SWEEPS (I=1,N)
C
      NP1=NP+1
      NPK=NP+KP
      DO 140 I=1,NP
      IP1=I+1
```

```
C
C     SEARCH FOR NEXT PIVOT ROW (I-TH PIVOT IS IN COL. I)
C
      ATPE=0.
      DO 60 J=I,NP
      IF (DABS(C(J,I))-ATPE) 60,50,50
   50 ATPE=DABS(C(J,I))
      IPIV=J
   60 CONTINUE
C
C     OPERATE ON THE PIVOT ROW
C
      IF (ATPE) 230,230,70
   70 DO 80 J=IP1,NPK
   80 PIV(J)=C(IPIV,J)/C(IPIV,I)
C
C     PERFORM ELIMINATIONS BELOW THE DIAGONAL (COL. I)
C
      IFROM=NP
      ITO=NP
   90 IF (IFROM-IPIV) 100,120,100
  100 RM=-C(IFROM,I)
      DO 110 J=IP1,NPK
  110 C(ITO,J)=C(IFROM,J)+RM*PIV(J)
      ITO=ITO-1
  120 IFROM=IFROM-1
      IF (IFROM-I) 130,90,90
C
C     PUT THE I-TH PIVOT ROW IN THE VACATED ROW I
C
  130 DO 140 J=IP1,NPK
  140 C(I,J)=PIV(J)
C
C     NOW DO THE BACK SOLUTION
C
      I=NP
  150 IP1=I
      I=I-1
      IF (I) 180,180,160
  160 DO 170 J=NP1,NPK
      DO 170 L=IP1,NP
  170 C(I,J)=C(I,J)-C(I,L)*C(L,J)
      GO TO 150
C
C     MOVE THE SOLUTION TO ARRAY X(I,J)
C
  180 DO 190 J=1,KP
      NPJ=NP+J
      DO 190 I=1,NP
  190 X(I,J)=C(I,NPJ)
      ISD = 20
      DO 210  I=1,N
      AB = ABS(1.0 - X(I,2))
      IF  (AB .NE. 0.0)   GO TO 200
      AB = 1.0E-16
  200 ISCH = -ALOG10(AB)
  210 ISD = MINO(ISD, ISCH)
      RETURN
C
  220 WRITE (6,240) NP,KP
      CALL FXEM
  230 WRITE (6,250)
      CALL FXEM
      RETURN
  240 FORMAT(3HON=I12,5H   K=I12,35H ARE INCORRECT FOR SUBROUTINE MTXEQ)
  250 FORMAT (37HODET(A)=0 IN CALL TO SUBROUTINE MTXEQ)
      END
```

```
$IBFTC EIGVA    DECK
      SUBROUTINE EIG
      DIMENSION A(50,50),Z(50,50),S(50,50),D(50)
      DIMENSION TITLE(8)
      DOUBLE PRECISION A
   10 FORMAT(1H1,20X,42HEIGENVALUE SOLUTION FOR DET(KE+LAMBDA*KG)=0////)
   20 FORMAT(1X,13HTHE MATRIX KG////)
   30 FORMAT(1X,10E13.6//)
   40 FORMAT(1X,///)
   50 FORMAT(1X,13HTHE MATRIX KE////)
   60 FORMAT(1X,15HTHE EIGENVALUES////)
      WRITE (6,10)
      REWIND 17
      READ (17) TITLE
      READ (17) N,I1,((A(I,J),J=1,I),I=1,N)
      DO 70 I=1,N
      DO 70 J=1,I
   70 A(J,I)=A(I,J)
      WRITE(6,50)
      DO 80 I=1,N
      WRITE (6,30) (A(I,J),J=1,N)
   80 WRITE (6,40)
      CALL JACOB(N,A,S,D)
      DO 90 J=1,N
      DEN=1.0/DSQRT(A(J,J))
      DO 90 I=1,N
   90 Z(I,J)=S(I,J)*DEN
      READ (17) N,I1,((A(I,J),J=1,I),I=1,N)
      DO 100 I=1,N
      DO 100 J=1,I
  100 A(J,I)=A(I,J)
      WRITE (6,20)
      DO 110 I=1,N
      WRITE (6,30) (A(I,J),J=1,N)
  110 WRITE (6,40)
      WRITE (6,60)
      DO 130 J=1,N
      DO 130 I=1,N
      SUM=0.0
      DO 120 K=1,N
  120 SUM=SUM+A(I,K)*Z(K,J)
  130 S(I,J)=SUM
      DO 150 I=1,N
      DO 150 J=1,N
      SUM=0.0
      DO 140 K=1,N
  140 SUM=SUM+Z(K,I)*S(K,J)
  150 A(I,J)=SUM
      CALL JACOB(N,A,S,D)
      DO 160 I=1,N
  160 D(I)=1./D(I)
      WRITE (6,30) (D(I),I=1,N)
      REWIND 17
      RETURN
      END
$IBFTC JACOBI   DECK
      SUBROUTINE JACOB(N,A,S,D)
      DIMENSION A(50,50),S(50,50),D(50)
      DOUBLE PRECISION A
      OFF=.1E-10
      INDIC=0
      DO 10 I=1,N
      DO 10 J=1,N
   10 S(I,J)=0.0
      DO 20 I=1,N
   20 S(I,I)=1.0
```

```
      SUM=0.0
      NLES1=N-1
      DO 30 I=1,NLES1
      K=I+1
      DO 30 J=K,N
   30 SUM=SUM+A(I,J)*A(I,J)
      VF=SQRT(SUM*2.0)
      BOX=1.0
   40 KQ=2
   50 JP=1
   60 IF(VF-DABS(A(JP,KQ))) 80,70,70
   70 GO TO 190
   80 INDIC=1
      Y=-A(JP,KQ)
      ZI=0.5*(A(JP,JP)-A(KQ,KQ))
      W=Y/(SQRT(Y*Y+ZI*ZI))
      IF(ZI) 90,100,100
   90 W=-W
  100 SN=W/(SQRT(2.0*(1.0+SQRT(1.0-W*W))))
      CS=SQRT(1.0-SN*SN)
      IF(BOX-.5) 120,120,110
  110 S(JP,JP)=CS
      S(KQ,JP)=-SN
      S(JP,KQ)=SN
      S(KQ,KQ)=CS
  120 HOLD1=A(JP,JP)*CS*CS+A(KQ,KQ)*SN*SN-2.0*A(JP,KQ)*SN*CS
      HOLD2=A(JP,JP)*SN*SN+A(KQ,KQ)*CS*CS+2.0*A(JP,KQ)*SN*CS
      DO 130 I=1,N
      D(I)=A(I,JP)*CS-A(I,KQ)*SN
      A(I,KQ)=A(I,JP)*SN+A(I,KQ)*CS
  130 A(I,JP)=D(I)
      IF(BOX-0.5) 150,150,140
  140 BOX=0.0
      GO TO 170
  150 DO 160 I=1,N
      D(I)=S(I,JP)*CS-S(I,KQ)*SN
      S(I,KQ)=S(I,JP)*SN+S(I,KQ)*CS
  160 S(I,JP)=D(I)
  170 A(JP,JP)=HOLD1
      A(KQ,KQ)=HOLD2
      A(JP,KQ)=0.0
      DO 180 I=1,N
      A(JP,I)=A(I,JP)
  180 A(KQ,I)=A(I,KQ)
  190 IF(JP-KQ+1) 200,210,210
  200 JP=JP+1
      GO TO 60
  210 IF(KQ-N) 220,230,250
  220 KQ=KQ+1
      GO TO 50
  230 TEST=INDIC
      IF(TEST-0.5) 250,240,240
  240 INDIC=0
      GO TO 40
  250 IF(OFF-VF) 260,270,270
  260 VF=VF/10.0
      GO TO 40
  270 DO 280 I=1,N
  280 D(I)=A(I,I)
      RETURN
      END
```

APPENDIX III

SAMPLE INPUT DATA

```
cc   1    5    10     15   20   25          30-80
     1    0    14     18   50    1           6-BAY VIERENDEEL TRUSS
     2    1   111                       5.0
     2    2   111
     2    3                5.0          5.0
     2    4                5.0
     2    5               10.0          5.0
     2    6               10.0
     2    7               15.0          5.0
     2    8               15.0
     2    9               20.0          5.0
     2   10               20.0
     2   11               25.0          5.0
     2   12               25.0
     2   13               30.0          5.0
     2   14               30.0                          -5.0
     3    1     1            3            3          29.0D06      .0625     .000020345
     3    2     3            4            3          29.0D06      .0625     .000020345
     3    3     2            4            3          29.0D06      .0625     .000020345
     3    4     3            5            3          29.0D06      .0625     .000020345
     3    5     5            6            3          29.0D06      .0625     .000020345
     3    6     4            6            3          29.0D06      .0625     .000020345
     3    7     5            7            3          29.0D06      .0625     .000020345
     3    8     7            8            3          29.0D06      .0625     .000020345
     3    9     6            8            3          29.0D06      .0625     .000020345
     3   10      7           9            3          29.0D06      .0625     .000020345
     3   11     9           10            3          29.0D06      .0625     .000020345
     3   12     8           10            3          29.0D06      .0625     .000020345
     3   13     9           11            3          29.0D06      .0625     .000020345
     3   14    11           12            3          29.0D06      .0625     .000020345
     3   15    10           12            3          29.0D06      .0625     .000020345
     3   16    11           13            3          29.0D06      .0625     .000020345
     3   17    13           14            3          29.0D06      .0625     .000020345
     3   18    12           14            3          29.0D06      .0625     .000020345
     9
```

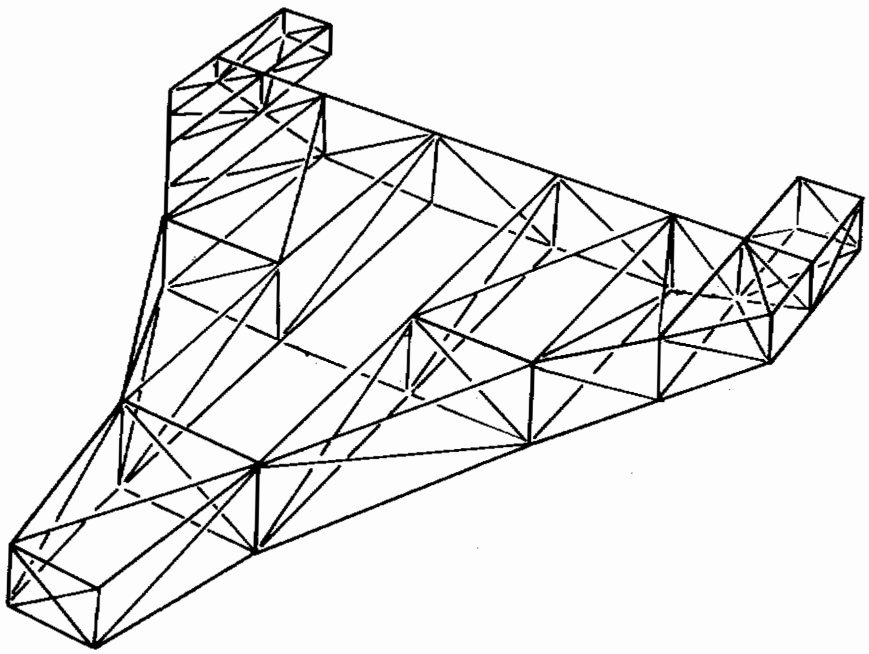

Figure 1. Three-Dimensional Truss Structure (Lifting reentry body)

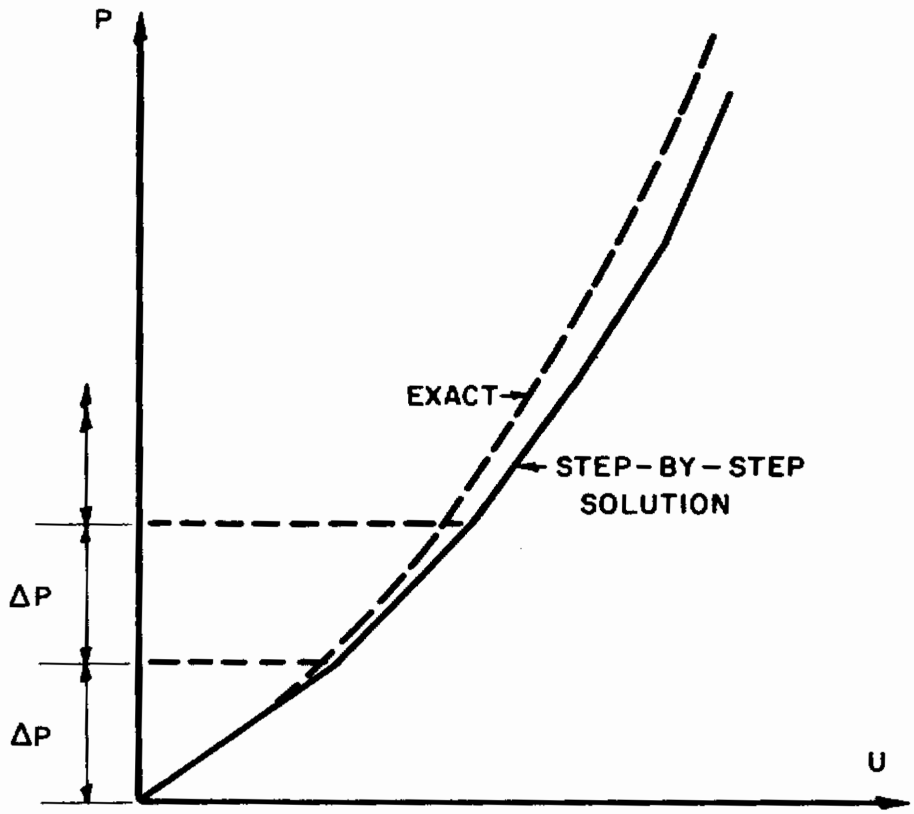

Figure 2. Nonlinear Force-Displacement Relationship

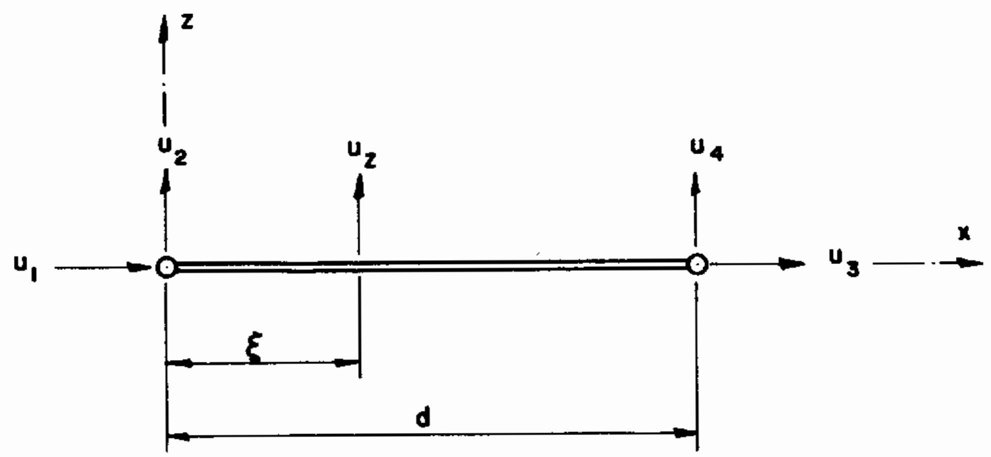

Figure 3. Pin-Jointed Bar Element in Local Coordinate System

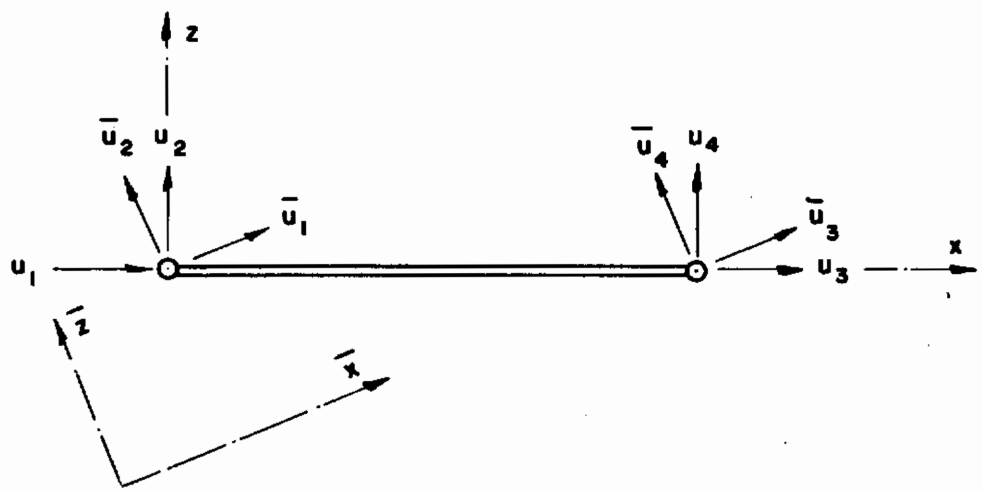

Figure 4. Pin-Jointed Bar Element in Datum Coordinate System

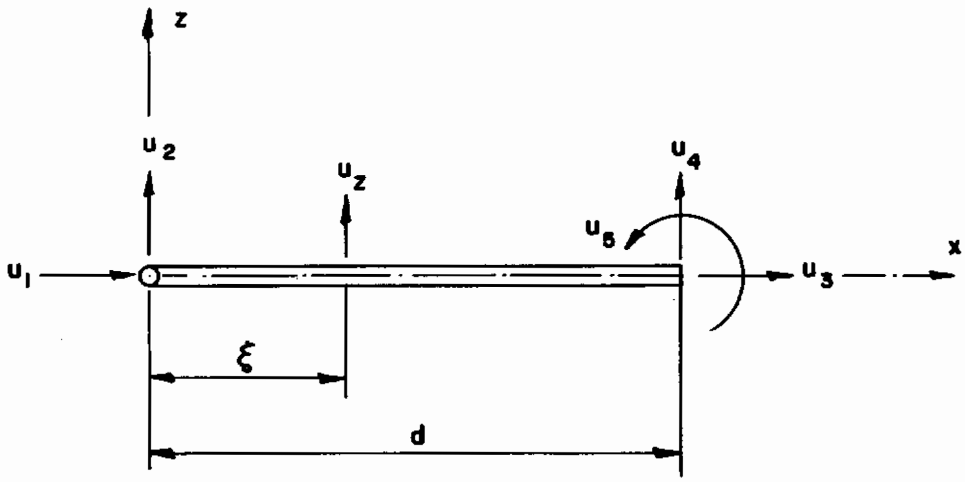

Figure 5. Pinned-Rigid Beam Element

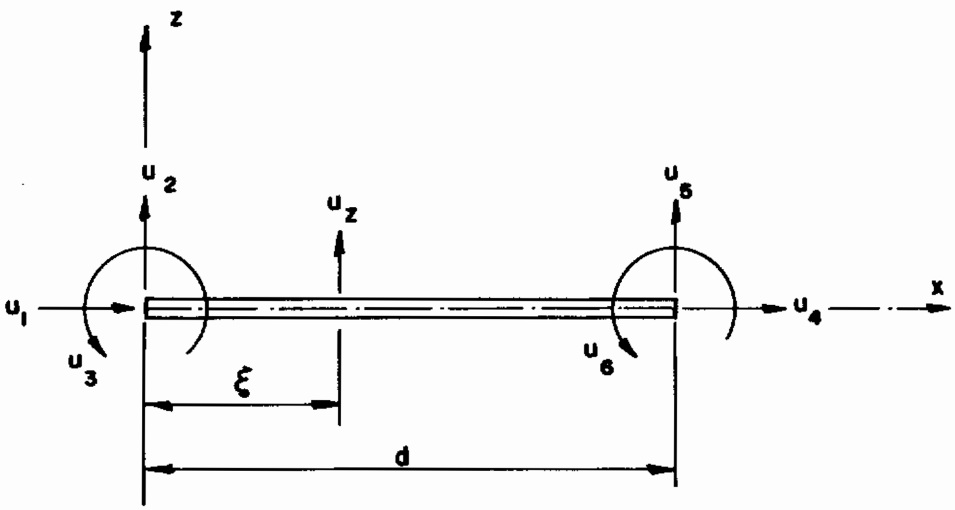

Figure 6. Beam Element

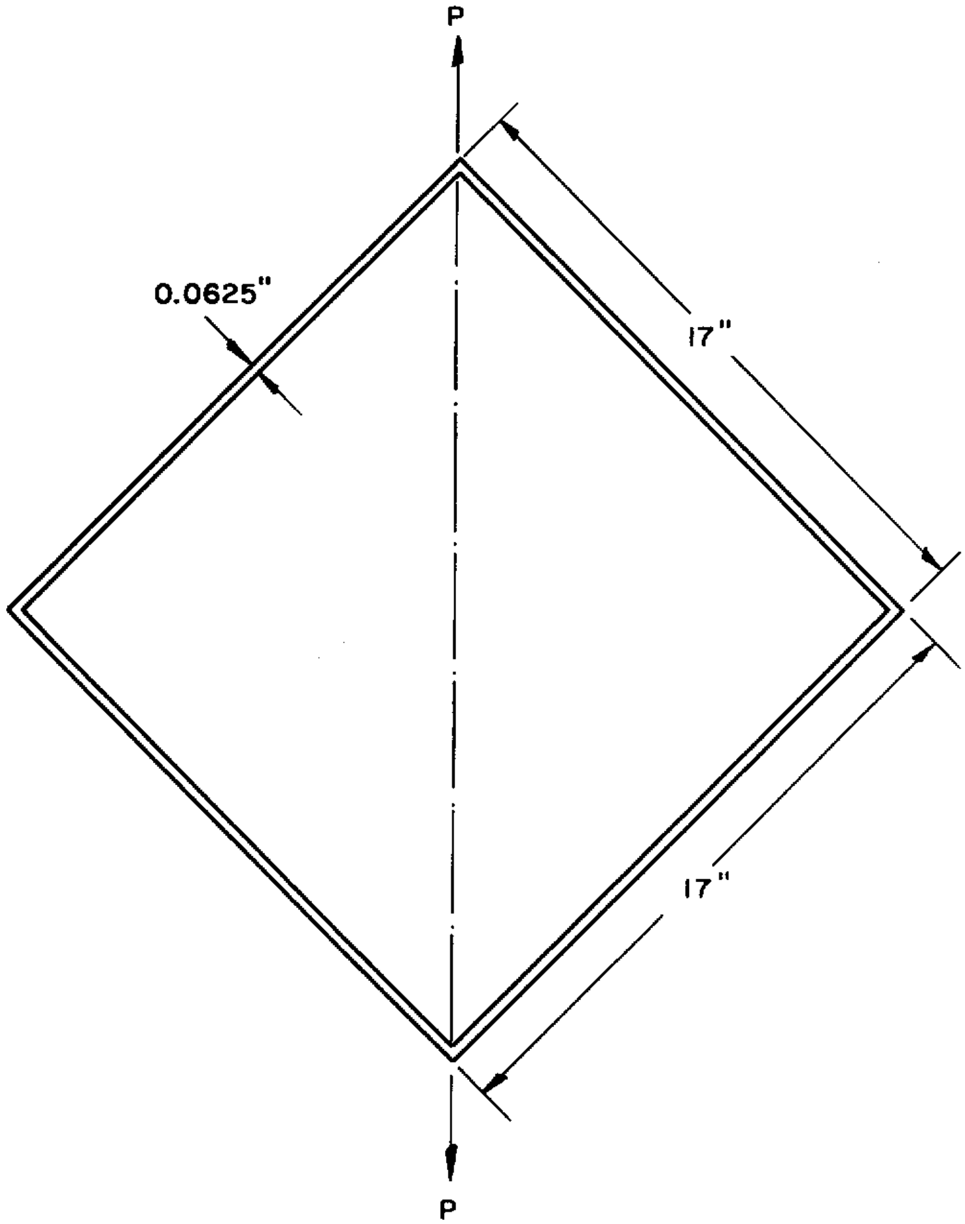

Figure 7. Square Frame Loading and Geometry

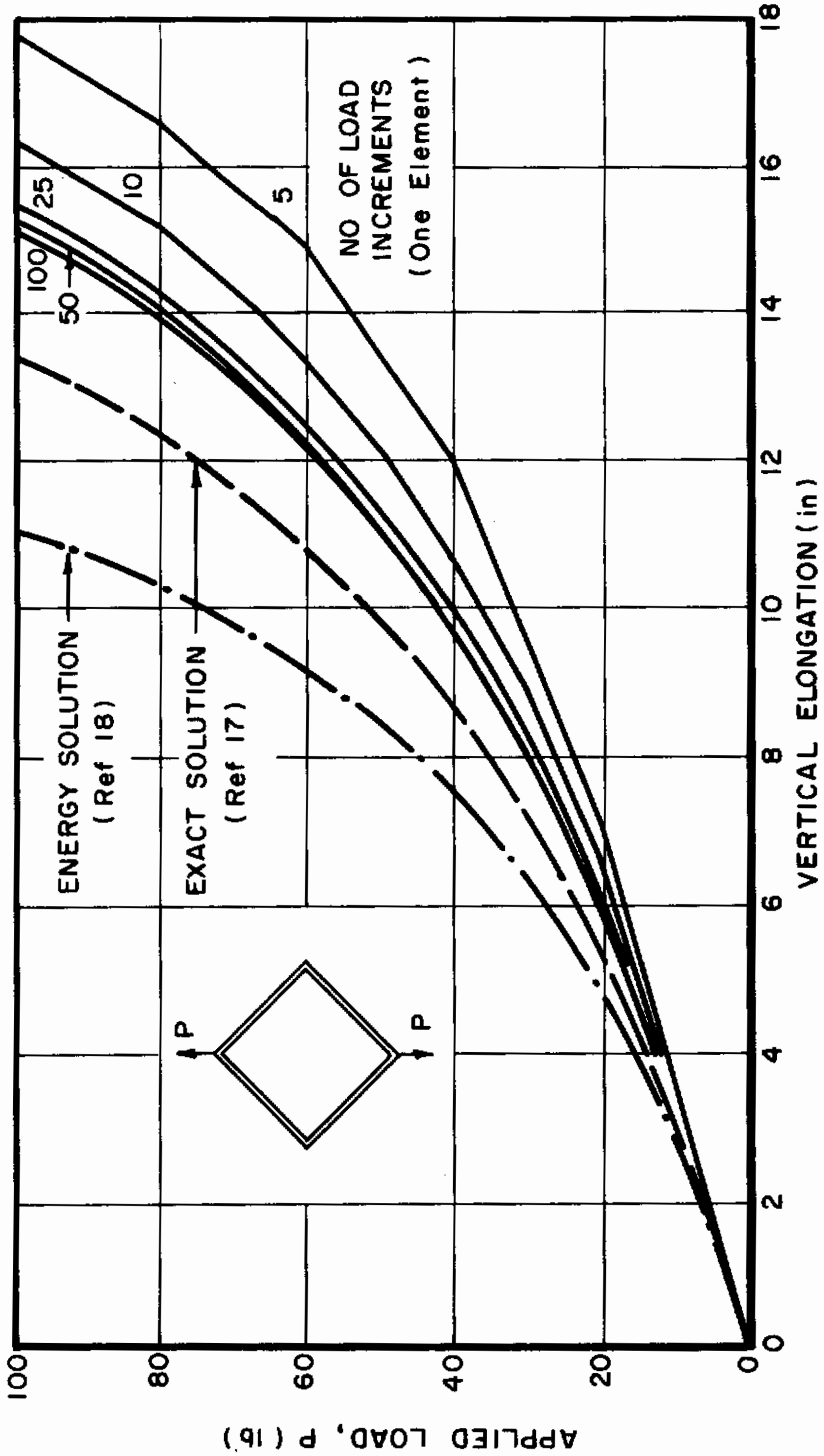

Figure 8. Vertical Elongation for Different Load Increments

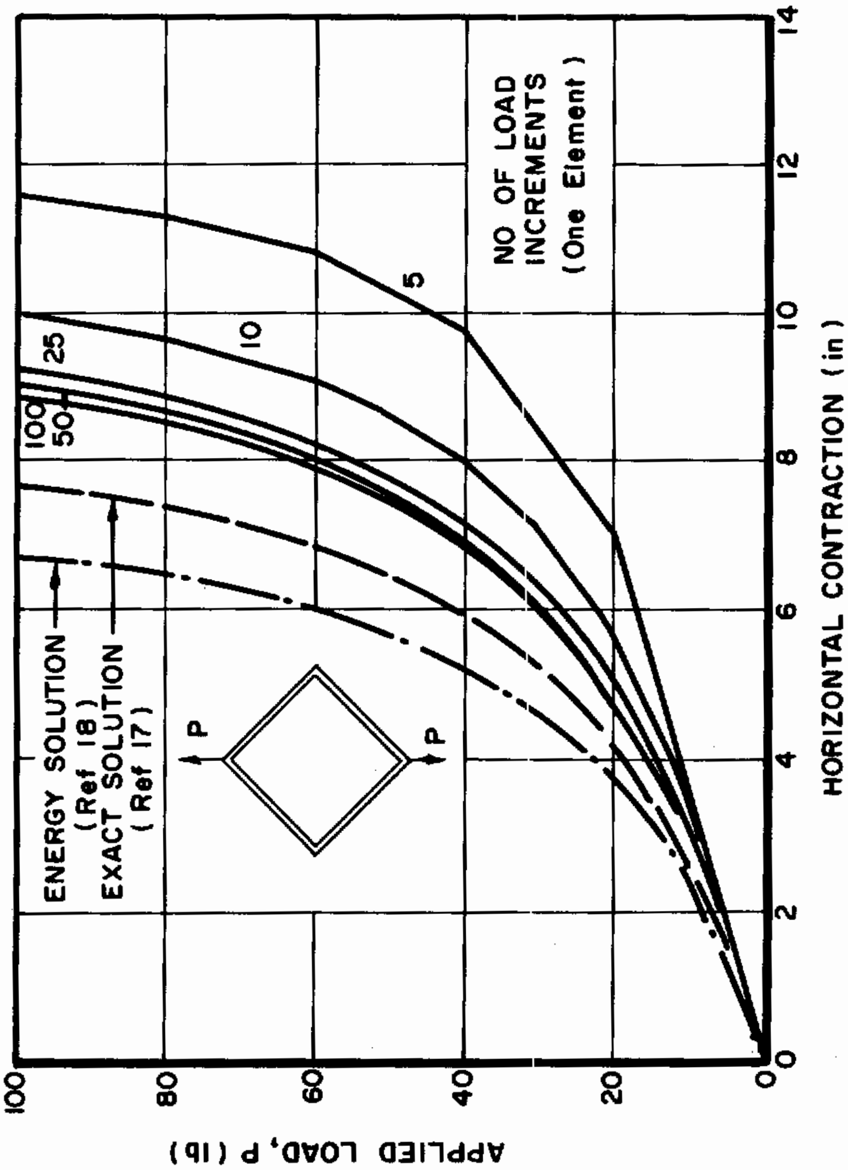

Figure 9. Horizontal Contraction for Different Load Increments

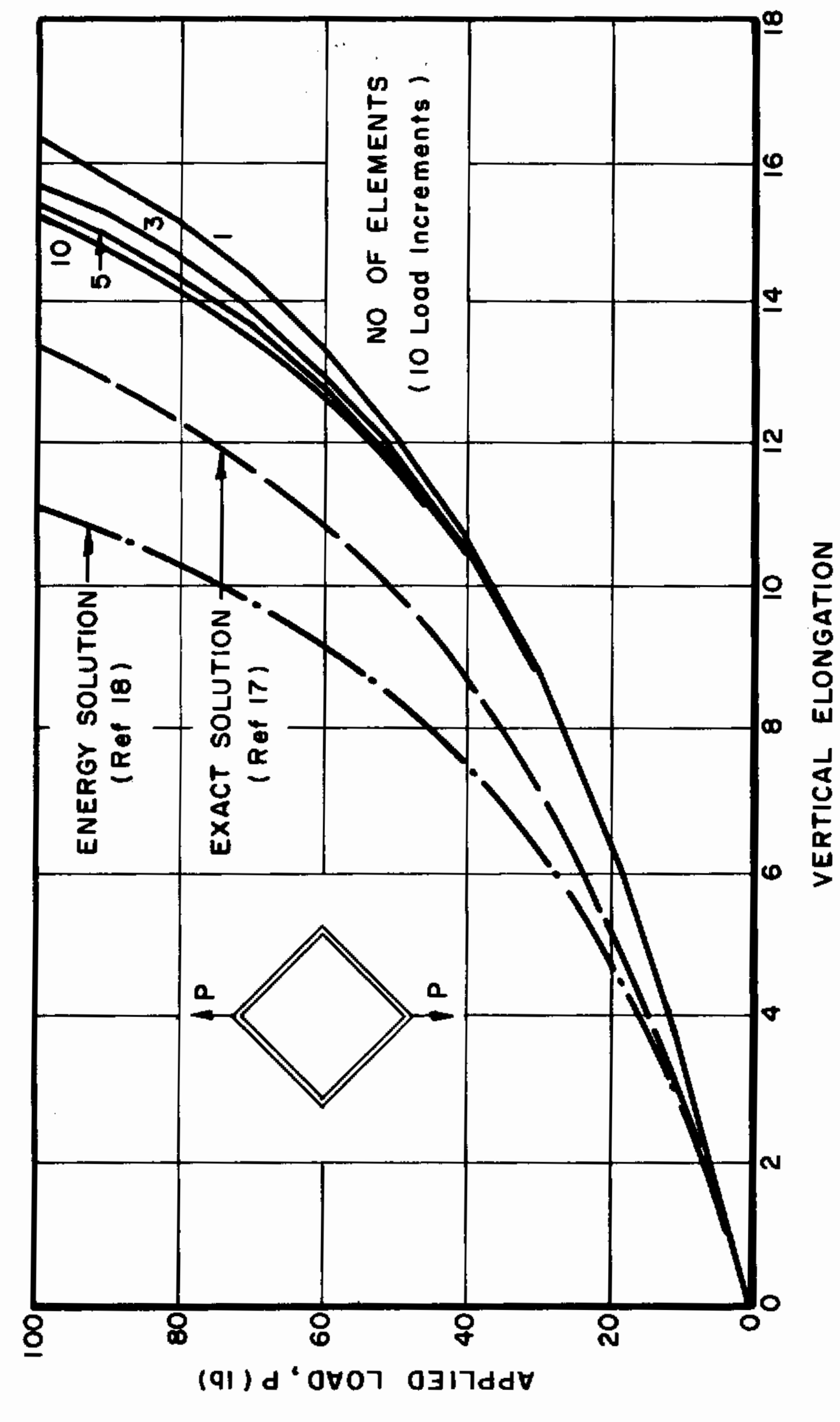

Figure 10. Vertical Elongation for Different Number of Elements

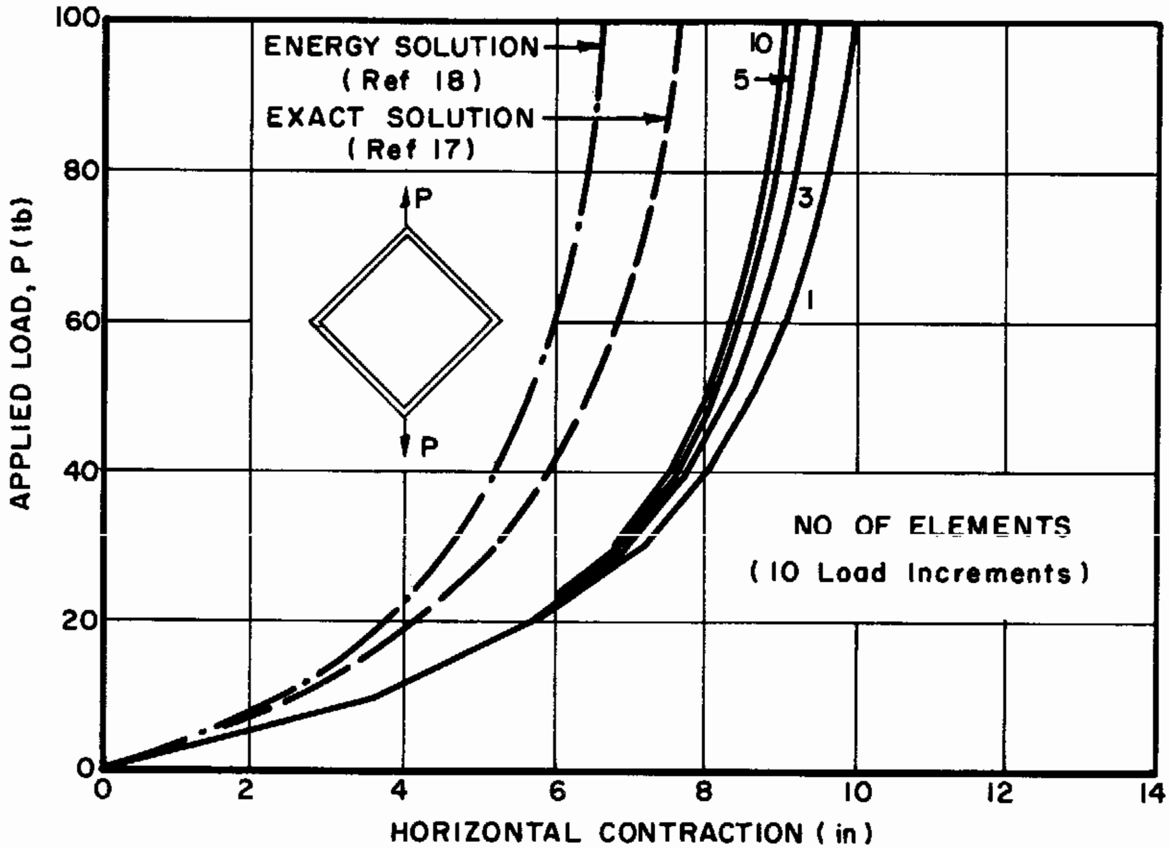

Figure 11. Horizontal Contraction for Different Number of Elements

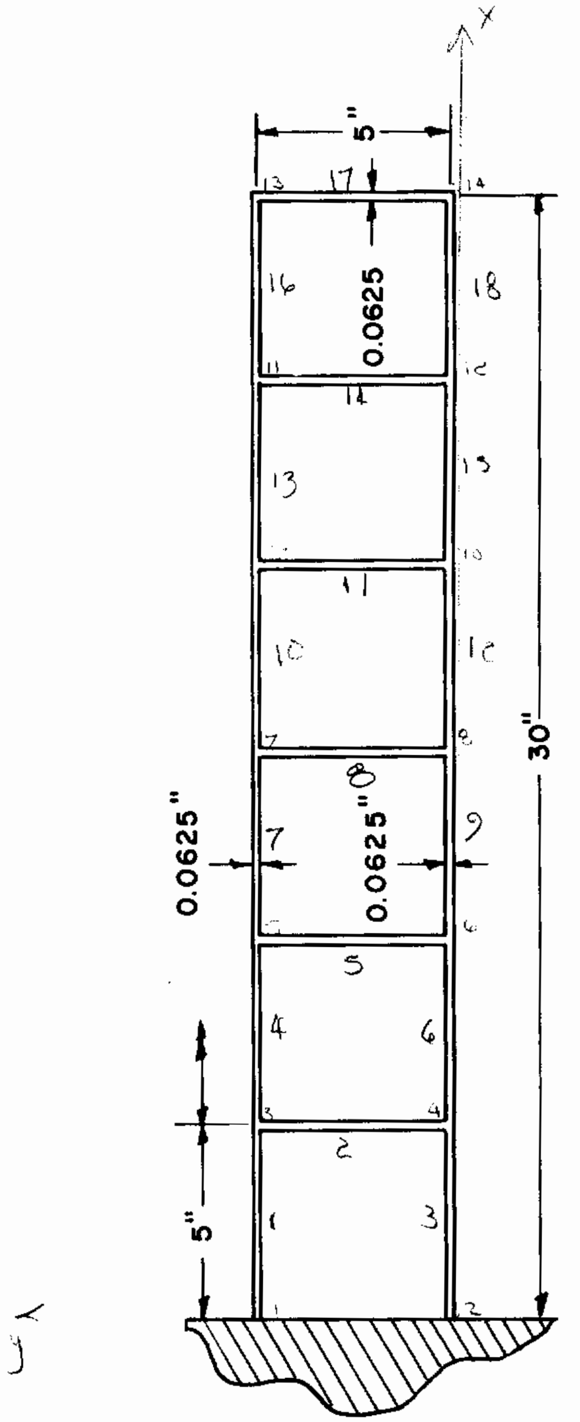

Figure 12. Six-Bay Frame Geometry

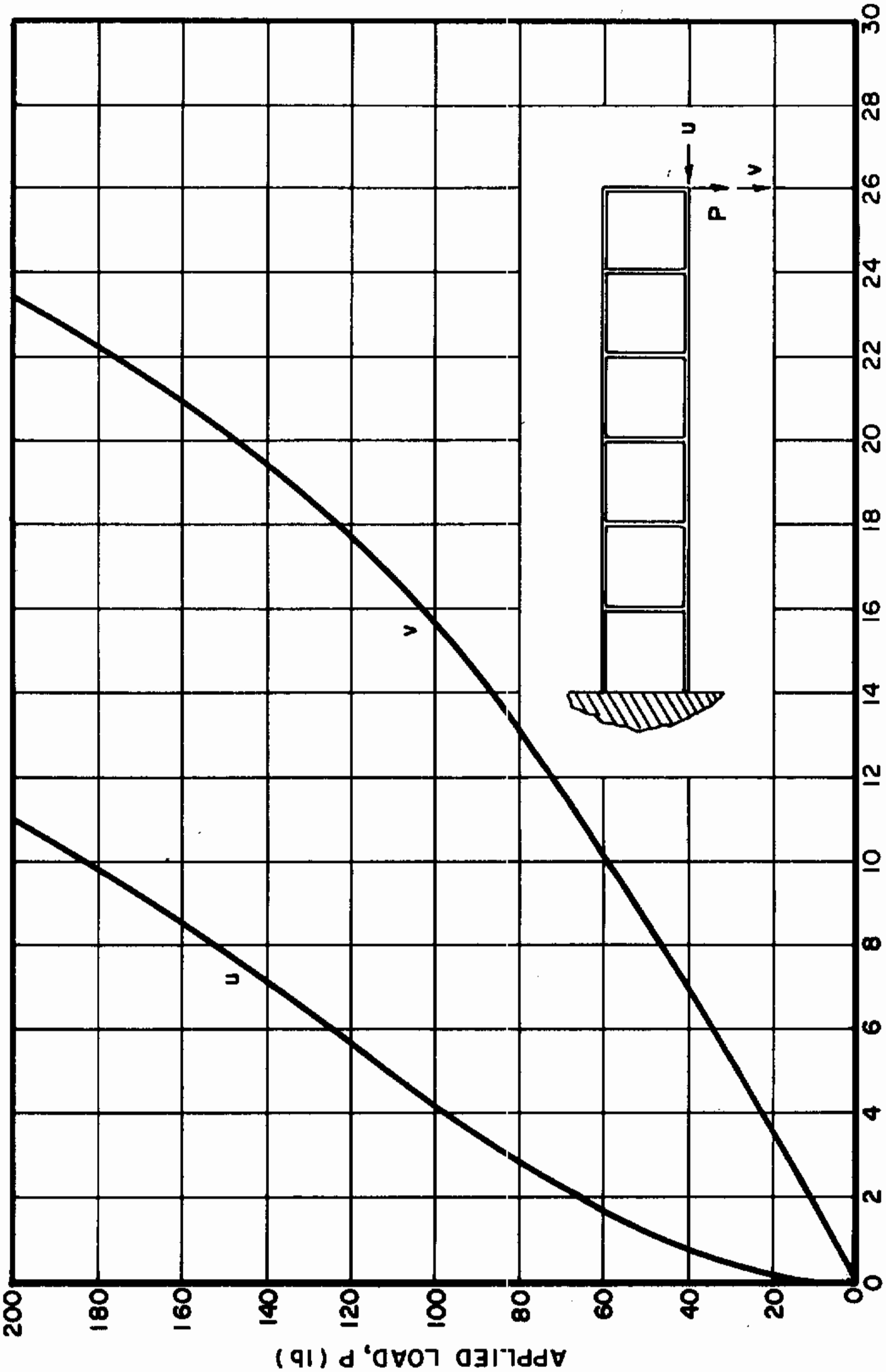

Figure 13. Horizontal and Vertical Deflections in a Six-Bay Frame

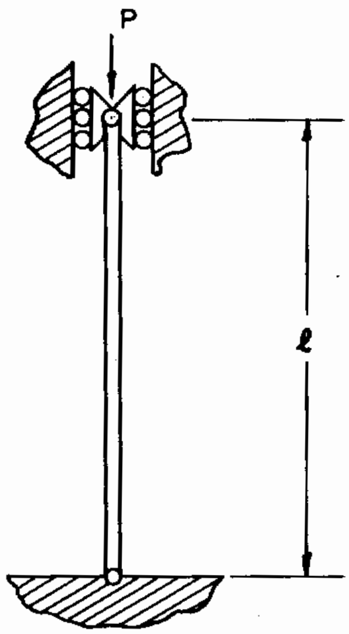

Figure 14. Simply-Supported Column

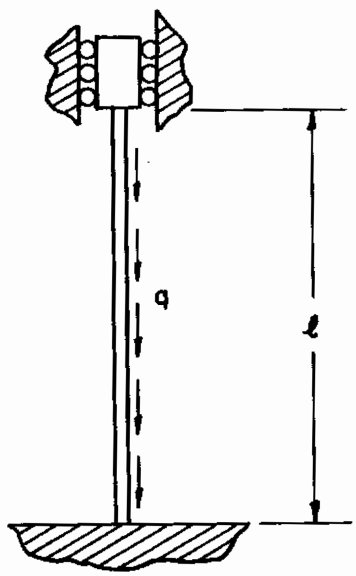

Figure 15. Built-in Column Under Gravity Loading

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b GROUP N/A

3. REPORT TITLE

"LARGE DEFLECTION AND STABILITY ANALYSIS OF TWO-DIMENSIONAL TRUSS AND FRAME STRUCTURES"

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

May 15 - 20, 1967

5. AUTHOR(S) *(Last name, first name, initial)*

Przemieniecki, John S.
Purdy, David M.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1968	67	20

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 1467	AFFDL-TR-68-38
c. Task No. 146701	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	N/A

10. AVAILABILITY/LIMITATION NOTICES

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Air Force Flight Dynamics Laboratory (FDTR), Wright-Patterson AFB, Ohio 45433

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Air Force Flight Dynamics Laboratory (FDTR) Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433

13. ABSTRACT

Frame structures with either pinned or fixed joints are frequently used in aerospace applications as the primary structure supporting light secondary panels or other structural assemblies. In the analysis of such structural designs it is very often necessary to consider loading conditions for which the deflections are large enough to cause significant changes in the geometry of the structure so that the equations of equilibrium must be formulated for the deformed configuration. In the present paper the general analysis for large deflections of frame structures is presented using the concept of discrete element idealization. The solution for deflections and stresses is presented as a step-by-step matrix method based on load increments and is particularly suitable for computer programming. As a bi-product of the large deflection analysis the eigenvalue equations for structural stability are also formulated. The theoretical results of the nonlinear, large deflection matrix solution are compared with the exact analytical results for a square frame. In addition, the results for deflections of a six-bay truss and buckling of columns with either constant axial load or gravity loading are also presented. The computer program listing and instructions for the preparation of input data are included.

DD FORM 1 JAN 64 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Frame Structures						
Nonlinear						
Matrix Method						
Large Deflections						
Stability						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.