A DISCRETE-ELEMENT SOLUTION OF PLATES AND PAVEMENT SLABS USING A VARIABLE-INCREMENT-LENGTH MODEL

by

Charles M. Pearre, III
W. Ronald Hudson

Research Report Number 56-11

Development of Methods for Computer Simulation
of Beam-Columns and Grid-Beam and Slab Systems

Research Project 3-5-63-56

conducted for

The Texas Highway Department

in cooperation with the
U. S. Department of Transportation
Federal Highway Administration
Bureau of Public Roads

by the

CENTER FOR HIGHWAY RESEARCH
THE UNIVERSITY OF TEXAS AT AUSTIN

April 1969

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the Bureau of Public Roads.

PREFACE

This report presents the results of an analytical study undertaken to improve the method for the direct computer solution of plates and pavement slabs. The basic procedure for the structural analysis is an extension of a discrete method developed by Hudson and Matlock and a direct solution technique developed by Stelzer and Hudson.

This is the eleventh in a series of reports that describe the work in Research Project No. 3-5-63-56, entitled "Development of Methods for Computer Simulation of Beam-Columns and Grid-Beam and Slab Systems." The project, under the general direction of Professor Hudson Matlock, is divided into two parts. Part I is concerned primarily with bridge structures. Part II deals with pavement slabs. The reader may find it advantageous to review Report Nos. 56-1, 56-4, 56-6, and 56-9, as they provide background for this report.

This is the third report in the series that deals directly with pavement slabs. Several subsequent reports concerning pavement solutions will be submitted as they are developed.

Duplicate copies of the program deck and test data cards for the example problems may be obtained from the Center for Highway Research, The University of Texas at Austin.

The excellent facilities of the Computation Center of The University of Texas and the cooperation of its staff have contributed significantly to this report. Thanks are due to Art Frakes, Don Fenner, Joye Linkous, Eva Miller, and others who assisted with the preparation of the manuscript.

Charles M. Pearre, III
W. Ronald Hudson

April 1969

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

LIST OF REPORTS

Report No. 56-1, "A Finite-Element Method of Solution for Linearly Elastic Beam-Columns" by Hudson Matlock and T. Allan Haliburton, presents a finite-element solution for beam-columns that is a basic tool in subsequent reports.

Report No. 56-2, "A Computer Program to Analyze Bending of Bent Caps" by Hudson Matlock and Wayne B. Ingram, describes the application of the beam-column solution to the particular problem of bent caps.

Report No. 56-3, "A Finite-Element Method of Solution for Structural Frames" by Hudson Matlock and Berry Ray Grubbs, describes a solution for frames with no sway.

Report No. 56-4, "A Computer Program to Analyze Beam-Columns under Movable Loads" by Hudson Matlock and Thomas P. Taylor, describes the application of the beam-column solution to problems with any configuration of movable non-dynamic loads.

Report No. 56-5, "A Finite-Element Method for Bending Analysis of Layered Structural Systems" by Wayne B. Ingram and Hudson Matlock, describes an alternating-direction iteration method for solving two-dimensional systems of layered grids-over-beams and plates-over-beams.

Report No. 56-6, "Discontinuous Orthotropic Plates and Pavement Slabs" by W. Ronald Hudson and Hudson Matlock, describes an alternating-direction iteration method for solving complex two-dimensional plate and slab problems with emphasis on pavement slabs.

Report No. 56-7, "A Finite-Element Analysis of Structural Frames" by T. Allan Haliburton and Hudson Matlock, describes a method of analysis for rectangular plane frames with three degrees of freedom at each joint.

Report No. 56-8, "A Finite-Element Method for Transverse Vibrations of Beams and Plates" by Harold Salani and Hudson Matlock, describes an implicit procedure for determining the transient and steady-state vibrations of beams and plates, including pavement slabs.

Report No. 56-9, "A Direct Computer Solution for Plates and Pavement Slabs" by C. Fred Stelzer, Jr., and W. Ronald Hudson, describes a direct method for solving complex two-dimensional plate and slab problems.

Report No. 56-10, "A Finite-Element Method of Analysis for Composite Beams" by Thomas P. Taylor and Hudson Matlock, describes a method of analysis for composite beams with any degree of horizontal shear interaction.

Report No. 56-11, "A Discrete-Element Solution of Plates and Pavement Slabs Using a Variable-Increment-Length Model" by Charles M. Pearre, III, and W. Ronald Hudson, presents a method of solving for the deflected shape of freely discontinuous plates and pavement slabs subjected to a variety of loads.

Report No. 56-12, "A Discrete-Element Method of Analysis for Combined Bending and Shear Deformations of a Beam" by David F. Tankersley and William P. Dawkins, presents a method of analysis for the combined effects of bending and shear deformations.

Report No. 56-13, "A Discrete-Element Method of Multiple-Loading Analysis for Two-Way Bridge Floor Slabs" by John J. Panak and Hudson Matlock, presents a procedure which was developed for analysis of two-way bridge floor slabs.

ABSTRACT

A method of solving for the deflected shape of freely discontinuous plates and pavement slabs subjected to a variety of loads including transverse loads, in-plane forces, and externally applied couples is presented. The method is applicable to uniform and discontinuous plates and pavement slabs with freely-variable foundation support, including holes in the subgrade.

This is a direct method for solving the discrete-element equations, which are an exact mathematical representation of the discrete-element model. Other reports (Refs 4 and 8) have described previous solutions. In order to improve those solutions, however, a method of varying the size of the discrete grid of the problem was needed.

The model presented contains this capability by including variable increment lengths between stations. This gives the capability of greater accuracy in the solution near loads and changes in the stiffness of the slab. A computer program is presented which utilizes the equations developed from the variable increment model. Several example problems illustrate the method's advantages over previous methods. The results compare well with closed-form solutions and with other discrete-element methods.

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

TABLE OF CONTENTS

PREFACE iii

LIST OF REPORTS v

ABSTRACT vii

NOMENCLATURE xi

CHAPTER 1. INTRODUCTION

The Problem 1
Description of Report 1

CHAPTER 2. THEORY OF ELASTIC PLATES AND SLABS

General Plate Theory 3
The Isotropic Plate Equation 3
Slabs on Foundations 5
Discrete-Element Models 5

CHAPTER 3. VARIABLE GRID PATTERNS

Constant-Increment Pattern 9
Double Solution Pattern 9
Short Beam Pattern 11
Two-Dimensional Variable-Increment-Length Pattern 13
One-Dimensional Variable-Increment-Length Pattern 13

CHAPTER 4. THE VARIABLE-INCREMENT-LENGTH MODEL

The Physical Model 15
Input Values of the Model 15
Effects of Modeling 18

CHAPTER 5. FORMULATION AND SOLUTION OF EQUATIONS

Free-Body Analysis 19
Matrix Representation of General Equation 34
Method of Solution 36

CHAPTER 6. THE COMPUTER PROGRAM

The FORTRAN Program 43
Input Information 46
Output Information 47
Special Programming for Multiple Loading 48
Error Checks 48

CHAPTER 7. EXAMPLE PROBLEMS AND VERIFICATION OF THE METHOD

Example Problem 1 - Complex Bridge Approach Slab 51
Example Problem 2 - Slab on Foundation 54
Example Problem 3 - Simply-Supported Steel Plate 56
Example Problem 4 - Small-Scale Slab-on-Foundation Test . . . 56
Example Problem 5 - Simply-Supported Plate with Cut 58
Example Problem 6 - Continuous Bridge Slab 58
Results of Example Problems 61

CHAPTER 8. SUMMARY 63

REFERENCES 65

APPENDICES

Appendix 1. Operating Manual for Program VISAB 3 71
Appendix 2. Flow Diagram for Program VISAB 3 101
Appendix 3. Program Listing of VISAB 3 151
Appendix 4. Hand and Machine Listings of Coded Data Input - Example Programs 241
Appendix 5. Sample Output Forms - Table 8 Only 253
Appendix 6. Sample Computer Solutions - Example Problems . . . 259

NOMENCLATURE

Symbol	Typical Units	Definition
A_j	-	Continuity coefficient
a, b, c, d	-	Temporary bar numbering used in derivations
$a_{i,j}$	lb/in	Term of stiffness matrix
$AA1_j$ to $AA5_j$	lb/in	Submatrix of stiffness matrix
$AA6_j$	lb	Submatrix of load matrix
B_j	-	Continuity coefficient
$b_{i,j}$	lb/in	Term of stiffness matrix
C_j	-	Continuity coefficient
C_x , C_y	$\frac{\text{in-lb}}{\text{rad}}$	Torsional stiffness of slab element
$C^x_{i,j}$	$\frac{\text{in-lb}}{\text{rad}}$	Torsional stiffness of Slab Element i,j about the x-axis
$C^y_{i,j}$	$\frac{\text{in-lb}}{\text{rad}}$	Torsional stiffness of Slab Element i,j about the y-axis
$C^{x'}_{i,j}$	in-lb	Torque exerted on the x-beam due to the relative rotation in Torsion Element i,j
$C^{y'}_{i,j}$	in-lb	Torque exerted on the y-beam due to the relative rotation in Torsion Element i,j
$c_{i,j}$	lb/in	Term of stiffness matrix
D	in-lb	Bending stiffness of plate
D_x , D_y	in-lb	Bending stiffness of plate
$D^x_{i,j}$	in-lb	Bending stiffness of an orthotropic plate in the x-direction

Symbol	Typical Units	Definition
$D^{y}_{i,j}$	in-lb	Bending stiffness of an orthotropic plate in the y-direction
D_j	-	Continuity coefficient
$d_{i,j}$	lb/in	Term of stiffness matrix
E_x	lb/in^2	Modulus of elasticity in x-direction
E_y	lb/in^2	Modulus of elasticity in y-direction
E_j	-	Continuity coefficient
$e_{i,j}$	lb/in	Term of stiffness matrix
$f_{i,j}$	lb/in	Term of stiffness matrix
G_o	lb/in^2	Approximate orthotropic shear modulus $G_o = \frac{E_x E_y}{E_y(1 + \nu_{xy}) + E_x(1 + \nu_{yx})}$
$g_{i,j}$	lb/in	Term of stiffness matrix
$h_{i,j}$	lb/in	Term of stiffness matrix
h^{x}_{i}	in.	The increment length along the x-beams
h^{y}_{j}	in.	The increment length along the y-beams
i	-	An integer used to index stations, and bars in the x-direction
j	-	An integer used to index stations, and bars in the y-direction
K	lb/in	Stiffness matrix
k	lb/in^3	Modulus of subgrade support
M_x	in-lb	Bending moment acting on an element of the plate in the x-direction

Symbol	Typical Units	Deflection
M_y	in-lb	Bending moment acting on an element of the plate in the y-direction
M_{xy}	in-lb	Twisting moment tending to rotate the element about the x-axis (clockwise positive)
M_{yx}	in-lb	Twisting moment tending to rotate the element about the y-axis (clockwise positive)
$M^{x'}_{i,j}$	in-lb	The bending moment in the x-beam at Station i,j (equals $h_y M^x_{i,j}$)
$M^{y'}_{i,j}$	in-lb	The bending moment in the y-beam at Station i,j (equals $h_x M^y_{i,j}$)
$M^x_{i,j}$	lb	Unit bending moment in the slab in the x-direction at Station i,j
$M^y_{i,j}$	lb	Unit bending moment in the slab in the y-direction at Station i,j
m	-	Counting integer
m_x	-	Number of increments in the x-direction
m_y	-	Number of increments in the y-direction
n	-	Counting integer
$P^x_{i,j}$	lb	Axial load per beam in the x-direction
$P^y_{i,j}$	lb	Axial load per beam in the y-direction
$\Delta P^x_{i,j}$	lb	Change in axial load in the x-beam occurring at Station i,j
$P_{i,j}$	lb/in	Term of stiffness matrix
Q	lb	Concentrated lateral load
$Q_{i,j}$	lb	Externally applied load at Point i,j
q	lb/in^2	Distributed lateral load

Symbol	Typical Units	Definition
$q_{i,j}$	lb/in	Term of stiffness matrix
$r_{i,j}$	lb/in	Term of stiffness matrix
$S_{i,j}$	lb/in	Elastic restraint used to represent the foundation in the discrete-element model
$s_{i,j}$	lb/in	Term in stiffness matrix
$T^{x}_{i,j}$	lb/in	External torque applied to Bar i on the j^{th} x-beam
$T^{y}_{i,j}$	lb/in	External torque applied to Bar j on the i^{th} y-beam
$t_{i,j}$	lb/in	Term of stiffness matrix
$u_{i,j}$	lb/in	Term of load matrix
$V^{x}_{a,j}$	lb	Shear in Bar a of the j^{th} x-beam
$V^{x}_{i,c}$	lb	Shear in Bar a of the i^{th} y-beam
W_{j}	in.	Submatrix of deflection matrix
$w_{i,j}$	in.	Lateral deflection
x, y, z	-	Standard Cartesian coordinate directions
ν	-	Poisson's ratio
ν_{xy}	-	Poisson's ratio which results in strain in the y-direction if stress is applied in the x-direction
ν_{yx}	-	Poisson's ratio which results in strain in the x-direction if stress is applied in the y-direction
$\Sigma F_{v_{i,j}}$	lb	Summation of vertical forces at Joint i,j

CHAPTER 1. INTRODUCTION

The Problem

Two excellent methods for the solution of discontinuous plates and pavement slabs have been described by Hudson (Refs 3 and 4) and Stelzer (Refs 7 and 8). The principal feature of these discrete-element methods is a physical model representation of the plate or slab as a system of orthogonal beams. Hudson's method utilizes an alternating-direction iterative technique for solving for the deflections of the plate or slab. Stelzer's method utilizes a direct matrix manipulation technique to obtain the deflections. Investigation has shown the direct solution method to be the more efficient of the two methods, based on computation time. The modeling accuracy of both methods of solution increases with the number of increments used in the model. In investigating accuracy it was found that the increased number of increments generally affected the solution only near points of abrupt or rapid changes in load, support, or stiffness of the plate. Also, it was noted that when complex loading patterns occurred, the load could not always be placed at the correct location. Thus, a method was needed which permits the use of small increments near abrupt changes but allows large increments to be used in areas where detailed accuracy is not required.

Description of Report

This report describes a model which improves previous methods of solution. The development of the method starts in Chapter 2 with a brief discussion of the basic theory of plates and slabs and the discrete-element theory. The possible grid patterns for use in the variable-increment-length model are discussed in Chapter 3. The variable-increment-length model is presented in Chapter 4. A free-body analysis of the model is used in Chapter 5 to derive the general equation in finite-difference form. The method by which the equations are solved is also presented in Chapter 5. Chapter 6 presents a general description of the computer program. Verification of the method is handled in

Chapter 7 by solving several problems and comparing their solution with previous solutions. Further uses of the model are demonstrated by solving several additional example problems.

CHAPTER 2. THEORY OF ELASTIC PLATES AND SLABS

General Plate Theory

The bending of a plate depends to a large degree on its thickness in comparison with its other dimensions. Timoshenko (Ref 9) notes three kinds of plate bending: (1) thin plates with small deflections, (2) thin plates with large deflections, and (3) thick plates. This report deals only with the first, making the following assumptions:

(1) There is no deformation in the plate's middle plane.

(2) Planes of the plate which initially lie normal to the middle surface of the plate remain normal to the middle surface of the plate after bending.

(3) The normal stresses in the direction transverse to the plate can be disregarded.

With these assumptions, all components of stress can be expressed in terms of the deflected shape of the plate. This function has to satisfy a linear partial differential equation which, together with the boundary conditions, completely defines the deflection w. The solution of this differential equation gives all necessary information for calculating the stresses at any point in the plate.

The Isotropic Plate Equation

The deflected surfaces of plates and slabs which are subjected to loads applied perpendicular to their surfaces can be described by the biharmonic equation. Timoshenko's equation with the up direction (z-axis) changed to positive is given below.

$$\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_{yx}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} - \frac{\partial^2 M_{xy}}{\partial x \partial y} = q \qquad (2.1)$$

In Eq 2.1, M_x is the bending moment acting on an element of the plate in the x-direction, M_y is the bending moment acting on an element of the plate in the y-direction, M_{xy} is a twisting moment tending to rotate the element about the x-axis (clockwise positive), and M_{yx} is a twisting moment tending to rotate the element about the y-axis. Observing that for equilibrium $M_{xy} = -M_{yx}$, the equation can be condensed into the following form:

$$\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_y}{\partial y^2} - \frac{2\partial^2 M_{xy}}{\partial x \partial y} = q \tag{2.2}$$

It is shown in Ref 3 that for the special case of isotropy, the moment equations can be stated as follows:

$$M_x = D\left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \tag{2.3}$$

$$M_y = D\left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \tag{2.4}$$

$$M_{xy} = -M_{yx} = -D(1 - \nu) \frac{\partial^2 w}{\partial x \partial y} \tag{2.5}$$

where D is the bending stiffness of the plate, ν is Poisson's ratio, and and w is the deflection of the plate in the z-direction. The other terms have been previously defined.

These terms for the moments can be used to evaluate Eq 2.2, giving

$$D\left(\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} \right) = q \tag{2.6}$$

For a more complete discussion of Eq 2.6 see Chapter 2 of Ref 3. Closed-form solutions to Eq 2.6 are available for a number of special cases. These cases are solutions of homogeneous, isotropic plates, which are generally round with a finite radius or square with infinite dimensions in the x and y-directions. The loading conditions in most closed-form solutions are either uniform loads over the entire plate or a concentrated load in the center of the plate. As the plate becomes more complex, closed-form solutions are generally not available and an approximate numerical method must be used to solve the problem.

Slabs on Foundations

Although much work has been done on the bending of slabs on foundations, the most significant was accomplished by Westergaard (Refs 10, 11, and 12). In his solution to Eq. 2.6 Westergaard made a number of limiting assumptions (see Ref 8, p 5). The assumptions imposed by Westergaard in his solution to plate or slab bending are not met by most pavement slab problems. In the first place, Westergaard assumed an infinite plate, but pavement slabs in reality have finite dimensions. Second, concrete is not the homogeneous, isotropic, elastic solid which Westergaard assumed in his solution. Third, the subgrade which Westergaard assumed to be uniform and linearly elastic is not always uniform, due to settlements, and soil is not linearly elastic. Fourth, the reactions of the subgrade are not vertical only, and they are not always proportional to the deflection, as Westergaard assumed. Fifth, the slab is often not of a uniform thickness, as Westergaard assumed, since the subgrade is generally not perfectly level when the slab is poured.

Discrete-Element Models

The theories discussed above are based on infinitesimal calculus. Since many complex engineering problems do not properly fulfill the conditions governing the use of such calculus and cannot be solved by resorting to it, so-called "numerical" methods have been developed. In these methods, the differential equation concerned is replaced by its finite-difference equivalent, and the problem reduces to solving a large number of simultaneous algebraic equations instead of one complex differential equation. The finite-difference equations can be derived from a physical model of the real system.

In Ref 5, Matlock presents a discrete-element model which satisfies the differential equation of a beam-column when infinitesimal increments are used. The model (Fig 2.1) is made up of a system of rigid bars connected by springs which represent the stiffness of the beam. The differential equation of a beam-column corresponding to Eq 2.6 for a slab is

$$D \frac{\partial^4 w}{\partial x^4} = q \tag{2.7}$$

The discrete-element equation which describes the bending of the model of a beam-column is

$$\frac{D}{h^4} (w_{i-2} - 4w_{i-1} + 6w_i - 4w_{i+1} + w_{i+2}) = q \tag{2.8}$$

where h is the length of the rigid bars. Equation 2.8 is the equation for the bending of a uniform beam-column. The equation can be derived from a free-body analysis of the bar-spring model. If each spring is allowed to have a different stiffness, Eq 2.8 can be rederived as the following:

$$\frac{1}{h^4} \Big[D_{i-1} w_{i-2} - 2(D_{i-1} + D_i) w_{i-1} + (D_{i-1} + 4D_i + D_{i+1}) w_i$$

$$-2(D_i + D_{i+1}) w_{i+1} + D_{i+1} w_{i+2} \Big] = q \tag{2.9}$$

With Eq 2.9, the requirement that the stiffness of the beam-column be a constant has been removed. For a more complete development of the beam-column equations, it is suggested that Ref 5 be read.

In the development of the discrete-element slab model, Hudson (Ref 4) extended the bar-spring model to a slab. Using the slab bar-spring model (Fig 4.1), the discrete-element equation equivalent to Eq 2.6 was developed.

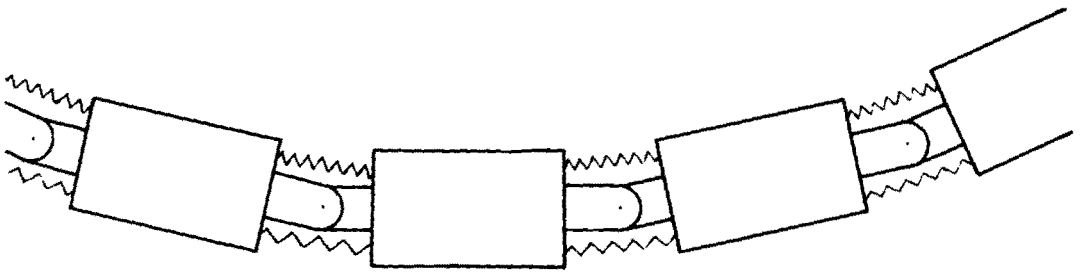


Fig 2.1. Discrete-element beam model.

Hudson used an alternating-direction iterative method to solve for the deflected shape of the slab model. Several problems with closed-form solutions were solved using the slab model to prove that the model was equivalent to the real problem. The deflections obtained from the solutions of the model were in good agreement with the closed-form deflections. As the increment length was decreased, the model solution came closer to being exactly the same as the closed-form solution. Because of the troublesome closure parameters required in the iterative method, Stelzer (Ref 8) developed a model for the direct solution of the slab model. Stelzer's discrete-element equation was derived from a free-body analysis of the model as a single system. The direct solution method produced the same results as the iterative method for any given problem, but was generally much faster.

Since the model has a number of nodal points (or stations), the properties of the slab can be allowed to vary at each of these points. This eliminates a number of the limitations on the properties of the slab and its supports. The model has been shown to become more accurate as the increment length is decreased. The problem with trying to increase the accuracy by decreasing the increment length was that the amount of computer time used increased faster than the increment length decreased. Therefore, it is necessary to find a model which will allow the accuracy of the solution to be increased without materially increasing the amount of computer time required for the solution. This could best be accomplished by some method which would allow the increment length to be variable from bar to bar. The following chapter presents a number of possible grid patterns that might be used to increase the accuracy of the model.

CHAPTER 3. VARIABLE GRID PATTERNS

In order to make best use of the discrete-element solution, some method is needed to permit better resolution of the problem near loads or other abrupt changes. This requires smaller increment lengths in selected locations. A variable-increment-length model was chosen in order to allow the use of these smaller increment lengths. In choosing a variable-increment-length model, five possible variable grid patterns were considered. The four grid patterns that are not used are discussed before the one-dimensional variable increment pattern that was selected for the model.

Constant-Increment Pattern

In his model Hudson uses a constant increment-length for all bars in a given direction. The station pattern of this model is shown in Fig 3.1. The model can be made more accurate by increasing the number of stations used. Complex loading cases can be solved with the loads located in exact positions using the constant-increment model if enough increments are used.

Since the constant-increment-length model requires all bars in one direction to be the same length, the model requires a very large number of stations to solve slabs with complex loading or to increase the accuracy of the solution near loads. Since the time of solution increases as a cubic function of the number of increments, the use of large numbers of increments is inefficient. Also, the constant-increment-length model has been shown to be inefficient since the solutions at stations some distance from the load are not affected by the increase in the number of increments. Therefore, in order to improve the accuracy of the solution some other grid pattern is needed.

Double Solution Pattern

The second possible grid pattern is a double grid. This method would involve first solving for the deflections of the slab using the constant-increment model with large increments. Then a section (Fig 3.2) would be cut out around

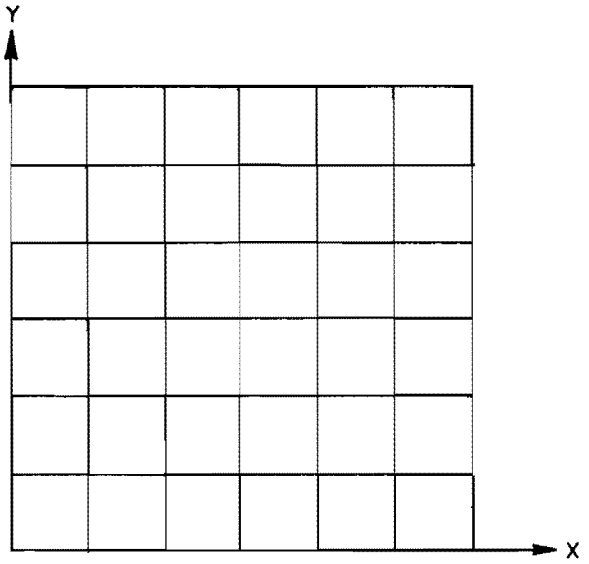


Fig 3.1. Constant-increment grid pattern.

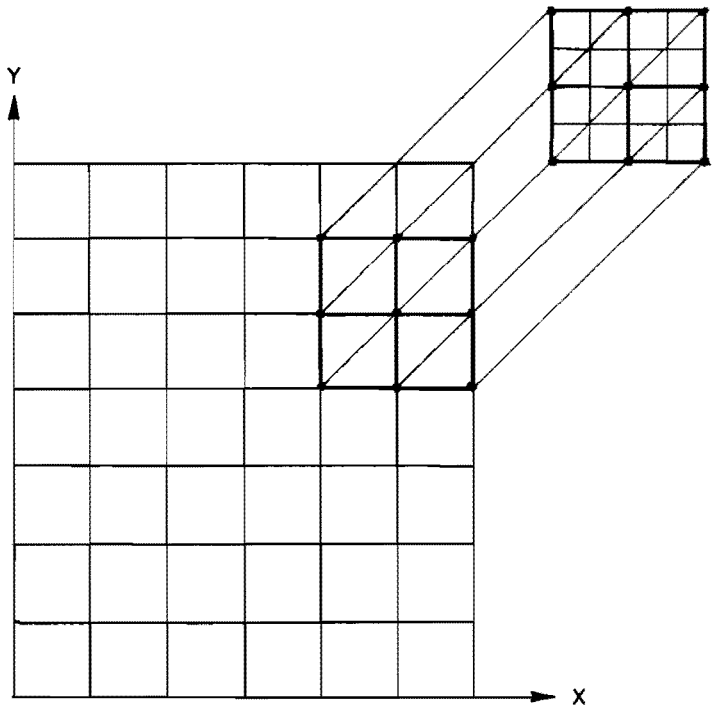


Fig 3.2. Double solution grid pattern.

the load or area in question and this section would be solved with boundary conditions applied around the edges based on the solution of the total slab. This section would have smaller increment lengths than the remainder of the slab. With the moments and deflections of the section known, equivalent loads would be calculated for the stations of the total slab that are also part of the section. The total slab would then be re-solved with these equivalent loads placed on it. If the boundary conditions that were applied to the section during its first solution change when the total slab is re-solved, the section must be re-solved. Thus, an iterative solution develops in which the closure is based on boundary conditions and equivalent loads remaining constant from one iteration to the next.

This type of solution has several disadvantages which prevent it from being an efficient method to use in solving the problem. The first is that the method requires two full solutions of the basic stiffness matrices. The disadvantage increases if the stiffness matrix of the cut section has to be solved for each iteration. The stiffness matrix of the section would have to be re-solved each time if the boundary conditions for the section could not be modeled as loads. The method also requires that boundary conditions and equivalent loads be found for the section at each iteration. These disadvantages mean an increased amount of computer time is required for the solution. The method also has the disadvantage that two sets of variable arrays must be used to store the information about the total slab and the section. If the secondary problem technique (saving the recursion coefficients and not re-solving the stiffness matrices) is used, the double model method requires four external tape units to solve the problem, instead of the two external tape units required by the constant-increment-length method.

Short Beam Pattern

The short beam model would introduce additional mesh points connected by bars in the area near loads or other points of interest. An example of this station system is shown in Fig 3.3. This method requires fewer station numbers since the beams do not extend across the entire slab and no stations are duplicated. Based on computation time, therefore, this is the most efficient method of establishing variable increment lengths.

The method is not used because the standard slab finite-difference operator (Fig 3.4) cannot be used with this system. The operator would require

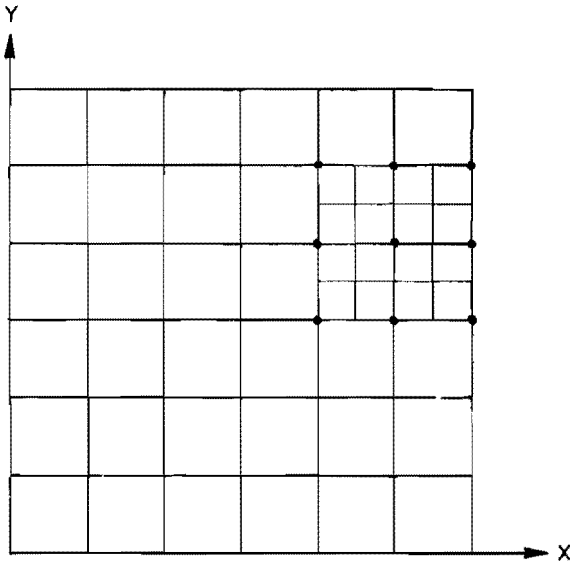


Fig 3.3. Short beam grid pattern.

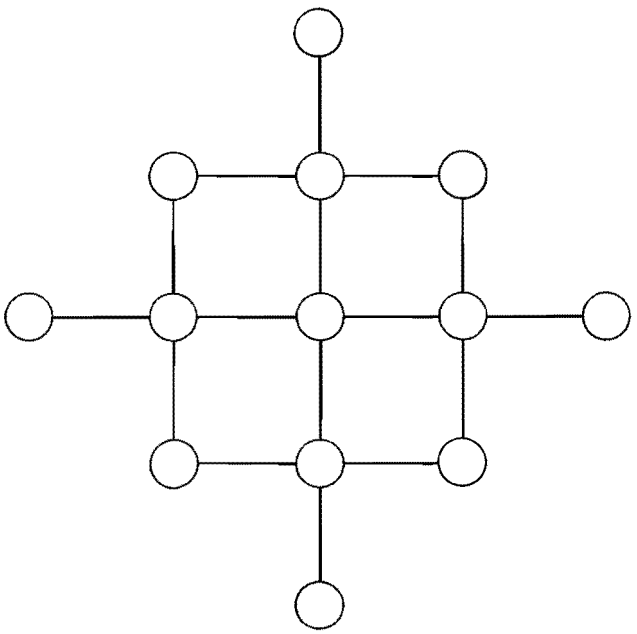


Fig 3.4. Finite-difference operator.

internal boundary conditions (special equations) at the ends of the short beams in order to be used. Also, the model numbering system used by Hudson and Stelzer cannot be applied to this model.

Two-Dimensional Variable-Increment-Length Pattern

A fourth possible arrangement of the stations on the slab would be to allow the lengths of all bars to vary independently of each other, requiring only that the beams be continuous across the slab (Fig 3.5). The disadvantage of this system is that the equations for a nonstraight beam have not been fully developed. In many respects such nonrectangular elements would resemble the triangular finite elements developed at the University of California. This approach has not been investigated in this report but bears further study.

The so-called two-dimensional variable-increment-length pattern was not chosen because of the complexity of the resulting equations and because the relationship of the model to the prototype slab is difficult to visualize.

One-Dimensional Variable-Increment-Length Pattern

The pattern that was chosen uses a grid system which allows the increment length along a beam in one direction to vary, but requires all beams in that direction to have the same pattern of increment lengths. Figure 3.6 shows an example of this grid system. The length of the increments is denoted by h_a^x and h_b^y, where x and y name the direction of the beam and a and b are increment numbers.

An advantage of this pattern is that it uses the standard slab operator (Fig 3.4) and the solution process for this pattern can be similar to Stelzer's solution process. When h_a^x and h_b^y are input as constants across the pattern, the pattern becomes the constant-increment-length pattern. The method requires less data storage than the other methods, so it is possible to solve larger problems. The solution is obtained with a single pass technique which requires less computer time than iterative techniques. An explanation of how the one-dimensional variable-increment-length station system is used in the bar-spring model is given in Chapter 4.

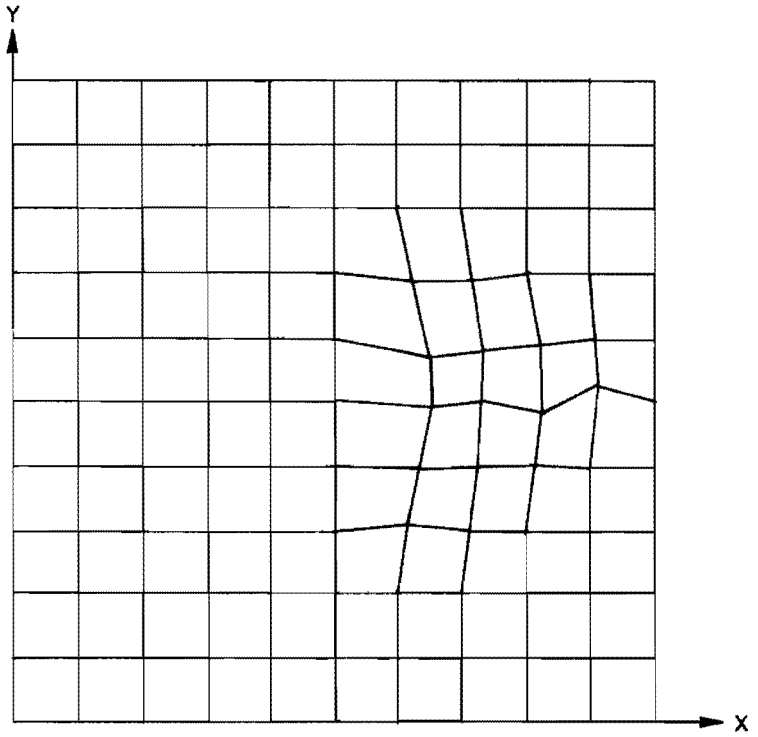


Fig 3.5. Two-dimensional variable-increment grid pattern.

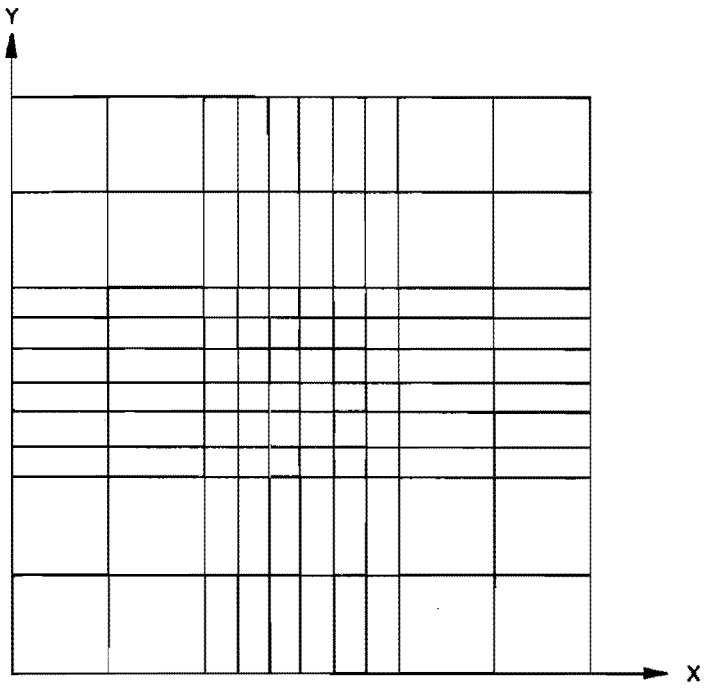


Fig 3.6. One-dimensional variable-increment grid pattern.

CHAPTER 4. THE VARIABLE-INCREMENT-LENGTH MODEL

This chapter applies the grid pattern selected in Chapter 3 to a discrete-element model. The input values for the variable-increment-length model are discussed, together with their effect on the problem.

The Physical Model

The method used here is to model the slab or plate physically by a system of discrete elements whose behavior can be described with algebraic equations. The physical model seems preferable to other numerical methods because it facilitates visualization of the problem and formulation of proper boundary and loading conditions. The difference equation can then be derived to describe the bending moment in the model.

Figure 4.1 is a pictorial representation of the discrete-element model of the slab as suggested by Hudson (Ref 3). This model is modified by changing the constant grid pattern to a one-dimensional variable-increment-length pattern in order to form the variable-increment-length model. A pictorial representation of the variable-increment-length model is shown in Fig 4.2. The Poisson's ratio effects and the bending stiffness of the plate are represented by elastic blocks at node points of the slab. The elastic blocks have a stress-strain ratio equivalent to the real plate and have Poisson's ratio equal to that of the plate. The torsion elements represent the real torsional stiffness of the plate. Proper values of torsional stiffness must always be input in plate and slab problems.

Input Values of the Model

It is necessary to relate the model to the real problem. The plate is divided into increments in the x and y-directions with increments h_i^x and h_j^y, respectively. These "beam" increments are designated with i in the x-direction and j in the y-direction. The station or joint at the positive end of each increment is numbered with the same number as that increment.

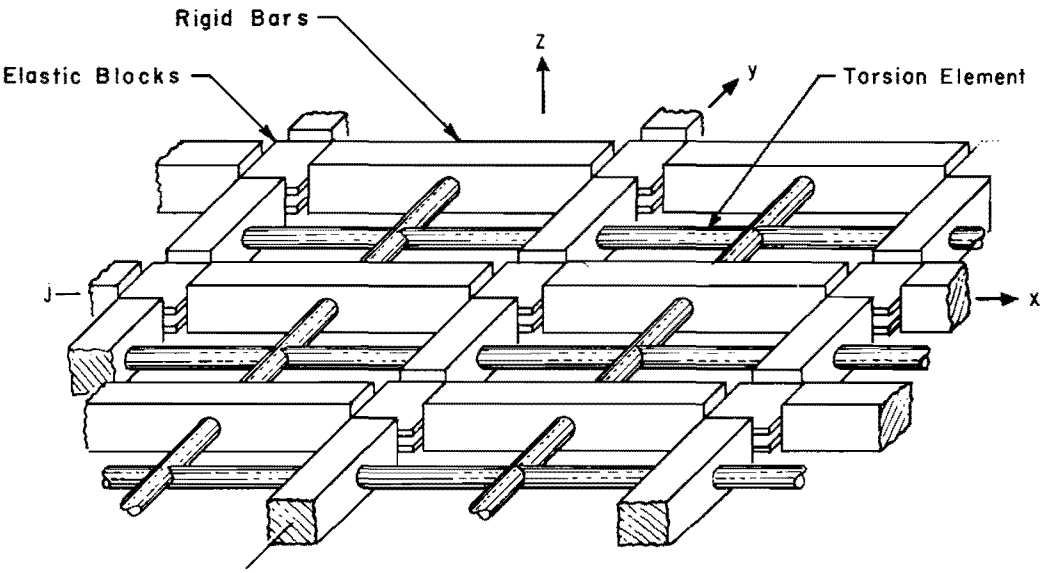


Fig 4.1. Constant-increment model.

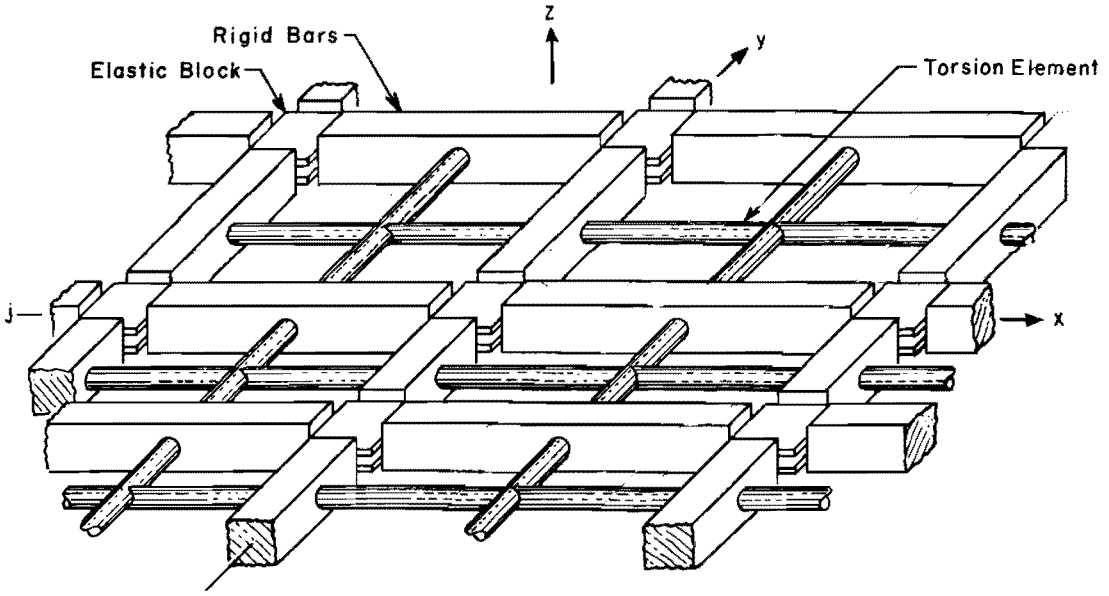


Fig 4.2. Variable-increment model.

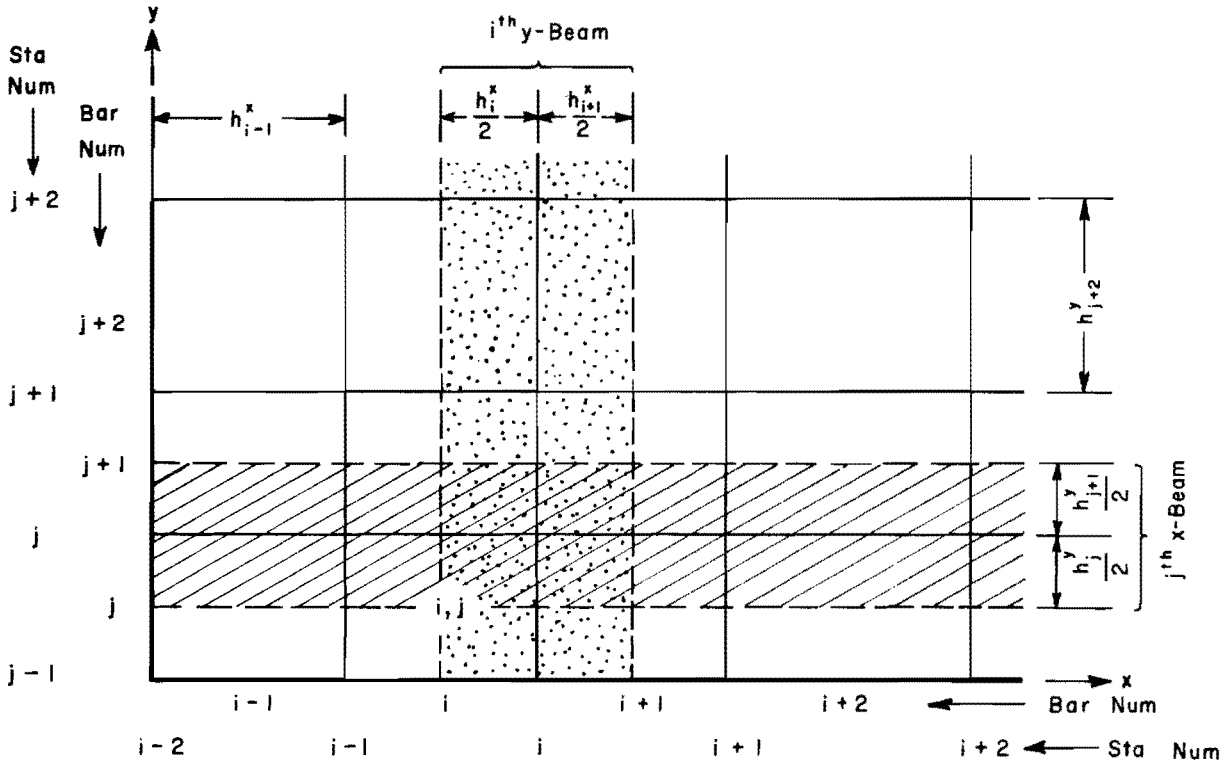


Fig 4.3. Plan view of plate segment divided into x and y-beams.

This numbering system then gives the i,j grid indicated in the plate segment in Fig 4.3. It can be noted from Fig 4.3 that for all x-beams the length of the i^{th} increment is h_i^x ; and, therefore, the increment length can be defined by a one-dimensional variable. The bending stiffness $D_{i,j}$ for a plate is a unit value per inch of width. It is convenient for use in computations to input the average stiffness over a full increment width. $D^x_{i,j}$ represents the average stiffness in the x-direction, that is, the average stiffness over an area centered at Station i,j which is $(h^y_j + h^y_{j+1})/2$ wide and $(h^x_i + h^x_{i+1})/2$ long. The torsional stiffness $C_{i,j}$ of the plate segment is a unit value per inch of width. The torsion elements are numbered with the joint number of the positive end joint of the beam at the positive end of the torsion elements. Axial loads P are also input into the bars with the changes ΔP considered to occur at stations.

Effects of Modeling

Errors resulting from this method are caused by approximating the real slab with a model. Special attention must be paid to the ratios of h^x_i to h^x_{i+1} and h^y_j to h^y_{j+1} . The ratio of adjacent increment lengths should not be too great. A number of problems have been run to check the ratio of increment lengths and it has been found that the maximum allowable ratio depends on the problem being solved. As a general rule, the solution should be carefully checked if h^x_i/h^x_{i+1} or h^y_j/h^y_{j+1} exceeds five or is less than 0.20. The ratio of h^x_i to h^x_{i+n} , where n is greater than one, is not limited by this five-to-one ratio. The algebraic solution is exact for the model within computer accuracy. Therefore, the closer the model duplicates the real slab, the more precise the answers computed by this method will be. The number of increments to be used will depend on the size of the problem and the load complexity as well as the accuracy required for the particular solution.

CHAPTER 5. FORMULATION AND SOLUTION OF EQUATIONS

The purpose of this chapter is to formulate from a free-body analysis the equations needed to solve for the bending of the slab and to present the method of solution for these equations. Equations are presented concisely rather than in a complete mathematical development.

Free-Body Analysis

In order to derive the equations for the solution of the bending of a slab, it is helpful to refer to a free-body of the model. Consider first a section of the assembled slab model centered at any Station i,j (Fig 5.1). Temporarily, the four bars intersecting at Station i,j will be known as Bars a, b, c, and d as shown in Fig 5.1. The lengths of Bars a, b, c, and d are, respectively, h_a^x , h_b^x , h_c^y , and h_d^y .

Figure 5.2 represents a free-body of the slab joint with all appropriate internal and external forces and reactions shown. Summing vertically at Joint i,j with up taken as positive gives

$$\Sigma F_{v_{i,j}} = Q_{i,j} + V^x_{a,j} + V^y_{i,c} - V^x_{b,j} - V^y_{i,d} - S_{i,j}w_{i,j} = 0 \quad (5.1)$$

By taking the summation of moments on a bar about a joint it is seen that

$$-h^x_a V^x_{a,j} = C^{x'}_{i,j} + C^{x'}_{i,j+1} + T^x_{a,j} + M^{x'}_{i-1,j} - M^{x'}_{i,j}$$

$$+ P^x_{a,j}(-w_{i-1,j} + w_{i,j}) \quad (5.2)$$

$$-h^x_b V^x_{b,j} = C^{x'}_{i+1,j} + C^{x'}_{i+1,j+1} + T^x_{b,j} + M^{x'}_{i,j} - M^{x'}_{i+1,j}$$

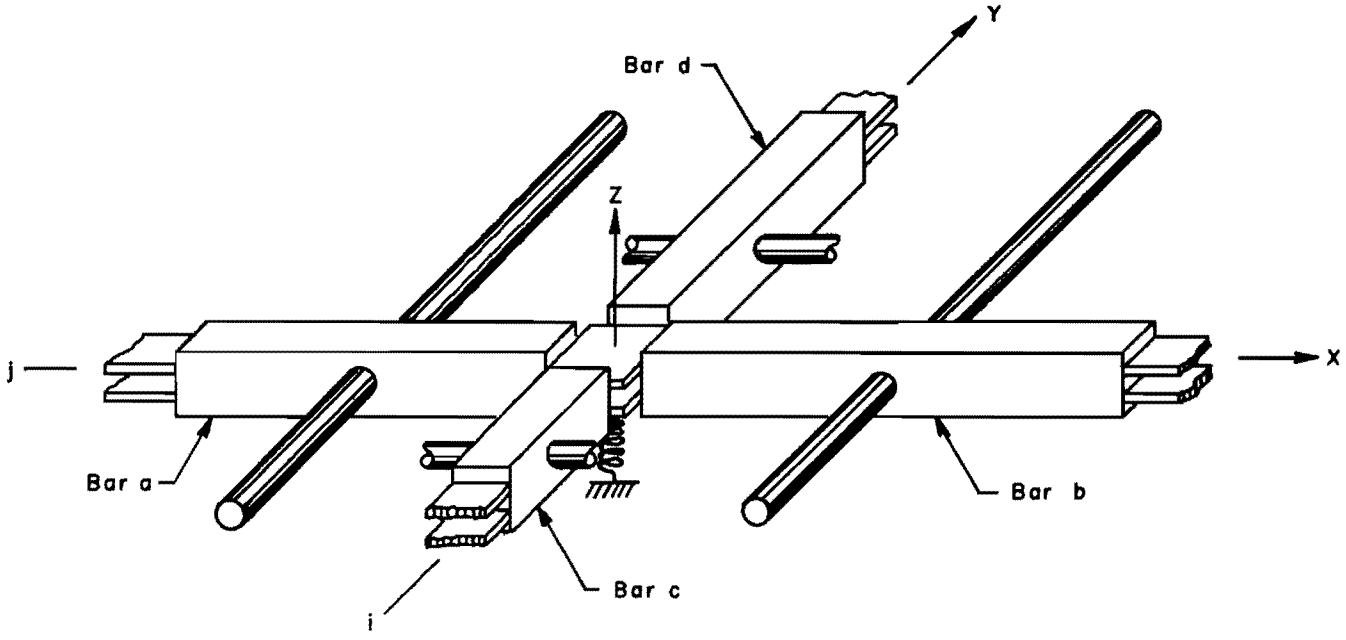


Fig 5.1. Typical Joint i,j taken from discrete-element slab model.

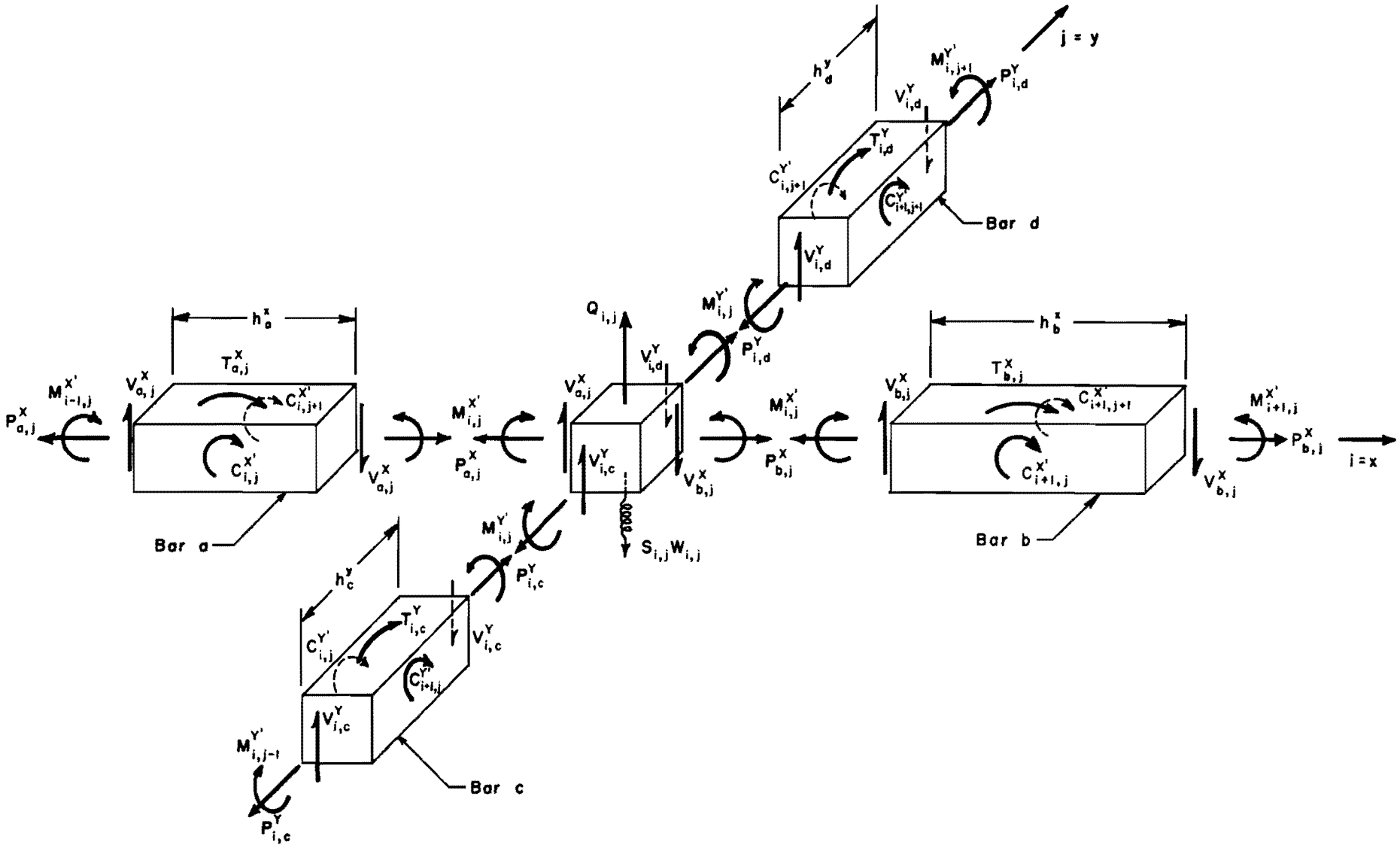


Fig 5.2. Free-body of slab joint.

$$+ P^{x}_{b,j}(-w_{i,j} + w_{i+1,j}) \tag{5.3}$$

$$-h^{y}_{c}V^{y}_{i,c} = C^{y'}_{i,j} + C^{y'}_{i+1,j} + T^{y}_{i,c} + M^{y'}_{i,j-1} - M^{y'}_{i,j}$$

$$+ P^{y}_{i,c}(-w_{i,j-1} + w_{i,j}) \tag{5.4}$$

$$-h^{y}_{d}V^{y}_{i,d} = C^{y'}_{i,j+1} + C^{y'}_{i+1,j+1} + T^{y}_{i,d} + M^{y'}_{i,j} - M^{y'}_{i,j+1}$$

$$+ P^{y}_{i,d}(-w_{i,j} + w_{i,j+1}) \tag{5.5}$$

Substituting the values for $V^{x}_{a,j}$, $V^{x}_{b,j}$, $V^{y}_{i,c}$, and $V^{y}_{i,d}$ as defined above into Eq 5.1 and solving for $Q_{i,j}$ results in the following equation:

$$Q_{i,j} = \frac{1}{h^{x}_{a}}\left[C^{x'}_{i,j} + C^{x'}_{i,j+1} + T^{x}_{a,j} + M^{x'}_{i-1,j} - M^{x'}_{i,j}\right.$$

$$\left. + P^{x}_{a,j}(-w_{i-1,j} + w_{i,j}) \right] + \frac{1}{h^{y}_{c}}\left[C^{y'}_{i,j} + C^{y'}_{i+1,j} + T^{y}_{i,c}\right.$$

$$\left. + M^{y'}_{i,j-1} - M^{y'}_{i,j} + P^{y}_{i,c}(-w_{i,j-1} + w_{i,j}) \right] + \frac{1}{h^{x}_{b}}\left[-C^{x'}_{i+1,j}\right.$$

$$\left. - C^{x'}_{i+1,j+1} - T^{x}_{b,j} - M^{x'}_{i,j} + M^{x'}_{i+1,j} - P^{x}_{b,j}(-w_{i,j}\right.$$

$$\left. + w_{i+1,j}) \right] + \frac{1}{h^{y}_{d}}\left[-C^{y'}_{i,j+1} - C^{y'}_{i+1,j+1} - T^{y}_{i,d}\right.$$

(equation continued)

$$- M^{y'}_{i,j} + M^{y'}_{i,j+1} - P^{y}_{i,d}(-w_{i,j} + w_{i,j+1}) \Big]$$

$$+ S_{i,j} w_{i,j} \tag{5.6}$$

At this point an additional note of clarification may be helpful. It is convenient in computation to use the same indexing systems for bars and torsion elements as for joints. So far, bars have been referred to as Bars a, b, c, and d. Figure 5.3 shows the numbering system used in the computer program. It may be seen that the index a becomes i , b becomes i+1 , c becomes j , and d becomes j+1 . Therefore, for example, h^x_a becomes h^x_i , $T^x_{b,j}$ becomes $T_{i+1,j}$, $P^y_{i,c}$ becomes $P^x_{i,j}$, etc.

The expressions for the $C^{x'}$ and $C^{y'}$ terms are derived directly from the model. The complete formulation for constant increment length can be found on page 139 of Ref 4. As noted in Ref 4 each $C^{x'}$ term involves an h^y term. Since the area between any four joints is always a rectangle, the h^y term for any $C^{x'}$ term is the length of the y-bar named by the j subscript of the $C^{x'}$ term. The $C^{x'}$ and $C^{y'}$ terms are shown below.

$$C^{x'}_{i,j} = \frac{C^x_{i,j}}{h^y_j}(w_{i-1,j-1} - w_{i,j-1} - w_{i-1,j} + w_{i,j}) \tag{5.7}$$

$$C^{x'}_{i,j+1} = \frac{C^x_{i,j+1}}{h^y_{j+1}}(-w_{i-1,j} + w_{i,j} + w_{i-1,j+1} - w_{i,j+1}) \tag{5.8}$$

$$C^{x'}_{i+1,j} = \frac{C^x_{i+1,j}}{h^y_j}(w_{i,j-1} - w_{i+1,j-1} - w_{i,j} + w_{i+1,j}) \tag{5.9}$$

$$C^{x'}_{i+1,j+1} = \frac{C^x_{i+1,j+1}}{h^y_{j+1}}(-w_{i,j} + w_{i+1,j} + w_{i,j+1} - w_{i+1,j+1}) \tag{5.10}$$

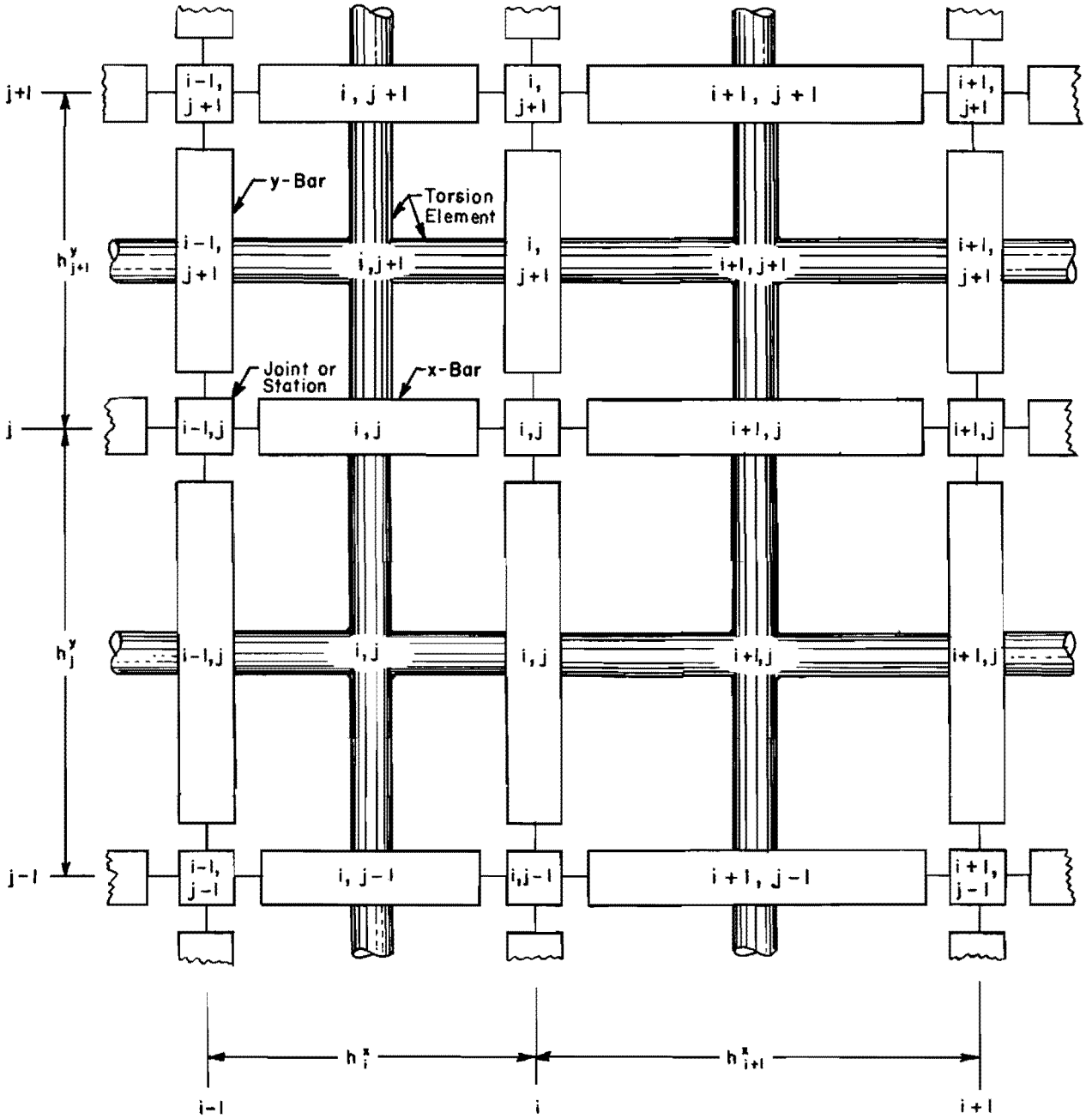


Fig 5.3. Plan view of the slab model showing all parts with generalized numbering system.

$$C^{y'}_{i,j} = \frac{C^{y}_{i,j}}{h^{x}_{i}} (w_{i-1,j-1} - w_{i-1,j} - w_{i,j-1} + w_{i,j}) \tag{5.11}$$

$$C^{y'}_{i+1,j} = \frac{C^{y}_{i+1,j}}{h^{x}_{i+1}} (-w_{i,j-1} + w_{i,j} + w_{i+1,j-1} - w_{i+1,j}) \tag{5.12}$$

$$C^{y'}_{i,j+1} = \frac{C^{y}_{i,j+1}}{h^{x}_{i}} (w_{i-1,j} - w_{i-1,j+1} - w_{i,j} + w_{i,j+1}) \tag{5.13}$$

$$C^{y'}_{i+1,j+1} = \frac{C^{y}_{i+1,j+1}}{h^{x}_{i+1}} (-w_{i,j} + w_{i,j+1} + w_{i+1,j} - w_{i+1,j+1}) \tag{5.14}$$

$M^{x'}$ and $M^{y'}$ expressions are found by introducing the finite-difference approximations for the second derivative.

$$M^{x'}_{i-1,j} = D^{x}_{i-1,j} \left(\frac{h^{y}_{j} + h^{y}_{j+1}}{2} \right)$$

$$\left[\frac{w_{i-2,j} h^{x}_{i} - w_{i-1,j}(h^{x}_{i-1} + h^{x}_{i}) + w_{i,j} h^{x}_{i-1}}{h^{x}_{i-1} h^{x}_{i} (h^{x}_{i-1} + h^{x}_{i})\frac{1}{2}} \right.$$

$$\left. + \nu_{yx} \frac{w_{i-1,j-1} h^{y}_{j+1} - w_{i-1,j}(h^{y}_{j} + h^{y}_{j+1}) + w_{i-1,j+1} h^{y}_{j}}{h^{y}_{j} h^{y}_{j+1} (h^{y}_{j} + h^{y}_{j+1})\frac{1}{2}} \right] \tag{5.15}$$

$$M^{x'}_{i,j} = D^{x}_{i,j} \left(\frac{h^{y}_{j} + h^{y}_{j+1}}{2} \right)$$

(equation continued)

$$\left[\frac{w_{i-1,j} h^x_{i+1} - w_{i,j}(h^x_i + h^x_{i+1}) + w_{i+1,j} h^x_i}{h^x_i h^x_{i+1}(h^x_i + h^x_{i+1})^{\frac{1}{2}}} \right.$$

$$\left. + \nu_{yx} \frac{w_{i,j-1} h^y_{j+1} - w_{i,j}(h^y_j + h^y_{j+1}) + w_{i,j+1} h^y_j}{h^y_j h^y_{j+1}(h^y_j + h^y_{j+1})^{\frac{1}{2}}} \right] \qquad (5.16)$$

$$M^{x'}_{i+1,j} = D^x_{i+1,j} \left(\frac{h^y_j + h^y_{j+1}}{2} \right)$$

$$\left[\frac{w_{i,j} h^x_{i+2} - w_{i+1,j}(h^x_{i+1} + h^x_{i+2}) + w_{i+2,j} h^x_{i+1}}{h^x_{i+1} h^x_{i+2}(h^x_{i+1} + h^x_{i+2})^{\frac{1}{2}}} \right.$$

$$\left. + \nu_{yx} \frac{w_{i+1,j-1} h^y_{j+1} - w_{i+1,j}(h^y_j + h^y_{j+1}) + w_{i+1,j+1} h^y_j}{h^y_{j+1} h^y_j (h^y_j + h^y_{j+1})^{\frac{1}{2}}} \right] \qquad (5.17)$$

$$M^{y'}_{i,j-1} = D^y_{i,j-1} \left(\frac{h^x_i + h^x_{i+1}}{2} \right)$$

$$\left[\frac{w_{i,j-2} h^y_j - w_{i,j-1}(h^y_{j-1} + h^y_j) + w_{i,j} h^y_{j-1}}{h^y_{j-1} h^y_j (h^y_{j-1} + h^y_j)^{\frac{1}{2}}} \right.$$

$$\left. + \nu_{xy} \frac{w_{i-1,j-1} h^x_{i+1} - w_{i,j-1}(h^x_i + h^x_{i+1}) + w_{i+1,j-1} h^x_i}{h^x_{i+1} h^x_i (h^x_i + h^x_{i+1})^{\frac{1}{2}}} \right] \qquad (5.18)$$

$$M^{y'}_{i,j} = D^y_{i,j} \left(\frac{h^x_i + h^x_{i+1}}{2} \right)$$

(equation continued)

$$\left[\frac{w_{i,j-1}h^y_{j+1} - w_{i,j}(h^y_j + h^y_{j+1}) + w_{i,j+1}h^y_j}{h^y_j h^y_{j+1}(h^y_j + h^y_{j+1})^{\frac{1}{2}}} \right.$$

$$\left. + \nu_{xy} \frac{w_{i-1,j}h^x_{i+1} - w_{i,j}(h^x_i + h^x_{i+1}) + w_{i+1,j}h^x_i}{h^x_{i+1}h^x_i(h^x_i + h^x_{i+1})^{\frac{1}{2}}} \right] \qquad (5.19)$$

$$M^{y'}_{i,j+1} = D^y_{i,j+1} \left(\frac{h^x_i + h^x_{i+1}}{2} \right)$$

$$\left[\frac{w_{i,j}h^y_{j+2} - w_{i,j+1}(h^y_{j+1} + h^y_{j+2}) + w_{i,j+2}h^y_{j+1}}{h^y_{j+1}h^y_{j+2}(h^y_{j+1} + h^y_{j+2})^{\frac{1}{2}}} \right.$$

$$\left. + \nu_{xy} \frac{w_{i-1,j+1}h^x_{i+1} - w_{i,j+1}(h^x_i + h^x_{i+1}) + w_{i+1,j+1}h^x_i}{h^x_{i+1}h^x_i(h^x_i + h^x_{i+1})^{\frac{1}{2}}} \right] \qquad (5.20)$$

Substituting for these values of C'_x, C'_y, M'_x, and M'_y into Eq 5.6, the full form of the equation is obtained.

$$Q_{i,j} = \frac{1}{h^x_i} \left\{ \frac{C^x_{i,j}}{h^y_j} (w_{i-1,j-1} - w_{i,j-1} - w_{i-1,j} + w_{i,j}) \right.$$

$$+ \frac{C^x_{i,j+1}}{h^y_{j+1}} (-w_{i-1,j} + w_{i,j} + w_{i-1,j+1} - w_{i,j+1})$$

$$\left. + D^x_{i-1,j} \left(\frac{h^y_j + h^y_{j+1}}{2} \right) \right.$$

(equation continued)

$$\left[\frac{w_{i-2,j}h_i^x - w_{i-1,j}(h_{i-1}^x + h_i^x) + w_{i,j}h_{i-1}^x}{h_{i-1}^x h_i^x (h_{i-1}^x + h_i^x)\frac{1}{2}} \right.$$

$$\left. + \nu_{yx} \frac{w_{i-1,j-1}h_{j+1}^y - w_{i-1,j}(h_j^y + h_{j+1}^y) + w_{i-1,j+1}h_j^y}{h_j^y h_{j+1}^y (h_j^y + h_{j+1}^y)\frac{1}{2}} \right]$$

$$- D_{i,j}^x \left(\frac{h_j^y + h_{j+1}^y}{2} \right) \left[\frac{w_{i-1,j}h_{i+1}^x - w_{i,j}(h_i^x + h_{i+1}^x) + w_{i+1,j}h_i^x}{h_i^x h_{i+1}^x (h_i^x + h_{i+1}^x)\frac{1}{2}} \right.$$

$$\left. + \nu_{yx} \frac{w_{i,j-1}h_{j+1}^y - w_{i,j}(h_j^y + h_{j+1}^y) + w_{i,j+1}h_j^y}{h_j^y h_{j+1}^y (h_j^y + h_{j+1}^y)\frac{1}{2}} \right] + T_{i,j}^x$$

$$\left. + P_{i,j}^x(-w_{i-1,j} + w_{i,j}) \right\} + \frac{1}{h_j^y} \left\{ \frac{C_{i,j}^y}{h_i^x} (w_{i-1,j-1} - w_{i-1,j} \right.$$

$$\left. - w_{i,j-1} + w_{i,j}) + \frac{C_{i+1,j}^y}{h_{i+1}^x} (-w_{i,j-1} + w_{i,j} + w_{i+1,j-1} - w_{i+1,j}) \right.$$

$$+ D_{i,j-1}^y \left(\frac{h_i^x + h_{i+1}^x}{2} \right) \left[\frac{w_{i,j-2}h_j^y - w_{i,j-1}(h_{j-1}^y + h_j^y) + w_{i,j}h_{j-1}^y}{h_{j-1}^y h_j^y (h_{j-1}^y + h_j^y)\frac{1}{2}} \right.$$

$$\left. + \nu_{xy} \frac{w_{i-1,j-1}h_{i+1}^x - w_{i,j-1}(h_i^x + h_{i+1}^x) + w_{i+1,j-1}h_i^x}{h_{i+1}^x h_i^x (h_i^x + h_{i+1}^x)\frac{1}{2}} \right]$$

$$- D_{i,j}^y \left(\frac{h_i^x + h_{i+1}^x}{2} \right) \left[\frac{w_{i,j-1}h_{j+1}^y - w_{i,j}(h_j^y + h_{j+1}^y) + w_{i,j+1}h_j^y}{h_j^y h_{j+1}^y (h_j^y + h_{j+1}^y)\frac{1}{2}} \right.$$

(equation continued)

$$+ \nu_{xy} \frac{w_{i-1,j} h^x_{i+1} - w_{i,j}(h^x_i + h^x_{i+1}) + w_{i+1,j} h^x_i}{h^x_i h^x_{i+1} (h^x_i + h^x_{i+1})\frac{1}{2}} \Big] + T^y_{i,j}$$

$$+ P^y_{i,j}(-w_{i,j-1} + w_{i,j}) \Big\} + \frac{1}{h^x_{i+1}} \Big\{ - \frac{C^x_{i+1,j}}{h^y_j} (w_{i,j-1} - w_{i+1,j-1}$$

$$- w_{i,j} + w_{i+1,j}) - \frac{C^x_{i+1,j+1}}{h^y_{j+1}} (-w_{i,j} + w_{i+1,j} + w_{i,j+1}$$

$$- w_{i+1,j+1}) - D^x_{i,j} \left(\frac{h^y_j + h^y_{j+1}}{2} \right)$$

$$\Big[\frac{w_{i-1,j} h^x_{i+1} - w_{i,j}(h^x_i + h^x_{i+1}) + w_{i+1,j} h^x_i}{h^x_i h^x_{i+1} (h^x_i + h^x_{i+1})\frac{1}{2}}$$

$$+ \nu_{yx} \frac{w_{i,j-1} h^y_{j+1} - w_{i,j}(h^y_j + h^y_{j+1}) + w_{i,j+1} h^y_j}{h^y_j h^y_{j+1} (h^y_j + h^y_{j+1})\frac{1}{2}} \Big]$$

$$+ D^x_{i+1,j} \left(\frac{h^y_j + h^y_{j+1}}{2} \right)$$

$$\Big[\frac{w_{i,j} h^x_{i+2} - w_{i+1,j}(h^x_{i+1} + h^x_{i+2}) + w_{i+2,j} h^x_{i+1}}{h^x_{i+1} h^x_{i+2} (h^x_{i+1} + h^x_{i+2})\frac{1}{2}}$$

$$+ \nu_{yx} \frac{w_{i+1,j-1} h^y_{j+1} - w_{i+1,j}(h^y_j + h^y_{j+1}) + w_{i+1,j+1} h^y_j}{h^y_i h^y_{j+1} (h^y_j + h^y_{j+1})\frac{1}{2}}$$

(equation continued)

$$- T^x_{i+1,j} - P^x_{i+1,j}(-w_{i,j} + w_{i+1,j}) \Big\} + \frac{1}{h^y_{j+1}} \Big\{ - \frac{C^y_{i,j+1}}{h^x_i} (w_{i-1,j}$$

$$- w_{i-1,j+1} - w_{i,j} + w_{i,j+1}) - \frac{C^y_{i+1,j+1}}{h^x_{i+1}} (-w_{i,j} + w_{i,j+1}$$

$$+ w_{i+1,j} - w_{i+1,j+1}) - D^y_{i,j} \left(\frac{h^x_i + h^x_{i+1}}{2} \right)$$

$$\left[\frac{w_{i,j-1} h^y_{j+1} - w_{i,j}(h^y_j + h^y_{j+1}) + w_{i,j+1} h^y_j}{h^y_{j+1} h^y_j (h^y_{j+1} + h^y_j)^{\frac{1}{2}}} \right.$$

$$\left. + \nu_{xy} \frac{w_{i-1,j} h^x_{i+1} - w_{i,j}(h^x_i + h^x_{i+1}) + w_{i+1,j} h^x_i}{h^x_i h^x_{i+1} (h^x_i + h^x_{i+1})^{\frac{1}{2}}} \right] + D^y_{i,j+1}$$

$$\left(\frac{h^x_i + h^x_{i+1}}{2} \right) \left[\frac{w_{i,j} h^y_{j+2} - w_{i,j+1}(h^y_{j+1} + h^y_{j+2}) + w_{i,j+2} h^y_{j+1}}{h^y_{j+1} h^y_{j+2} (h^y_{j+1} + h^y_{j+2})^{\frac{1}{2}}} \right.$$

$$\left. + \nu_{xy} \frac{w_{i-1,j+1} h^x_{i+1} - w_{i,j+1}(h^x_i + h^x_{i+1}) + w_{i+1,j+1} h^x_i}{h^x_i h^x_{i+1} (h^x_i + h^x_{i+1})^{\frac{1}{2}}} \right]$$

$$- T^y_{i,j+1} - P^y_{i,j+1}(-w_{i,j} + w_{i,j+1}) \Big\} + S_{i,j} w_{i,j} \qquad (5.21)$$

Now, rearranging Eq 5.21 into its final form as shown below

$$a_{i,j}w_{i,j-2} + b_{i,j}w_{i-1,j-1} + c_{i,j}w_{i,j-1} + d_{i,j}w_{i+1,j-1}$$

$$+ e_{i,j}w_{i-2,j} + f_{i,j}w_{i-1,j} + g_{i,j}w_{i,j} + h_{i,j}w_{i+1,j}$$

$$+ p_{i,j}w_{i+2,j} + q_{i,j}w_{i-1,j+1} + r_{i,j}w_{i,j+1} + s_{i,j}w_{i+1,j+1}$$

$$+ t_{i,j}w_{i,j+2} = u_{i,j} \tag{5.22}$$

where

$$a_{i,j} = \frac{h^x_i + h^x_{i+1}}{(h^y_{j-1} + h^y_j)h^y_{j-1}h^y_j} D^y_{i,j-1} \tag{5.23}$$

$$b_{i,j} = \frac{1}{h^x_i h^y_j} (\nu_{yx}D^x_{i-1,j} + \nu_{xy}D^y_{i,j-1} + C^x_{i,j} + C^y_{i,j}) \tag{5.24}$$

$$c_{i,j} = -\frac{h^x_i + h^x_{i+1}}{h^y_j}\left(\frac{D^y_{i,j-1}}{h^y_{j-1}h^y_j} + \frac{D^y_{i,j}}{h^y_j h^y_{j+1}}\right) - \frac{h^x_i + h^x_{i+1}}{h^x_i h^x_{i+1}h^y_j}$$

$$(\nu_{xy}D^y_{i,j-1} + \nu_{yx}D^x_{i,j}) - \frac{C^x_{i,j} + C^y_{i,j}}{h^x_i h^y_j} - \frac{C^x_{i+1,j} + C^y_{i+1,j}}{h^x_{i+1}h^y_j}$$

$$- \frac{P^y_{i,j}}{h^y_j} \tag{5.25}$$

$$d_{i,j} = \frac{1}{h^x_{i+1}h^y_j} (\nu_{yx}D^x_{i+1,j} + \nu_{xy}D^y_{i,j-1} + C^x_{i+1,j} + C^y_{i+1,j}) \tag{5.26}$$

$$e_{i,j} = \frac{h^y_j + h^y_{j+1}}{(h^x_i + h^x_{i-1})h^x_i h^x_{i-1}} D^x_{i-1,j}$$

$$f_{i,j} = -\frac{h^y_j + h^y_{j+1}}{h^x_i}\left(\frac{D^x_{i-1,j}}{h^x_{i-1}h^x_i} + \frac{D^x_{i,j}}{h^x_i h^x_{i+1}}\right) - \frac{h^y_j + h^y_{j+1}}{h^y_j h^y_{j+1} h^x_i}(\nu_{yx} D^x_{i-1,j}$$

$$+ \nu_{xy} D^y_{i,j}) - \frac{C^x_{i,j} + C^y_{i,j}}{h^x_i h^y_j} - \frac{C^x_{i,j+1} + C^y_{i,j+1}}{h^x_i h^y_{j+1}} - \frac{P^x_{i,j}}{h^x_i} \qquad (5.28)$$

$$g_{i,j} = (h^y_j + h^y_{j+1})\left[\frac{D^x_{i-1,j}}{(h^x_{i-1} + h^x_i)(h^x_i)^2} + \frac{D^x_{i,j}(h^x_i + h^x_{i+1})}{(h^x_i)^2(h^x_{i+1})^2}\right.$$

$$\left. + \frac{D^x_{i+1,j}}{(h^x_{i+1} + h^x_{i+2})(h^x_{i+1})^2}\right] + (h^x_i + h^x_{i+1})\left[\frac{D^y_{i,j-1}}{(h^y_{j-1} + h^y_j)(h^y_j)^2}\right.$$

$$\left. + \frac{D^y_{i,j}(h^y_j + h^y_{j+1})}{(h^y_j)^2(h^y_{j+1})^2} + \frac{D^y_{i,j+1}}{(h^y_{j+1} + h^y_{j+2})(h^y_{j+1})^2}\right]$$

$$+ \frac{(h^y_j + h^y_{j+1})(h^x_i + h^x_{i+1})}{h^x_i h^x_{i+1} h^y_j h^y_{j+1}}(\nu_{yx} D^x_{i,j} + \nu_{xy} D^y_{i,j}) + \frac{C^x_{i,j} + C^y_{i,j}}{h^x_i h^y_j}$$

$$+ \frac{C^x_{i,j+1} + C^y_{i,j+1}}{h^x_i h^y_{j+1}} + \frac{C^x_{i+1,j} + C^y_{i+1,j}}{h^x_{i+1} h^y_j} + \frac{C^x_{i+1,j+1} + C^y_{i+1,j+1}}{h^x_{i+1} h^y_{j+1}}$$

$$+ \frac{P^x_{i,j}}{h^x_i} + \frac{P^x_{i+1,j}}{h^x_{i+1}} + \frac{P^y_{i,j}}{h^y_j} + \frac{P^y_{i,j+1}}{h^y_{j+1}} + S_{i,j} \qquad (5.29)$$

$$h_{i,j} = -\frac{h^y_j + h^y_{j+1}}{h^x_{i+1}}\left(\frac{D^x_{i,j}}{h^x_i h^x_{i+1}} + \frac{D^x_{i+1,j}}{h^x_{i+1}h^x_{i+2}}\right) - \frac{(h^y_j + h^y_{j+1})}{h^x_{i+1}h^y_j h^y_{j+1}}$$

$$(\nu_{yx}D^x_{i+1,j} + \nu_{xy}D^y_{i,j}) - \frac{C^x_{i+1,j} + C^y_{i+1,j}}{h^x_{i+1}h^y_j} - \frac{C^x_{i+1,j+1} + C^y_{i+1,j+1}}{h^x_{i+1}h^y_{j+1}}$$

$$- \frac{P^x_{i+1,j}}{h^x_{i+1}} \tag{5.30}$$

$$p_{i,j} = \frac{h^y_j + h^y_{j+1}}{(h^x_{i+1} + h^x_{i+2})h^x_{i+1}h^x_{i+2}} D^x_{i+1,j} \tag{5.31}$$

$$q_{i,j} = \frac{1}{h^x_i h^y_{j+1}}(\nu_{yx}D^x_{i-1,j} + \nu_{xy}D^y_{i,j+1} + C^x_{i,j+1} + C^y_{i,j+1}) \tag{5.32}$$

$$r_{i,j} = -\frac{h^x_i + h^x_{i+1}}{h^y_{j+1}}\left(\frac{D^y_{i,j}}{h^y_j h^y_{j+1}} + \frac{D^y_{i,j+1}}{h^y_{j+1}h^y_{j+2}}\right)$$

$$- \frac{h^x_i + h^x_{i+1}}{h^x_i h^x_{i+1} h^y_{j+1}}(\nu_{yx}D^x_{i,j} + \nu_{xy}D^y_{i,j+1}) - \frac{C^x_{i,j+1} + C^y_{i,j+1}}{h^x_i h^y_{j+1}}$$

$$- \frac{C^x_{i+1,j+1} + C^y_{i+1,j+1}}{h^x_{i+1}h^y_{j+1}} - \frac{P^y_{i,j+1}}{h^y_{j+1}} \tag{5.33}$$

$$s_{i,j} = \frac{1}{h^x_{i+1} h^y_{j+1}} (\nu_{yx} D^x_{i+1,j} + \nu_{xy} D^y_{i,j+1} + C^x_{i+1,j+1}$$

$$+ C^y_{i+1,j+1}) \tag{5.34}$$

$$t_{i,j} = \frac{h^x_i + h^x_{i+1}}{(h^y_{j+1} + h^y_{j+2}) h^y_{j+1} h^y_{j+2}} D^y_{i,j+1} \tag{5.35}$$

$$u_{i,j} = Q_{i,j} - \frac{T^x_{i,j}}{h^x_i} + \frac{T^x_{i+1,j}}{h^x_{i+1}} - \frac{T^y_{i,j}}{h^y_j} + \frac{T^y_{i,j+1}}{h^y_{j+1}} \tag{5.36}$$

Matrix Representation of General Equation

The matrix representation of Eq 5.22 is

$$\left[K \right] \left[W \right] = \left[Q \right] \tag{5.37}$$

Figure 5.4 shows the form of the K , W , and Q matrices, about which several important things should be noted. The K (stiffness) matrix is symmetrical about its major diagonal and is also banded, that is, the terms lie in lines parallel to the major diagonal. The stiffness matrix can be partitioned into submatrices, as shown by the dashed lines. It is noted that each submatrix is banded and there are no more than five non-zero submatrices on any row. If the slab to be solved has been divided into m increments in the x-direction and n increments in the y-direction, the K matrix will have n+3 rows and n+3 columns of submatrices. The submatrices will have m+3 rows and m+3 columns of terms. Solution of slab problems involves manipulating the submatrices. It is more efficient to solve rectangular slab problems if m is less than or equal to n ; therefore m is required to be less than or equal to n . It is important to notice that terms in the five-wide banded submatrix do not appear in the submatrix which contains the three-wide band and vice versa and that the terms of a three-wide banded submatrix do not appear in the submatrix which contains the

GENERAL SLAB EQUATION:

$$a_{i,j}w_{i,j-2} + b_{i,j}w_{i-1,j-1} + c_{i,j}w_{i,j-1} + d_{i,j}w_{i+1,j-1} + e_{i,j}w_{i-2,j} + f_{i,j}w_{i-1,j} + g_{i,j}w_{i,j} + h_{i,j}w_{i+1,j} + p_{i,j}w_{i+2,j} + q_{i,j}w_{i-1,j+1} + r_{i,j}w_{i,j+1} + s_{i,j}w_{i+1,j+1} + t_{i,j}w_{i,j+2} = u_{i,j}$$

OR IN MATRIX FORM:

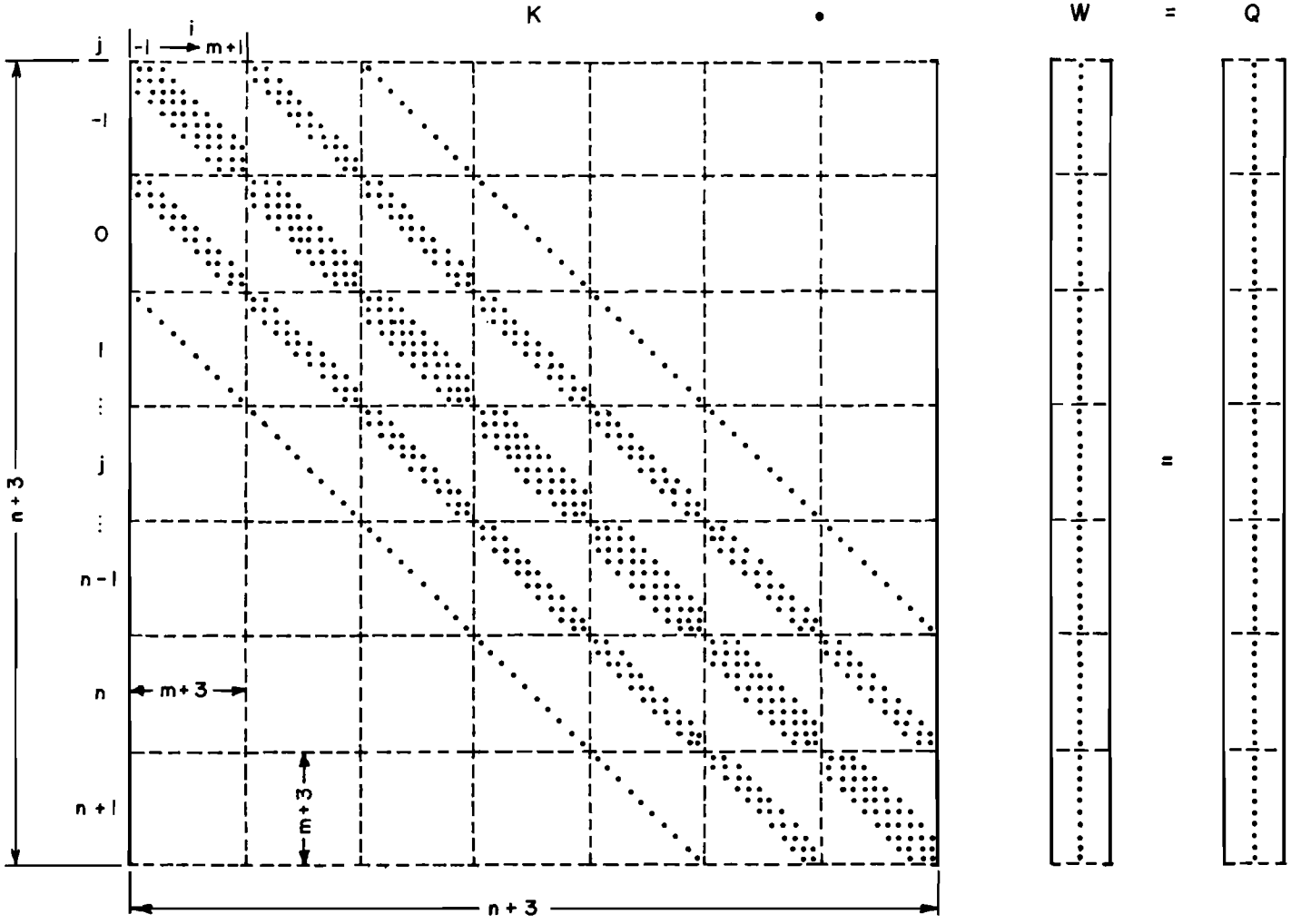


Fig 5.4. Form of K , W , and Q matrices.

single band and vice versa. The W (deflection) matrix and Q (load) matrix are column matrices. Figure 5.5 shows the part played by the terms of Eq 5.22 in a typical row of the submatrices.

Method of Solution

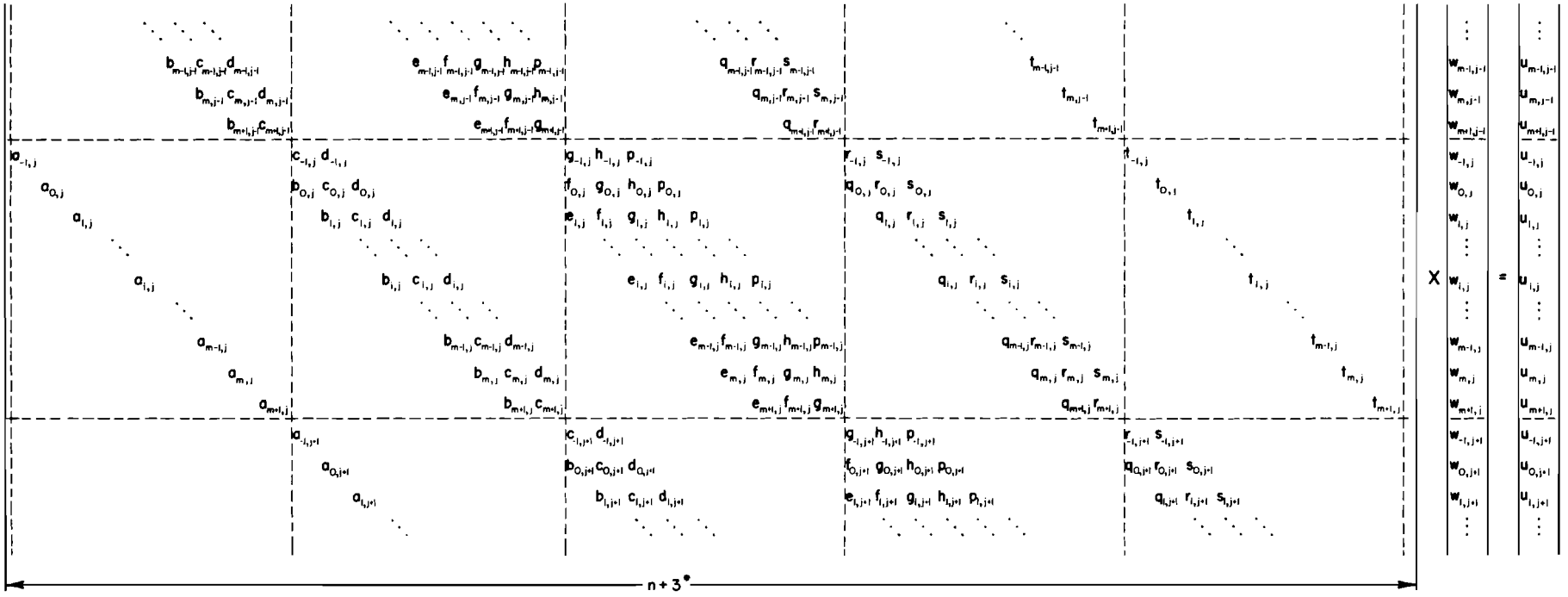
The general equation in matrix form is solved by a direct solution technique as described by Stelzer (Refs 7 and 8) with modifications by Endres (Ref 2). The solution method applies a recursive technique analogous to Matlock's method of solving beams and columns (Ref 5) to the matrix of five diagonal submatrices.

Referring to Figs 5.4 and 5.5 it is readily seen that the multitude of individual terms in the K , W , and Q matrices could be redefined as shown in Fig 5.6. In Fig 5.6 $AA1_j$, $AA2_j$, ... , $AA5_j$ are submatrices of the stiffness matrix, W_j is a submatrix of the deflection matrix, and Q_j is a submatrix of the load matrix. Applying this submatrix notation to the submatrices in Fig 5.5, the redefined matrices K , W , and Q can be shown as in Fig 5.7. From Fig 5.7 it is seen that the following equation is valid:

$$AA1_j W_{j-2} + AA2_j W_{j-1} + AA3_j W_j + AA4_j W_{j+1} + AA5_j W_{j+2} = AA6_j \qquad (5.38)$$

As shown in Fig 5.7, this equation results in a five-wide banded diagonal coefficient matrix, termed the "stiffness matrix," which when multiplied by the single-column "deflection matrix" is equal to a single-column "load matrix." Matlock in Ref 5 discusses a convenient method for the solution of an equation such as Eq 5.38 and states that this system of equations is most easily solved by a back-and-forth recursion process. Proceeding from $j = -1$ to $j = n+1$, two unknown deflections (W_{j-2} and W_{j-1} in Eq 5.38) are eliminated from the equation, resulting in another diagonally-banded system of equations of the form

$$W_j - B_j W_{j+1} - C_j W_{j+2} = A_j \qquad (5.39)$$



*The stiffness matrix is actually n+3 sub-matrices wide; only the five pertinent sub-matrices are shown in this figure

Fig 5.5. Terms in a typical row of submatrices.

$$AA1_j = \begin{bmatrix} a_{-1,j} & & & & & & & \\ & a_{0,j} & & & & & & \\ & & a_{1,j} & & & & & \\ & & & \ddots & & & & \\ & & & & a_{i,j} & & & \\ & & & & & \ddots & & \\ & & & & & & a_{m-1,j} & \\ & & & & & & & a_{m,j} \\ & & & & & & & & a_{m+1,j} \end{bmatrix}$$

$$AA2_j = \begin{bmatrix} c_{-1,j} & d_{-1,j} & & & & & & \\ b_{0,j} & c_{0,j} & d_{0,j} & & & & & \\ & b_{1,j} & c_{1,j} & d_{1,j} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & b_{i,j} & c_{i,j} & d_{i,j} & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & b_{m-1,j} & c_{m-1,j} & d_{m-1,j} \\ & & & & & & b_{m,j} & c_{m,j} & d_{m,j} \\ & & & & & & & b_{m+1,j} & c_{m+1,j} \end{bmatrix}$$

$$AA3_j = \begin{bmatrix} g_{-1,j} & h_{-1,j} & p_{-1,j} & & & & & & \\ f_{0,j} & g_{0,j} & h_{0,j} & p_{0,j} & & & & & \\ e_{1,j} & f_{1,j} & g_{1,j} & h_{1,j} & p_{1,j} & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & e_{i,j} & f_{i,j} & g_{i,j} & h_{i,j} & p_{i,j} & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & e_{m-1,j} & f_{m-1,j} & g_{m-1,j} & h_{m-1,j} & p_{m-1,j} \\ & & & & & e_{m,j} & f_{m,j} & g_{m,j} & h_{m,j} \\ & & & & & & e_{m+1,j} & f_{m+1,j} & g_{m+1,j} \end{bmatrix}$$

$$AA4_j = \begin{bmatrix} r_{-1,j} & s_{-1,j} & & & & & & \\ q_{0,j} & r_{0,j} & s_{0,j} & & & & & \\ & q_{1,j} & r_{1,j} & s_{1,j} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & q_{i,j} & r_{i,j} & s_{i,j} & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & q_{m-1,j} & r_{m-1,j} & s_{m-1,j} \\ & & & & & & q_{m,j} & r_{m,j} & s_{m,j} \\ & & & & & & & q_{m+1,j} & r_{m+1,j} \end{bmatrix}$$

$$AA5_j = \begin{bmatrix} t_{-1,j} & & & & & & & \\ & t_{0,j} & & & & & & \\ & & t_{1,j} & & & & & \\ & & & \ddots & & & & \\ & & & & t_{i,j} & & & \\ & & & & & \ddots & & \\ & & & & & & t_{m-1,j} & \\ & & & & & & & t_{m,j} \\ & & & & & & & & t_{m+1,j} \end{bmatrix}$$

$$W_j = \begin{bmatrix} w_{-1,j} \\ w_{0,j} \\ w_{1,j} \\ \vdots \\ w_{i,j} \\ \vdots \\ w_{m-1,j} \\ w_{m,j} \\ w_{m+1,j} \end{bmatrix} \qquad AA6_j = \begin{bmatrix} u_{-1,j} \\ u_{0,j} \\ u_{1,j} \\ \vdots \\ u_{i,j} \\ \vdots \\ u_{m-1,j} \\ u_{m,j} \\ u_{m+1,j} \end{bmatrix}$$

Fig 5.6. Submatrices.

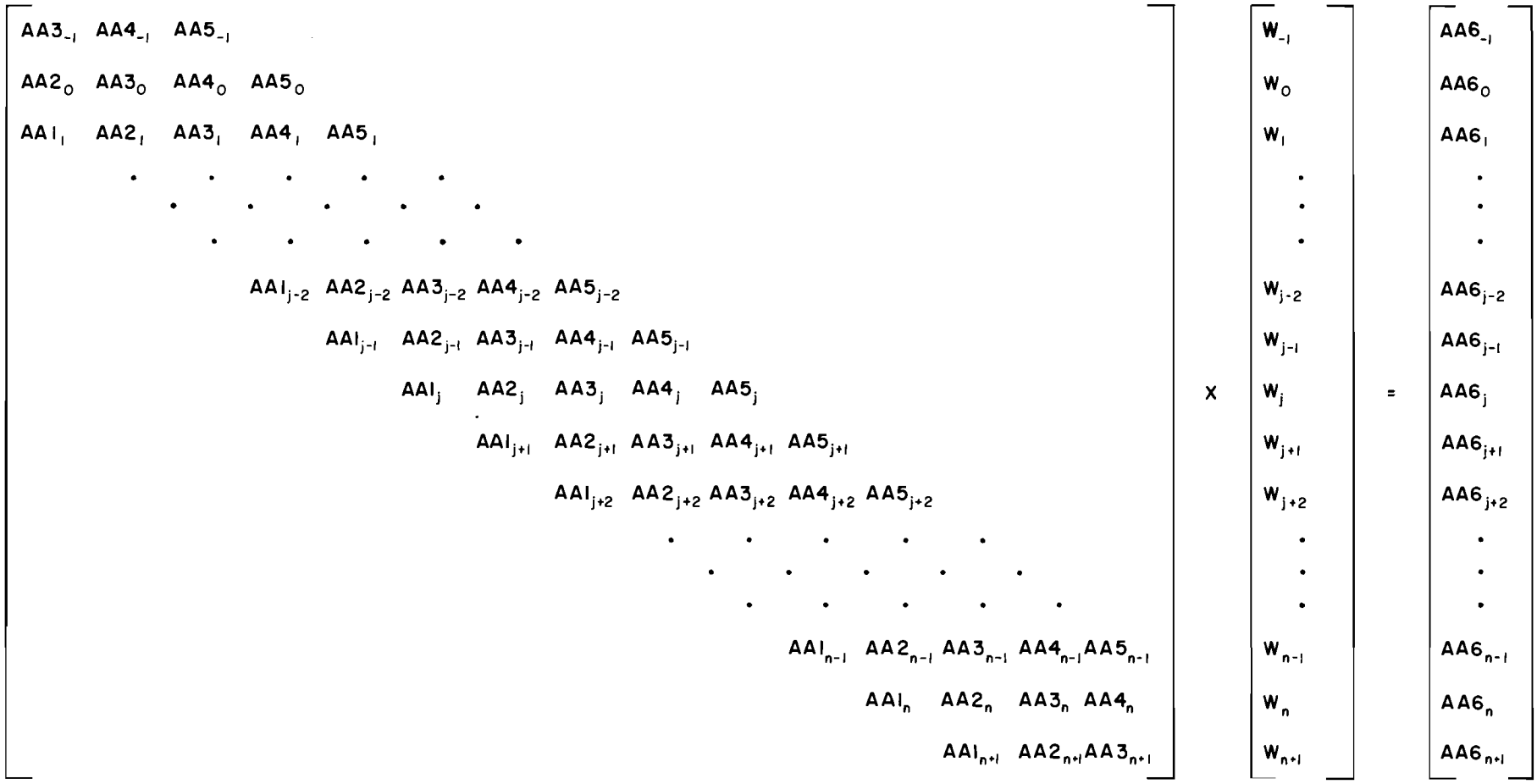


Fig 5.7. Redefinition of terms in K , W , and Q matrices.

where

$$A_j = D_j(E_jA_{j-1} + AA1_jA_{j-2} - AA6_j) \tag{5.40}$$

$$B_j = D_j(E_jC_{j-1} + AA4_j) \tag{5.41}$$

$$C_j = D_jAA5_j \tag{5.42}$$

$$D_j = -(E_jB_{j-1} + AA1_jC_{j+2} + AA3_j)^{-1} \tag{5.43}$$

$$E_j = AA1_jB_{j-2} + AA2_j \tag{5.44}$$

To complete the solution for all of the unknown deflection W_j, a reverse pass is made by applying the following version of Eq 5.39 at each station:

$$W_j = A_j + B_jW_{j+1} + C_jW_{j+2} \tag{5.45}$$

By the time the reverse pass is made, the deflections W_{j+1} and W_{j+2} will be known. The coefficients A_j, B_j, and C_j are called "continuity coefficients." The development of Eqs 5.40 through 5.45 is given in Ref 5.

For the purpose of starting the elimination process and turning around for the reverse pass, auxiliary fictitious stations are employed beyond the boundaries of the slab. The increment length of the fictitious bars connecting the fictitious stations to the real slab is set equal to the increment length of the first bars in the slab. This is done to avoid a division by zero which would occur if no increment length were given to the fictitious bars. Recall that the stiffness matrix is produced by applying Eq 5.22 at each station, including one fictitious station beyond the boundary of the slab. $AA1_{-1}$, $AA2_{-1}$, $AA1_0$, $AA5_n$, $AA4_{n+1}$, and $AA5_{n+1}$ (Fig 5.7) would then be automatically calculated as zero, since no load or stiffness data can be input for the fictitious mesh points beyond the slab. In the computation of continuity coefficients (Eqs 5.40 through 5.44) these zero matrices serve to bind the equations together and to eliminate any extraneous effects that might be thought of as existing beyond the boundaries of the slab. Since the fictitious stations beyond the boundaries of the model slab

have no flexural stiffness, they act as multiple hinges and thus isolate the model slab and the recursion equations describing its behavior from consideration of any effects beyond the boundaries.

Matlock uses this recursive technique for the solution of a one-dimensional system (beams and columns) where each of the terms of Eqs 5.40 through 5.45 refers to only one value, but the process is mathematically valid for the five-wide banded matrix described by Eq 5.38 even though each of the individual terms refers to a matrix of numbers. Instead of the normal algebraic manipulations, matrix manipulations will be used to solve for the deflections of the slab by the method outlined above. The matrix manipulation routines used in the computer program use the most efficient method for doing the manipulation by taking into account the fact that the submatrices are banded matrices.

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

CHAPTER 6. THE COMPUTER PROGRAM

The equations and solution technique derived in Chapter 5 are not useful for hand calculations, but they are extremely well-adapted for digital computer methods. The computer program developed during this investigation uses solution techniques originally developed by Stelzer (Refs 7 and 8) and improvements developed by Endres (Ref 2).

The program is written in FORTRAN compiler language for the Control Data Corporation 6600 Digital Computer. The compiler is compatible with FORTRAN II and IV. The compile time for the basic program is about twelve seconds. The internal storage requirements of the program are shown in Fig 6.1. The program utilizes two external magnetic tape units for secondary storage.

The time required to run problems varies with the size of the system, that is, the number of increments involved. The program is written so that problems can be input only with the number of increments in the x-direction less than or equal to the number of increments in the y-direction since, as noted in Chapter 5, this is the most efficient way to input rectangular problems.

The FORTRAN Program

A summary flow diagram for the VISAB Program is given in Fig 6.2. Detailed flow diagrams and a listing of the program VISAB 3 are provided in Appendices 2 and 3. Appendix 1 is a self-contained instruction and operating manual for VISAB 3. It includes instructions on the operation of the program, and detailed input forms and descriptions.

The program is written as a driver program which controls a number of subroutines. If it is desirable to redimension the program, only the dimension statement in the main program needs to be changed, as all the subroutines are variably dimensioned.

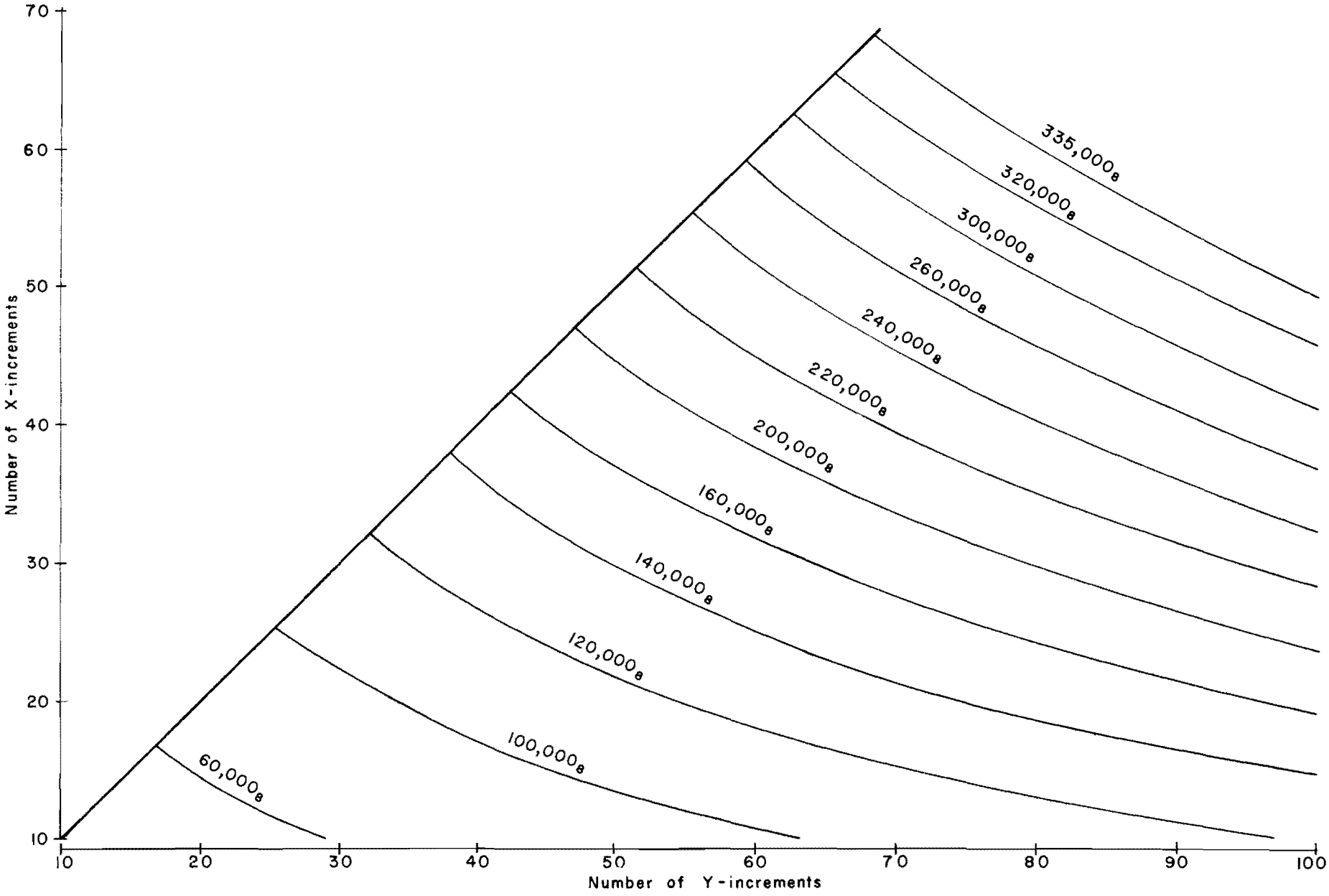


Fig 6.1. Internal storage requirements for Program VISAB 3.

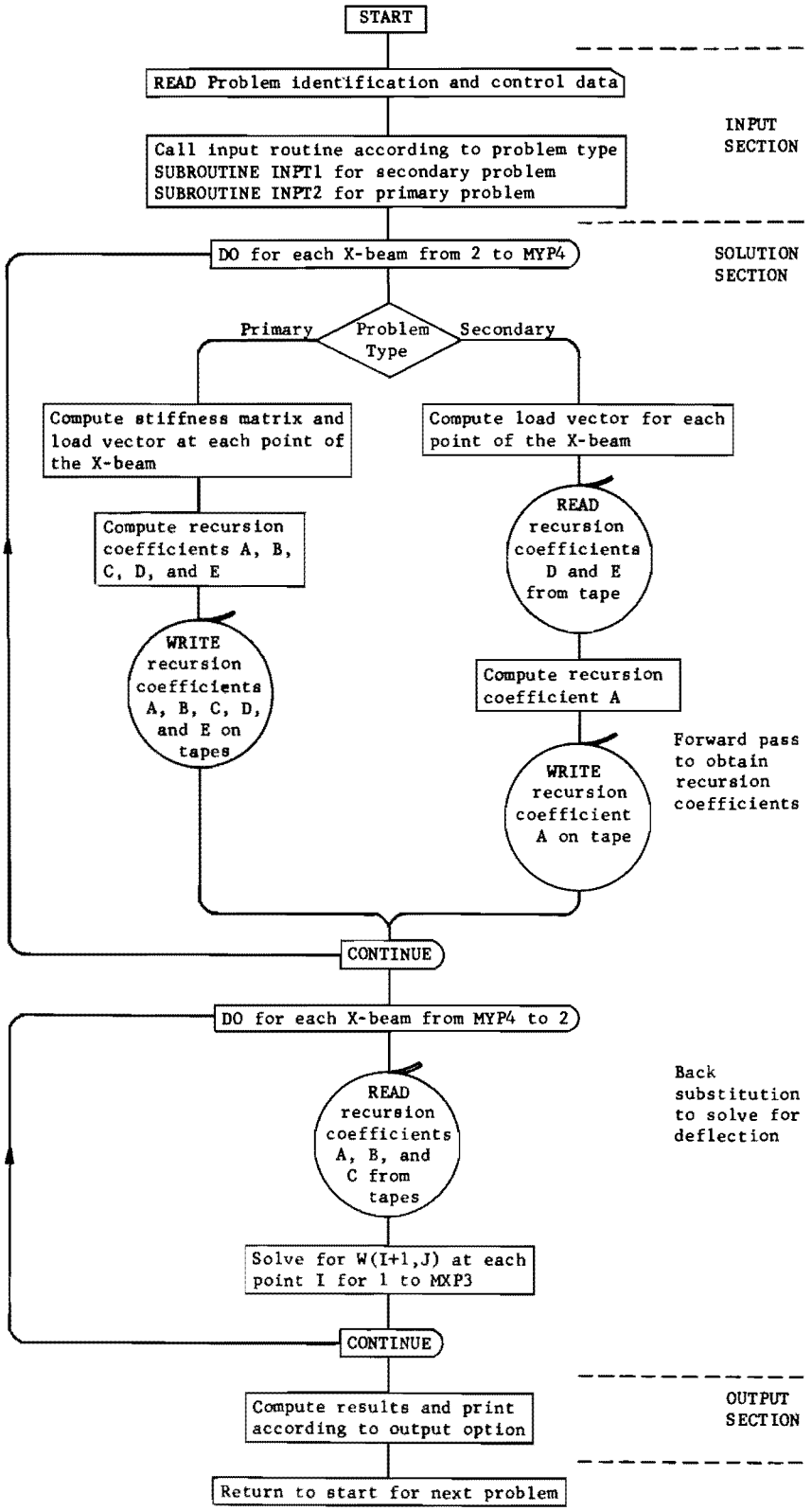


Fig 6.2. Summary flow diagram.

Input Information

The format used for inputting data into the program is arranged to be as convenient as possible. The problem input deck starts with two cover cards which identify the program and the particular run being made. The information on them is alphanumeric and denotes projects, coding data, personnel performing the keypunching, description of the problems being run, etc. The program will not operate correctly without these two cards. They are followed by

(1) Problem number card with alphanumeric description of the problem.

(2) Table 1 - Program Control Data and Constants - 1 card. Information on this card includes the number of cards to be read in Tables 2 through 7, number of increments, problem type, output format desired, Poisson's ratio, and slab thickness.

(3) Table 2 - Increment Length Data - The number of cards in this table is variable depending on the number required to specify the increment length.

(4) Table 3 - Bending Stiffness Data - The number of cards in this table is variable depending on the number required to specify the bending stiffness of the slab.

(5) Table 4 - Twisting Stiffness Data - The number of cards in this table is variable depending on the number required to specify the twisting stiffness of the slab.

(6) Table 5 - Transverse Load and Support Data - The number of cards in this table is variable depending on the number required to specify the load and support springs.

(7) Table 6 - x-Direction Load Data - The number of cards in this table is variable depending on the number required to specify external couples and axial loads in the x-direction.

(8) Table 7 - y-Direction Load Data - The number of cards in this table is variable depending on the number required to specify external couples and axial loads in the y-direction.

Appendix 1 gives a complete description of input forms and their use. Appendices 4, 5, and 6 contain numerical examples of input and output for some of the example problems in Chapter 7.

Output Information

The program output is arranged to be useful to the user. A format which can be trimmed to the standard 8-1/2-inch by 11-inch size is provided. For convenience and help in identifying problems, the program prints out all original input data at the beginning of each problem, in Tables 1 through 7. By use of an optional input parameter in Table 1, the array in which the input data is stored may be printed out. This information is generally not called for except in very complex problems where it may be desirable to have an additional check on the input data. The results, Table 8, may be printed in one of two forms. If the output parameter in Table 1 is blank or zero, the results are printed on the two-part, long form and if the parameter in Table 1 is one, the results are printed on the one-part, short form. The first form (or long form) prints Table 8 in two parts. Part one of this form prints external station numbers, deflections, bending moments in the x and y-directions, twisting moments in the x and y-directions, and the reaction of the slab at each point. The second part of the long form prints out external station numbers, the principal moments, the maximum twisting moment, the angle from the x-direction to the major principal moment, and, if a slab thickness is input, the principal stresses and the shear stress. The second (or short) form of Table 8 prints out the results in a shortened form. The short form prints out external station numbers, bending moments in the x and y-directions, twisting moment in the x-direction, the principal moment (or stress if a slab thickness is input) with the largest absolute value, the reaction of the slab, and the angle from the x-direction to the direction of the largest principal moment (or stress) at each point. Examples of these output forms are shown in Appendix 5.

As with all finite mathematical techniques, there are approximations in this program. It is not possible to determine both values of a double-valued function by numerical differentiation. Twisting moments are such double-valued functions, being a maximum just inside the plate boundary and zero just outside the boundary, and the best approximation in finite-difference techniques is a half-value or the average between the maximum and zero. The same half-value approximations result for bending moments at fixed ends for cantilevered structures (Ref 5). The bending moment-stiffness diagram is correct for this case since bending stiffness is input as a half-value at

the edges. Bending moments at free or simply-supported edges are calculated correctly by this method. Third derivatives which are related to the shear forces meet the Kirchoff boundary conditions at free edges (Ref 9, p 84).

Special Programming for Multiple Loading

Occasionally there is a need to solve a slab or plate with a number of different loading conditions. For these cases, two types of solutions have been developed. The first type is the single problem. In the single problem, only the recursion coefficients B and C (see Chapter 5) are stored on tape. The single problem type of solution requires that all input data be input for each set of loads and that the stiffness matrix be solved for each set of loads. The second type of solution is the multiple load type. When a problem with more than one set of loads is solved, the first set of loads can be solved as a primary problem and the additional sets of loads can be solved as secondary problems. When a primary problem is solved, the recursion coefficients B , C , D , and E are stored on the external tapes. The primary problem is the first of a set of problems with the same stiffness matrix, that is, the same slab or plate. By saving these coefficients, the input of the stiffness matrix is eliminated from additional problems. The secondary problem in this type of solution solves the slab or plate with different loads using the recursion coefficients B , C , D , and E from the primary problem which precedes the first secondary problem in the set of secondary problems. The use of the precalculated recursion coefficients eliminates a large percent of the matrix calculations in the problem. The percentage decrease for a secondary problem over a primary problem ranges from 60 percent for small problems to over 90 percent for very large problems. For additional details of the differences in primary and secondary problems see Ref 2 and the detailed flow chart in Appendix 2.

Error Checks

Since the program will run with almost any data the user inputs, the input data should be carefully checked. A number of error checks are provided in the program to aid in checking the input data. The program checks the input data station (or joint) numbers to make sure that the beginning number in the sequence (the FROM number - see Guide for Data Input) is less than or equal to the ending number (the THRU number in the sequence).

It also checks that all input stations are within the range of the dimension statement. When an error in stationing is found in the input data, the remaining cards in the problem are read and ignored. Then the program goes on to the next problem without solving the erroneous problem. The error checks cannot catch an incorrect count of the number of cards in the problem or data (stiffness, load, and support). The operating manual (Appendix 1) gives additional information on the error checks written into Program VISAB 3.

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

CHAPTER 7. EXAMPLE PROBLEMS AND VERIFICATION OF THE METHOD

This chapter provides the solution to six example problems to demonstrate Program VISAB 3 and its use in engineering calculations. As stated before, this work is an attempt to improve the efficiency of Hudson's method (Ref 3) and of Stelzer's method (Ref 7). The verification of the method of solution in this report can be conveniently accomplished by re-solving some of the examples of Refs 3 and 7. The first three example problems correspond to examples in Hudson's and Stelzer's reports. The other three example problems demonstrate further uses of the variable-increment-length method. Sample input and output in Appendices 4 and 6 provide the reader with a step-by-step example of the program in use.

Example Problem 1 - Complex Bridge Approach Slab

In order to show that the variable-increment-length solution is the same as the constant-increment-length solution when h^x and h^y are input as constants, a complex bridge approach slab problem was solved. This problem had previously been solved on the DSLAB program (Problem 601 in Ref 8) and is presented here for comparison. Figure 7.1 illustrates this slab. The slab is a 10-inch-thick reinforced concrete bridge approach slab. The slab is divided into 12 two-foot-long increments in the x-direction and 16 two-foot-long increments in the y-direction. The bending stiffness in the x and y-directions (D_x and D_y, respectively) can be calculated as shown in Ref 4 through the use of Eqs 7.1 and 7.2.

$$D_x = \frac{E_x t^3}{12(1 - \nu^2)} \tag{7.1}$$

$$D_y = \frac{E_y t^3}{12(1 - \nu^2)} \tag{7.2}$$

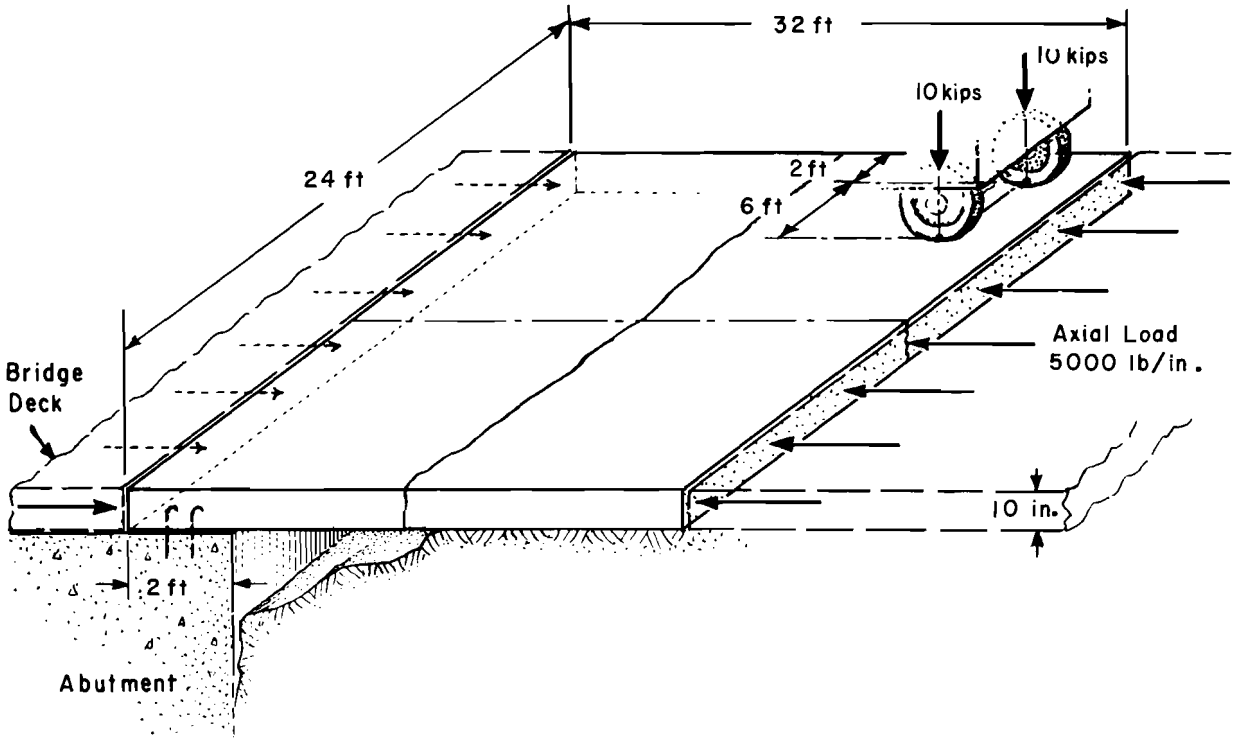


Fig 7.1. Bridge approach slab.

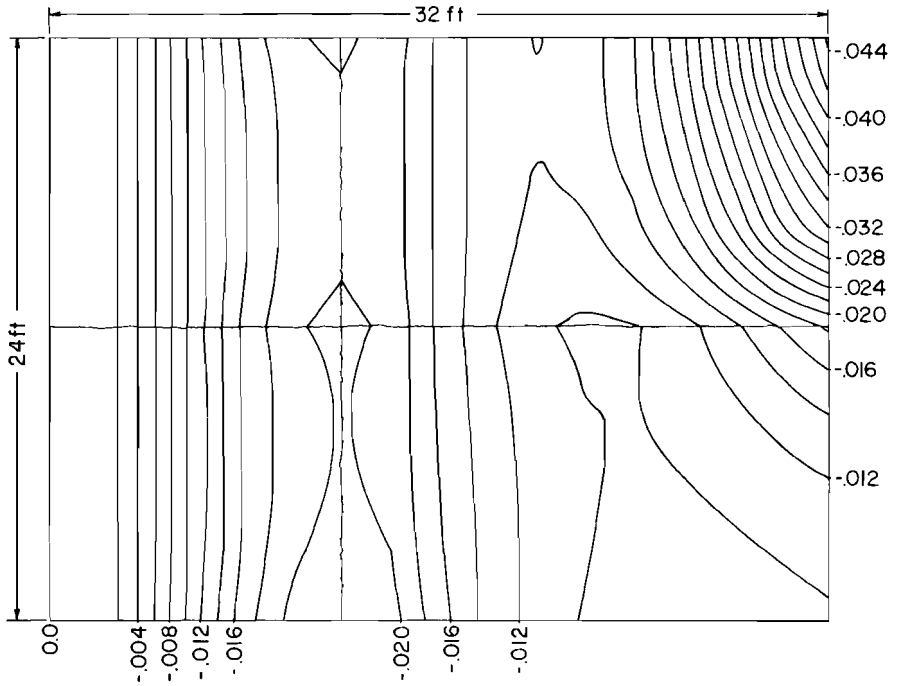


Fig 7.2. Deflection contours of bridge approach slab (deflections in inches).

where

t = thickness of the slab,

E_x = modulus of elasticity in the x-direction, and

E_y = modulus of elasticity in the y-direction.

Using $E_x = E_y = 4{,}000{,}000$ psi and $\nu = 0.25$, it is found from Eqs 7.1 and 7.2 that $D_x = D_y = 3.556 \times 10^8$ in-lb.

For all practical purposes the torsional stiffnesses about the x and y-axes (C_x and C_y, respectively) are equal although the resulting forces $C^{x'}$ and $C^{y'}$ depend on the dimensions of the increment.

For isotropic plates of uniform thickness ($E_x = E_y$),

$$C_x = C_y = \frac{E_x t^3}{12(1 + \nu)} \tag{7.3}$$

For orthotropic plates a modified torsional stiffness should be used (Ref 4, p 124):

$$G_o = \frac{E_x E_y}{E_y(1 + \nu_{xy}) + E_x(1 + \nu_{yx})} \tag{7.4}$$

$$C_x = C_y = \frac{Gt^3}{6} \tag{7.5}$$

For complex geometric orthotropy, an independent test is suggested (Ref 4, p 124).

The torsional stiffness for the example slab is $C_x = C_y = 2.667 \times 10^8$ in-lb/rad.

One end of the slab is supported by the bridge abutment and the other end by the embankment. The soil under the interior of the slab has settled leaving a section unsupported. The slab has a centerline joint and a crack

which developed from a combination of shrinkage and previous overstress. For a nonuniformly supported slab such as this, the dead weight of the slab must be considered when evaluating moments and stresses. This weight acts as a uniform load of 1.042 psi on the entire slab. Two 10-kip wheel loads were considered in this example. An axial load of 5,000 pounds per inch has been induced by the expansion of the adjoining pavement.

The resulting deflected shape is shown in Fig 7.2. The VISAB solution to this problem compares exactly with the solution given by Stelzer (Ref 8) to four significant figures. The maximum difference for moments in the two solutions was 0.5 of one percent. This difference is due primarily to additional computer round-off error due to the extra calculations from the stiffness matrix in this method. The very small amount of error shows that the VISAB 3 solution with constant increment lengths is equivalent to the solution obtained by Stelzer. Other examples not given here have also verified this.

Example Problem 2 - Slab on Foundation

Westergaard solved for the deflections of a slab on foundation with center load by assuming the slab to be infinite. Hudson (Ref 4, p 79) shows that Westergaard's slab can be modeled with a 24-foot-square concrete slab. Using the slab shown in Fig 7.3 and a load of 10^5 pounds in the center, a number of constant-increment-length solutions were run using from 8 to 60 increments. The maximum deflection for each of these cases is shown in Fig 7.4. Westergaard's formula gives a deflection of $w = 0.0553$ inch for this case.

To check this problem with VISAB 3, an increment length of four inches was chosen near the load. If constant increments are used, 72 increments are needed in each direction to obtain the 4-inch increment length. With VISAB 3 it is possible to have these 4-inch increments near the load with only 16 increments in each direction. The increments other than the four 4-inch increments around the load were chosen in such a way as to make the slab exactly 24 feet x 24 feet. The deflection under the load in this case was $w = 0.0555$ inch. This, when compared with a constant increment solution (DSLAB) with 16 increments, giving $w = 0.0604$ inch, shows that the difference between Westergaard's solution and the DSLAB solution with 16 increments is twenty-five times the difference between Westergaard's solution and the VISAB 3

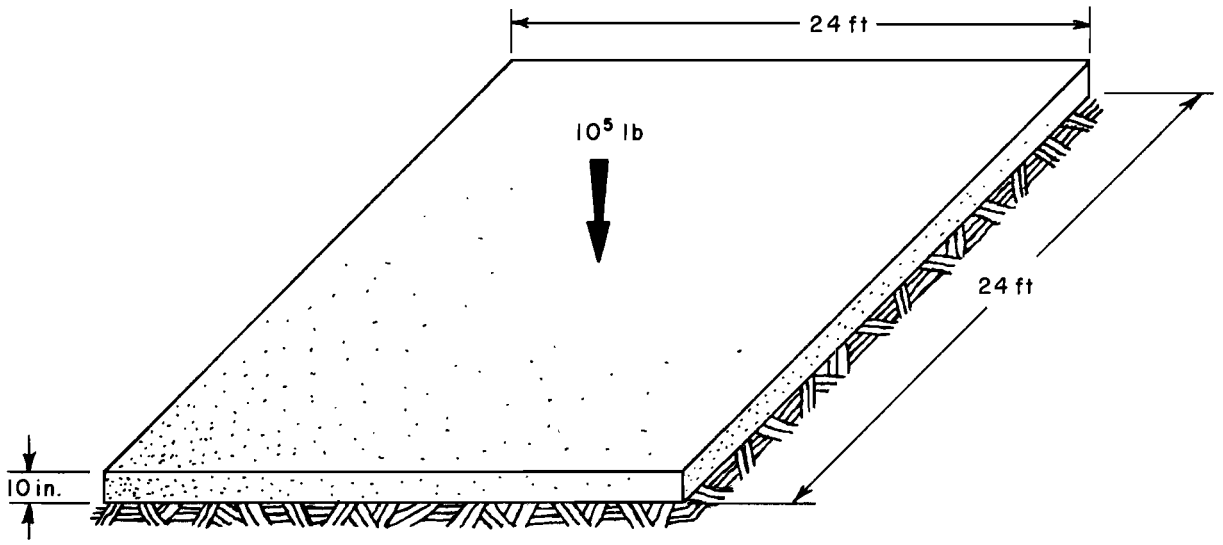


Fig 7.3. Slab on foundation.

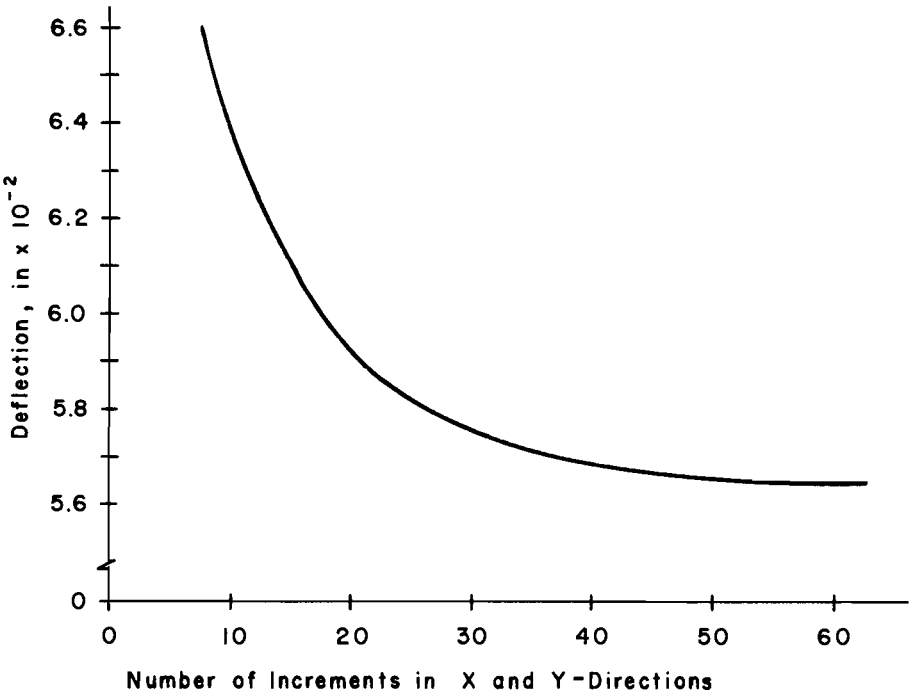


Fig 7.4. Variance in deflection under load with change in number of increments.

solution with 16 increments. This shows a significant increase in accuracy by use of variable increment lengths. The time required to run a 16 x 16-increment problem on VISAB 3 is 8.4 seconds. This compares to 24.0 seconds for a 16 x 16-increment problem on the latest DSLAB version (Ref 2).

By comparison to Fig 7.4, it can be seen that the VISAB 3 solution to this problem with 16 increments is a more accurate solution than the 60 x 60-increment constant-increment-length program. The difference in storage required for a 16 x 16-increment problem and that required for a 60 x 60-increment problem is approximately 70,000 words on the CDC 6600 computer. The 60 x 60-increment problem required approximately 11 minutes of computer time for a full solution, or approximately 100 times the amount required by the 16 x 16-increment problem. This proves that the variable increment model can save computer storage and computation time.

Example Problem 3 - Simply-Supported Steel Plate

To further illustrate the improved accuracy of the variable-increment-length model, a simply-supported steel plate example was solved. This problem was the second problem of a series solved by Hudson (Ref 4, p 71) and Stelzer (Ref 8, p 35). The plate is a 48-inch-square simply-supported steel plate one inch thick (Fig 7.5). The plate has a concentrated load in the center and a uniform axial load in the y-direction. The closed-form solution to this problem is a deflection of $w = 0.787$ inch at the center. Hudson and Stelzer using eight 6-inch increments in each direction computed a deflection of $w = 0.854$ inch. Using variable increment lengths with four 4-inch increments at the center and two 8-inch increments on the edges in each direction, a deflection of $w = 0.821$ was computed for the center point. The error in the VISAB 3 solution is therefore only 50 percent of the error in the previous solution. This problem shows that even with a small number of increments, the accuracy of the solution can be improved by adjusting the increment length.

Example Problem 4 - Small-Scale Slab-on-Foundation Test

The slab (Fig 7.6) used in the model tests by the Corps of Engineers was a 15 x 15-inch concrete slab on a natural rubber base. The deflections and stresses of the model were measured during the tests. The principal stresses

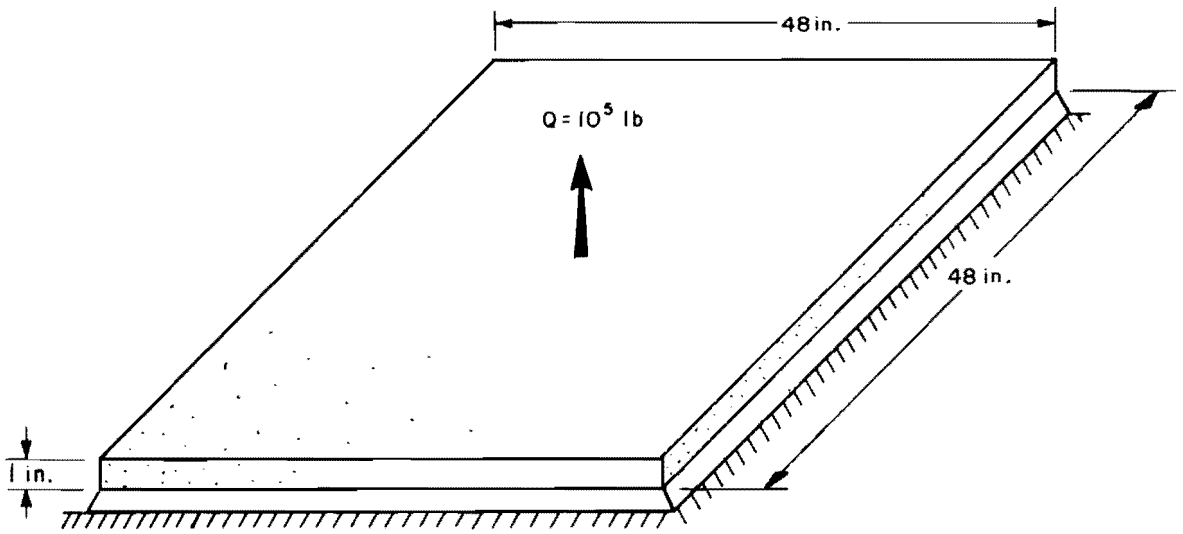


Fig 7.5. Square steel plate simply supported at all edges (Example Problem 101).

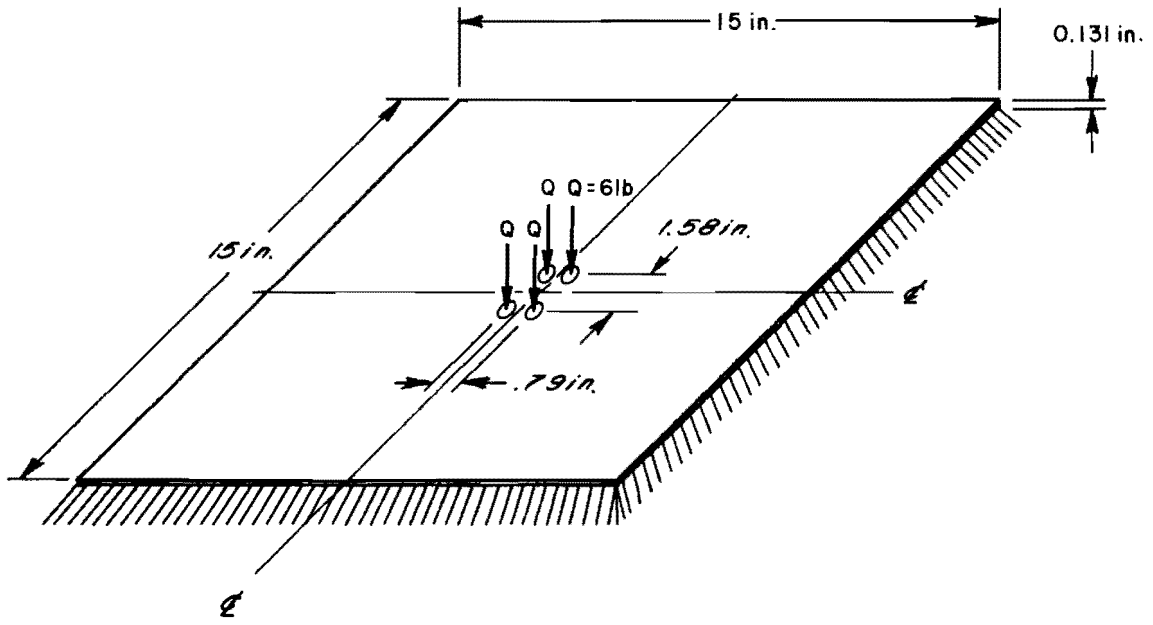


Fig 7.6. Corps of Engineers model test.

are reported in Ref 6. Using the constant-increment-length program, neither the location of the loads nor the size of the slab could be modeled exactly. For the case of twin-tandem the maximum principal stress measured was 885 psi. When solved on the computer, using the constant-increment-length program with 50 increments in each direction, the maximum principal stress calculated was 1017 psi or approximately 15 percent larger than the measured values. With VISAB 3, the variable-increment-length model allowed the slab and loads to be modeled exactly. Using 18 increments in the x-direction and 20 increments in the y-direction, the problem was run on VISAB 3. The computed solution for the maximum principal stress was 919 psi or only about 3 percent larger than the stress measured on the model. This problem shows that with more exact modeling of the problem, even with fewer stations, accuracy of the solutions improves. The variable-increment-length model is shown by this problem to be able to model the problem better than the constant-increment-length model.

Example Problem 5 - Simply-Supported Plate with Cut

The simply-supported plate shown in Fig 7.7 was tested at the Center for Highway Research (Ref 1). The plate is a quarter-inch-thick aluminum plate. The supports are rigid point supports at each corner. The plate was tested with a load of 40 pounds. The cut is one-half inch wide and is six inches long. The deflection measured at the point of load was 0.04175 inch. With a constant increment length of one-half inch and 50 increments in each direction, the deflection calculated at the point of load was 0.04404 inch or a difference of 5.5 percent. Using a variable grid system with 39 increments in the x-direction and 40 increments in the y-direction, the deflection at the load point was calculated to be 0.04217 inch or a difference of 1.0 percent. This again shows that the variable-increment-length model can give better accuracy with fewer increments.

Example Problem 6 - Continuous Bridge Slab

The continuous bridge slab shown in Fig 7.8 is too large to be solved with the one-foot constant increment length that is needed to locate the load correctly. The problem can be solved with the constant-increment-length programs available if one of the 18-foot panels is removed and the computer program is specially modified. The variable-increment-length program (VISAB 3)

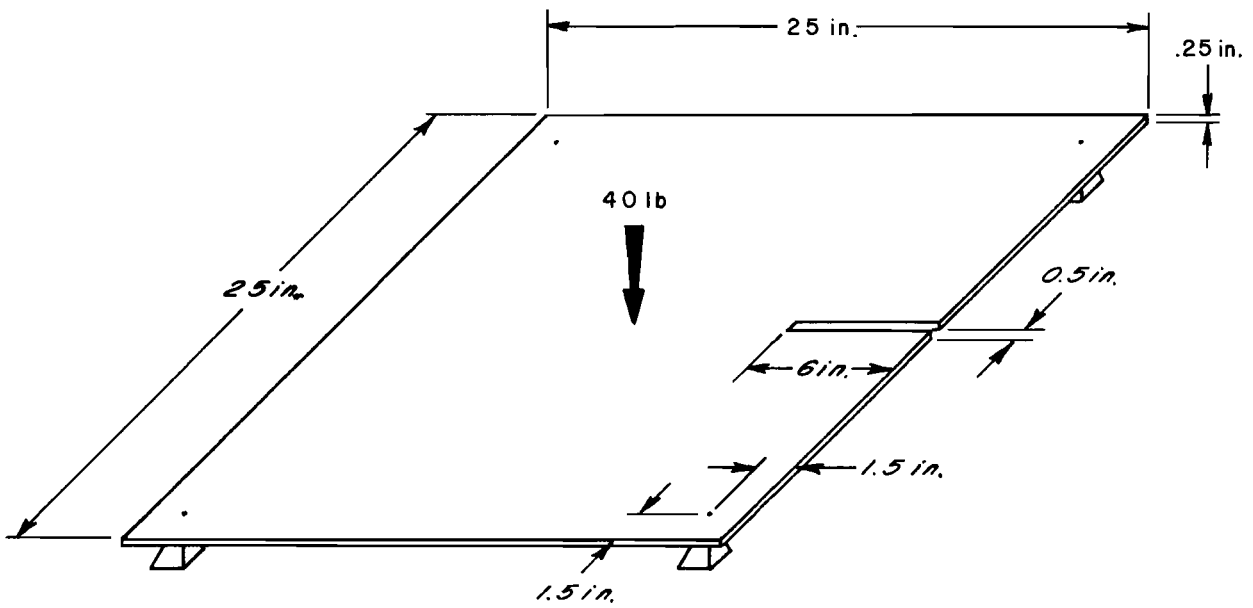


Fig 7.7. Simply-supported plate with cut.

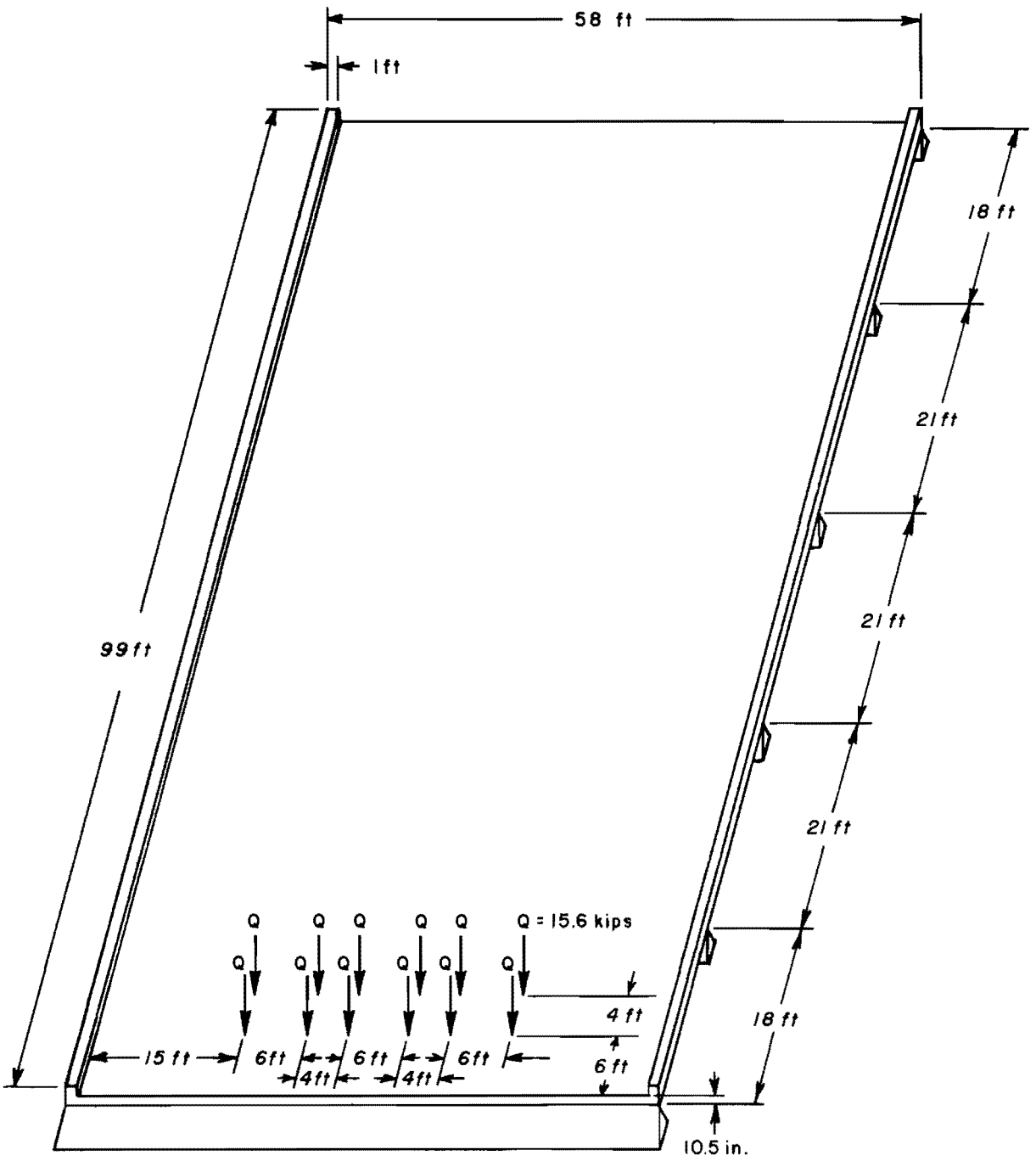


Fig 7.8. Continuous bridge slab.

can handle the problem with ease. To reduce the number of increments in the x-direction, eight 2-foot increments are placed on the x-beams. To reduce the number of increments in the y-direction, 3-foot increments are used in the two panels at the end away from the load. The problem now can be solved using a 50×73 grid system instead of the 58×99 system needed by the constant-increment-length system. The maximum bending moment in the slab was 20.4 ksi which is within one percent of the moment calculated by the constant-increment-length method. Two loadings in addition to the one in Fig 7 were run to show the usefulness of the multiple load technique. The time required for each secondary problem was only about 11 percent of the time required for the primary problem.

Results of Example Problems

The six example problems presented illustrate that the variable-increment-length model can be used to improve the accuracy of the direct solution slab programs. The errors in deflection near loads can be cut as much as 80 percent without increasing the solution time. The variable-increment-length solutions obtained by VISAB 3 are generally equivalent to constant-increment-length solutions using two to three times as many increments in each direction. This represents a time saving by VISAB 3 of approximately 95 percent over the previous methods. At the same time it can be seen that the user needs to pay closer attention to the method of modeling the problem than with the previous solution.

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

CHAPTER 8. SUMMARY

Useful methods for the solution of plates and pavement slabs have been described by Hudson in Ref 3 and Stelzer in Ref 7. The accuracy of the solutions in these methods depends on the choice of the increment length for use in the model. There is clearly a need for a model that will allow the use of variable increment-lengths in the model. This report provides such a model.

The model chosen is based on Hudson's model, which is helpful in visualizing the problem and forming the solution. The increment lengths on the x-beams and on the y-beams are allowed to vary along the beams, but the increment lengths on each x-beam are required to be identical to the corresponding increment length on the other x-beams, and the same is true for the y-beams.

The computer program VISAB 3 is limited to a maximum size of 68 increments in the x and y-directions when the number of increments in each direction is equal. This amount of storage is only slightly more than the amount required by Stelzer's method (Ref 7) as modified by Endres (Ref 2).

This method has application to a broad variety of problems since complex conditions can be met more exactly than with previous methods. The method is more accurate than previous methods when the same number of increments are used in all methods and therefore more economical than the previous methods.

Application of Results

This report is one of a series of reports which describe continually improving techniques for solution of pavement and bridge slabs and plates. The application of these methods will provide highway and bridge designers with a series of tools which can ultimately be used to design better slabs.

Several computer programs developed and reported in previous reports have been made operational on the Texas Highway Department computer facility. Five of the programs are in daily use by the department. For ultimate benefit, it will be desirable for the computer program described herein to be converted for use directly on the Texas Highway Department facility.

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

REFERENCES

1. Agarwal, Sohan L., personal notes on model tests, The University of Texas at Austin, March 15, 1968.

2. Endres, Frank L., personal notes on improvement of matrix calculations for DSLAB programs, The University of Texas at Austin, February 1968.

3. Hudson, W. Ronald, "Discontinuous Orthotropic Plates and Pavement Slabs," Ph.D. Dissertation, The University of Texas, Austin, Texas, August 1965.

4. Hudson, W. Ronald and Hudson Matlock, "Discontinuous Orthotropic Plates and Pavement Slabs," Research Report No. 56-6, Center for Highway Research, The University of Texas at Austin, May 1966.

5. Matlock, Hudson and T. A. Haliburton, "A Finite-Element Method of Solution for Linearly Elastic Beam-Columns," Research Report 56-1, Center for Highway Research, The University of Texas at Austin, September 1966.

6. "A Model Study of the Load-Distributing Characteristics on Twin, Twin-Tandem and Twin-Twin Gear Configurations for Rigid Pavement," Technical Report No. 4-8, Ohio River Division Laboratories, U.S. Army Corps of Engineers, Cincinnati, Ohio, October 1958.

7. Stelzer, C. Fred, "A Direct Computer Solution for Plates and Pavement Slabs," Master's Thesis, The University of Texas at Austin, January 1967.

8. Stelzer, C. Fred and W. Ronald Hudson, "A Direct Computer Solution for Plates and Pavement Slabs," Research Report 56-9, Center for Highway Research, The University of Texas at Austin, October 1967.

9. Timoshenko, S. and L. Woinowsky-Krieger, Theory of Plates and Shells, (Engineering Society Monographs), 2nd Edition, McGraw-Hill, New York, New York, 1959.

10. Westergaard, H. M., "Analytical Tools for Judging Results of Structural Tests of Concrete Pavements," Public Roads, Vol. 14, 1933.

11. Westergaard, H. M., "Computation of Stresses in Concrete Roads," Public Roads, Vol. 7, No. 2, April 1926.

12. Westergaard, H. M., "New Formulas for Stresses in Concrete Pavement of Airfields," Proceedings, American Society of Civil Engineers, Vol. 73, No. 5, May 1947.

APPENDICES

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

APPENDIX 1

OPERATING MANUAL FOR PROGRAM VISAB 3

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

OPERATING MANUAL FOR PROGRAM VISAB 3

extract from

A DISCRETE-ELEMENT SOLUTION OF PLATES AND PAVEMENT SLABS
USING A VARIABLE-INCREMENT-LENGTH MODEL

Report No. 56-11

by

Charles M. Pearre, III, and W. Ronald Hudson

April 1969

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

VISAB 3 is a computer program written to solve problems involving orthotropic plates and pavement slabs. The development of the equations and the overall method of solution are discussed in Chapters 4, 5, and 6 of the basic report. The purpose of this Appendix is to provide the program user with a concise manual which can be extracted for daily use with the program.

Program Operation

The general procedures following in the program are described in the flow charts (Appendix 2). A problem number card at the beginning of each problem controls the start of the solution. Unless there is an undercount of the number of cards in one or more of the tables, the program will work any number of problems. The problems are solved in sequence except when unacceptable data occur and a problem is rejected. The program finally stops when a problem number card which has the first 10 spaces blank is encountered.

The data deck starts with two cover cards used to identify the program and the particular run being made. The problems to be solved together in one run are stacked behind the cover cards in sequence as illustrated in Fig A1.1. Each problem consists of (1) one problem number card with alphanumeric description of the problem; (2) Table 1, Program Control Data and Constants, one card containing necessary control data and constants for the problem; (3) Table 2, Increment Length Data, which contains the increment lengths; (4) Table 3, Bending Stiffness Data, which contains the bending stiffness of the plate; (5) Table 4, Twisting Stiffness Data, which contains the torsional stiffness of the plate; (6) Table 5, Transverse Load and Support Data, which contains the loads and support springs for the problem; (7) Table 6, x-Direction Load Data, which contains axial loads and couples in the x-direction; and (8) Table 7, y-Direction Load Data, which contains axial loads and couples in the y-direction. The number of cards in Tables 2 through 7 is variable and must be properly specified in Table 1 as indicated in the Input Form.

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

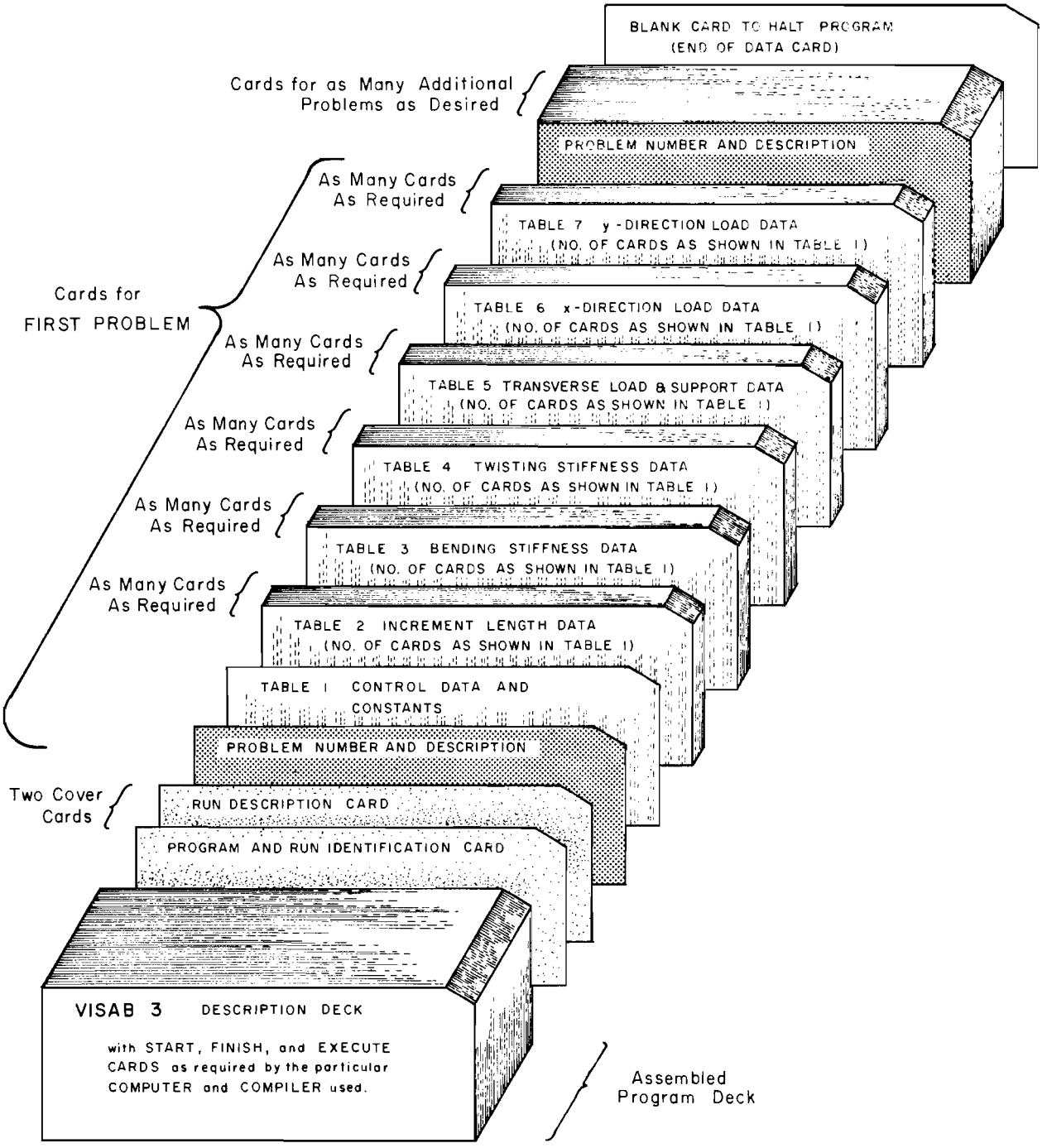


Fig A1.1. Assembly order for VISAB 3 program deck with data, ready for run.

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

Guide for Data Input

The following pages provide a Guide for Data Input. It should be expected that revisions of these forms and instructions will be developed in the future and may supersede the present versions. Example problems are discussed in Chapter 7. Appendix 4 includes example input data for several of these example problems. By comparing these example inputs with the description of the real problem the user can gain practical experience in preparation of input data. Proficiency in the use of the program can be gained only through actual coding of problems and solution in the computer. Recoding and re-solution of the example problems should prove to be helpful.

GENERAL PROGRAM NOTES

The data cards must be stacked in proper order for the program to run.

A consistent system of units must be used for all input data, for example, pounds and inches.

All 3 to 5-space words are understood to be integers or whole decimal numbers . . . . . | + 4 3 2 1 |

All 10-space words are floating-point decimal numbers . . . . . | + 4 . 3 2 1 E + 0 3 |

All numbers must be right justified.

The problem number may be alphanumeric.

The second 5-space word (KCHEK) on the problem number card must be blank for the program to run.

TABLE 1. PROGRAM CONTROL DATA AND CONSTANTS (1 card)

The number of input cards for Tables 2 through 7 must be shown separately and should be carefully checked.

The number of increments in each direction is input as an integer. For the problem to run MY cannot be less than MX .

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

The multiple load option allows the user to hold the recursion coefficients from the previous problem. If ML = 0 , the problem is considered to be a single problem and the recursion coefficients are not held. If ML = +1 , the problem is considered to be a primary problem and all input data are needed. If ML = -1 , the problem is considered to be a secondary problem using the recursion coefficients from the previous problem and only lateral loads and external couples can be input (Tables 5, 6, and 7).

The output options give the user two choices of output form. The option NOUT allows the user to choose a short form of the results if NOUT is equal to 1. If NOUT is zero, the long form of the results is printed. Examples of the two forms of results are shown in Appendix 5. The option NPRI allows the user to print out the input data arrays after all the input data are stored in them. If NPRI is equal to 1, the arrays are printed in Tables XX and YY. If NPRI is zero, Tables XX and YY are not printed.

Poisson's ratio will be taken as zero unless specified. It should always be positive.

Slab thickness is used only to calculate stresses from moments. If a thickness is not input stresses are not calculated.

TABLE 2. INCREMENT LENGTH DATA

HX and HY have units of length.

Increment lengths can be placed only on real bars (e.g., bar number zero is not real).

The HX and HY data from each card are added to the previous data for each X and Y-bar, respectively, from the first through the last X and Y-bar numbers given on the card.

TABLE 3. BENDING STIFFNESS DATA

DX and DY have units of force times length.

To distribute data over a rectangular area, the lower left hand and the upper right hand stations of

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

the area must be specified. The program automatically places the correct half-values along the edges of the area and the correct quarter-values at the corners.

To specify data at a single station, the station number must be given in both the "From Station" and "Thru Station" columns. The program then places a full value at this station.

Bending stiffness can be input at any real station.

TABLE 4. TWISTING STIFFNESS DATA

CX and CY have units of force times length.

To distribute data over a rectangular area, the lower left hand and the upper right hand torsion elements of the area must be specified. Twisting is input in full values at each torsion element.

To specify data in a single torsion element, the element number must be given in both the "From Station" and "Thru Station" columns.

Twisting stiffness can be input at any real torsion element (e.g., torsion element 1,1 is the first real torsion element in a slab which starts with station 0,0).

TABLE 5. TRANSVERSE LOAD AND SUPPORT DATA

Q has units of force (concentrated lateral load).

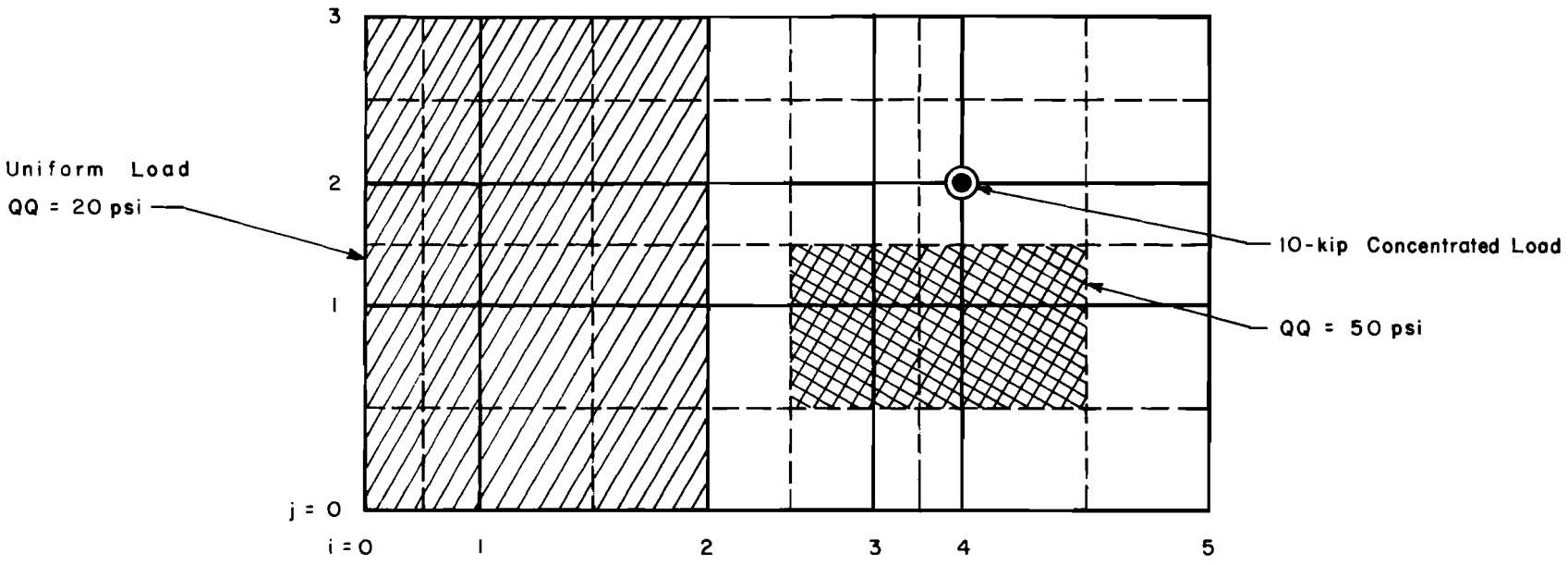
QQ has units of force per area (distributed lateral load).

S has units of force per length (concentrated support spring).

SS has units of force per length times area (distributed support spring).

To distribute data over a rectangular area, the lower left hand and the upper right hand stations must be specified. The program adds Q and S directly at each station in the area except along the edges where half-values are added and at the corners where quarter-values are added. The values QQ and SS are modified by the appropriate area around the station before they are added to the stored data. Figure A1.2 shows an example of load input.

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team



From Sta		Thru Sta		Load Q	Load QQ	
4	2	4	2	1.000 E + 04		Concentrated value at Station 4,2.
3	1	3	1		5.000 E + 01	Distributed load centered over Stations 3,1
4	1	4	1		5.000 E + 01	and 4,1 such that each gets a full value.
0	0	2	3		2.000 E + 01	Uniform load in rectangle 0,0 - 2,3 added in one-quarter values at a time to provide half-values at edges and quarter-values at corners of the area by the input routine of the program.

Fig A1.2. Example load input.

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

To specify data at a single station, the station number must be specified in both the "From Station" and "Thru Station" columns.

Data in this table can be input at any real station.

TABLE 6. x-DIRECTION LOAD DATA

TX has units of force times length (concentrated couple).

TTX has units of force times length (distributed couple).

PX has units of force (concentrated axial load).

PPX has units of force per length (distributed axial load).

To distribute data over a rectangular area, the lower left hand and the upper right hand bar numbers of the area must be specified. The program adds TX and PX directly at each bar in the area except for the J1 and J2 bars (edges) where half-values are added. The values TTX and PPX are multiplied by the width of the x-beam before being added to the previously stored data.

To specify data at a single bar, the bar number must be specified in both the "From X-Bar" and "Thru X-Bar" columns.

Data in this table can be input in any real bar in an x-beam. TX and TTX can be input in the fictitious bars (0,J) if and only if they are also input in the first real bar (1,J) on the same card.

TABLE 7. y-DIRECTION LOAD DATA

TY has units of force times length (concentrated couple).

TTY has units of force times length per length (distributed couple).

PY has units of force (concentrated axial load).

PPY has units of force per length (distributed axial load).

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

To distribute data over a rectangular area, the lower left hand and the upper right hand bar number of the area must be specified. The program adds TY and PY directly at each bar in the area except for the I1 and I2 bars (edges) where half-values are added. The values TTY and PPY are multiplied by the width of the y-beam before being added to the previously stored data.

To specify data at a single bar, the bar number must be specified in both the "From Y-Bar" and "Thru Y-Bar" columns.

Data in this table can be input in any real bar in a y-beam. TY and TTY can be input in the fictitious bars (I,0) if and only if they are also input in the first real bar (I,1) on the same card.

GENERAL NOTE ON TABLES 2 THROUGH 7

The problem will be rejected if any "Thru" number (I2 or J2) is less than the corresponding "From" number (I1 or J1).

Dimension Guide

Since the size and the storage capacity of computers vary and since some computer charges are based on storage required, it is necessary to make the size of a program variable. This is accomplished by use of a Dimension Statement, which is an integral part of the program deck. In VISAB 3, it is also necessary to change the Dimension Statement to run long rectangular problems. The Dimension Statement appears in the main program after the section of comment cards which make up the dimension guide. The Dimension Guide, shown in Fig A1.3, shows the variables of the Dimension Statement using an X and Y notation. To change the dimension statement the user repunches the dimension statement substituting X and Y into the variable arrays. Figure A1.4 shows the dimension statement set for X = 10 and Y = 20 . It should be noted that Y cannot be less than X for the program to operate.

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

```
C-----DIMENSION GUIDE FOR VISAB 3
C
C     FOR DIFFERENT SIZE PROBLEMS CHANGE THE DIMENSIONED ARRAYS
C         ACCORDING TO THE FOLLOWING EXAMPLE TYPES, WHERE N IS 1, 2,
C         3, ETC.   X SHOULD ALWAYS BE LESS THAN OR EQUAL TO Y.
C
C              AN(X+3)              N = 1, 2, 3
C              BN(X+3,1)            N = 1, 2, 3, 4, 5
C              CN(X+3,3)            N = 1, 2
C              DN(X+3,5)            N = 1
C              EN(X+3,X+3)          N = 1, 2, - - - ,  8
C              PN(X+5,Y+5)          N = 1, 2, - - - , 23
C              QN(X+6)              N = 1
C              RN(Y+6)              N = 1
C
C         ALSO, THE LN CONSTANTS MUST BE CHANGED AS FOLLOWS.
C
C              L1 = X + 5
C              L2 = Y + 5
C
```

Fig A1.3. Dimension guide.

```
 DIMENSION     A1(13),    A2(13),        A3(13),        B1(13,1),        01
1          B2(13,1),     B3(13,1),      B4(13,1),      B5(13,1),        02
2          C1(13,3),     C2(13,3),      D1(13,5),      E1(13,13),       03
3          E2(13,13),    E3(13,13),     E4(13,13),     E5(13,13),       04
4          E6(13,13),    E7(13,13),     E8(13,13),     P1(15,25),       05
5          P2(15,25),    P3(15,25),     P4(15,25),     P5(15,25),       06
6          P6(15,25),    P7(15,25),     P8(15,25),     P9(15,25),       07
7          P10(15,25),   P11(15,25),    P12(15,25),    P13(15,25),      08
8          P14(15,25),   P15(15,25),    P16(15,25),    P17(15,25),      09
9          P18(15,25),   P19(15,25),    P20(15,25),    P21(15,25),      10
A          P22(15,25),   P23(15,25),    Q1(16),        R1(26)           11

           L1  =  15                                                    12
           L2  =  25                                                    13
```

Fig A1.4. Example dimension statement for x = 10 and y = 20 .

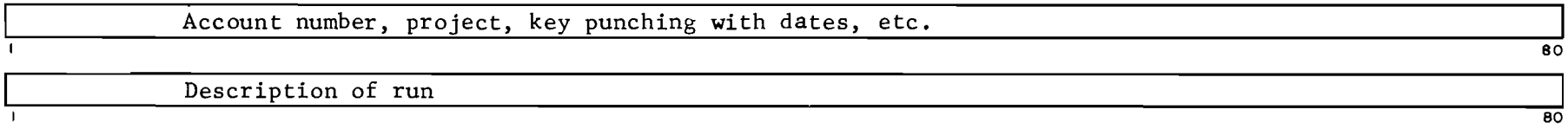
This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

The variables L1 and L2 are in this Dimension Guide to be used to check problems to make sure the problems will fit within the requested storage. L1 and L2 must be changed whenever the dimension statement is changed.

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

VISAB 3 GUIDE FOR DATA INPUT - CARD FORMS

IDENTIFICATION OF PROGRAM AND RUN (2 alphanumeric cards per run)



IDENTIFICATION OF PROBLEM (one card each problem; KCHEK must be blank;
program stops if NPROB is also blank)

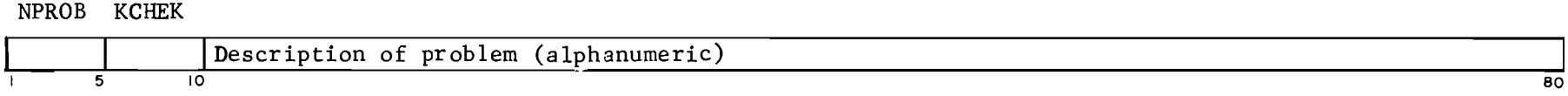
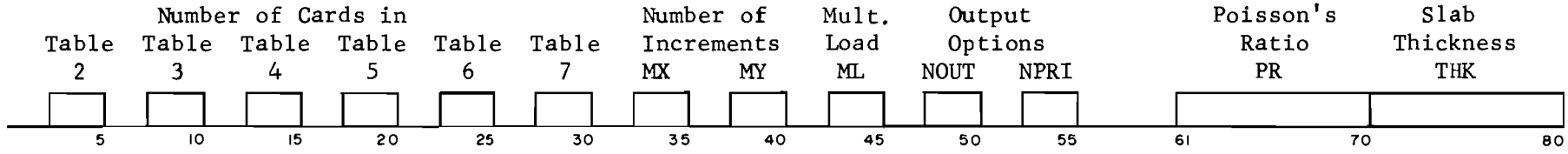
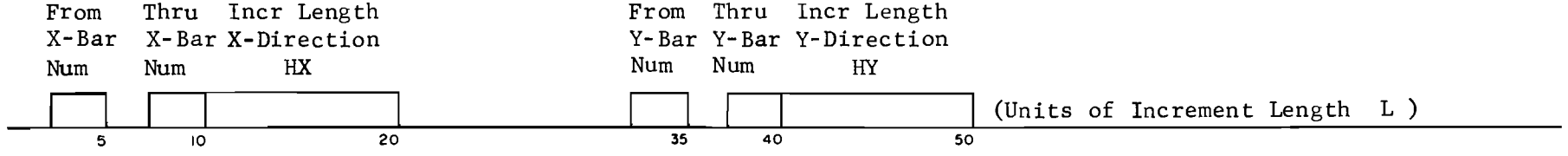


TABLE 1. PROGRAM CONTROL DATA AND CONSTANTS (1 card)



NOTE: For Multiple Load enter 0 for single problems, +1 for primary problems, and -1 for secondary problems.
For Output Option NOUT enter 0 for long form of output and 1 for short form of output.
For Output Option NPRI enter 1 if a listing of the entire input array is desired, and 0 for no listing.
For Slab Thickness (used only in stress calculations) enter the slab thickness if principal stresses are desired; if blank only principal moments will be calculated.

TABLE 2. INCREMENT LENGTH DATA (any number of cards as shown in Table 1)



This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

TABLE 3. BENDING STIFFNESS DATA (any number of cards as shown in Table 1)

From Station (X) I1	From Station (Y) J1	Thru Station (X) I2	Thru Station (Y) J2	Bending Stiffness (F×L) DX	Bending Stiffness (F×L) DY
5	10	15	20	30	40

TABLE 4. TWISTING STIFFNESS DATA (any number of cards as shown in Table 1)

From Station (X) I1	From Station (Y) J1	Thru Station (X) I2	Thru Station (Y) J2	Twisting Stiffness (F×L) CX	Twisting Stiffness (F×L) CY
5	10	15	20	30	40

TABLE 5. TRANSVERSE LOAD AND SUPPORT DATA (any number of cards as shown in Table 1)

From Station (X) I1	From Station (Y) J1	Thru Station (X) I2	Thru Station (Y) J2	Load (F) Q	Unit Load (F/L^2) QQ	Spring (F/L) S	Unit Spring (F/L^3) SS
5	10	15	20	30	40	50	60

TABLE 6. x-DIRECTION LOAD DATA (any number of cards as shown in Table 1)

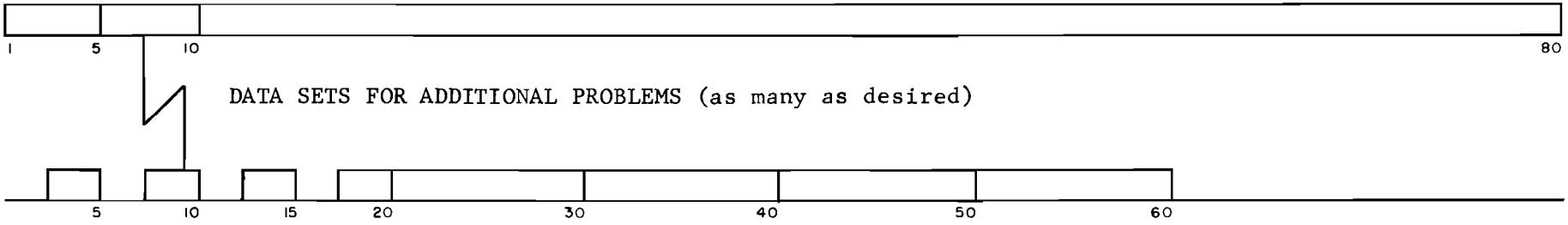
From X-Bar (X) I1	From X-Bar (Y) J1	Thru Y-Bar (X) I2	Thru Y-Bar (Y) J2	External Couple (F×L) TX	External Couple (F×L/L) TTX	Axial Tension (F) PX	Axial Tension (F/L) PPX
5	10	15	20	30	40	50	60

TABLE 7. y-DIRECTION LOAD DATA (any number of card as shown in Table 1)

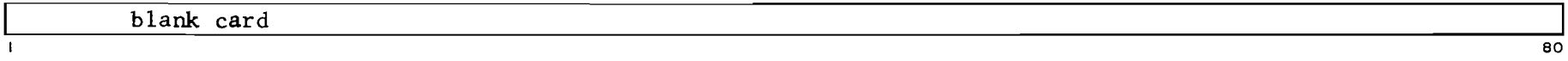
From X-Bar (X) I1	From X-Bar (Y) J1	Thru Y-Bar (X) I2	Thru Y-Bar (Y) J2	External Couple (F×L) TY	External Couple (F×L/L) TTY	Axial Tension (F) PY	Axial Tension (F/L) PPY
5	10	15	20	30	40	50	60

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

ADDITIONAL PROBLEMS MAY BE STACKED BEHIND THE FIRST PROBLEM; EACH MUST START WITH AN IDENTIFICATION OF PROBLEM CARD FOLLOWED BY TABLES 1 THRU 7 AS SHOWN ABOVE



BLANK CARD STOPS PROGRAM



NOTE: Units for variables in Tables 2 thru 7 are shown in parenthesis using F to denote force units and L to denote length units.

This page replaces an intentionally blank page in the original --- CTR Library Digitization Team

APPENDIX 2

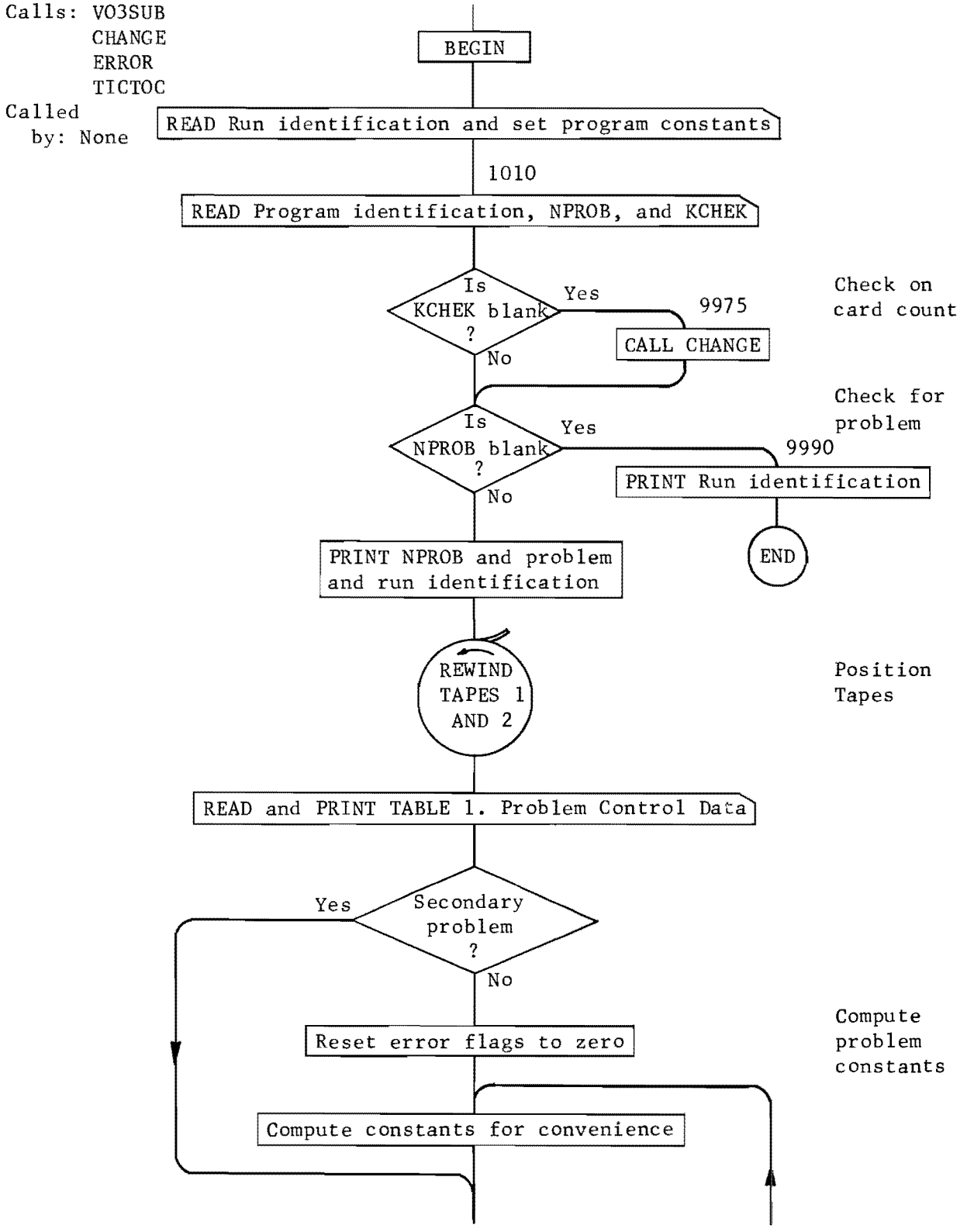
FLOW DIAGRAM FOR PROGRAM VISAB 3

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

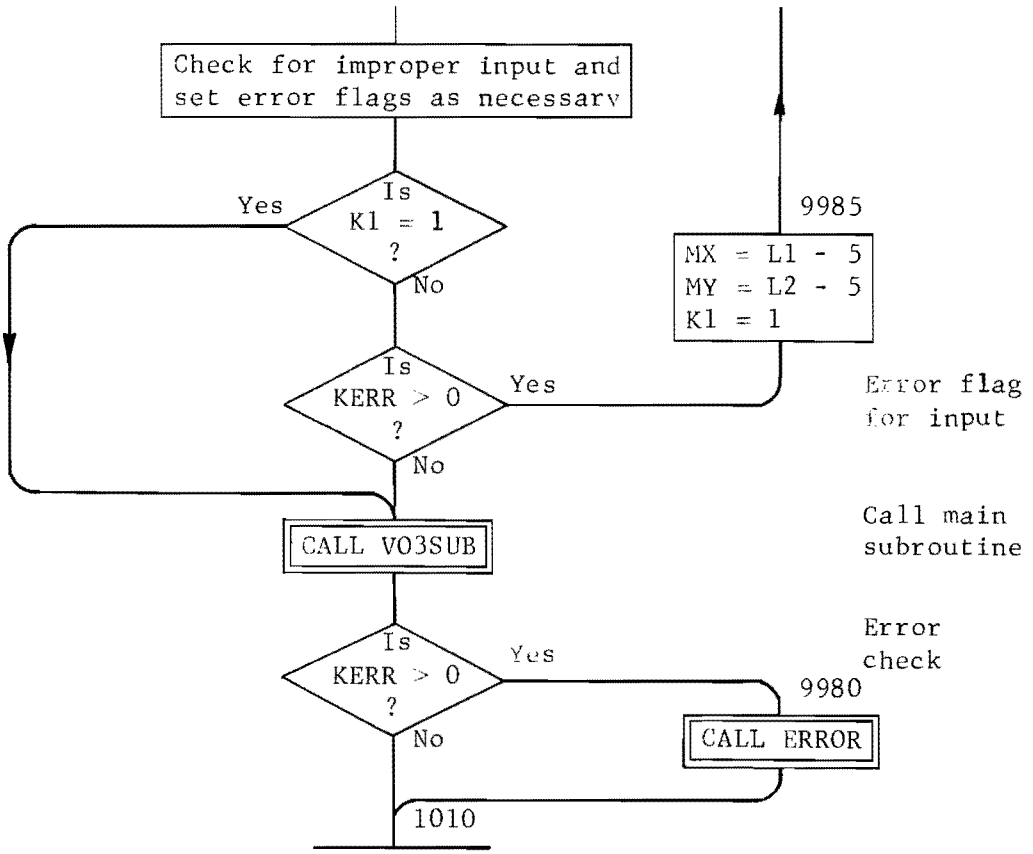
FLOW DIAGRAM FOR PROGRAM VISAB 3

Calls: VO3SUB
CHANGE
ERROR
TICTOC

Called by: None

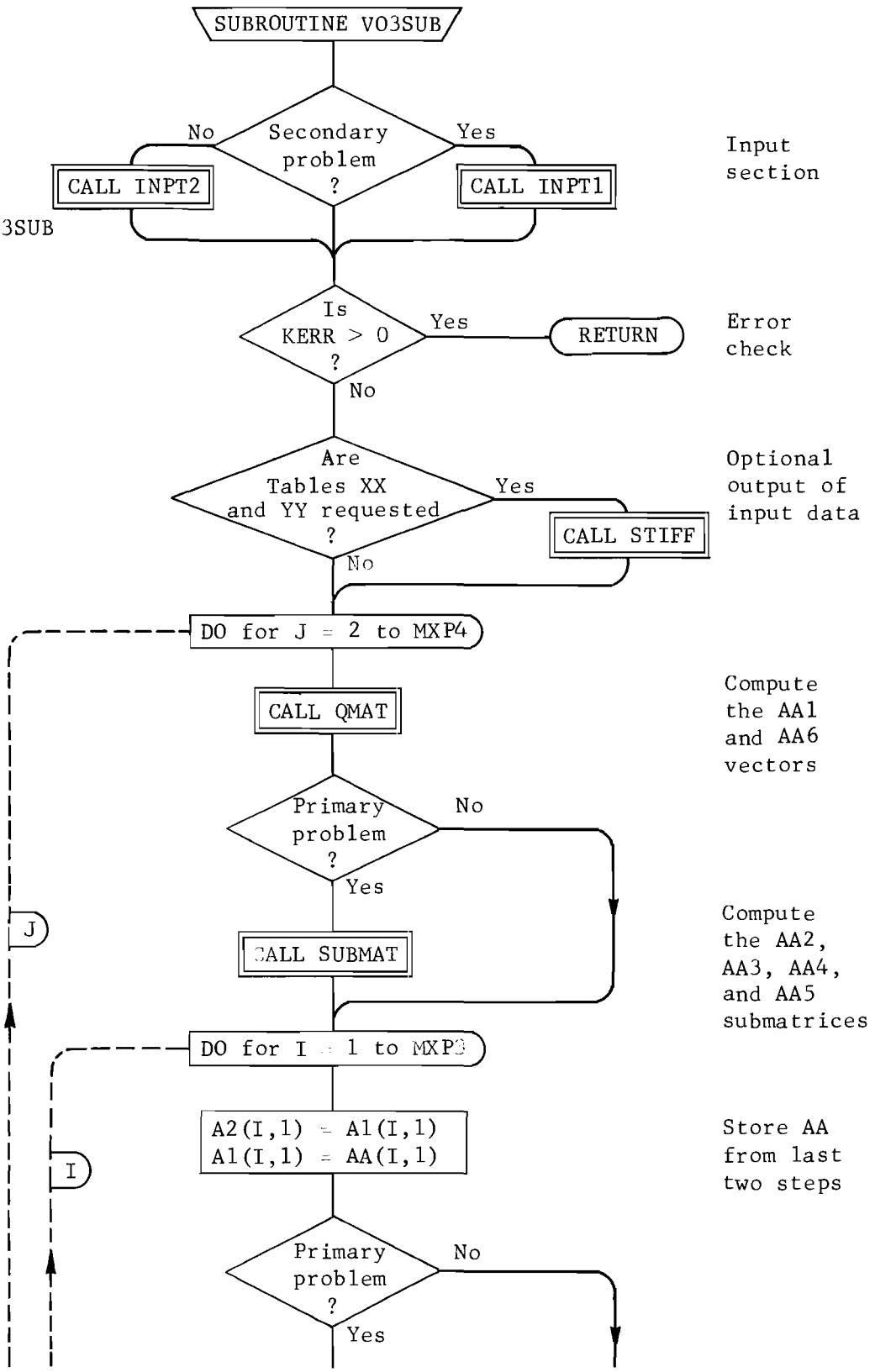


NOTE: For a listing of PROGRAM VISAB 3 see page 151.

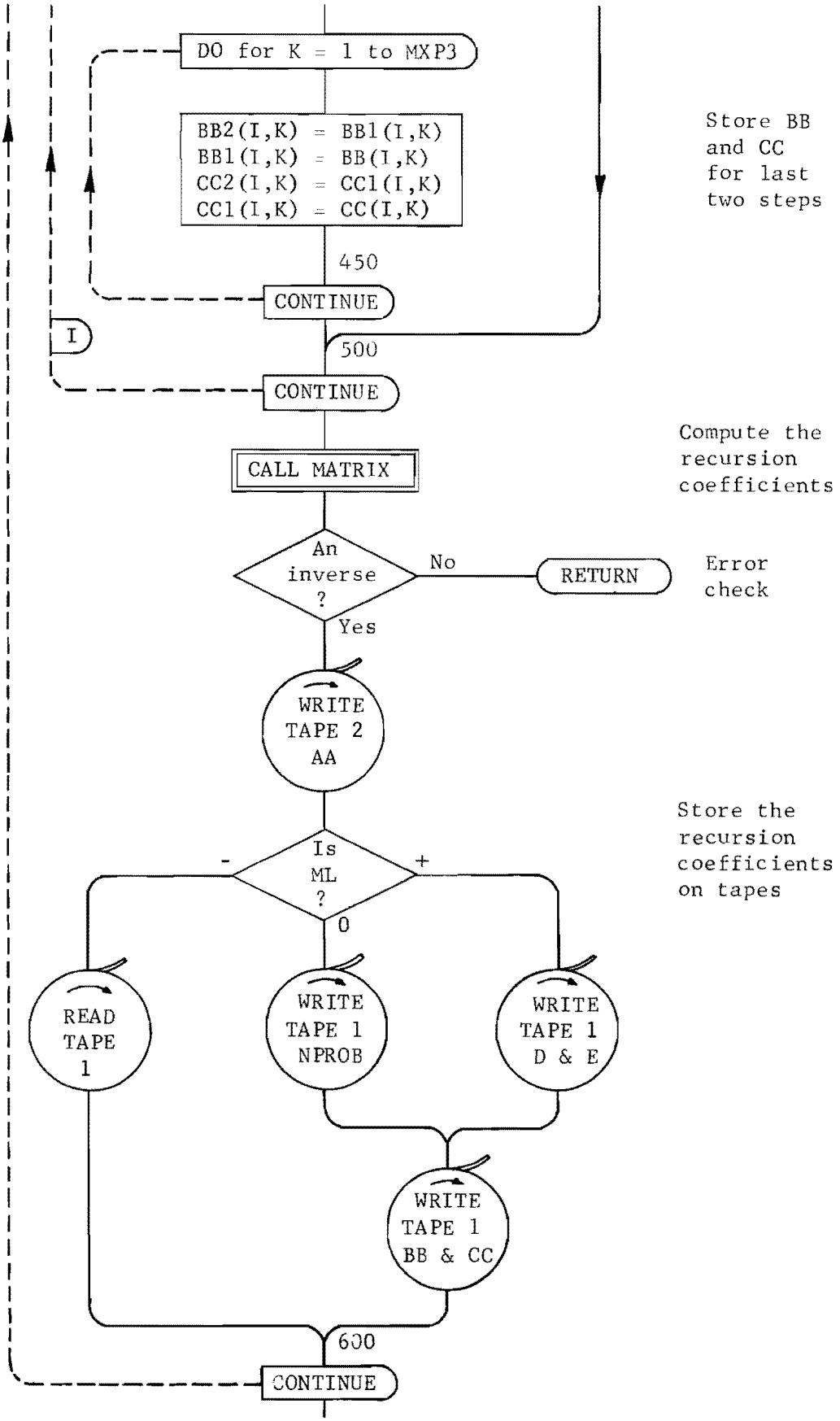

Check for improper input and set error flags as necessary
Is K1 = 1 ?
Yes
No
9985
MX = L1 - 5
MY = L2 - 5
K1 = 1
Is KERR > 0 ?
Yes
No
Error flag for input
CALL VO3SUB
Call main subroutine
Is KERR > 0 ?
Yes
No
Error check
9980
CALL ERROR
1010

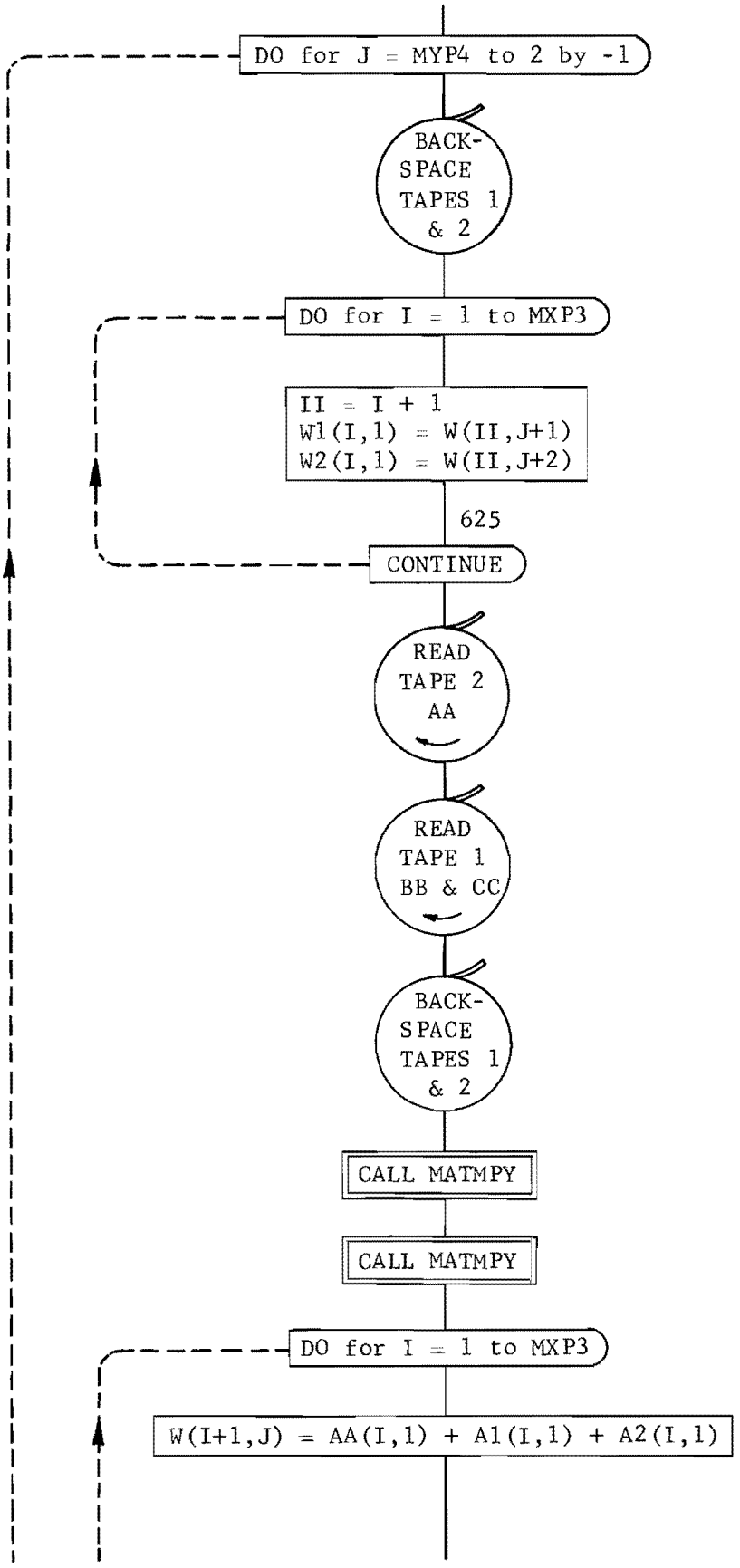
FLOW DIAGRAM FOR SUBROUTINE VO3SUB

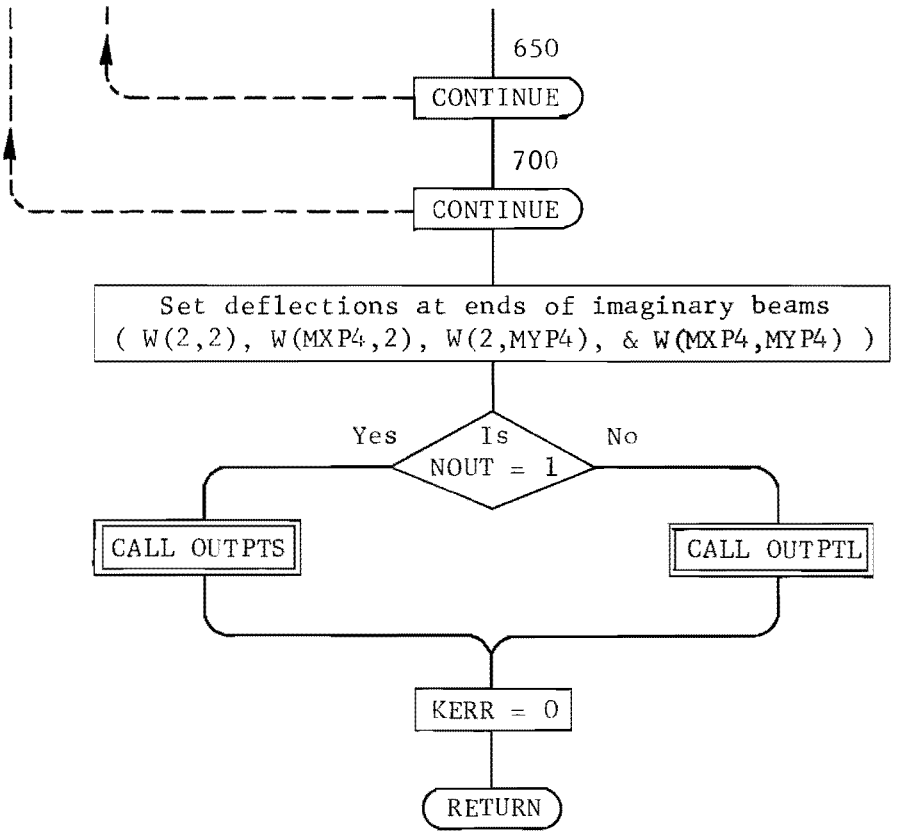
Calls: INPT1
INPT2
STIFF
QMAT
SUBMAT
MATRIX
MATMPY
OUTPTL
OUTPTS
Called by: VO3SUB



NOTE: For a listing of SUBROUTINE VO3SUB see page 156.

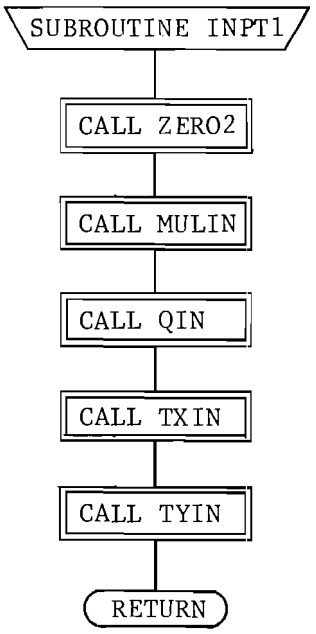

DO for K = 1 to MXP3
BB2(I,K) = BB1(I,K)
BB1(I,K) = BB(I,K)
CC2(I,K) = CC1(I,K)
CC1(I,K) = CC(I,K)
Store BB and CC for last two steps
450
CONTINUE
I
500
CONTINUE
Compute the recursion coefficients
CALL MATRIX
An inverse ?
No
RETURN
Error check
Yes
WRITE TAPE 2 AA
Store the recursion coefficients on tapes
Is ML ?
-
+
0
READ TAPE 1
WRITE TAPE 1 NPROB
WRITE TAPE 1 D & E
WRITE TAPE 1 BB & CC
600
CONTINUE


DO for J = MYP4 to 2 by -1
BACK-SPACE TAPES 1 & 2
Position tapes
DO for I = 1 to MXP3
II = I + 1
W1(I,1) = W(II,J+1)
W2(I,1) = W(II,J+2)
Set W vectors at J+1 and J+2
625
CONTINUE
READ TAPE 2 AA
READ TAPE 1 BB & CC
Read the recursion coefficients AA, BB, and CC from tapes
BACK-SPACE TAPES 1 & 2
Position tapes
CALL MATMPY
A1 = BB * W1
CALL MATMPY
A2 = CC * W2
DO for I = 1 to MXP3
W(I+1,J) = AA(I,1) + A1(I,1) + A2(I,1)
Solve for W vector at J


650
CONTINUE
700
CONTINUE
Set deflections at ends of imaginary beams
(W(2,2), W(MXP4,2), W(2,MYP4), & W(MXP4,MYP4))
Yes
Is
NOUT = 1
No
CALL OUTPTS
CALL OUTPTL
Output section
KERR = 0
Reset error flag
RETURN

FLOW DIAGRAM FOR SUBROUTINE INPT1

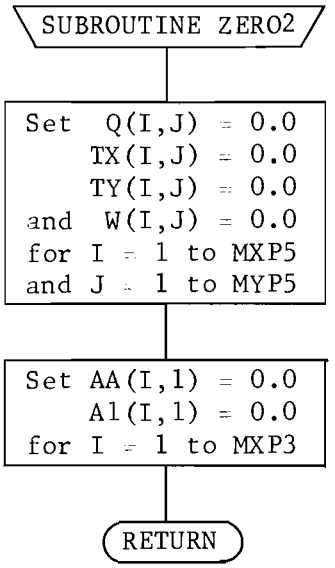
Calls: ZERO2
MULIN
QIN
TXIN
TYIN
Called by: VO3SUB



NOTE: For a listing of SUBROUTINE INPT1 see page 162.

FLOW DIAGRAM FOR SUBROUTINE ZERO2

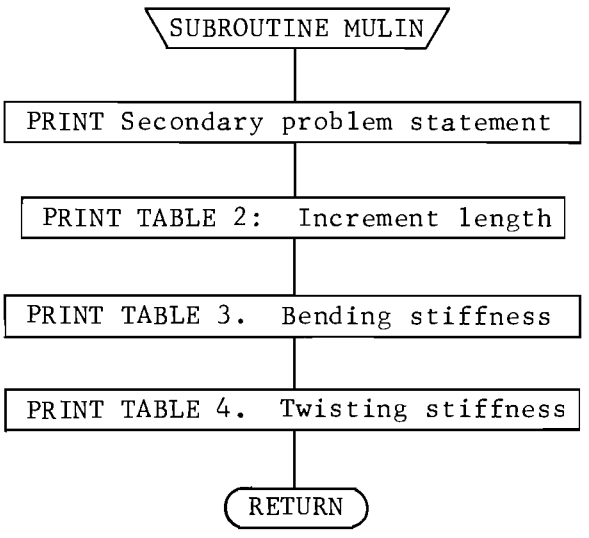
Calls: None
Called by: INPT1



NOTE: For a listing of SUBROUTINE ZERO2 see page 163.

FLOW DIAGRAM FOR SUBROUTINE MULIN

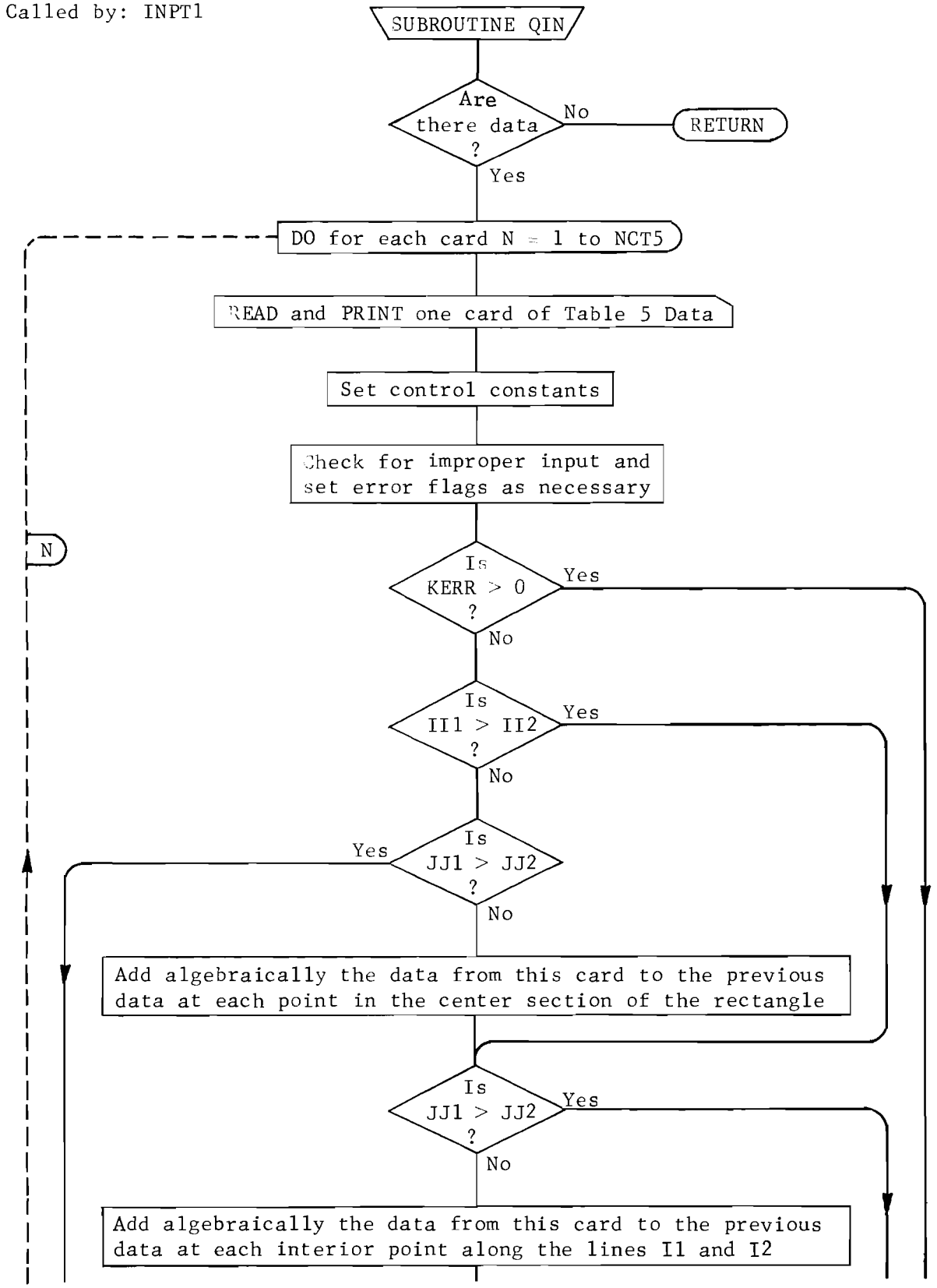
Calls: None
Called by: INPT1



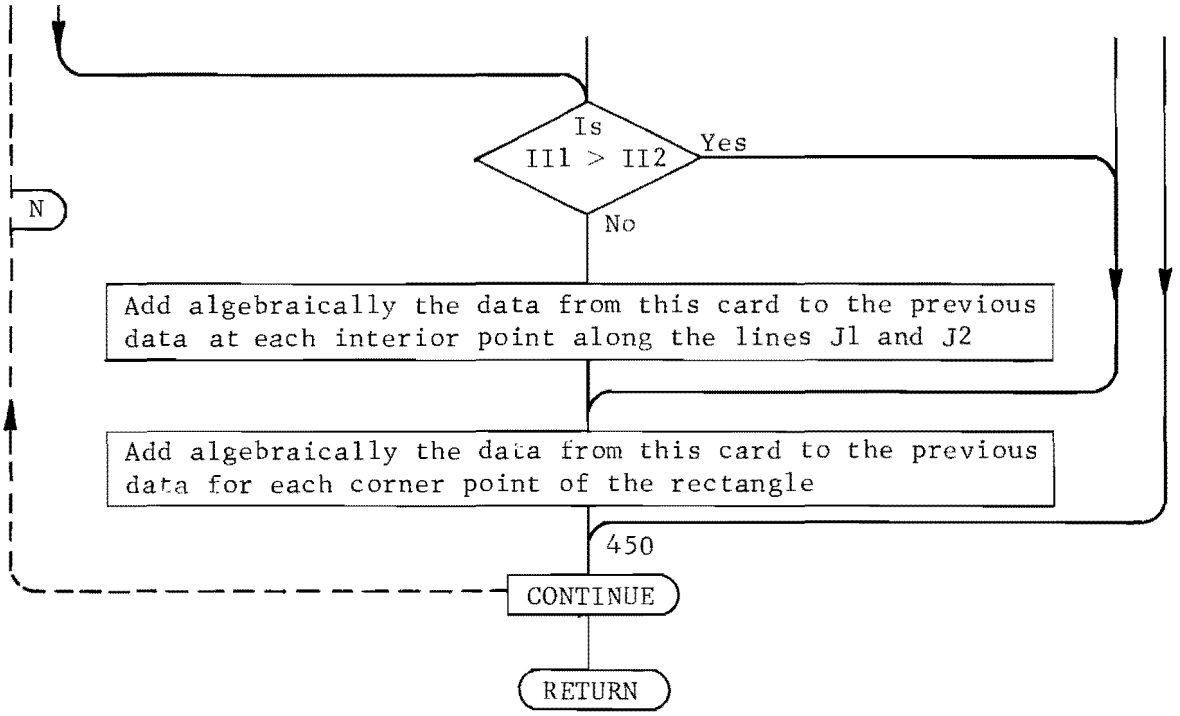
NOTE: For a listing of SUBROUTINE MULIN see page 164.

FLOW DIAGRAM FOR SUBROUTINE QIN

Calls: None
Called by: INPT1

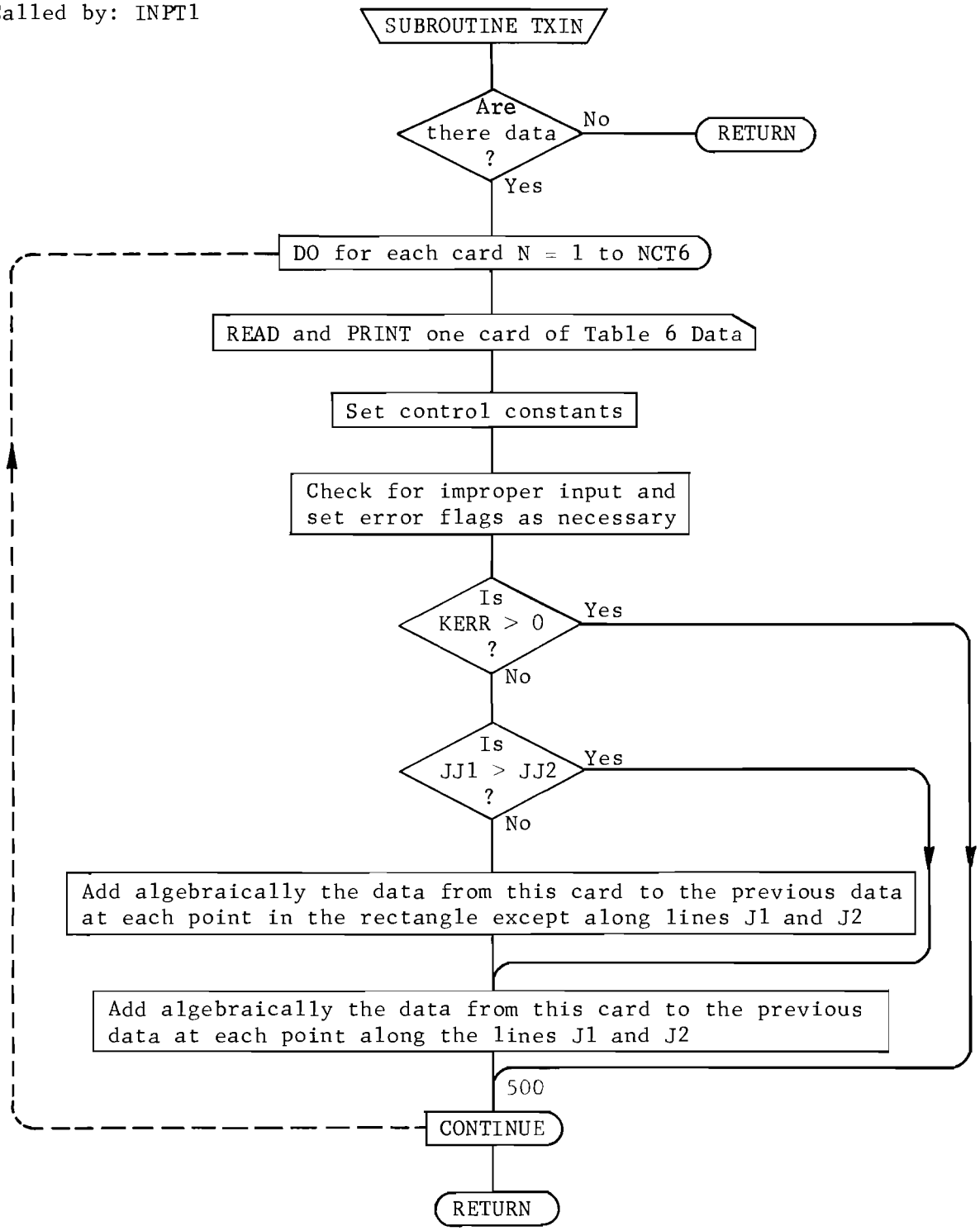


NOTE: For a listing of SUBROUTINE QIN see page 165.


Is
II1 > II2
Yes
No
N
Add algebraically the data from this card to the previous data at each interior point along the lines J1 and J2
Add algebraically the data from this card to the previous data for each corner point of the rectangle
450
CONTINUE
RETURN

FLOW DIAGRAM FOR SUBROUTINE TXIN

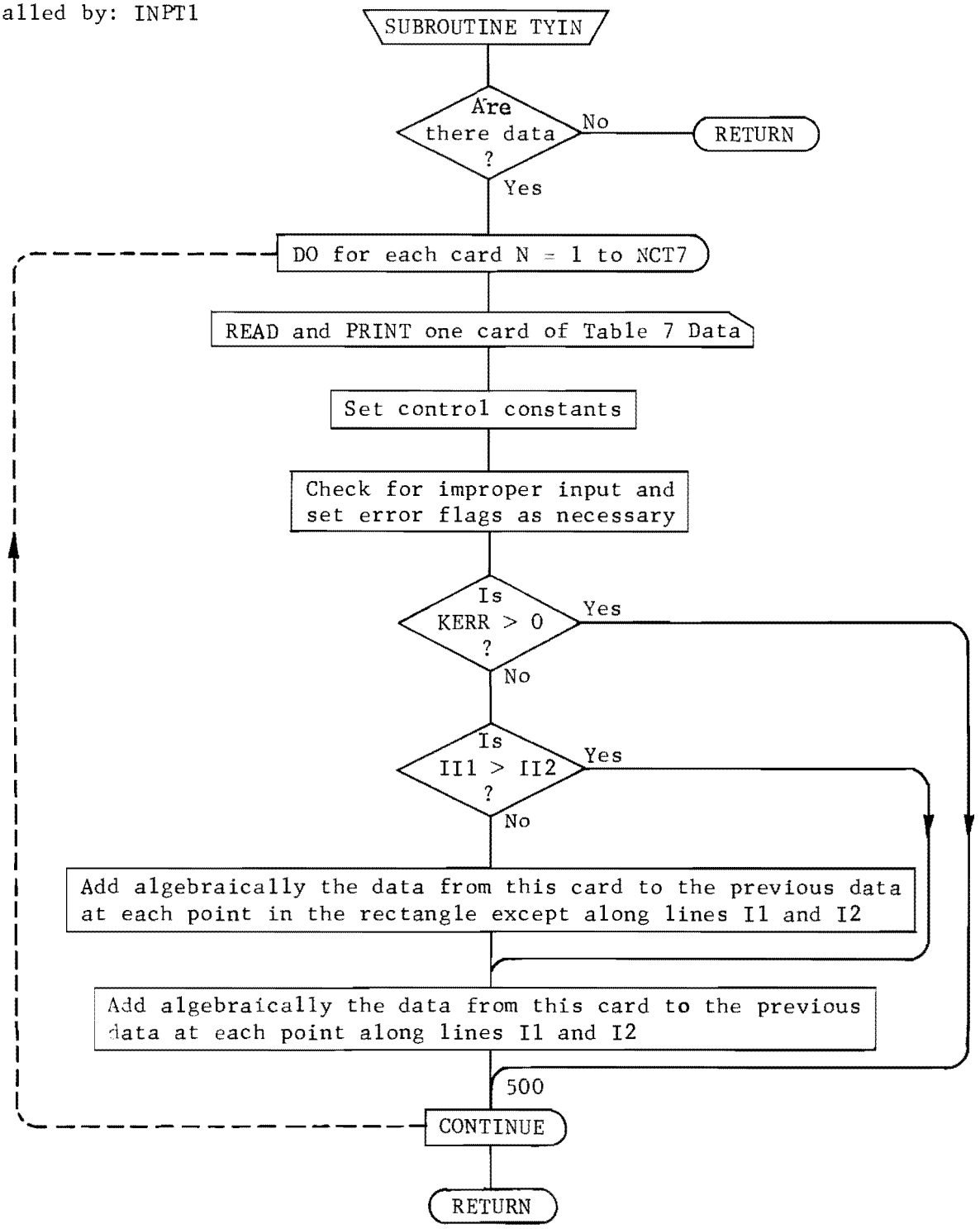
Calls: None
Called by: INPT1



NOTE: For a listing of SUBROUTINE TXIN see page 168.

FLOW DIAGRAM FOR SUBROUTINE TYIN

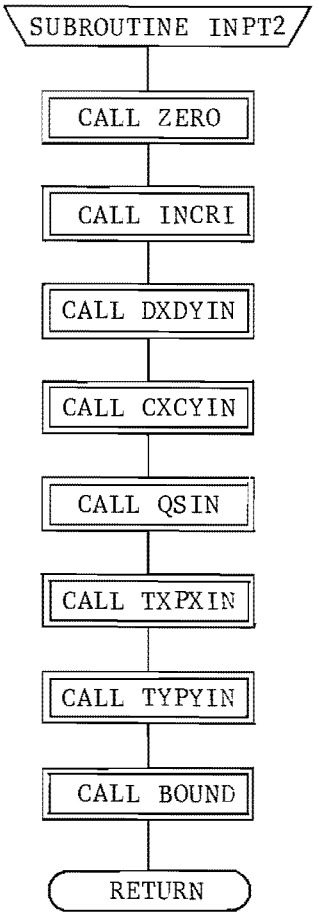
Calls: None
Called by: INPT1



NOTE: For a listing of SUBROUTINE TYIN see page 170.

FLOW DIAGRAM FOR SUBROUTINE INPT2

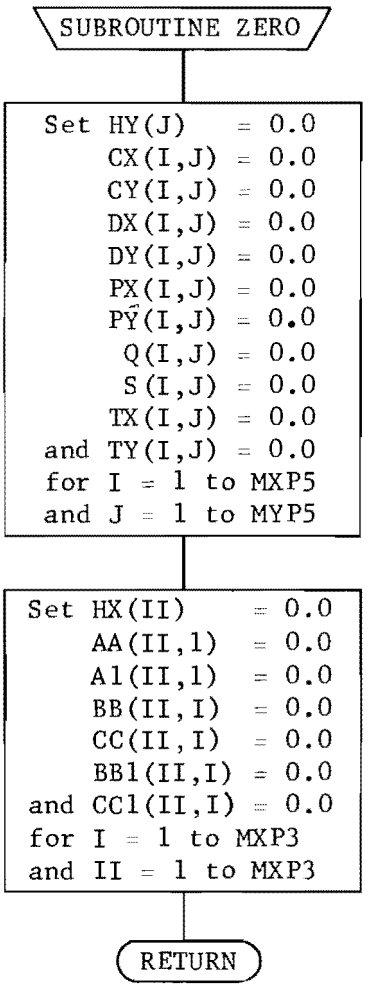
Calls: ZERO
INCRI
DXDYIN
CXCYIN
QSIN
TXPXIN
TYPYIN
BOUND
Called by: VO3SUB



NOTE: For a listing of SUBROUTINE INPT2 see page 172.

FLOW DIAGRAM FOR SUBROUTINE ZERO

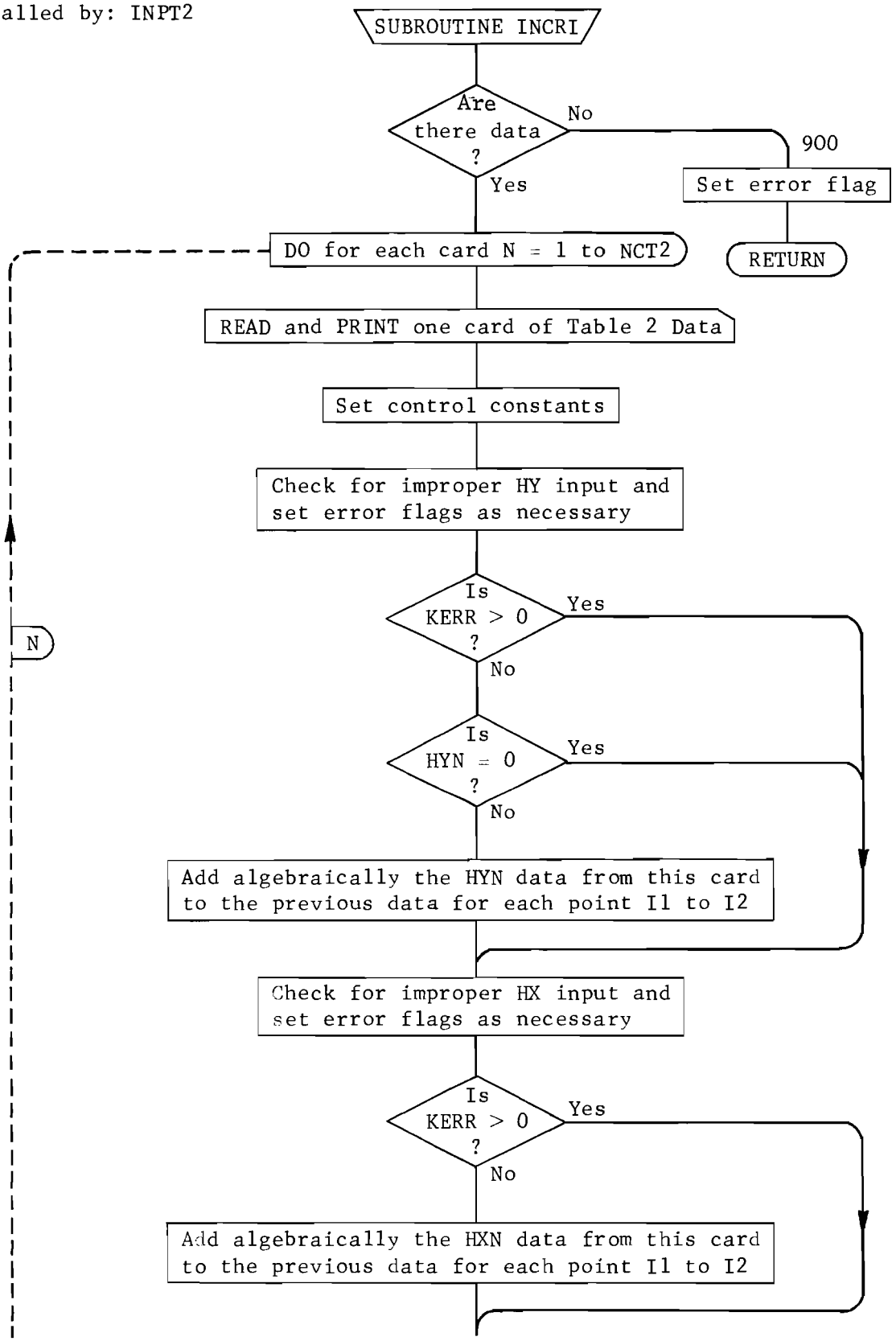
Calls: None
Called by: INPT2



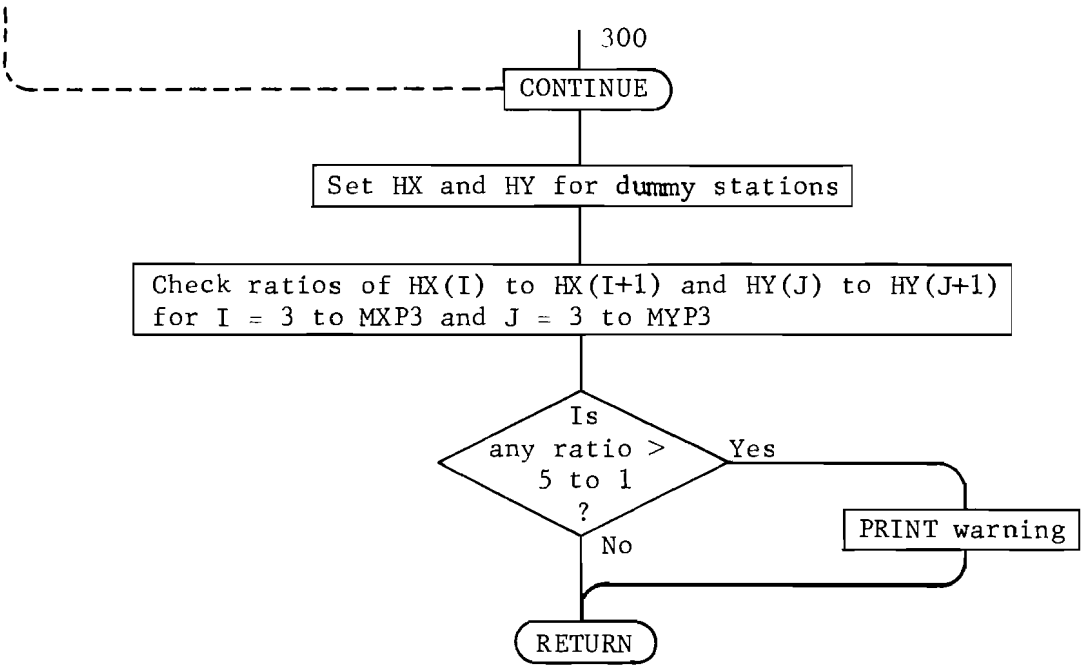
NOTE: For a listing of SUBROUTINE ZERO see page 174.

FLOW DIAGRAM FOR SUBROUTINE INCRI

Calls: None
Called by: INPT2

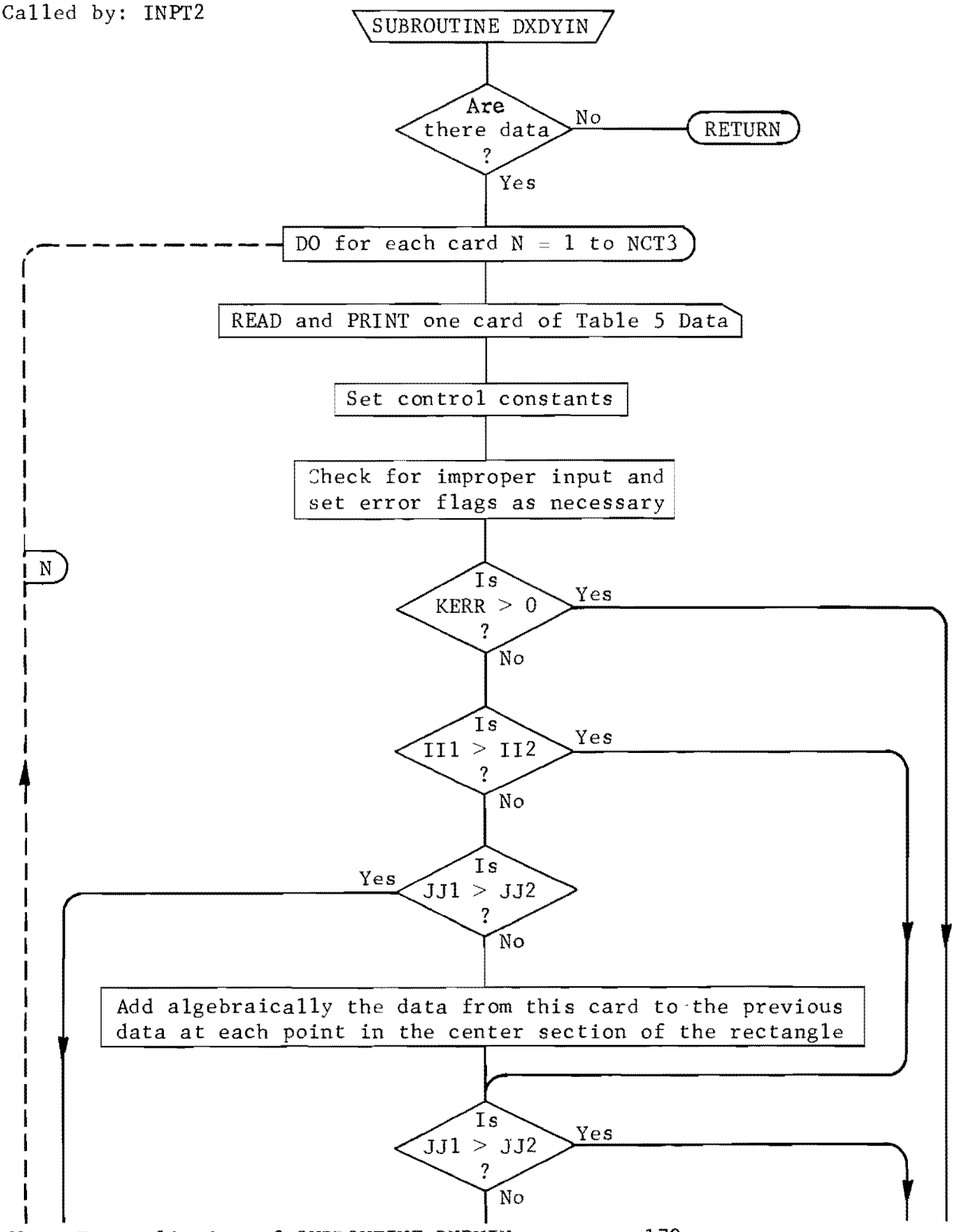


NOTE: For a listing of SUBROUTINE INCRI see page 176.

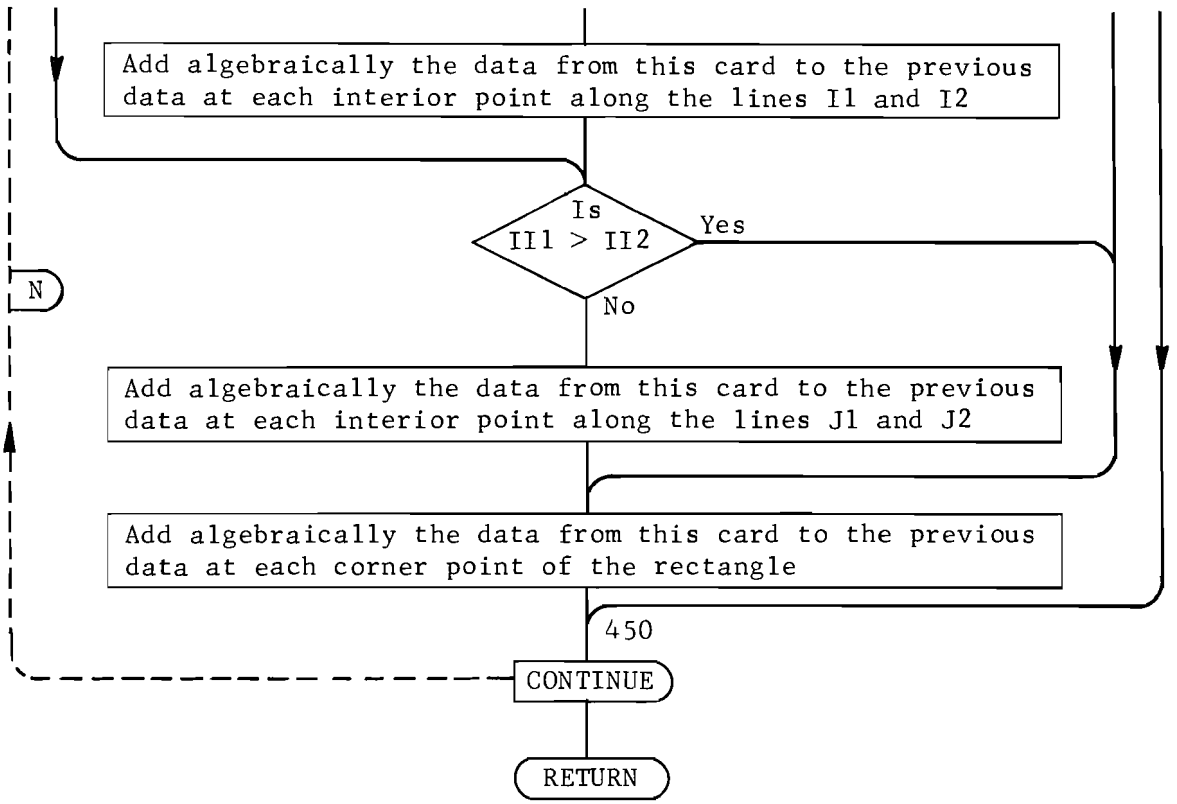

300
CONTINUE
Set HX and HY for dummy stations
Check ratios of HX(I) to HX(I+1) and HY(J) to HY(J+1)
for I = 3 to MXP3 and J = 3 to MYP3
Is
any ratio >
5 to 1
?
Yes
No
PRINT warning
RETURN

FLOW DIAGRAM FOR SUBROUTINE DXDYIN

Calls: None
Called by: INPT2

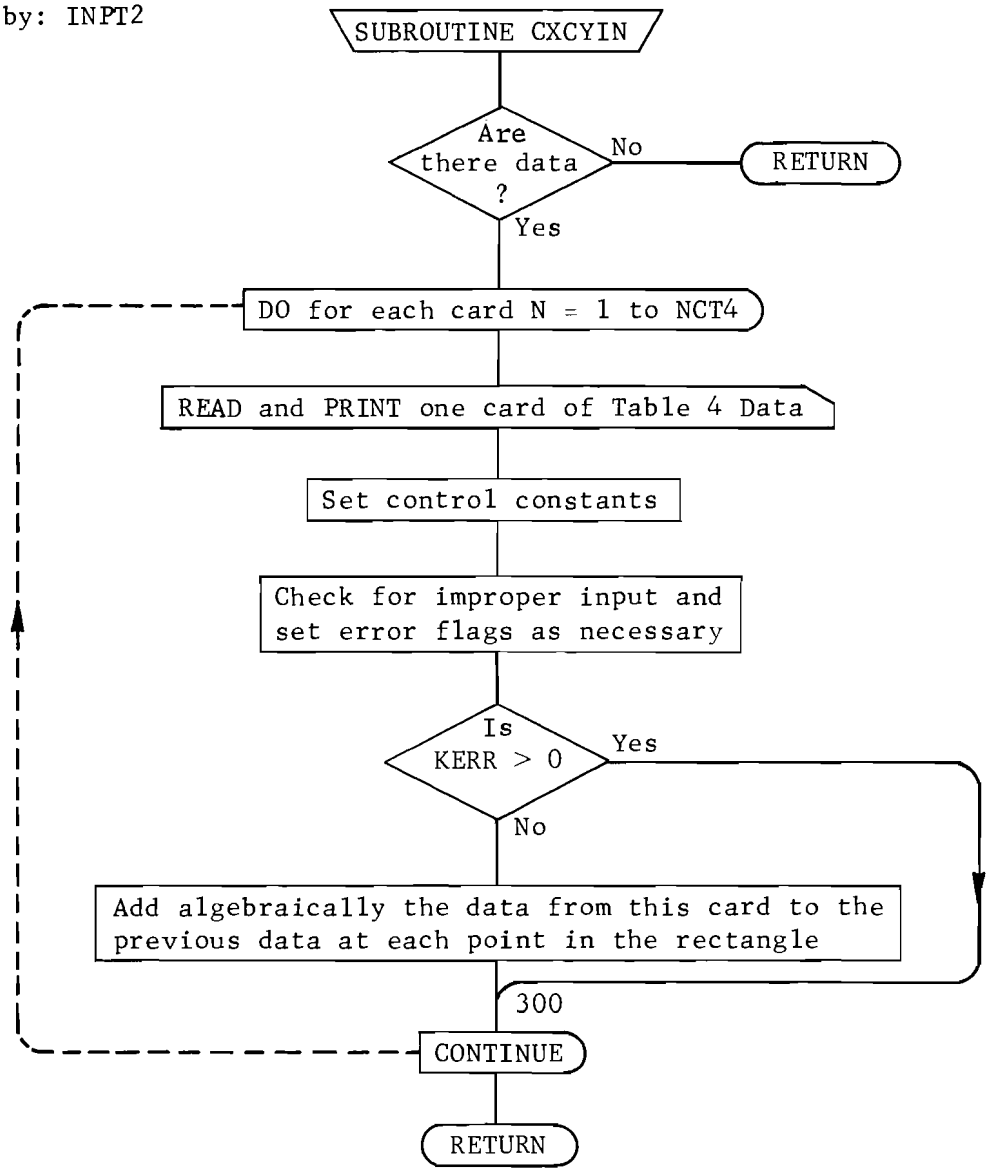


NOTE: For a listing of SUBROUTINE DXDYIN see page 179.


Add algebraically the data from this card to the previous data at each interior point along the lines I1 and I2
Is II1 > II2
Yes
No
N
Add algebraically the data from this card to the previous data at each interior point along the lines J1 and J2
Add algebraically the data from this card to the previous data at each corner point of the rectangle
450
CONTINUE
RETURN

FLOW DIAGRAM FOR SUBROUTINE CXCYIN

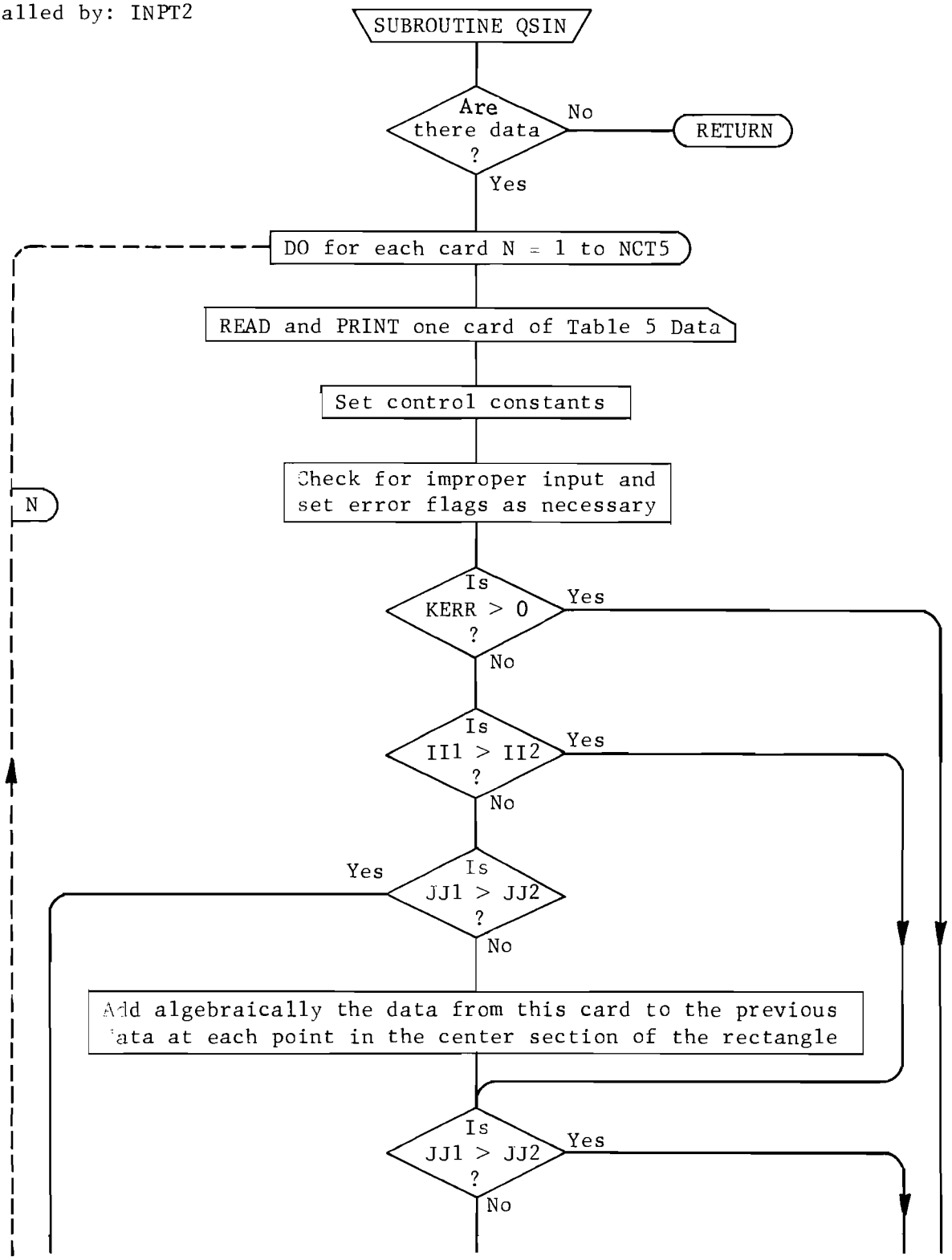
Calls: None
Called by: INPT2



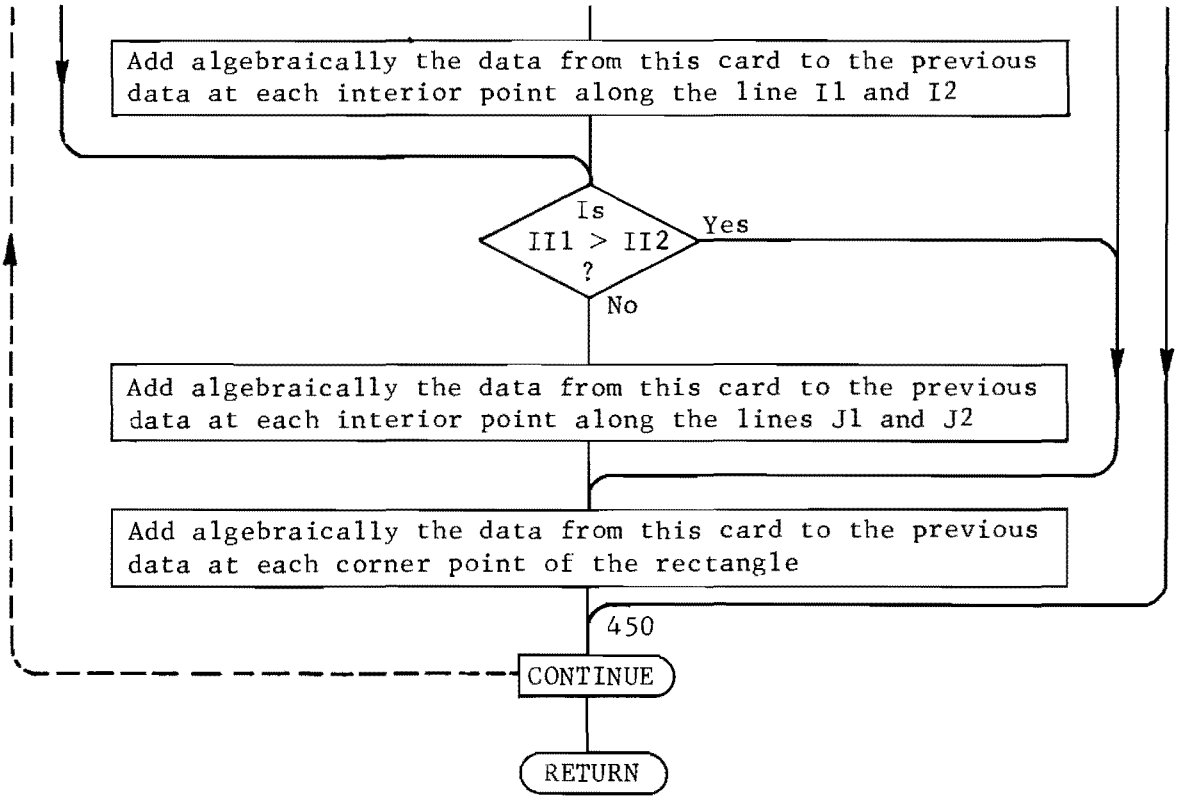
NOTE: For a listing of SUBROUTINE CXCYIN see page 182.

FLOW DIAGRAM FOR SUBROUTINE QSIN

Calls: None
Called by: INPT2

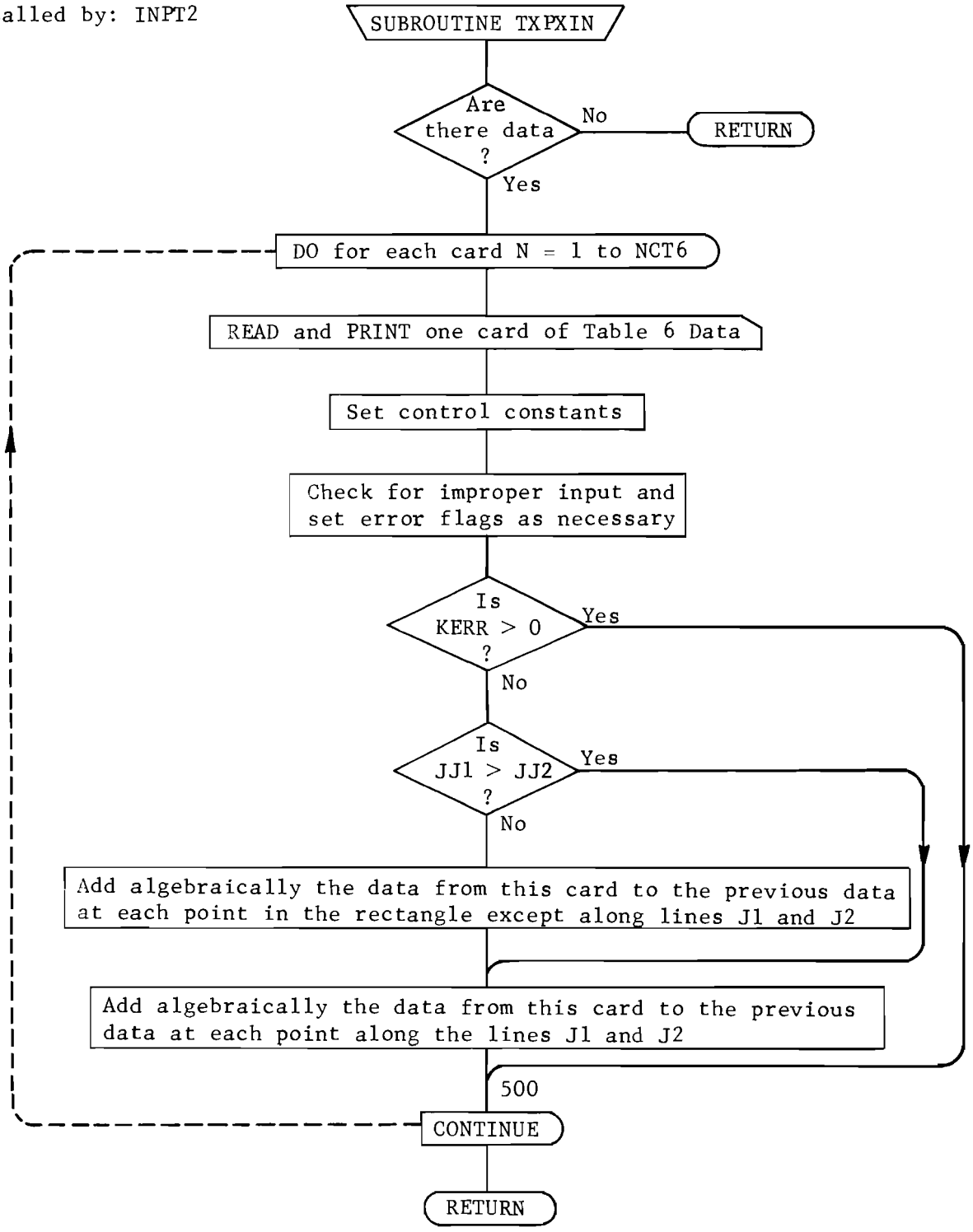


NOTE: For a listing of SUBROUTINE QSIN see page 184.


Add algebraically the data from this card to the previous data at each interior point along the line I1 and I2
Is II1 > II2 ?
Yes
No
Add algebraically the data from this card to the previous data at each interior point along the lines J1 and J2
Add algebraically the data from this card to the previous data at each corner point of the rectangle
450
CONTINUE
RETURN

FLOW DIAGRAM FOR SUBROUTINE TXPXIN

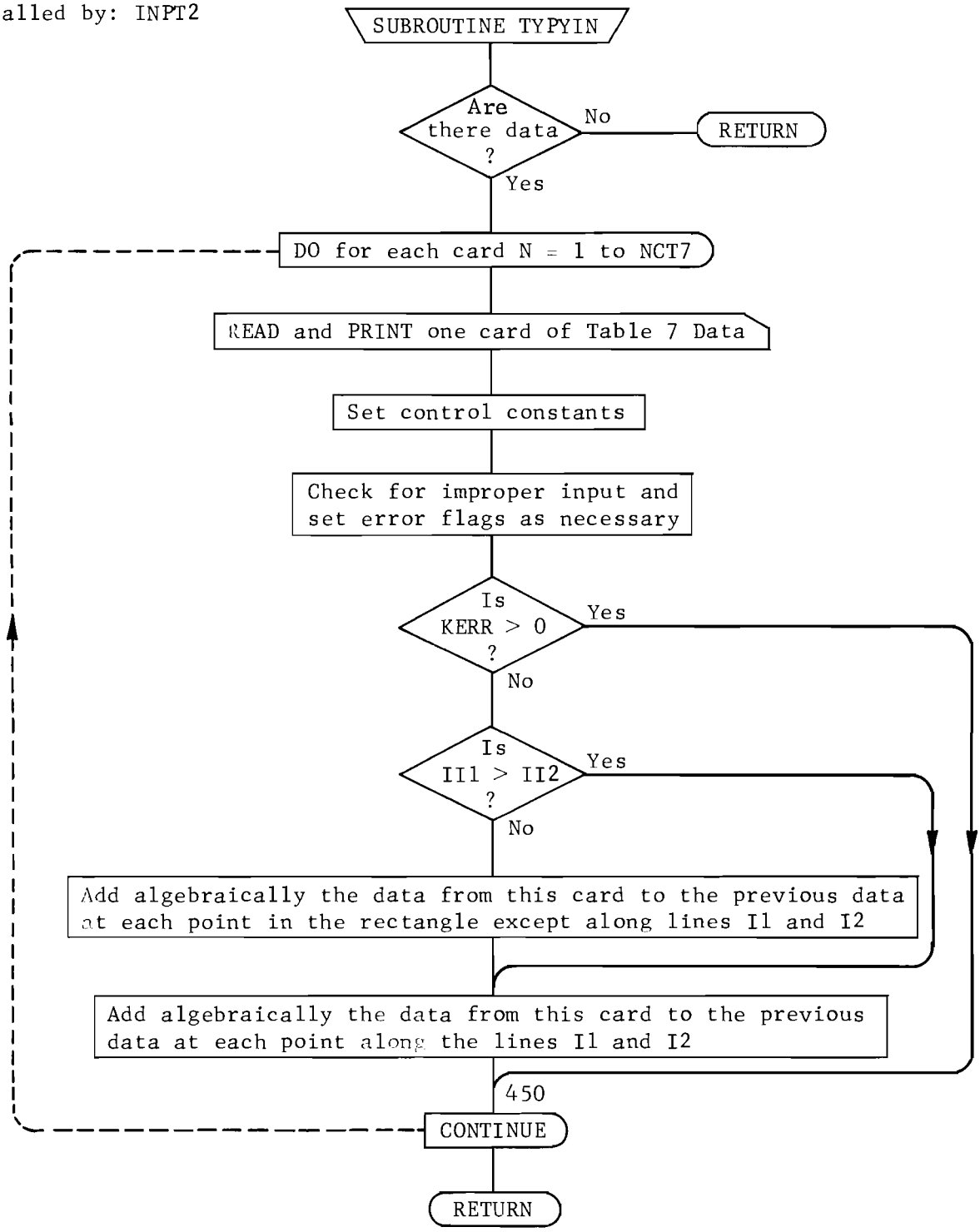
Calls: None
Called by: INPT2



NOTE: For a listing of SUBROUTINE TXPXIN see page 187.

FLOW DIAGRAM FOR SUBROUTINE TYPYIN

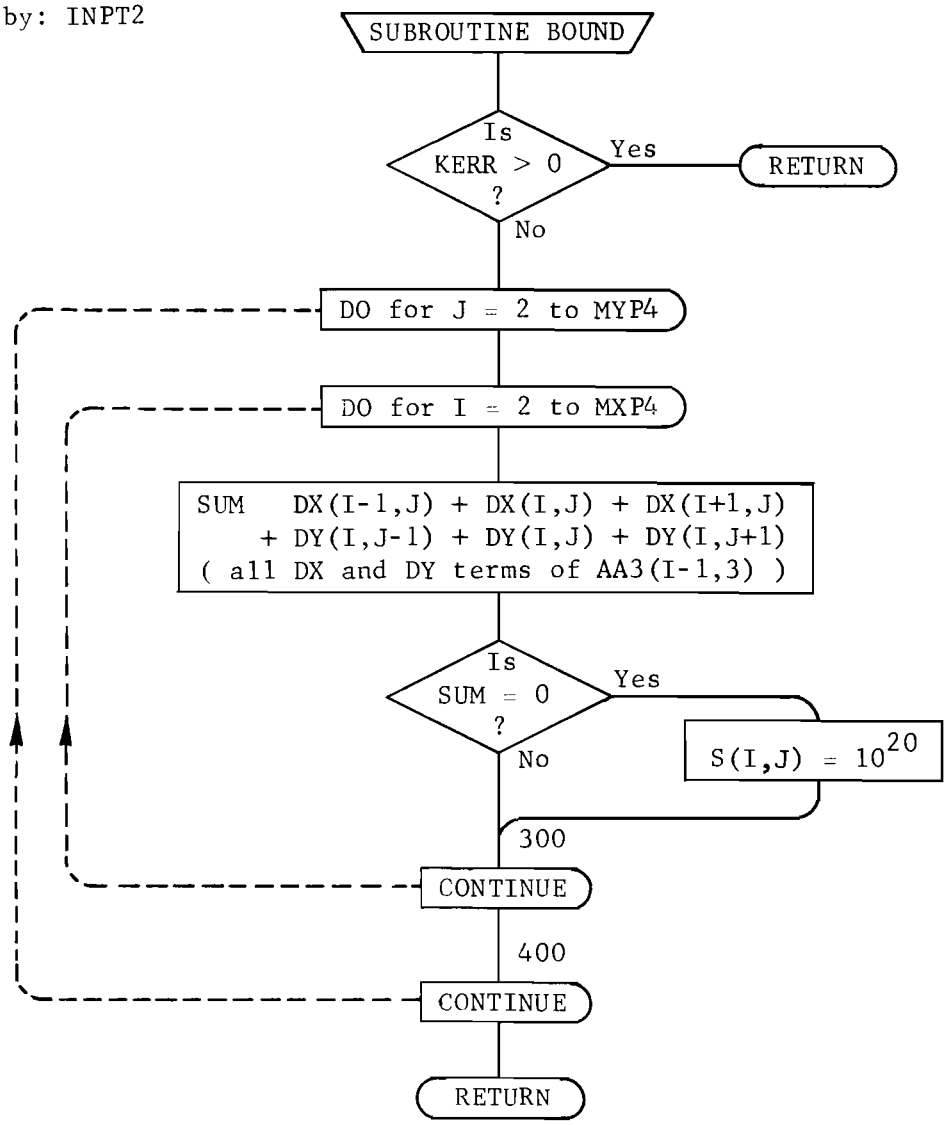
Calls: None
Called by: INPT2



NOTE: For a listing of SUBROUTINE TYPYIN see page 190.

FLOW DIAGRAM FOR SUBROUTINE BOUND

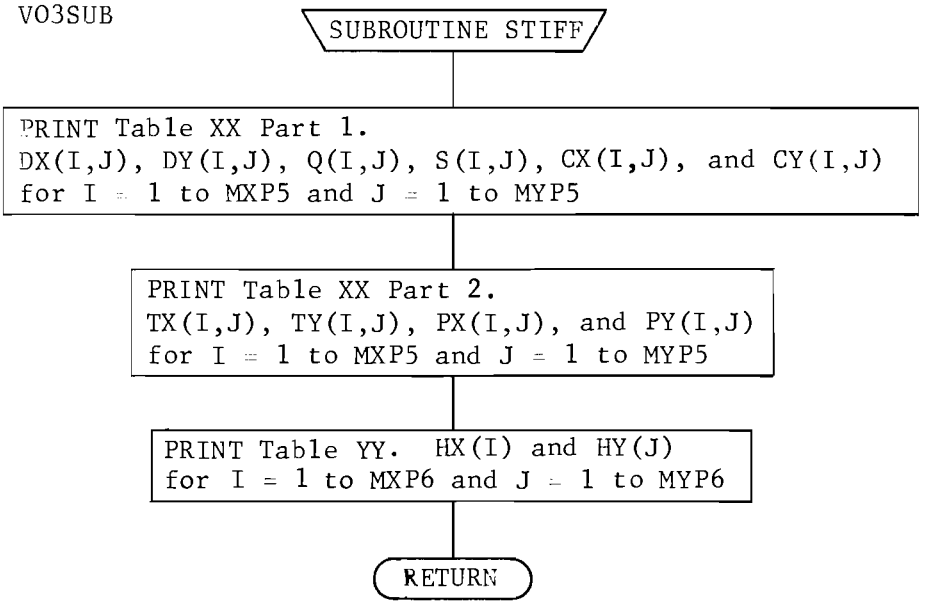
Calls: None
Called by: INPT2



NOTE: For a listing of SUBROUTINE BOUND see page 193.

FLOW DIAGRAM FOR SUBROUTINE STIFF

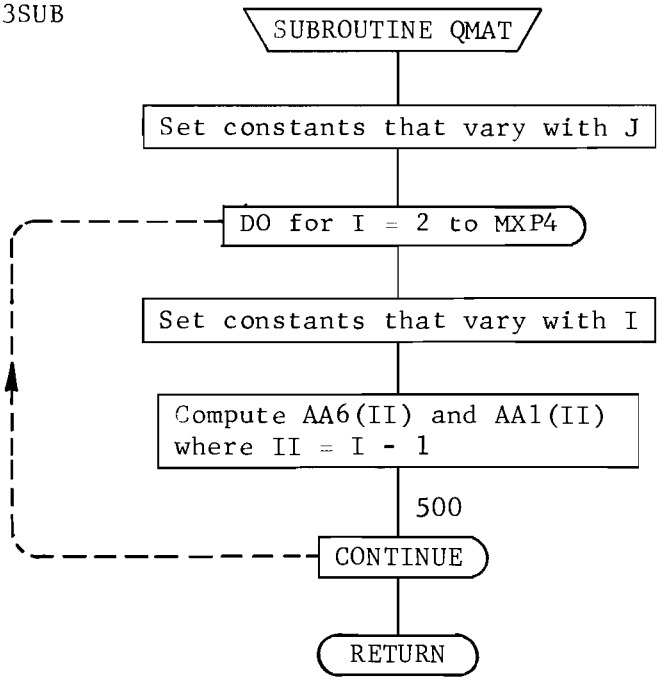
Calls: None
Called by: VO3SUB



NOTE: For a listing of SUBROUTINE STIFF see page 194.

FLOW DIAGRAM FOR SUBROUTINE QMAT

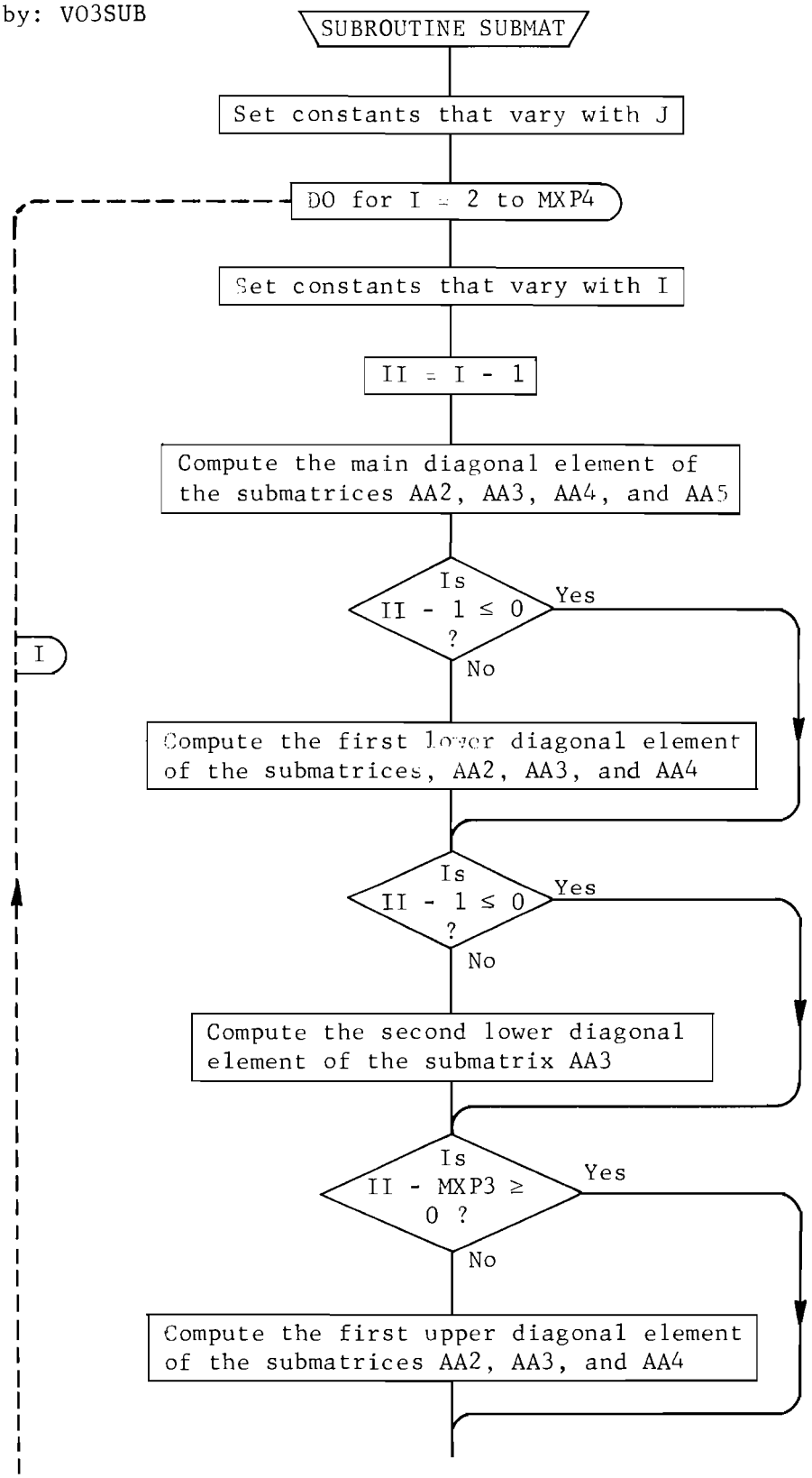
Calls: None
Called by: VO3SUB



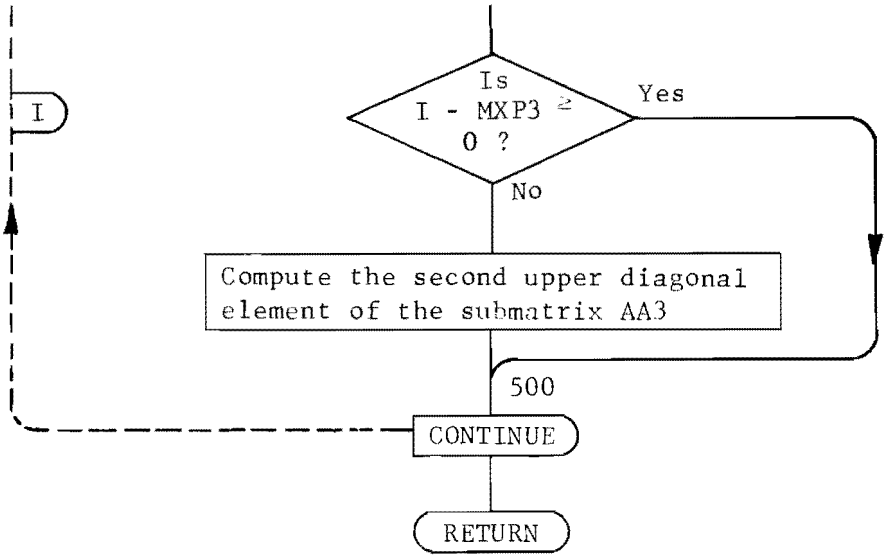
NOTE: For a listing of SUBROUTINE QMAT see page 197.

FLOW DIAGRAM FOR SUBROUTINE SUBMAT

Calls: None
Called by: VO3SUB

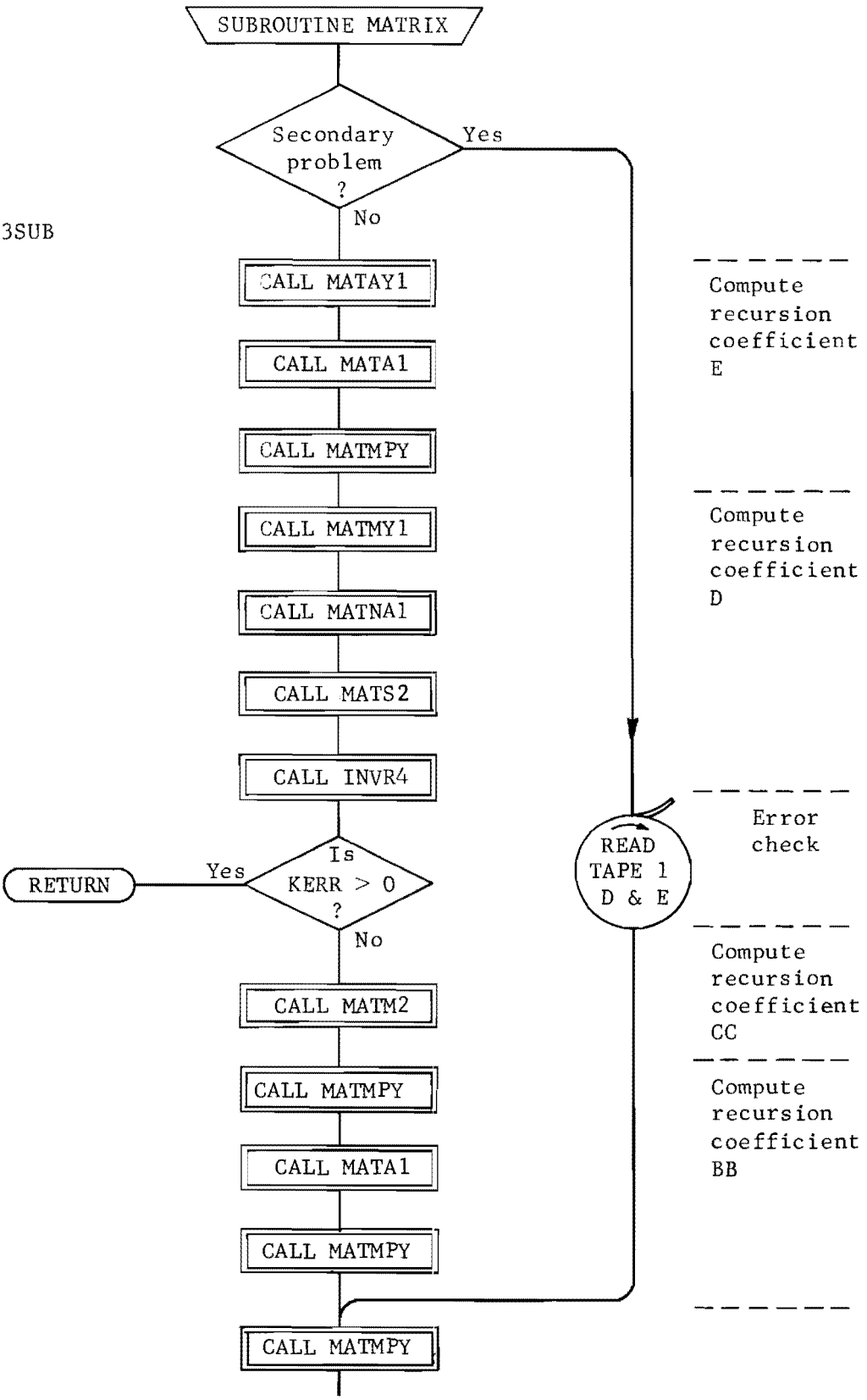


NOTE: For a listing of SUBROUTINE SUBMAT see page 199.

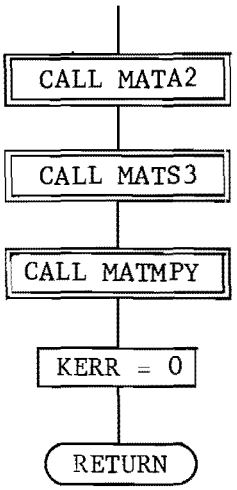

I
Is
I - MXP3 ≥
0 ?
Yes
No
Compute the second upper diagonal
element of the submatrix AA3
500
CONTINUE
RETURN

FLOW DIAGRAM FOR SUBROUTINE MATRIX

Calls: MATMY1
MATA1
MATMPY
MATNA1
MATS2
INVR4
MATM2
MATA2
MATS3
Called by: VO3SUB



NOTE: For a listing of SUBROUTINE MATRIX see page 203.

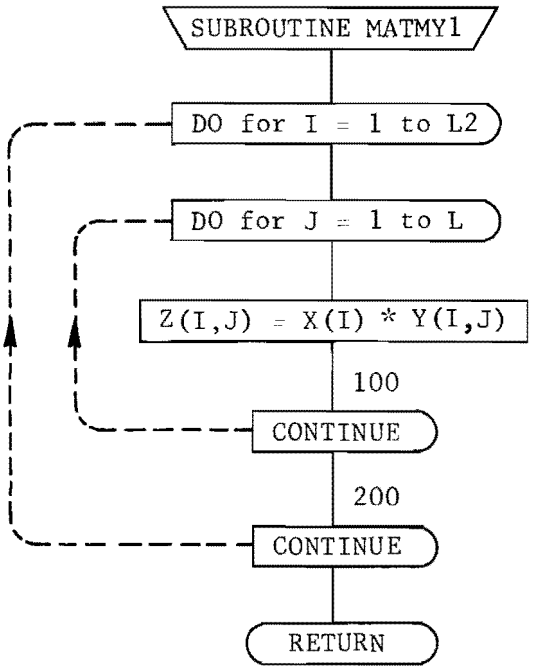


Compute recursion coefficient AA

Reset error flag

FLOW DIAGRAM FOR SUBROUTINE MATMY1

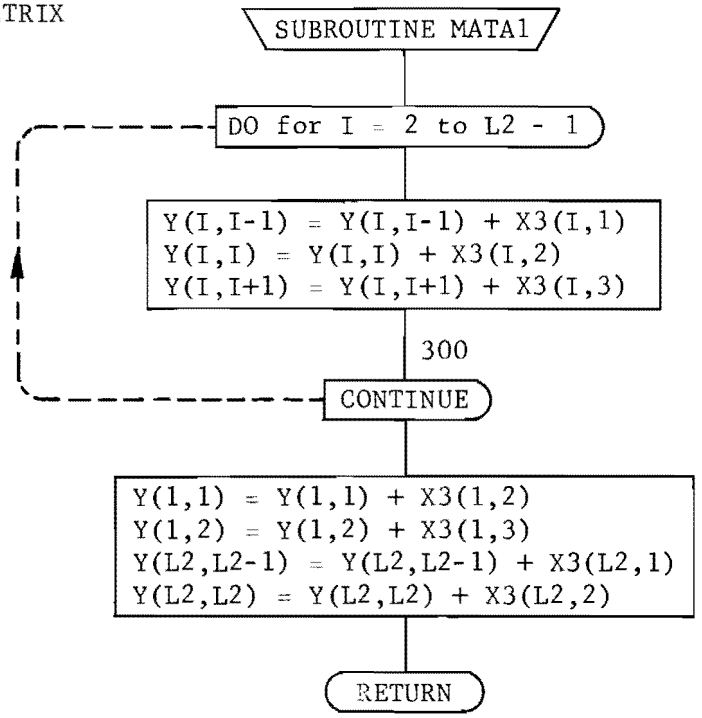
Calls: None
Called by: MATRIX



NOTE: For a listing of SUBROUTINE MATMY1 see page 206.

FLOW DIAGRAM FOR SUBROUTINE MATA1

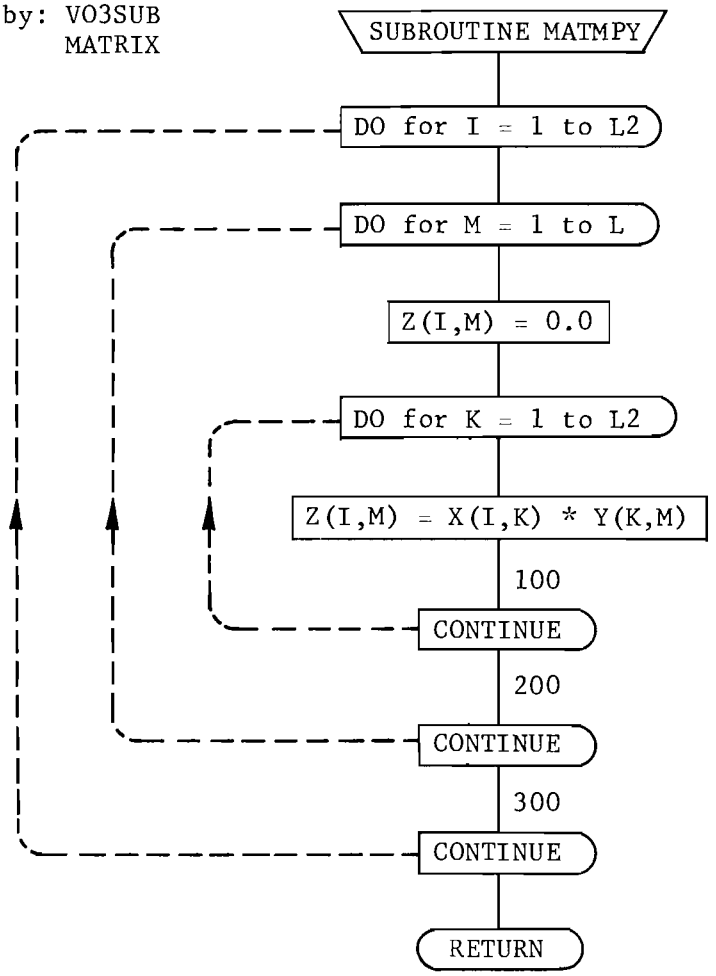
Calls: None
Called by: MATRIX



NOTE: For a listing SUBROUTINE MATA1 see page 207.

FLOW DIAGRAM FOR SUBROUTINE MATMPY

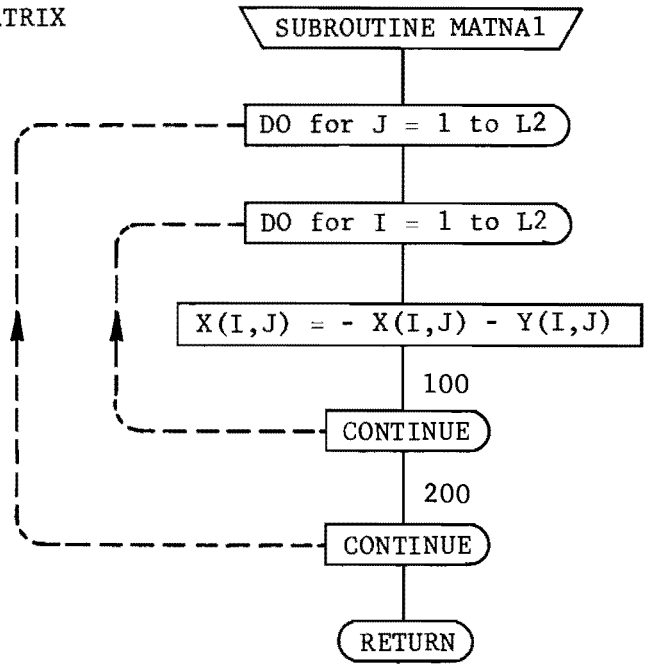
Calls: None
Called by: VO3SUB
MATRIX



NOTE: For a listing of SUBROUTINE MATMPY see page 208.

FLOW DIAGRAM FOR SUBROUTINE MATNA1

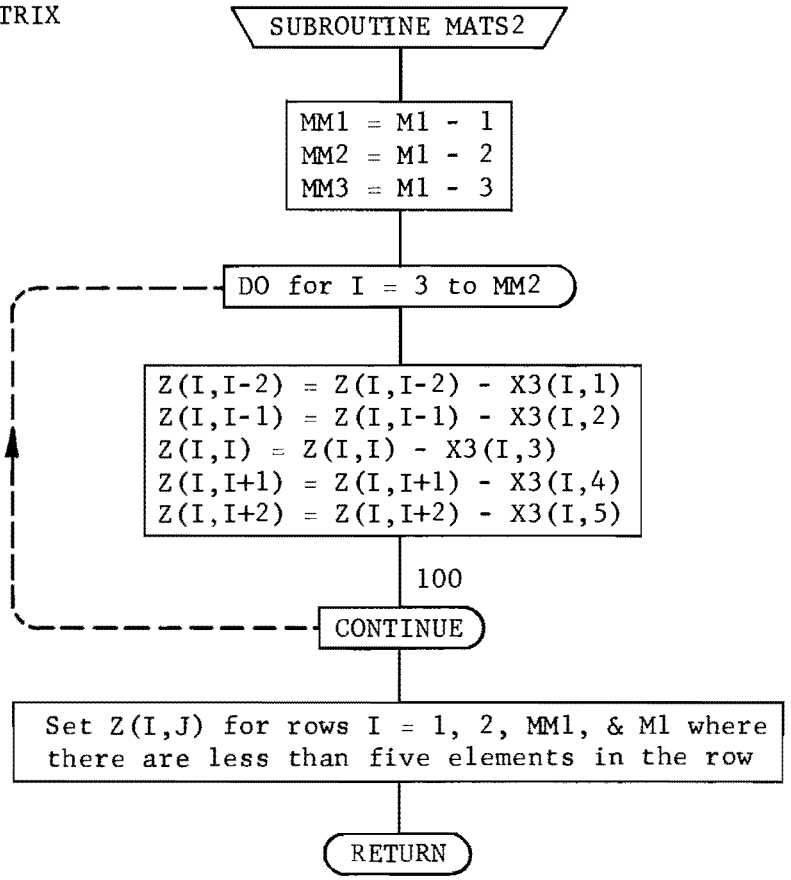
Calls: None
Called by: MATRIX



NOTE: For a listing of SUBROUTINE MATNA1 see page 209.

FLOW DIAGRAM FOR SUBROUTINE MATS2

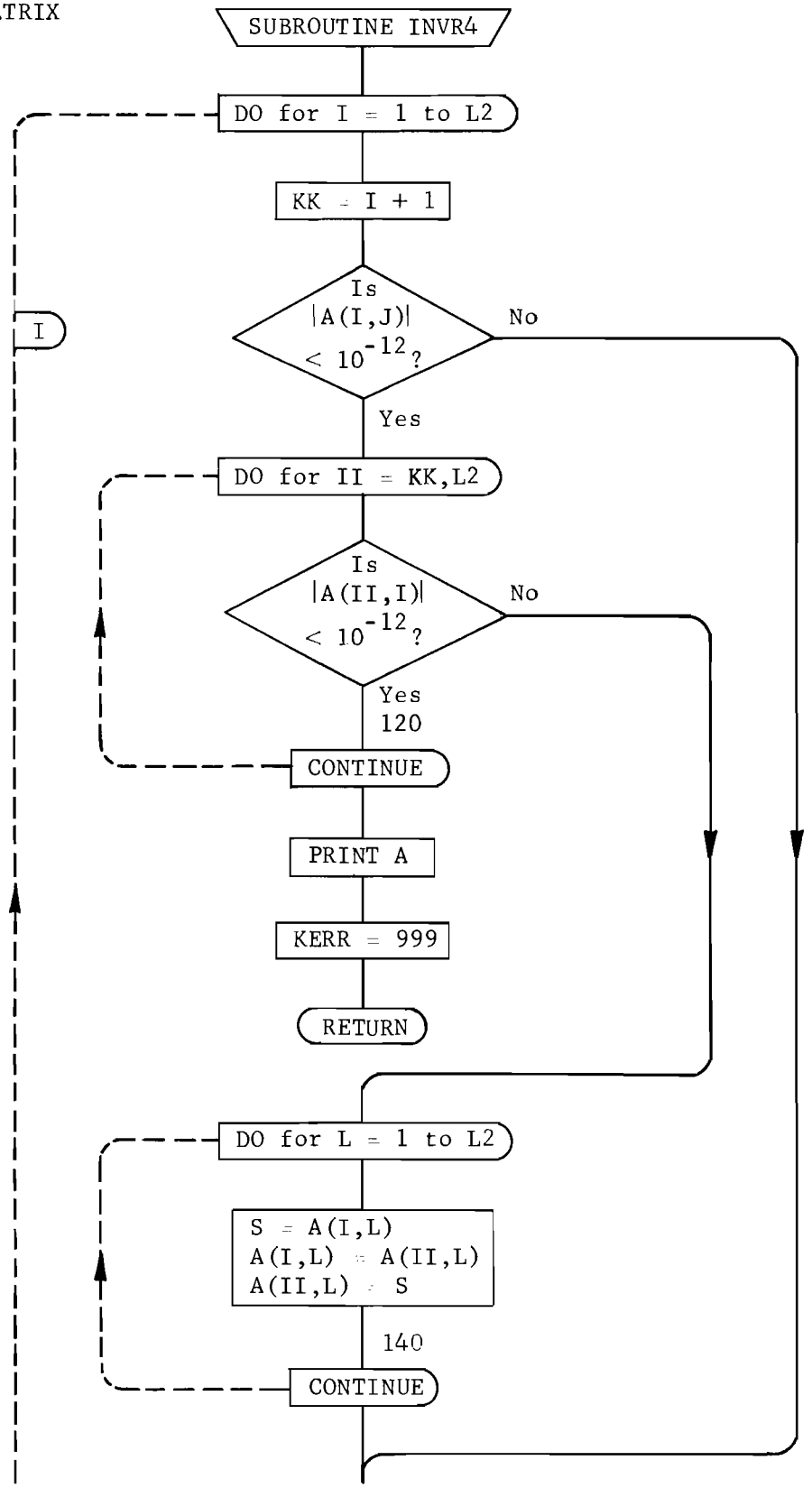
Calls: None
Called by: MATRIX



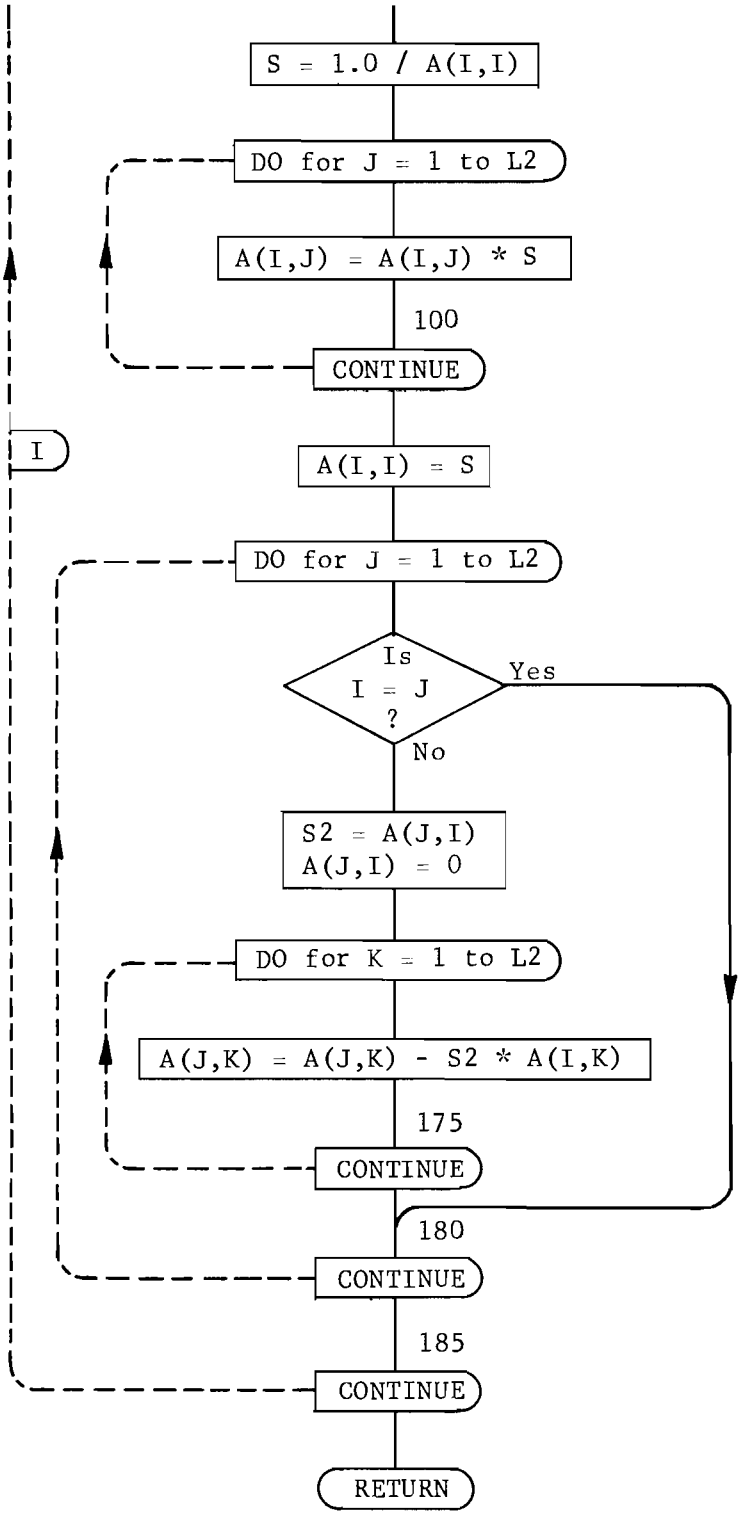
NOTE: For a listing of SUBROUTINE MATS2 see page 210.

FLOW DIAGRAM FOR SUBROUTINE INVR4

Calls: None
Called by: MATRIX

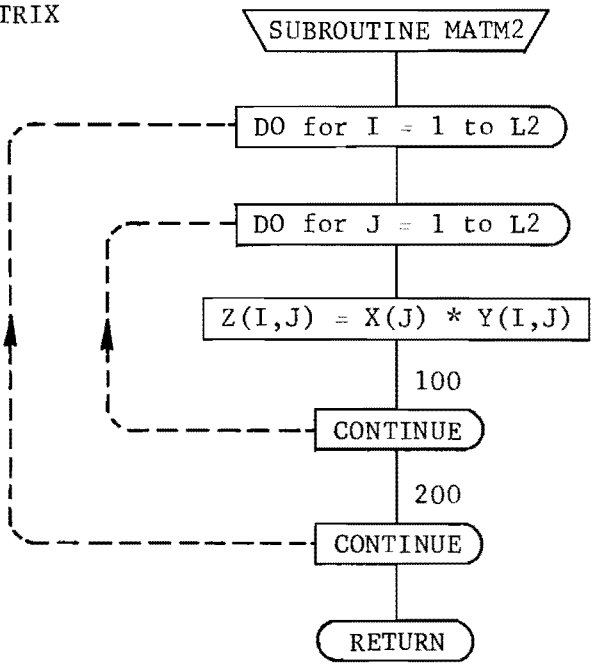


NOTE: For a listing of SUBROUTINE INVR4 see page 211.


S = 1.0 / A(I,I)
DO for J = 1 to L2
A(I,J) = A(I,J) * S
100
CONTINUE
I
A(I,I) = S
DO for J = 1 to L2
Is
I = J
?
Yes
No
S2 = A(J,I)
A(J,I) = 0
DO for K = 1 to L2
A(J,K) = A(J,K) - S2 * A(I,K)
175
CONTINUE
180
CONTINUE
185
CONTINUE
RETURN

FLOW DIAGRAM FOR SUBROUTINE MATM2

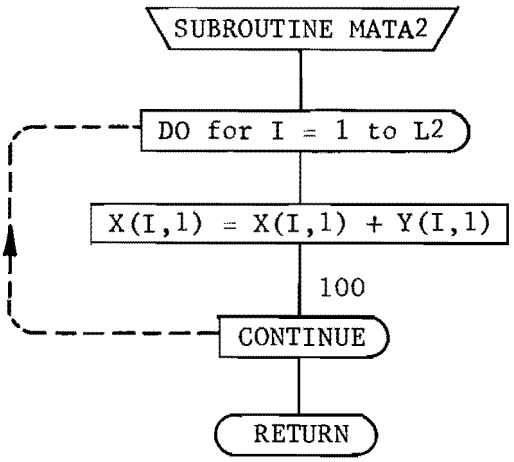
Calls: None
Called by: MATRIX



NOTE: For a listing of SUBROUTINE MATM2 see page 213.

FLOW DIAGRAM FOR SUBROUTINE MATA2

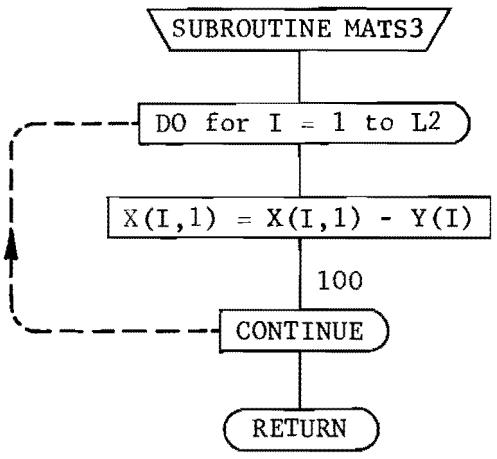
Calls: None
Called by: MATRIX



NOTE: For a listing of SUBROUTINE MATA2 see page 214.

FLOW DIAGRAM FOR SUBROUTINE MATS3

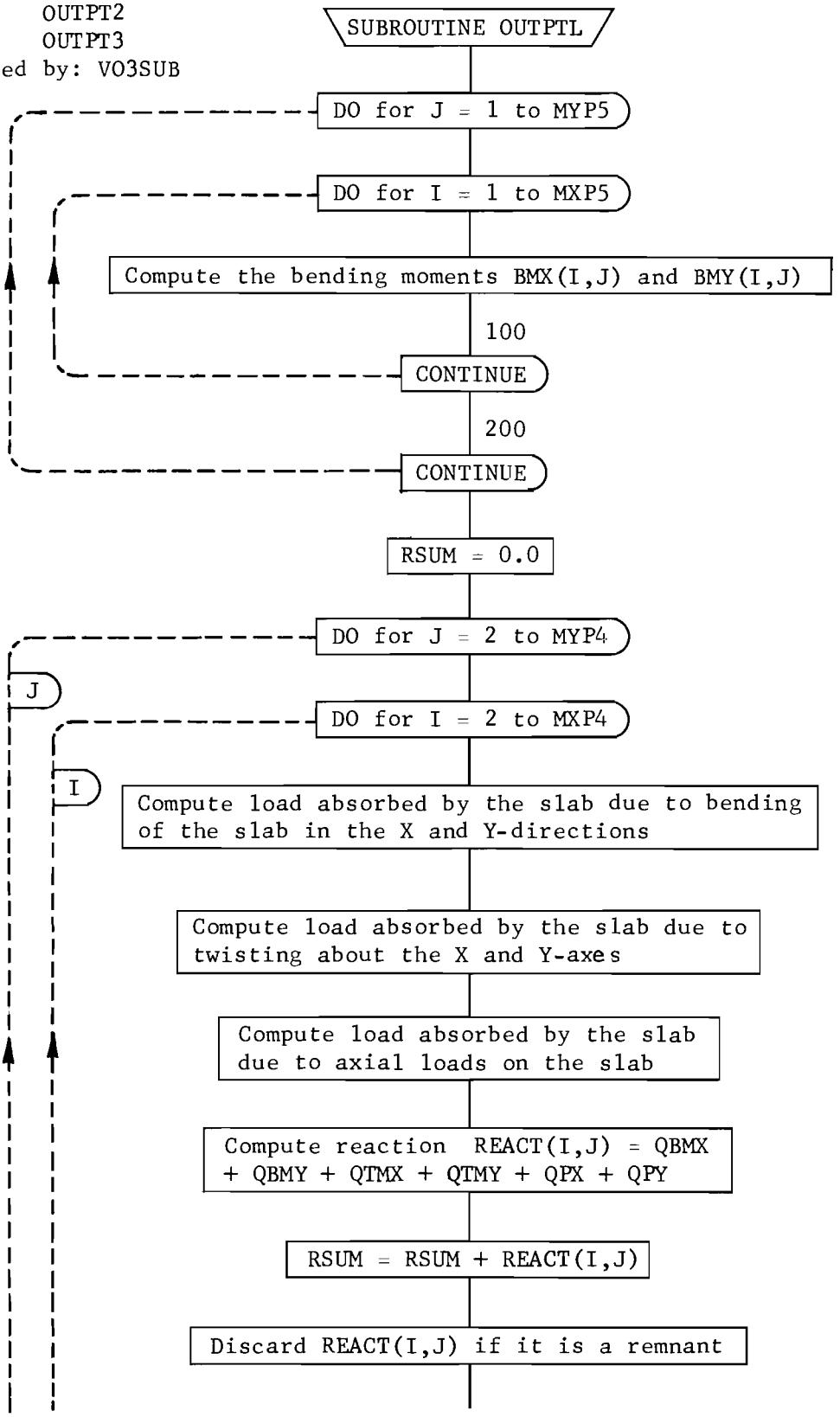
Calls: None
Called by: MATRIX



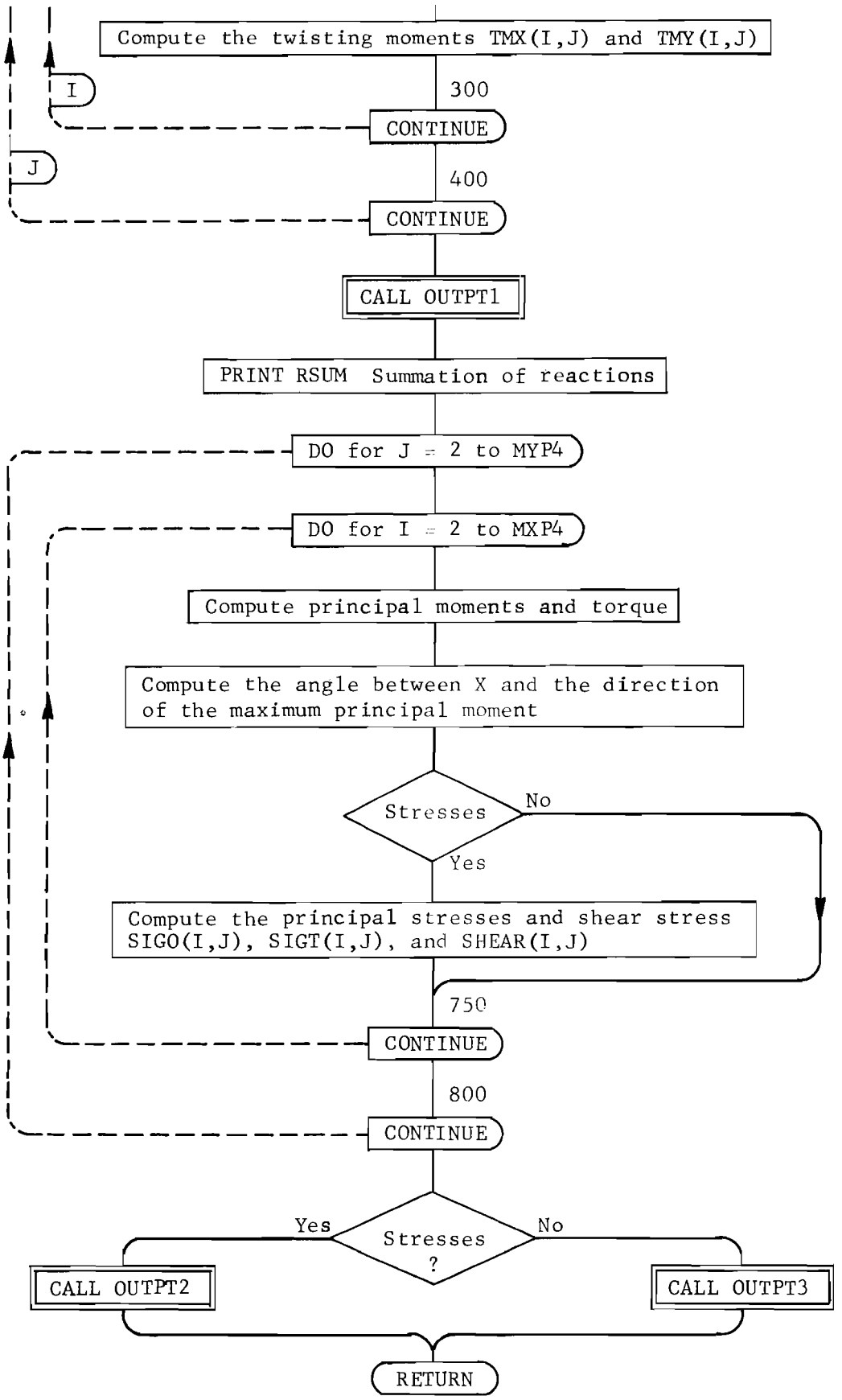
NOTE: For a listing of SUBROUTINE MATS3 see page 215.

FLOW DIAGRAM FOR SUBROUTINE OUTPTL

Calls: OUTPT1
OUTPT2
OUTPT3
Called by: VO3SUB

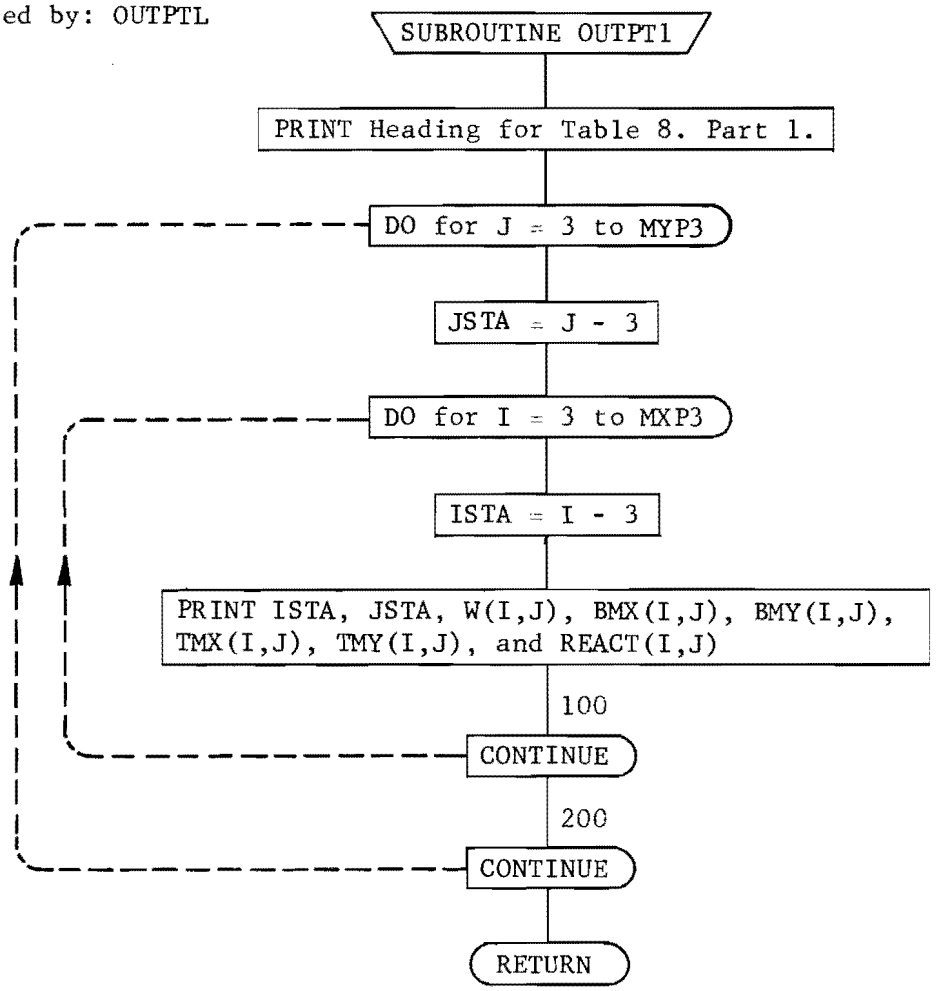


NOTE: For a listing of SUBROUTINE OUTPTL see page 216.


Compute the twisting moments TMX(I,J) and TMY(I,J)
I
300
CONTINUE
J
400
CONTINUE
CALL OUTPT1
PRINT RSUM Summation of reactions
DO for J = 2 to MYP4
DO for I = 2 to MXP4
Compute principal moments and torque
Compute the angle between X and the direction of the maximum principal moment
Stresses
No
Yes
Compute the principal stresses and shear stress SIGO(I,J), SIGT(I,J), and SHEAR(I,J)
750
CONTINUE
800
CONTINUE
Yes
Stresses ?
No
CALL OUTPT2
CALL OUTPT3
RETURN

FLOW DIAGRAM FOR SUBROUTINE OUTPT1

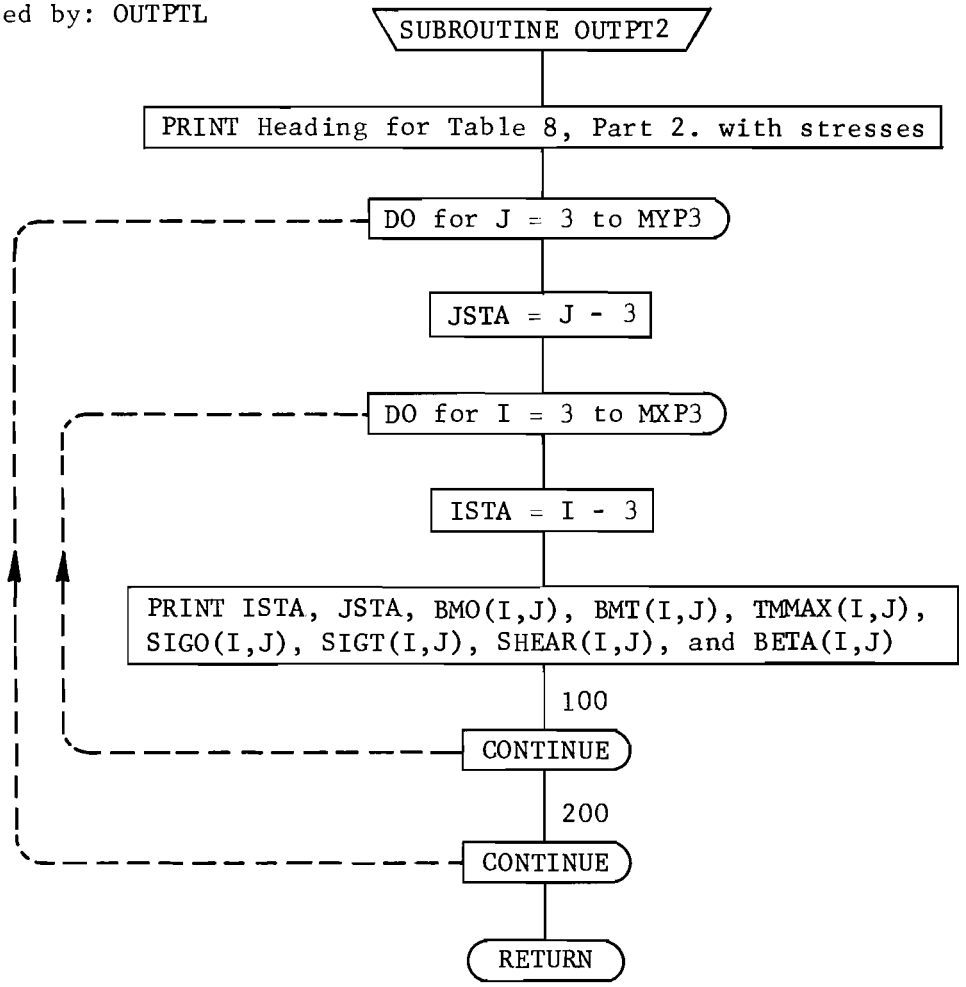
Calls: None
Called by: OUTPTL



NOTE: For a listing of SUBROUTINE OUTPT1 see page 221.

FLOW DIAGRAM FOR SUBROUTINE OUTPT2

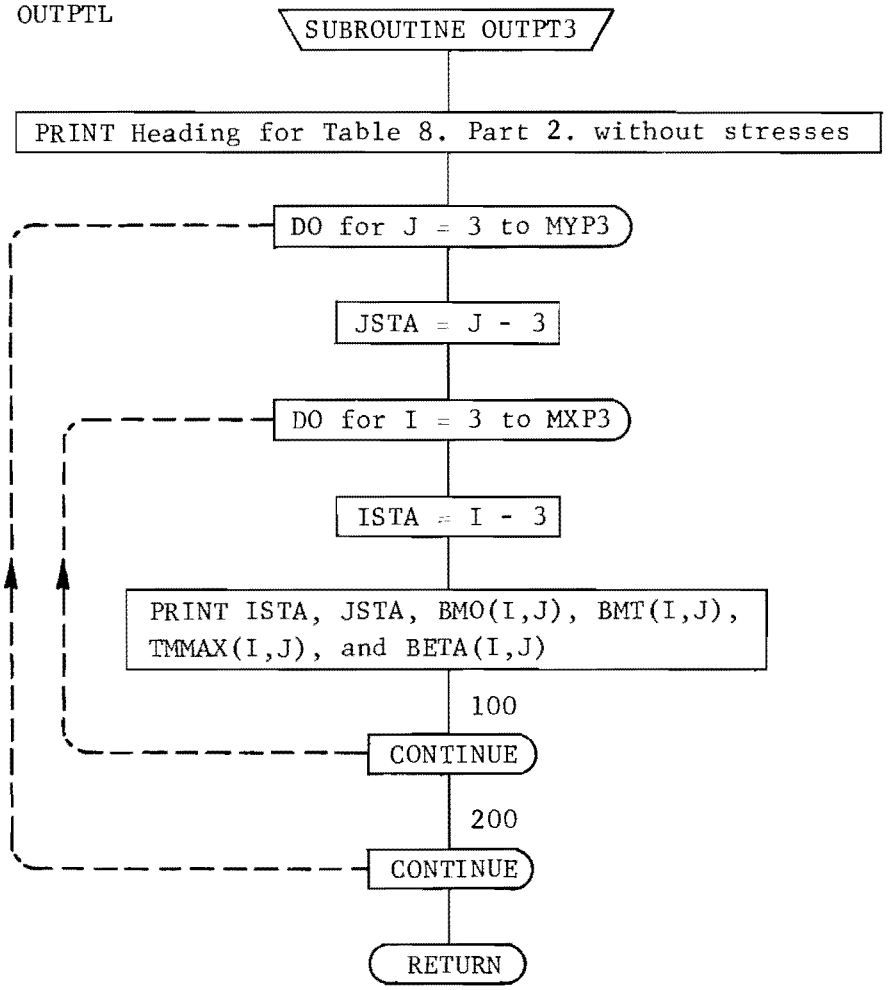
Calls: None
Called by: OUTPTL



NOTE: For a listing of SUBROUTINE OUTPT2 see page 222.

FLOW DIAGRAM FOR SUBROUTINE OUTPT3

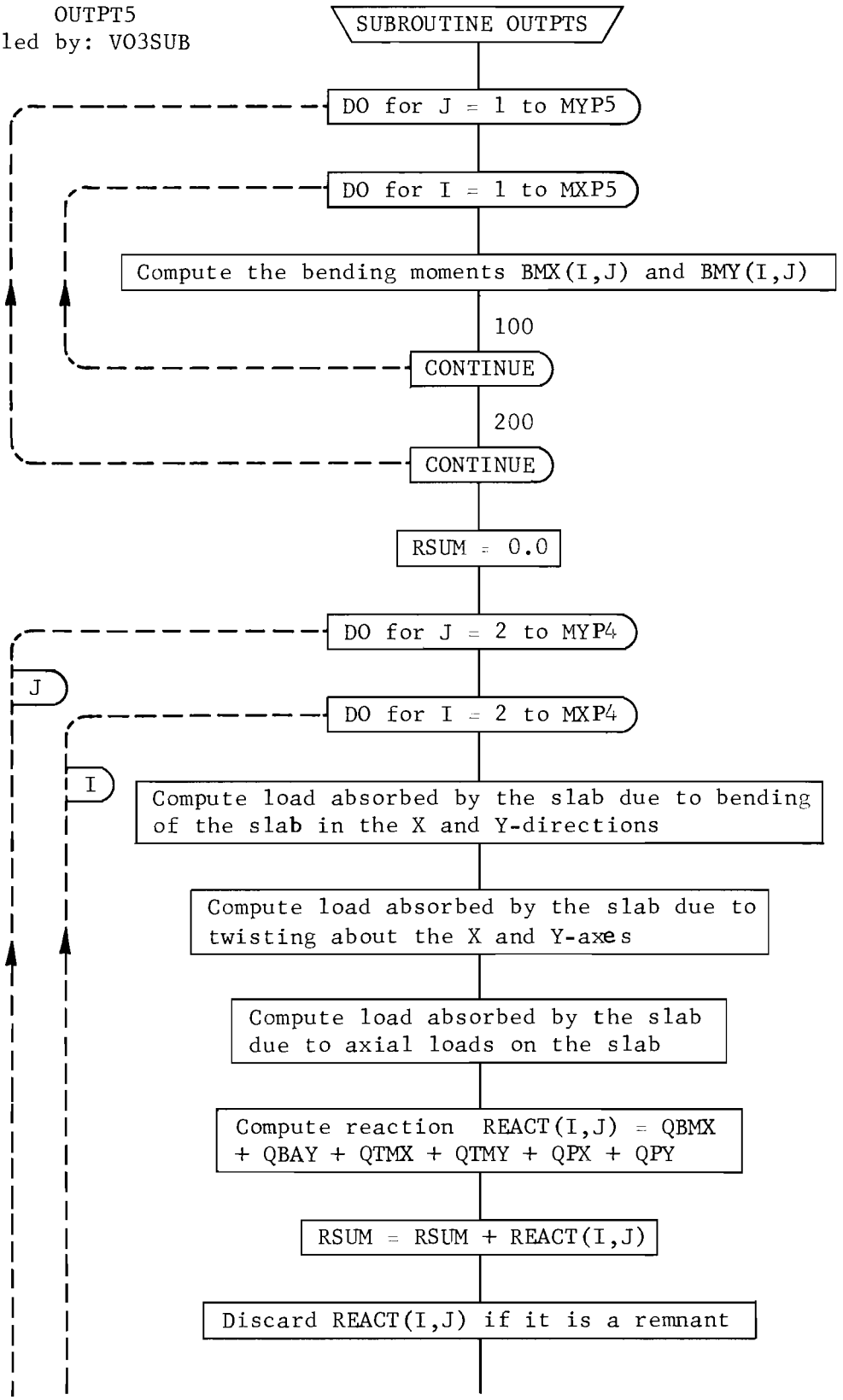
Calls: None
Called by: OUTPTL



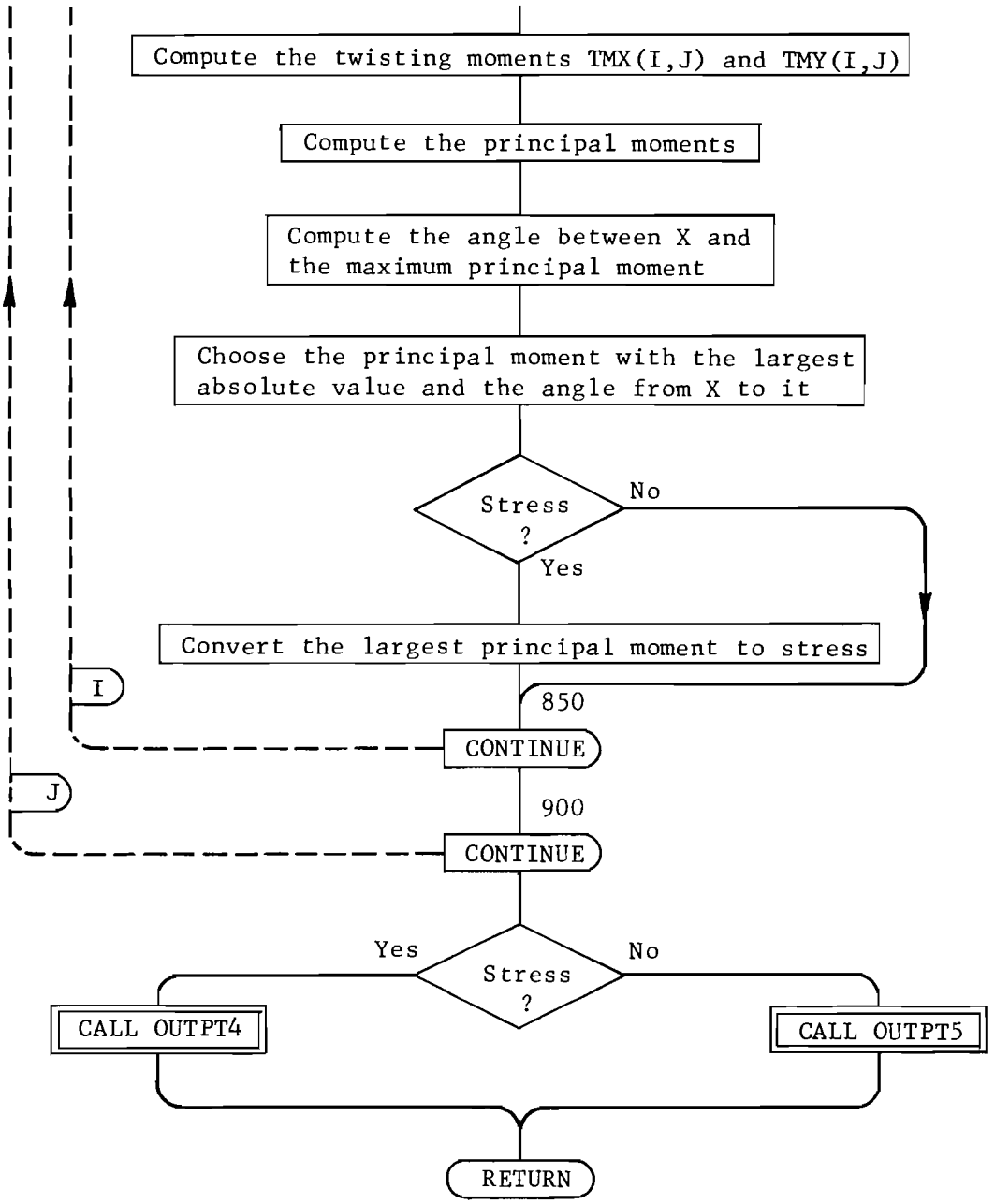
NOTE: For a listing of SUBROUTINE OUTPT3 see page 224.

FLOW DIAGRAM FOR SUBROUTINE OUTPTS

Calls: OUTPT4
OUTPT5
Called by: VO3SUB

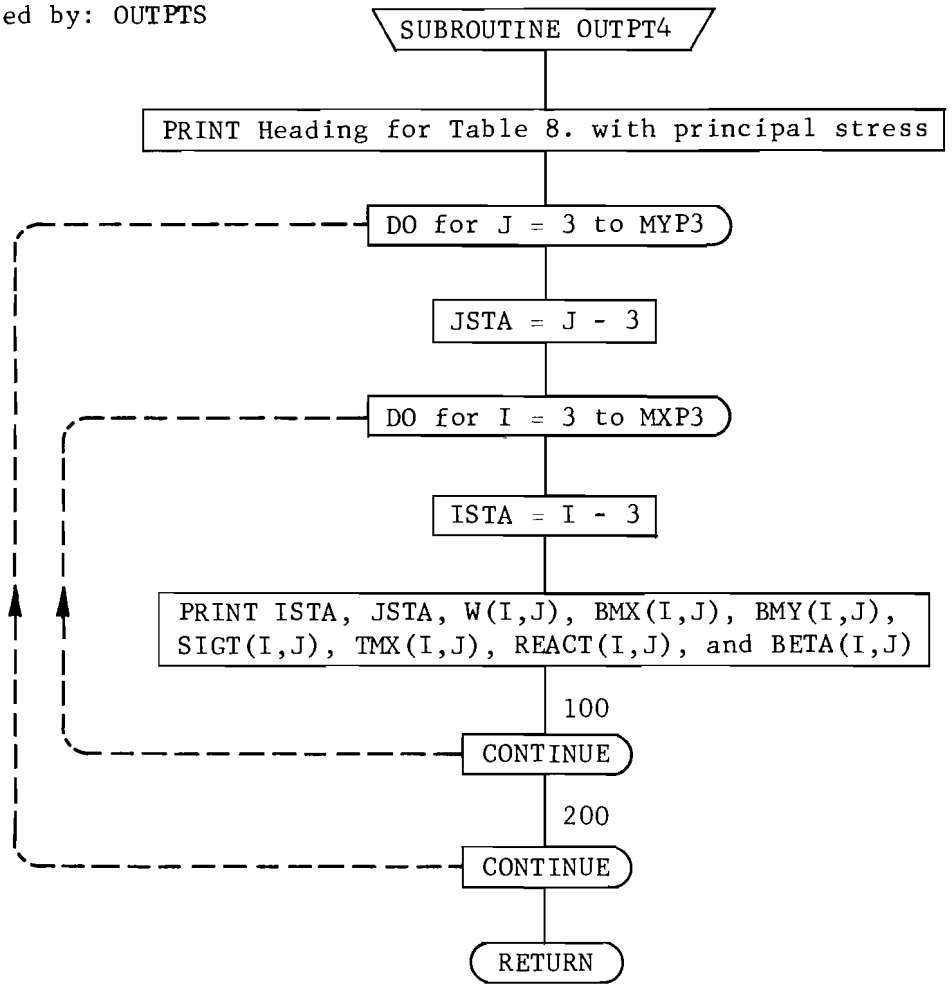


NOTE: For a listing of SUBROUTINE OUTPTS see page 225.


Compute the twisting moments TMX(I,J) and TMY(I,J)
Compute the principal moments
Compute the angle between X and the maximum principal moment
Choose the principal moment with the largest absolute value and the angle from X to it
Stress ?
No
Yes
Convert the largest principal moment to stress
I
850
CONTINUE
J
900
CONTINUE
Yes
Stress ?
No
CALL OUTPT4
CALL OUTPT5
RETURN

FLOW DIAGRAM FOR SUBROUTINE OUTPT4

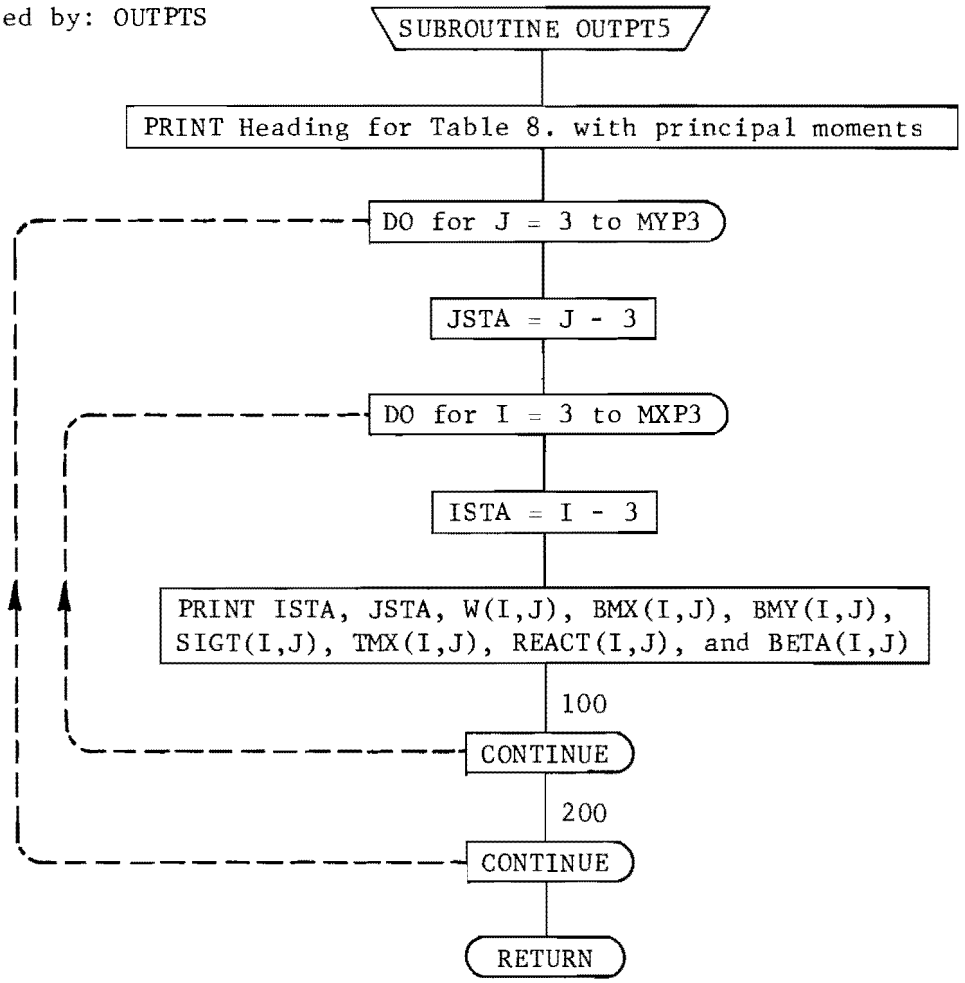
Calls: None
Called by: OUTPTS



NOTE: For a listing of SUBROUTINE OUTPT4 see page 230.

FLOW DIAGRAM FOR SUBROUTINE OUTPT5

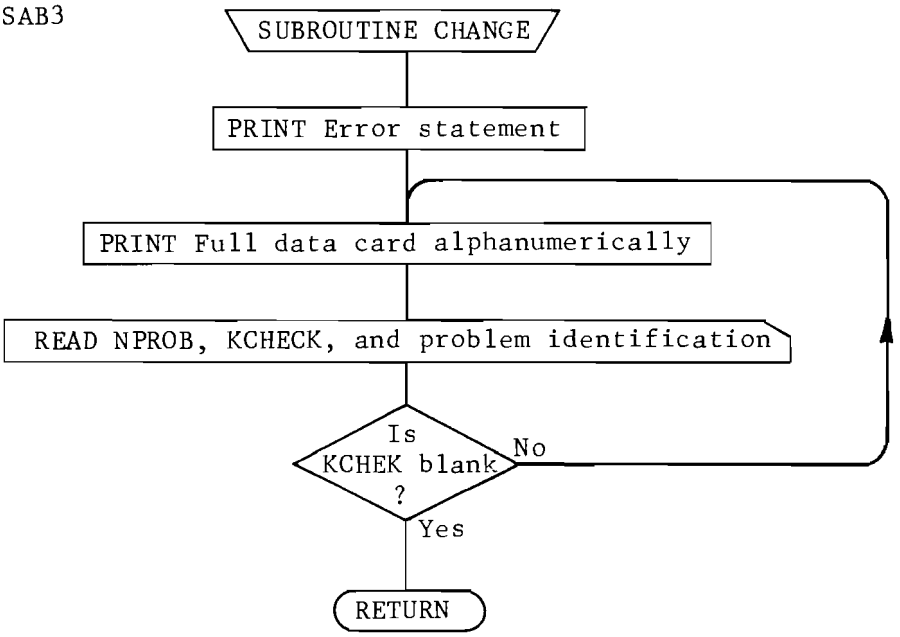
Calls: None
Called by: OUTPTS



NOTE: For a listing of SUBROUTINE OUTPT5 see page 232.

FLOW DIAGRAM FOR SUBROUTINE CHANGE

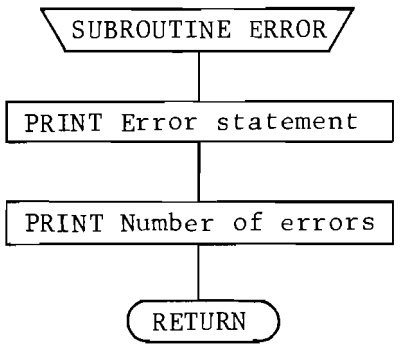
Calls: TICTOC
Called by: VISAB3



NOTE: For a listing of SUBROUTINE CHANGE see page 234.

FLOW DIAGRAM FOR SUBROUTINE ERROR

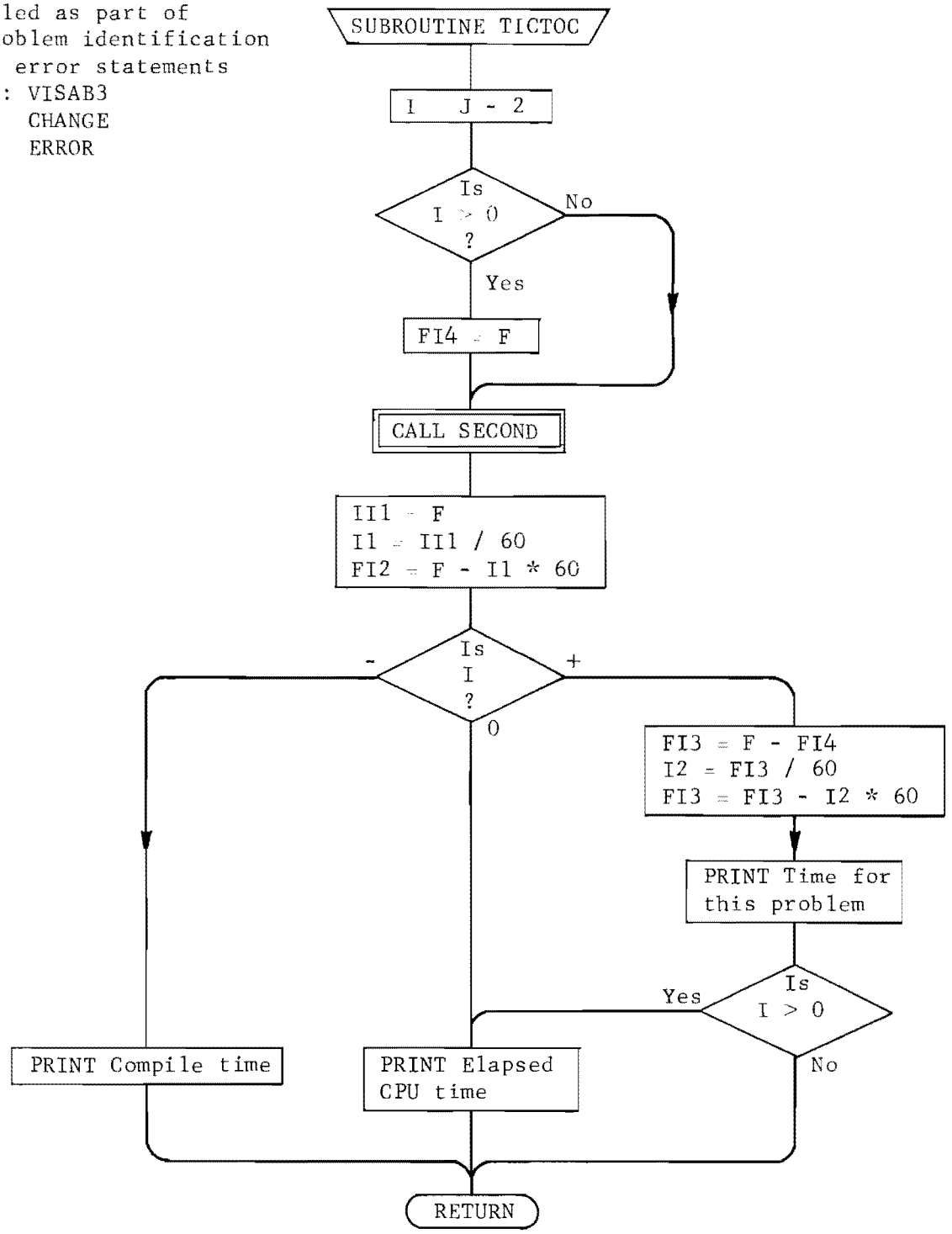
Calls: TICTOC
Called by: VISAB3



NOTE: For a listing of SUBROUTINE ERROR see page 235.

FLOW DIAGRAM FOR SUBROUTINE TICTOC

Calls: Second
Called as part of
problem identification
or error statements
by: VISAB3
CHANGE
ERROR



NOTE: For a listing of SUBROUTINE TICTOC see page 236.

APPENDIX 3

PROGRAM LISTING OF VISAB 3

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

```
VISAB3,17,60,57000,300.CF994005,PEARRE.
RUN(P)
REQUEST TAPE 1.     SCRATCH TAPE , 556 BPI.
REQUEST TAPE 2.     SCRATCH TAPE , 556 BPI.
REWIND,TAPE 1.
REWIND,TAPE 2.
LGO.
-
```

```
      PROGRAM VISAB 3 ( INPUT, OUTPUT, TAPE1, TAPE2 )
C
C-----THIS PROGRAM SOLVES PLATES AND SLABS USING A VARIBLE INCREMENT
C          LENGTH AND A DIRECT SOLUTION OF THE MATRIX EQUATIONS.
C
     1 FORMAT (    50H      PROGRAM VISAB 3 -    MASTER DECK     - PEARRE -, 02MR8
     1                            35H DRIVER REVISION DATE - 21 MAR 1968 )
C
C
C-----NOTATION FOR PROGRAM VISAB 3
C
C          AN1( ), AN2( )        ALPHANUMERIC REMARKS, INFORMATION ONLY
C          A1( ) --- A3( )       DIMENSION VARIABLES FOR USE IN SUBROUTINES
C          B1( ) --- B5( )       DIMENSION VARIABLES FOR USE IN SUBROUTINES
C          C1( ), C2( )          DIMENSION VARIABLES FOR USE IN SUBROUTINES
C          D1( )                 DIMENSION VARIABLE FOR USE IN SUBROUTINES
C          E1( ) --- E8( )       DIMENSION VARIABLES FOR USE IN SUBROUTINES
C          F                     ELAPSED CPU TIME
C          HUM, NUM              POSITIONING VARIABLES IN COMMON FIELD
C          ITEST                 BLANK FIELD FOR TESTING PROBLEM NUMBER
C          KCHEK                 PROBLEM NUMBER CARD IDENTIFIER
C          KERR, K1              ERROR FLAGS
C          L1, L2                CONSTANTS FOR CHECKING PROBLEM SIZE
C          ML                    MULTIPLE LOAD OPTION
C          MX                    NUMBER OF INCREMENTS IN X-DIRECTION
C          MXP3 --- MXP6         MX PLUS NUMBER AT END OF NAME
C          MY                    NUMBER OF INCREMENTS IN Y-DIRECTION
C          MYP2 --- MYP6         MY PLUS NUMBER AT END OF NAME
C          N                     COUNTER
C          NCT2 --- NCT7         NUMBER OF CARDS IN TABLE GIVEN BY NUMBER
C          NOUT                  OPTION ON OUTPUT FORM
C          NPRI                  OPTION ON INPUT PRINT OUT
C          NPROB                 PROBLEM NUMBER
C          PR                    POISSONS RATIO
C          P1( ) --- P23( )      DIMENSION VARIABLES FOR USE IN SUBROUTINES
C          Q1( )                 DIMENSION VARIABLE FOR USE IN SUBROUTINES
C          R1( )                 DIMENSION VARIABLE FOR USE IN SUBROUTINES
C          THK                   SLAB THICKNESS
C
C
C-----DIMENSION GUIDE FOR VISAB 3
C
C      FOR DIFFERENT SIZE PROBLEMS CHANGE THE DIMENSIONED ARRAYS
C          ACCORDING TO THE FOLLOWING EXAMPLE TYPES, WHERE N IS 1, 2,
C          3, ETC.   X SHOULD ALWAYS BE LESS THAN OR EQUAL TO Y.
C
C
C               AN(X+3)             N = 1, 2, 3
C               BN(X+3,1)           N = 1, 2, 3, 4, 5
C               CN(X+3,3)           N = 1, 2
C               DN(X+3,5)           N = 1
C               EN(X+3,X+3)         N = 1, 2, - - - ,  8
C               PN(X+5,Y+5)         N = 1, 2, - - - , 23
C               QN(X+6)             N = 1
C               RN(Y+6)             N = 1
```

```
C
C
C          ALSO, THE LN CONSTANTS MUST BE CHANGED AS FOLLOWS.
C
C
C               L1 = X + 5
C               L2 = Y + 5
C
C
C-----THE PROGRAM IS NOW DIMENSIONED FOR A 16 BY 16 REAL SLAB.
C
C
      DIMENSION     A1(19),    A2(19),         A3(19),         B1(19,1),          01
     1          B2(19,1),      B3(19,1),       B4(19,1),       B5(19,1),          02
     2          C1(19,3),      C2(19,3),       D1(19,5),       E1(19,19),         03
     3          E2(19,19),     E3(19,19),      E4(19,19),      E5(19,19),         04
     4          E6(19,19),     E7(19,19),      E8(19,19),      P1(21,21),         05
     5          P2(21,21),     P3(21,21),      P4(21,21),      P5(21,21),         06
     6          P6(21,21),     P7(21,21),      P8(21,21),      P9(21,21),         07
     7          P10(21,21),    P11(21,21),     P12(21,21),     P13(21,21),        08
     8          P14(21,21),    P15(21,21),     P16(21,21),     P17(21,21),        09
     9          P18(21,21),    P19(21,21),     P20(21,21),     P21(21,21),        10
     A          P22(21,21),    P23(21,21),     Q1(22),         R1(22)             11
C
      EQUIVALENCE    (A2,B4),          (A3,B5),          (E1,P2,P4),      07FE8
     1               (E2,P3,P5),       (E3,P13,P18),     (E4,P15),        07FE8
     2               (E5,P16,P19),     (E6,P17,P20),     (P1,P12),        07FE8
     3               (P14,P23)                                            07FE8
C
      COMMON / CON / L1, L2, ITEST, MXP6                                  23JA8
C
      COMMON    KERR, N, K1, NUM(13), HUM(36), NCT2, NCT3, NCT4, NCT5,    16DE7
     1          NCT6, NCT7, MX, MY, ML, NOUT, NPRI, PR, THK, MYP2, MXP3,  18JA8
     2          MYP3, MXP4, MYP4, MXP5, MYP5, MYP6, AN1(32), AN2(14),     18JA8
     3          NPROB, KCHEK, F, A1, B1, B2, B3, B4, B5, C1, C2, D1, E7,  18FE8
     4          E8, P1, P2, P3, P6, P7, P8, P9, P10, P11, P13, P14, P15,  18FE8
     5          P16, P17, P21, P22, Q1, R1                                18FE8
C
               L1  =  21                                                            12
               L2  =  21                                                            13
C
   11 FORMAT (    5H1     , 80X, 10H1-----TRIM )                           13NO7
   12 FORMAT (  16A5  )                                                    04MY3 ID
   13 FORMAT ( 5X, 16A5 )                                                  07NO7
   15 FORMAT (///10H     PROB , /5X, A5, 5X, 14A5 )                        19MR5 ID
   19 FORMAT (///45H     RETURN THIS PAGE TO TIME RECORD FILE        )     16JA8
   20 FORMAT ( 1115, 5X, 2E10.3 )                                          16JA8
   30 FORMAT ( //30H     TABLE 1.  CONTROL DATA    ,                /      15AP3
     1          / 30H          NUM CARDS TABLE 2    , 42X, I3,      /      20JL7
     2            30H          NUM CARDS TABLE 3    , 42X, I3,      /      20JL7
     3            30H          NUM CARDS TABLE 4    , 42X, I3,      /      09NO7
     4            30H          NUM CARDS TABLE 5    , 42X, I3,      /      09NO7
     5            30H          NUM CARDS TABLE 6    , 42X, I3,      /      09NO7
     6            30H          NUM CARDS TABLE 7    , 42X, I3,      /      16JA8
     7            30H          NUM INCREMENTS  MX   , 42X, I3,      /      24JA8
```

```
     8            30H           NUM INCREMENTS  MY   , 42X, I3,      /       16JA8
     9            30H           POISSONS RATIO       , 39X, F6.4,    /       16JA8
     A            30H           SLAB THICKNESS       , 35X, E10.3,   /       16JA8
     B          / 30H           MUL LOAD OPTION ( IF,                        16JA8
     C            30H + OR - 1 )                     , 12X, I3,      /       16JA8
     D          / 30H           OPTION ( IF = 1 ) FO,                        16JA8
     E            30HR SHORT FORM OF OUTPUT          , 12X, I3,      /       16JA8
     F          / 30H           OPTION ( IF = 1 ) TO                         16JA8
     G            30H PRINT INPUT VARIABLE ARRAYS  . 12X, I3,      /       16JA8
     H            30X,                               42X, I3,      /  )    18JA8
  40 FORMAT ( //40H     ********************************    , /             18JA8
     1            40H     *                                *    , /        18JA8
     2            40H     *     ERROR IN INPUT DATA        *    , /        18JA8
     3            40H     *                                *    , /        18JA8
     4            40H     *      MX SET TO  L1 - 5         *    , /        18JA8
     5            40H     *      MY SET TO  L2 - 5         *    , /        18JA8
     6            40H     *                                *    , /        18JA8
     7            40H     ********************************    , // )       24JA8
C
C
      CALL TIC TOC ( 1 )                                                    24JA8
C
C-----SET PROGRAM CONSTANTS
C
               ITEST = 5H                                                   19MR5 ID
C
C-----PROGRAM AND PROBLEM IDENTIFICATION
C
      READ  12, ( AN1(N), N = 1, 32 )                                       19MR5 ID
C
 1010 READ  12,  NPROB,  KCHEK,  ( AN2(N), N = 1, 14 )                     23JA8
C
C-----CHECK FOR PROBLEM NUMBER
C
          IF ( KCHEK - ITEST )          9975, 1015, 9975                    24JA8
C
 1015     IF ( NPROB - ITEST )          1020, 9990, 1020                    24JA8
C
C-----PRINT PROBLEM STATEMENT AND EXECUTE PROGRAM
C
 1020 PRINT 11                                                              26AG3 ID
      PRINT  1                                                              19MR5 ID
      PRINT 13, ( AN1(N), N = 1, 32 )                                       19MR5 ID
      PRINT 15, NPROB, ( AN2(N), N = 1, 14 )                                26AG3 ID
C
      REWIND 1                                                              05DE7
      REWIND 2                                                              05DE7
C
C-----INPUT TABLE 1
C
      READ  20, NCT2, NCT3, NCT4, NCT5, NCT6, NCT7, MX, MY, ML, NOUT,       16JA8
     1          NPRI, PR, THK                                               16JA8
      PRINT 30, NCT2, NCT3, NCT4, NCT5, NCT6, NCT7, MX, MY, PR, THK,        16JA8
     1          ML, NOUT, NPRI                                              16JA8
C
```

```
C-----RESET ERROR FLAGS IF ML = - 1
C
      IF ( ML )                        1100, 1040, 1040                31JA8
C
 1040           KERR  =  0                                              31JA8
                K1  =  0                                                16DE7
C
C-----COMPUTE FOR CONVENIENCE
C
 1050           MYP2  =  MY + 2                                         07NO7
                MXP3  =  MX + 3                                         07NO7
                MYP3  =  MY + 3                                         07NO7
                MXP4  =  MX + 4                                         07NO7
                MYP4  =  MY + 4                                         07NO7
                MXP5  =  MX + 5                                         07NO7
                MYP5  =  MY + 5                                         07NO7
                MXP6  =  MX + 6                                         09NO7
                MYP6  =  MY + 6                                         09NO7
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
 1100      IF ( MXP5 .GT. L1 )            KERR  =  KERR + 1            15DE7
           IF ( MYP5 .GT. L2 )            KERR  =  KERR + 1            15DE7
           IF ( MXP5 .GT. MYP5 )          KERR  =  KERR + 1            15DE7
C
           IF ( K1 .EQ. 1 )               GO TO 1200                   16DE7
           IF ( KERR .GT. 0 )             GO TO 9985                   15DE7
C
C-----CALL MAIN SUBROUTINE OF PROGRAM
C
 1200 CALL VO3SUB ( A1,  A2,  A3,  B1,  B2,  B3,  B4,  B5,  C1,  C2,    16JA7
     1              D1,  E1,  E2,  E3,  E4,  E5,  E6,  E7,  E8,  P1,    16JA8
     2              P2,  P3,  P4,  P5,  P6,  P7,  P8,  P9,  P10, P11,   16JA8
     3              P12, P13, P14, P15, P16, P17, P18, P19, P20, P21,   16JA8
     4              P22, P23, Q1,  R1,  MXP3,     MXP5,     MYP5,       22JA8
     5              MXP6,     MYP6     )                                22JA8
C
C-----IF INPUT DATA IS INCORRECT TERMINATE PROBLEM
C
           IF ( KERR .GT. 0 )             GO TO 9980                   15DE7
C
C-----PRINT TIME AND RETURN TO START FOR NEXT PROBLEM
C
      CALL TIC TOC (3)                                                 25SE6
C
          GO TO 1010                                                   26AG3 ID
C
C-----INCORRECT CARD COUNT, EJECT PROBLEM
C
 9975 PRINT 11                                                         31JA8
      PRINT 1                                                          31JA8
      PRINT 13, ( AN1(N), N = 1, 32 )                                  19MR5 ID
C
      CALL CHANGE                                                      31JA8
C
```

```
C-----RETURN TO START FOR NEXT PROBLEM
C
          GO TO 1015                                                    23JA8
C
C-----PRINT NUMBER OF DATA ERRORS
C
 9980 CALL ERROR                                                        23JA8
C
C-----RETURN TO START FOR NEXT PROBLEM
C
          GO TO 1010                                                    15DE7
C
C-----RESET MX AND MY DUE TO DATA ERROR
C
 9985           MX   =  L1 - 5                                         15DE7
                MY   =  L2 - 5                                         15DE7
C
      PRINT 40                                                          15DE7
C
                K1   =  1                                              13DE7
C
C-----RETURN TO COMPUTE FOR CONVENIENCE AND CHECK THE REMAINING INPUT
C          DATA FOR THIS PROBLEM
C
          GO TO 1050                                                    15DE7
C
C-----PRINT END OF RUN IDENTIFICATION
C
 9999 PRINT 11                                                          18JA8 ID
      PRINT 1                                                           21JL7
      PRINT 13, ( AN1(N), N = 1, 32 )                                   19MR5 ID
C
      CALL TIC TOC (2)                                                  26SE6
C
      PRINT 19                                                          26AG3 ID
C
C-----END COMPUTATION
C
      END
```

```
      SUBROUTINE V03SUB ( AA1,  AA5,  AA6,  AA,  A1,  A2,  W1,  W2,     16JA8
     1                    AA2,  AA4,  AA3,  BB,  BB1,  BB2,  CC,  CC1,  16JA8
     2                    CC2,  D,  E,  BETA,  BMO,  BMT,  BMX,  BMY,   22JA8
     3                    CX,  CY,  DX,  DY,  PX,  PY,  Q,  REACT,  S,  22JA8
     4                    SHEAR,  SIGO,  SIGT,  TMMAX,  TMX,  TMY,  TX, 22JA8
     5                    TY,  W,  HX,  HY,  L1,  L2,  L3,  L4,  L5 )   03FE8
C
C-----MAIN SECTION OF PROGRAM.
C
    1 FORMAT (   50H     PROGRAM VISAB 3 -   MASTER DECK   - PEARRE - , 02MR8
     1                          35HPROGRAM REVISION DATE - 21 MAR 1968 )
C
C
C-----NOTATION FOR SUBROUTINE V03SUB
C
C          AA( )                 RECURSION COEFFICIENT
C          AA1( ) --- AA6( )     SUBMATRICES OF STIFFNESS MATRIX
C          A1( ), A2( )          TEMPORARY AA( ) TERMS
C          AN1( ), AN2( )        ALPHANUMERIC REMARKS, INFORMATION ONLY
C          BB( )                 RECURSION COEFFICIENT
C          BB1( ), BB2( )        TEMPORARY BB( ) TERMS
C          BETA ( )              ANGLE FROM X TO MAXIMUM PRINCIPAL MOMENT
C          BMO( )                MAXIMIM PRINCIPAL MOMENT
C          BMT( )                MINIMUM PRINCIPAL MOMENT
C          BMX( )                X-BENDING MOMENT
C          BMY( )                Y-BENDING MOMENT
C          CC( )                 RECURSION COEFFICIENT
C          CC1( ), CC2( )        TEMPORARY CC( ) TERMS
C          CX( )                 TWISTING STIFFNESS X-DIRECTION
C          CY( )                 TWISTING STIFFNESS Y-DIRECTION
C          D( )                  RECURSION COEFFICIENT
C          DX( )                 BENDING STIFFNESS X-DIRECTION
C          DY( )                 BENDING STIFFNESS Y-DIRECTION
C          E( )                  RECURSION COEFFICIENT
C          HT1, HUM, NT1, NUM    POSITIONING VARIABLES IN COMMON FIELD
C          HX( )                 INCREMENT LENGTH X-BEAM
C          HY( )                 INCREMENT LENGTH Y-BEAM
C          I, II, J, J1, J2,     COUNTERS AND INDICIES IN DO LOOPS
C          K, LL, N
C          KERR                  ERROR FLAG
C          L1 --- L5             CONSTANTS FOR TRANSFERRING DIMENSION STATEMENT
C          ML                    MULTIPLE LOAD OPTION
C          MXP3 --- MXP6         MX PLUS NUMBER AT END OF NAME
C          MYP2 --- MYP6         MY PLUS NUMBER AT END OF NAME
C          NOUT                  OPTION ON OUTPUT FORM
C          NPRI                  OPTION ON INPUT PRINT OUT
C          NPROB                 PROBLEM NUMBER
C          PX( )                 AXIAL LOAD X-DIRECTION
C          PY( )                 AXIAL LOAD Y-DIRECTION
C          Q( )                  LATERAL LOAD
C          REACT( )              SLAB REACTION
C          S( )                  SUPPORT SPRING
C          SHEAR( )              SHEAR STRESS
C          SIGO( )               MAXIMUM PRINCIPAL STRESS
C          SIGT( )               MINIMUM PRINCIPAL STRESS
```

```
C          TMMAX( )              MAXIMUM TORQUE
C          TMX( )                X-TWISTING MOMENT
C          TMY( )                Y-TWISTING MOMENT
C          TX( )                 EXTERNAL COUPLE X-DIRECTION
C          TY( )                 EXTERNAL COUPLE Y-DIRECTION
C          W( )                  DEFLECTION OF SLAB
C          W1( ), W2( )          TEMPORARY VALUES OF W( )
C
C
      DIMENSION     AA(L1,1), AA1(L1),        AA2(L1,3),      AA3(L1,5),  16JA8
     1          AA4(L1,3),    AA5(L1),        AA6(L1),        A1(L1,1),   16JA8
     2          A2(L1,1),     BB(L1,L1),      BB1(L1,L1),     BB2(L1,L1), 24JA8
     3          BETA(L2,L3),  BMO(L2,L3),     BMT(L2,L3),     BMX(L2,L3), 22JA8
     4          BMY(L2,L3),   CC(L1,L1),      CC1(L1,L1),     CC2(L1,L1), 22JA8
     5          CX(L2,L3),    CY(L2,L3),      D(L1,L1),       DX(L2,L3),  22JA8
     6          DY(L2,L3),    E(L1,L1),       HX(L4),         HY(L5),     22JA8
     7          PX(L2,L3),    PY(L2,L3),      Q(L2,L3),       REACT(L2,L3)22JA8
     8          S(L2,L3),     SHEAR(L2,L3),   SIGO(L2,L3),    SIGT(L2,L3),22JA8
     9          TMMAX(L2,L3), TMX(L2,L3),     TMY(L2,L3),     TX(L2,L3),  23JA8
     A          TY(L2,L3),    W(L2,L3),       W1(L1,1),       W2(L1,1)    22JA8
C
C-----THE FOLLOWING VARIBLES ARE EQUIVALENCE IN THE MAIN PROGRAM.
C
C      EQUIVALENCE   (AA5,W1),          (AA6,W2),          (BB,BMO,BMX),
C     1              (BB1,BMT,BMY),     (BB2,REACT,TMMAX),   (CC,SHEAR),
C     2              (CC1,SIGO,TMX),    (CC2,SIGT,TMY),    (BETA,Q),
C     3              (S,W)
C
      COMMON    KERR, I, II, J, J1, J2, K, LL, N, NUM(7), HUM(36),        18JA8
     1          NT1(8), ML, NOUT, NPRI, HT1(2), MYP2, MXP3, MYP3, MXP4,  18JA8
     2          MYP4, MXP5, MYP5, MYP6, AN1(32), AN2(14), NPROB           18JA8
C
    6 FORMAT ( )                                                          04MY3
   10 FORMAT (    5H      , 80X, 10HI-----TRIM )                         03FE4 ID
   11 FORMAT (    5H1     , 80X, 10HI-----TRIM )                         03FE4 ID
   13 FORMAT (    5X, 16A5 )                                             26AG3 ID
   16 FORMAT (///17H     PROB (CONTD), /5X, A5, 5X, 14A5 )               19MR5 ID
C
C
C-----INPUT SECTION
C
C-----CALL INPUT SUBROUTINE BASED ON MULTIPLE LOAD OPTION
C
          IF ( ML )                     100, 200, 200                   24JA8
C
  100 CALL INPT1 ( AA, A1, HX, HY, Q, TX, TY, W, L1, L2, L3, L4, L5 )   03FE8
C
          GO TO 300                                                     18JA8
C
  200 CALL INPT2 ( AA, A1, BB, BB1, CC, CC1, CX, CY, DX, DY, HX, HY,    03FE8
     1             PX, PY, Q, S, TX, TY, L1, L2, L3, L4, L5 )           03FE8
C
  300     IF ( KERR .GT. 0 )            GO TO 9980                      18JA8
C
C-----PRINT OPTIONAL TABLES TO CHECK INPUT DATA
```

```
C
C
      IF ( NPRI .EQ. 1 )            CALL STIFF ( DX, DY, Q, S, CX,  18JA8
     1          CY, HX, HY, TX, TY, PX, PY, L2, L3, L4, L5 )        16JA8
C
C
C-----BEGIN MAIN SOLUTION
C
C-----FORM SUB-MATRICES
C
      DO  600  J = 2, MYP4                                          14NO7
C
      CALL QMAT ( AA1, AA6, HX, HY, DY, Q, TX, TY, L1, L2, L3, L4, L5 ) 16JA8
C
      IF ( ML )                     400, 350, 350                   24JA8
C
  350 CALL SUBMAT ( AA2, AA3, AA4, AA5, HX, HY, DX, DY, S, CX, CY, PX,  18JA8
     1              PY, L1, L2, L3, L4, L5 )                        16JA8
C
C-----SOLVE MATRICES
C
C-----SET TEMPORARY VARIBLES EQUAL TO RECURSION COEFFICIENTS FROM
C     PREVIOUS ROWS
C
  400 DO  500  I = 1, MXP3                                          18JA8
C
          A2(I,1)  =  A1(I,1)                                       16JA8
          A1(I,1)  =  AA(I,1)                                       16JA8
C
      IF ( ML )                     500, 425, 425                   24JA8
C
  425 DO  450  K = 1, MXP3                                          18JA8
C
          BB2(I,K)  =  BB1(I,K)                                     10JA8
          BB1(I,K)  =  BB(I,K)                                      10JA8
          CC2(I,K)  =  CC1(I,K)                                     10JA8
          CC1(I,K)  =  CC(I,K)                                      10JA8
C
  450 CONTINUE                                                      18JA8
C
  500 CONTINUE                                                      18JA8
C
C-----CALL MATRIX TO SOLVE FOR THE RECURSION COEFFICIENTS FOR THIS ROW
C
      CALL MATRIX ( L1, AA, AA1, AA2, AA3, AA4, AA5, AA6, A1, A2, BB,   16JA8
     1              BB1, BB2, CC, CC1, CC2, D, E )                  16JA8
C
C-----CHECK FOR NO INVERSE
C
      IF ( KERR .GT. 0 )            GO TO 9980                      22JA8
C
C-----STORE RECURSION COEFFICIENT AA TO TAPE 2
C
      WRITE ( 2 ) ( AA(I,1), I = 1, MXP3 )                          18A18
C
```

```
C-----TAPE CONTROL CHECK
C
          IF ( ML )                          525, 575, 550                    24JA8
C
C-----SPACE TAPE 1
C
  525 READ  ( 1 )                                                            05DE7
C
          GO TO 600                                                           05DE7
C
C-----STORE RECURSION COEFFICIENTS D AND E ON TAPE 1
C
  550 WRITE ( 1 ) ( (  D(I,K),  E(I,K), I = 1, MXP3 ), K = 1, MXP3 )         18JA8
C
          GO TO 580                                                           21JA8
C
C-----STORE NPROB ON TAPE 1 TO CREATE A SPACER FILE WHEN ML = 0
C
  575 WRITE ( 1 ) ( NPROB )                                                   18JA8
C
C-----STORE RECURSION COEFFICIENTS BB AND CC ON TAPE 1
C
  580 WRITE ( 1 ) ( ( BB(I,K), CC(I,K), I = 1, MXP3 ), K = 1, MXP3 )         18JA8
C
  600     CONTINUE                                                            04MY7
C
C-----COMPUTE DEFLECTIONS
C
          DO  700  LL = 2, MYP4                                               18JA8
C
               J  =  MYP6 - LL                                                29NO7
               J1  =  J + 1                                                   30NO7
               J2  =  J + 2                                                   30NO7
C
C-----POSITION TAPES
C
      BACKSPACE 1                                                             05DE7
      BACKSPACE 2                                                             21JA8
C
C-----SET VECTORS W1 AND W2
C
          DO  625  I = 1, MXP3                                                21JL7
C
               II  =  I + 1                                                   29NO7
C
               W1(I,1)  =  W(II,J1)                                           30NO7
               W2(I,1)  =  W(II,J2)                                           29NO7
C
  625     CONTINUE                                                            22JE7
C
C-----READ RECURSION COEFFICIENT AA FROM TAPE 2
C
      READ  ( 2 ) ( AA(I,1), I = 1, MXP3 )                                   18JA8
C
C-----READ RECURSION COEFFICIENTS BB AND CC FROM TAPE 1
```

```
C
      READ  ( 1 ) ( ( BB(I,K), CC(I,K), I = 1, MXP3 ), K = 1, MXP3 )     05DE7
C
C-----POSITION TAPES
C
      BACKSPACE 1                                                        05DE7
      BACKSPACE 1                                                        22JA8
      BACKSPACE 2                                                        21JA8
C
C-----MULTIPLY THE BB MATRIX BY THE W1 VECTOR
C
      CALL MATMPY ( L1, 1, BB, W1, A1 )                                  11DE7
C
C-----MULTIPLY THE CC MATRIX BY THE W2 VECTOR
C
      CALL MATMPY ( L1, 1, CC, W2, A2 )                                  11DE7
C
C-----COMPUTE DEFLECTIONS FOR THIS ROW
C
          DO  650  I = 1, MXP3                                           21JL7
C
               W(I+1,J)  =  AA(I,1) + A1(I,1) + A2(I,1)                  30NO7
C
  650     CONTINUE                                                       04MY7
C
  700     CONTINUE                                                       18JA8
C
C-----SET DEFLECTIONS AT THE ENDS OF THE IMAGINARY BEAMS
C
               W(2,2)  =  2.0 * W(2,3) - W(2,4)                          29NO7
               W(MXP4,2)  =  2.0 * W(MXP4,3) - W(MXP4,4)                 29NO7
               W(2,MYP4)  =  2.0 * W(2,MYP3) - W(2,MYP2)                 29NO7
               W(MXP4,MYP4)  =  2.0 * W(MXP4,MYP3) - W(MXP4,MYP2)        29NO7
C
C
C-----OUTPUT SECTION
C
      PRINT 11                                                           15SE6
      PRINT  1                                                           21JL7
      PRINT 13, ( AN1(N), N = 1, 32 )                                    19MR5 ID
      PRINT 16, NPROB, ( AN2(N), N = 1, 14 )                             28AG3 ID
C
C-----CHOOSE OUTPUT FORM
C
          IF ( NOUT .EQ. 1 )            GO TO 800                        18JA8
C
C-----COMPUTE AND PRINT LONG FORM OF RESULTS
C
      CALL OUTPTL ( BETA, BMO, BMT, BMX, BMY, CX, CY, DX, DY, HX, HY,    22JA8
     1              PX, PY, REACT, SHEAR, SIGO, SIGT, TMMAX, TMX, TMY,   22JA8
     2              W, L2, L3, L4, L5 )                                  22JA8
C
          GO TO 900                                                      18JA8
C
C-----COMPUTE AND PRINT SHORT FORM OF RESULTS
```

```
C
  800 CALL OUTPTS ( BETA, BMX, BMY, CX, CY, DX, DY, HX, HY, PX, PY,      24JA8
     1              REACT, SIGT, TMX, W, L2, L3, L4, L5 )               24JA8
C
C
C-----RETURN TO MAIN PROGRAM FOR NEXT PROBLEM
C
  900           KERR  =  0                                              18JA8
C
      RETURN                                                            17DE7
C
C-----ERROR IN INPUT DATA, RETURN TO MAIN PROGRAM
C
 9980 RETURN                                                            09NO7
C
      END                                                               09NO7
```

```
      SUBROUTINE INPT1 ( AA, A1, HX, HY, Q, TX, TY, W, L1, L2, L3, L4,  03FE8
     1                   L5 )                                           03FE8
C
C-----THIS SUBROUTINE CALLS THE INPUT ROUTINES FOR OFF-SPRING PROBLEMS.
C
C
C-----NOTATION FOR SUBROUTINE INPT1
C
C          AA( )               RECURSION COEFFICIENT
C          A1( )               TEMPORARY AA( ) TERM
C          HX( )               INCREMENT LENGTH X-BEAM
C          HY( )               INCREMENT LENGTH Y-BEAM
C          NT1, NUM
C          L1 --- L5           CONSTANTS FOR TRANSFERRING DIMENSION STATEMENT
C          Q( )                LATERAL LOAD
C          TX( )               EXTERNAL COUPLE X-DIRECTION
C          TY( )               EXTERNAL COUPLE Y-DIRECTION
C          W( )                DEFLECTION OF SLAB
C
C
      DIMENSION     AA(L1,1),       A1(L1,1),       HX(L4),             03JA8
     1              HY(L5),         Q(L2,L3),       TX(L2,L3),          03FE8
     2              TY(L2,L3),      W(L2,L3)                            03FE8
C
C
C-----ZERO LOAD DATA FROM PREVIOUS PROBLEM
C
      CALL ZERO2 ( AA, A1, Q, TX, TY, W, L1, L2, L3 )                   03FE8
C
C-----PRINT MULTIPLE LOAD STATEMENT AND TABLES 2, 3, AND 4
C
      CALL MULIN                                                        18JA8
C
C-----INPUT TABLE 5
C
      CALL QIN ( Q, HX, HY, L2, L3, L4, L5 )                           18JA8
C
C-----INPUT TABLE 6
C
      CALL TXIN ( TX, HY, L2, L3, L5 )                                  18JA8
C
C-----INPUT TABLE 7
C
      CALL TYIN ( TY, HX, L2, L3, L4 )                                  18JA8
C
C-----RETURN TO SUBROUTINE V03SUB
C
      RETURN                                                            22JA8
C
      END                                                               22JA8
```

```
      SUBROUTINE ZERO2 ( AA, A1, Q, TX, TY, W, L1, L2, L3 )              03FE8
C
C-----THIS SUBROUTINE ZEROS OUT PREVIOUS LOAD DATA.
C
C
C-----NOTATION FOR SUBROUTINE ZERO2
C
C          AA( )                 RECURSION COEFFICIENT
C          A1( )                 TEMPORARY AA( ) TERM
C          HT1, HUM, MXP,        POSITIONING VARIABLES IN COMMON FIELD
C          NT1, NUM
C          I, J                  COUNTERS IN DO LOOP
C          L1, L2, L3            TRANSFER CONSTANTS
C          MXP3                  MX PLUS THREE
C          MXP5                  MX PLUS FIVE
C          MYP5                  MY PLUS FIVE
C          Q( )                  LATERAL LOAD
C          TX( )                 EXTERNAL COUPLE X-DIRECTION
C          TY( )                 EXTERNAL COUPLE Y-DIRECTION
C          W( )                  DEFLECTION OF SLAB
C
C
      DIMENSION      AA(L1,1),       A1(L1,1),       Q(L2,L3),           03FE8
     1               TX(L2,L3),      TY(L2,L3),      W(L2,L3)            03FE8
C
      COMMON    KERR, I, J, NUM(13), HUM(36), NT1(11), HT1(2), MYP2,     18JA8
     1          MXP3, MXP(3), MXP5, MYP5                                 16DE7
C
C
          DO  200  J = 1, MYP5                                           18JA8
C
          DO  100  I = 1, MXP5                                           05DE7
C
                Q(I,J) = 0.0                                             05DE7
               TX(I,J) = 0.0                                             05DE7
               TY(I,J) = 0.0                                             05DE7
                W(I,J) = 0.0                                             05DE7
C
  100     CONTINUE                                                       05DE7
C
  200     CONTINUE                                                       18JA8
C
          DO  300  I  =  1, MXP3                                         18JA8
C
               AA(I,1)  =  0.0                                           18JA8
               A1(I,1)  =  0.0                                           03FE8
C
  300     CONTINUE                                                       18JA8
C
C-----RETURN TO SUBROUTINE INPT1
C
      RETURN                                                             05DE7
C
      END                                                                05DE7
```

```
      SUBROUTINE MULIN                                                  05DE7
C
C-----THIS SUBROUTINE PRINTS THE MULTIPLE LOAD STATEMENTS AND INPUT
C          TABLES 2, 3, AND 4 FOR OFF-SPRING PROBLEMS.
C
C
   15 FORMAT ( // 5X, 61( 1H* ), / 5X, 1H*, 59X, 1H*, / 5X, 1H*, 4X,    18JA8
     1           50H RECURSION COEFFICIENTS HELD FROM PREVIOUS PROBLEM, 18JA8
     2            5X, 1H*, / 5X, 1H*, 59X, 1H*, / 5X, 61( 1H* ), // )  18JA8
   20 FORMAT ( //35H     TABLE 2.  INCREMENT LENGTH    , //             05DE7
     1           40H          HELD FROM PREVIOUS PROBLEM    , / )       05DE7
   25 FORMAT ( //35H     TABLE 3.  BENDING STIFFNESS   , //             16JA8
     1           40H          HELD FROM PREVIOUS PROBLEM    , / )       16JA8
   30 FORMAT ( //35H     TABLE 4.  TWISTING STIFFNESS  , //             16JA8
     1           40H          HELD FROM PREVIOUS PROBLEM    , / )       16JA8
C
C
C-----PRINT MULTIPLE LOAD STATEMENT
C
      PRINT 15                                                          05DE7
C
C-----PRINT TABLE 2
C
      PRINT 20                                                          05DE7
C
C-----PRINT TABLE 3
C
      PRINT 25                                                          16JA8
C
C-----PRINT TABLE 4
C
      PRINT 30                                                          16JA8
C
C-----RETURN TO SUBROUTINE INPT1
C
      RETURN                                                            05DE7
C
      END                                                               05DE7
```

```
      SUBROUTINE QIN ( Q, HX, HY, L1, L2, L3, L4 )                          22JA8
C
C-----THIS SUBROUTINE INPUTS LOAD DATA.
C
C
C-----NOTATION FOR SUBROUTINE QIN
C
C          AREA, AREAX, AREAY  AREA FUNCTIONS COMPUTED FOR CONVENIENCE
C          HT1, HUM, NT, NT1   POSITIONING VARIABLES IN COMMON FIELD
C          HX( )               INCREMENT LENGTH X-BEAM
C          HY( )               INCREMENT LENGTH Y-BEAM
C          I, II1, II2, I1,    COUNTERS AND INDICES IN DO LOOPS
C          I2, J, JJ1, JJ2,
C          J1, J2, N
C          IN1, JN1, IN2, JN2  INPUT STATION NUMBERS
C          KERR                ERROR FLAG
C          L1 --- L4           TRANSFER CONSTANTS
C          MXP3                MX PLUS THREE
C          MYP3                MY PLUS THREE
C          NCT5                NUMBER OF CARDS IN TABLE 5
C          Q( )                LATERAL LOAD
C          QN                  TEMPORARY INPUT CONSTANT
C          QNH                 HALF OF QN
C          QNQ                 QUARTER OF QN
C          QQN                 TEMPORARY INPUT CONSTANT
C
C
      DIMENSION     Q(L1,L2),       HX(L3),         HY(L4)                  22JA8
C
      COMMON   KERR, I, II1, II2, IN1, IN2, I1, I2, J, JJ1, JJ2, JN1,     11DE7
     1         JN2, J1, J2, N, AREA, AREAX, AREAY, QN, QNH, QNQ, QQN,     03FE8
     2         HUM(29), NT(3), NCT5, NT1(7), HT1(2), MYP2, MXP3, MYP3     03FE8
C
   10 FORMAT ( 4 ( 2X, I3 ), 2E10.3 )                                       31JA8
   20 FORMAT ( //40H     TABLE 5.  LOAD DATA                               24JA8
     1          //50H     SUPPORT VALUES HELD FROM PREVIOUS PROBLEM        07FE8
     2          //40H       FROM   THRU       Q          QQ               07FE8
     3           /40H        STA    STA      (F)      (F/LL) , / )         07FE8
   30 FORMAT ( 5X, 2 ( 1X, I2, 1X, I3, ), 4E11.3 )                          16JA8
   40 FORMAT ( //40H          NO DATA INPUT FOR THIS TABLE  , / )           24JA8
C
C
C-----INPUT TABLE 5
C
      PRINT 20                                                              04NO7
C
          IF ( 1 .GT. NCT5 )            GO TO 500                         22JA8
C
          DO  450  N = 1, NCT5                                              22JA8
C
      READ  10, IN1, JN1, IN2, JN2, QN, QQN                                 16JA8
      PRINT 30, IN1, JN1, IN2, JN2, QN, QQN                                 16JA8
```

```
                J1   =   JN1 + 3                                                07NO7
                J2   =   JN2 + 3                                                07NO7
C
                II1  =   I1 + 1                                                 04NO7
                II2  =   I2 - 1                                                 04NO7
                JJ1  =   J1 + 1                                                 04NO7
                JJ2  =   J2 - 1                                                 04NO7
C
                QNH  =   QN * 0.50                                              03FE8
                QNQ  =   QN * 0.25                                              03FE8
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
           IF ( I1 .GT. I2 )              KERR  =  KERR + 1                     15DE7
           IF ( J1 .GT. J2 )              KERR  =  KERR + 1                     15DE7
           IF ( I1 .LT. 3 )               KERR  =  KERR + 1                     15DE7
           IF ( J1 .LT. 3 )               KERR  =  KERR + 1                     15DE7
           IF ( I2 .GT. MXP3 )            KERR  =  KERR + 1                     15DE7
           IF ( J2 .GT. MYP3 )            KERR  =  KERR + 1                     15DE7
C
           IF ( KERR .GT. 0 )             GO TO 450                             15DE7
C
C-----ADD INPUT DATA FROM THIS CARD TO PREVIOUSLY STORED DATA
C
           IF ( III .GT. II2 )            GO TO 210                             04NO7
           IF ( JJ1 .GT. JJ2 )            GO TO 310                             04NO7
C
           DO  200  I = III, II2                                                04NO7
C
           DO  100  J = JJ1, JJ2                                                22JA8
C
                AREA  =  ( HX(I) + HX(I+1) ) * ( HY(J) + HY(J+1) ) * 0.2503NO7
C
                Q(I,J)  =  Q(I,J) + QN + QQN * AREA                             16JA8
C
  100      CONTINUE                                                             22JA8
C
  200      CONTINUE                                                             04NO7
C
  210      IF ( JJ1 .GT. JJ2 )            GO TO 410                             04NO7
C
           DO  300  J = JJ1, JJ2                                                04NO7
C
                AREAY  =  ( HY(J) + HY(J+1) ) * 0.25                            07NO7
C
                AREA  =  AREAY * HX(I1+1)                                       04NO7
C
                Q(I1,J)  =  Q(I1,J) + QNH + QQN * AREA                          03FE8
C
                AREA  =  AREAY * HX(I2)                                         04NO7
C
                Q(I2,J)  =  Q(I2,J) + QNH + QQN * AREA                          03FE8
C
```

```
  310     IF ( II1 .GT. II2 )           GO TO 410                    13DE7
C
          DO  400  I = II1, II2                                      13DE7
C
               AREAX  =  ( HX(I) + HX(I+1) ) * 0.25                  04NO7
C
               AREA   =  AREAX * HY(J1+1)                            04NO7
C
               Q(I,J1)  =  Q(I,J1) + QNH + QQN * AREA                03FE8
C
               AREA   =  AREAX * HY(J2)                              04NO7
C
               Q(I,J2)  =  Q(I,J2) + QNH + QQN * AREA                03FE8
C
  400     CONTINUE                                                   04NO7
C
  410          AREA   =  HX(I1+1) * HY(J1+1) * 0.25                  10NO7
C
               Q(I1,J1)  =  Q(I1,J1) + QNQ + QQN * AREA              03FE8
C
               AREA   =  HX(I1+1) * HY(J2) * 0.25                    10NO7
C
               Q(I1,J2)  =  Q(I1,J2) + QNQ + QQN * AREA              03FE8
C
               AREA   =  HX(I2) * HY(J1+1) * 0.25                    10NO7
C
               Q(I2,J1)  =  Q(I2,J1) + QNQ + QQN * AREA              03FE8
C
               AREA   =  HX(I2) * HY(J2) * 0.25                      10NO7
C
               Q(I2,J2)  =  Q(I2,J2) + QNQ + QQN * AREA              03FE8
C
  450     CONTINUE                                                   04NO7
C
C-----RETURN TO SUBROUTINE INPT1
C
      RETURN                                                         16JA8
C
C-----PRINT NO DATA STATEMENT
C
  500 PRINT 40                                                       16JA8
C
C-----RETURN TO SUBROUTINE INPT1
C
      RETURN                                                         16JA8
C
      END                                                            22JA8
```

```
      SUBROUTINE TXIN ( TX, HY, L1, L2, L3 )                                22JA8
C
C-----THIS SUBROUTINE INPUTS TORQUES IN THE X-DIRECTION.
C
C
C-----NOTATION FOR SUBROUTINE TXIN
C
C         HT1, HUM, NT,       POSITIONING VARIABLES IN COMMON FIELD
C         NT1, NUM
C         HY( )               INCREMENT LENGTH Y-BEAM
C         HYN, HYN1           TEMPORARY FUNCTIONS OF HY
C         I, I1, I2, J, JJ1,  COUNTERS AND INDICES IN DO LOOP
C         JJ2, J1, J2, N
C         IN1, JN1, IN2, JN2  INPUT BAR NUMBERS
C         KERR                ERROR FLAG
C         L1, L2, L3          TRANSFER CONSTANTS
C         MXP3                MX PLUS THREE
C         MYP3                MY PLUS THREE
C         NCT6                NUMBER OF CARDS IN TABLE 6
C         TTXN                TEMPORARY INPUT CONSTANT
C         TX( )               EXTERNAL COUPLE X-DIRECTION
C         TXN                 TEMPORARY INPUT CONSTANT
C         TXNH                HALF OF TXN
C
C
      DIMENSION     TX(L1,L2),     HY(L3)                                   22JA8
C
      COMMON   KERR, I, IN1, IN2, I1, I2, J, JJ1, JJ2, JN1, JN2, J1,     24DE7
     1         J2, N, NUM(2), TXN, TXNH, TTXN, HYN, HYN1, HUM(31),       03FE8
     2         NT(4), NCT6, NT1(6), HT1(2), MYP2, MXP3, MYP3              03FE8
C
   10 FORMAT ( 4 ( 2X, I3 ), 2E10.3 )                                      31JA8
   20 FORMAT ( //40H     TABLE 6.  X-DIRECTIONAL LOAD DATA                 21MR8
     1         //50H     AXIAL LOAD VALUES HELD FROM PREVIOUS PROBLEM      07FE8
     2         //40H       FROM   THRU       TX          TTX              07FE8
     3          /40H      X-BAR  X-BAR      (FL)       (FL/L) , / )       21MR8
   30 FORMAT ( 5X, 2 ( 1X, I2, 1X, I3 ), 4E11.3 )                          22JA8
   40 FORMAT ( //40H          NO DATA INPUT FOR THIS TABLE  , / )          24JA8
C
C
C-----INPUT TABLE 6
C
      PRINT 20                                                             22JA8
C
          IF ( 1 .GT. NCT6 )            GO TO 600                        22JA8
C
          DO  500  N = 1, NCT6                                             22JA8
C
      READ  10, IN1, JN1, IN2, JN2, TXN, TTXN                              22JA8
      PRINT 30, IN1, JN1, IN2, JN2, TXN, TTXN                              22JA8
C
               I1  =  IN1 + 3                                              16NO7
               I2  =  IN2 + 3                                              07NO7
               J1  =  JN1 + 3                                              07NO7
               J2  =  JN2 + 3                                              07NO7
```

```
C
                JJ1  =  J1 + 1                                          04NO7
                JJ2  =  J2 - 1                                          04NO7
C
                TXNH  = TXN * 0.50                                      03FE8
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
           IF ( I1 .GT. I2 )              KERR  =  KERR + 1             15DE7
           IF ( J1 .GT. J2 )              KERR  =  KERR + 1             15DE7
           IF ( I1 .LT. 3 )               KERR  =  KERR + 1             15DE7
           IF ( J1 .LT. 3 )               KERR  =  KERR + 1             15DE7
           IF ( I2 .GT. MXP3 )            KERR  =  KERR + 1             15DE7
           IF ( J2 .GT. MYP3 )            KERR  =  KERR + 1             15DE7
C
           IF ( I1 .GT. 3 )               GO TO 100                     22JA8
C
           IF ( I1 .EQ. I2 )              KERR  =  KERR + 1             22JA8
C
  100      IF ( KERR .GT. 0 )             GO TO 500                     07FE8
C
C-----ADD INPUT DATA FROM THIS CARD TO PREVIOUSLY STORED DATA
C
           IF ( JJ1 .GT. JJ2 )            GO TO 310                     07FE8
C
           DO  300  J = JJ1, JJ2                                        22JA8
C
                HYN  =  ( HY(J) + HY(J+1) ) * 0.50                      22JA8
C
           DO  200  I = I1, I2                                          22JA8
C
                TX(I,J)  =  TX(I,J) + TXN + TTXN * HYN                  22JA8
C
  200      CONTINUE                                                     22JA8
C
  300      CONTINUE                                                     22JA8
C
  310           HYN  =  HY(J1+1) * 0.50                                22JA8
                HYN1  =  HY(J2) * 0.50                                  07FE8
C
           DO  400  I = I1, I2                                          22JA8
C
                TX(I,J1)  =  TX(I,J1) + TXNH + TTXN * HYN               03FE8
C
                TX(I,J2)  =  TX(I,J2) + TXNH + TTXN * HYN1              03FE8
C
  400      CONTINUE                                                     22JA8
C
  500      CONTINUE                                                     22JA8
C
C-----RETURN TO SUBROUTINE INPT1
C
      RETURN                                                            22JA8
C
C-----PRINT NO DATA STATEMENT
C
  600 PRINT 40                                                          22JA8
C
C-----RETURN TO SUBROUTINE INPT1
C
      RETURN                                                            22JA8
C
      END                                                               22JA8
```

```
      SUBROUTINE TYIN ( TY, HX, L1, L2, L3 )                                22JA8
C
C-----THIS SUBROUTINE INPUTS TORQUES IN THE Y-DIRECTION.
C
C
C-----NOTATION FOR SUBROUTINE TYIN
C
C          HT1, HUM, NT,        POSITIONING VARIABLES IN COMMON FIELD
C          NT1, NUM
C          HX( )                INCREMENT LENGTH X-BEAM
C          HXN, HXN1            TEMPORARY FUNCTIONS OF HX
C          I, II1, II2, I1,     COUNTERS AND INDICES IN DO LOOPS
C          I2, J, J1, J2, N
C          IN1, JN1, IN2, JN2   INPUT BAR NUMBERS
C          KERR                 ERROR FLAG
C          L1, L2, L3           TRANSFER CONSTANTS
C          MXP3                 MX PLUS THREE
C          MYP3                 MY PLUS THREE
C          TTYN                 TEMPORARY INPUT CONSTANT
C          TY( )                EXTERNAL COUPLE Y-DIRECTION
C          TYN                  TEMPORARY INPUT CONSTANT
C          TYNH                 HALF OF TYN
C
C
      DIMENSION      TY(L1,L2),      HX(L3)                                 22JA8
C
      COMMON    KERR, I, II1, II2, IN1, IN2, I1, I2, J, JN1, JN2, J1,     24DE7
     1          J2, N, NUM(2), TYN, TYNH, TTYN, HXN, HXN1, HUM(31),       03FE8
     2          NT(5), NCT7, NT1(5), HT1(2), MYP2, MXP3, MYP3             03FE8
C
   10 FORMAT ( 4 ( 2X, I3 ), 2E10.3 )                                      31JA8
   20 FORMAT ( //40H      TABLE 7.  Y-DIRECTIONAL LOAD DATA                21MR8
     1          //50H     AXIAL LOAD VALUES HELD FROM PREVIOUS PROBLEM     07FE8
     2          //40H       FROM   THRU       TY          TTY             07FE8
     3           /40H      Y-BAR  Y-BAR      (FL)       (FL/L) , / )      07FE8
   30 FORMAT ( 5X, 2 ( 1X, I2, 1X, I3 ), 4E11.3 )                          22JA8
   40 FORMAT ( //40H           NO DATA INPUT FOR THIS TABLE  , / )          24JA8
C
C
C-----INPUT TABLE 7
C
      PRINT 20                                                             22JA8
C
          IF ( 1 .GT. NCT7 )            GO TO 600                          22JA8
C
          DO  500  N = 1, NCT7                                             22JA8
C
      READ  10, IN1, JN1, IN2, JN2, TYN, TTYN                              22JA8
      PRINT 30, IN1, JN1, IN2, JN2, TYN, TTYN                              22JA8
C
                I1  =  IN1 + 3                                             16NO7
                I2  =  IN2 + 3                                             07NO7
                J1  =  JN1 + 3                                             07NO7
                J2  =  JN2 + 3                                             07NO7
C
```

```
                II1  =  I1 + 1                                          04NO7
                II2  =  I2 - 1                                          04NO7
C
                TYNH  =  TYN * 0.50                                     03FE8
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
          IF ( I1 .GT. I2 )             KERR  =  KERR + 1               15DE7
          IF ( J1 .GT. J2 )             KERR  =  KERR + 1               15DE7
          IF ( I1 .LT. 3 )              KERR  =  KERR + 1               15DE7
          IF ( J1 .LT. 3 )              KERR  =  KERR + 1               15DE7
          IF ( I2 .GT. MXP3 )           KERR  =  KERR + 1               15DE7
          IF ( J2 .GT. MYP3 )           KERR  =  KERR + 1               15DE7
C
          IF ( J1 .GT. 3 )              GO TO 100                       22JA8
C
          IF ( J1 .EQ. J2 )             KERR  =  KERR + 1               22JA8
C
  100     IF ( KERR .GT. 0 )            GO TO 500                       07FE8
C
C-----ADD INPUT DATA FROM THIS CARD TO PREVIOUSLY STORED DATA
C
          IF ( II1 .GT. II2 )           GO TO 310                       07FE8
C
          DO  300  I = II1, II2                                         22JA8
C
               HXN  =  ( HX(I) + HX(I+1) ) * 0.50                       04NO7
C
          DO  200  J = J1, J2                                           22JA8
C
               TY(I,J)  =  TY(I,J) + TYN + TTYN * HXN                   22JA8
C
  200     CONTINUE                                                      22JA8
C
  300     CONTINUE                                                      22JA8
C   .
  310          HXN  =  HX(I1+1) * 0.50                                  04NO7
               HXN1  =  HX(I2) * 0.50                                   04NO7
C
          DO  400  J = J1, J2                                           24JA8
C
               TY(I1,J)  =  TY(I1,J) + TYNH + TTYN * HXN                03FE8
C
               TY(I2,J)  =  TY(I2,J) + TYNH + TTYN * HXN1               03FE8
C
  400     CONTINUE                                                      22JA8
C
  500     CONTINUE                                                      22JA8
C
C-----RETURN TO SUBROUTINE INPT1
C
      RETURN                                                            22JA8
C
C-----PRINT NO DATA STATEMENT
C
  600 PRINT 40                                                          22JA8
C
C-----RETURN TO SUBROUTINE INPT1
C
      RETURN                                                            22JA8
C
      END                                                               22JA8
```

```
      SUBROUTINE INPT2 ( AA, A1, BB, BB1, CC, CC1, CX, CY, DX, DY, HX,  03FE8
     1                   HY, PX, PY, Q, S, TX, TY, L1, L2, L3, L4, L5 ) 03FE8
C
C-----THIS SUBROUTINE CALLS INPUT ROUTINE FOR AN ORIGINAL PROBLEM.
C
C
C-----NOTATION FOR SUBROUTINE INPT2
C
C          AA( )               RECURSION COEFFICIENT
C          A1( )               TEMPORARY AA( ) TERM
C          BB( )               RECURSION COEFFICIENT
C          BB1( )              TEMPORARY BB( ) TERMS
C          CC( )               RECURSION COEFFICIENT
C          CC1( )              TEMPORARY CC( ) TERMS
C          CX( )               TWISTING STIFFNESS X-DIRECTION
C          CY( )               TWISTING STIFFNESS Y-DIRECTION
C          DX( )               BENDING STIFFNESS X-DIRECTION
C          DY( )               BENDING STIFFNESS Y-DIRECTION
C          HX( )               INCREMENT LENGTH X-BEAM
C          HY( )               INCREMENT LENGTH Y-BEAM
C          L1 --- L5           CONSTANTS FOR TRANSFERRING DIMENSION STATEMENT
C          PX( )               AXIAL LOAD X-DIRECTION
C          PY( )               AXIAL LOAD Y-DIRECTION
C          Q( )                LATERAL LOAD
C          S( )                SUPPORT SPRING
C          TX( )               EXTERNAL COUPLE X-DIRECTION
C          TY( )               EXTERNAL COUPLE Y-DIRECTION
C
C
      DIMENSION     AA(L1,1),       A1(L1,1),       BB(L1,L1),          03FE8
     1              BB1(L1,L1),     CC(L1,L1),      CC1(L1,L1),         03FE8
     2              CX(L2,L3),      CY(L2,L3),      DX(L2,L3),          03FE8
     3              DY(L2,L3),      HX(L4),         HY(L5),             03FE8
     4              PX(L2,L3),      PY(L2,L3),      Q(L2,L3),           03FE8
     5              S(L2,L3),       TX(L2,L3),      TY(L2,L3)           03FE8
C
C
C-----ELIMINATE DATA FOR PREVIOUS PROBLEMS
C
      CALL ZERO ( AA, A1, BB, BB1, CC, CC1, CX, CY, DX, DY, HX, HY, PX, 03FE8
     1            PY, Q, S, TX, TY, L1, L2, L3, L4, L5 )                03FE8
C
C-----INPUT TABLE 2
C
      CALL INCRI ( HX, HY, L4, L5 )                                     18JA8
C
C-----INPUT TABLE 3
C
      CALL DXDYIN ( DX, DY, L2, L3 )                                    18JA8
C
C-----INPUT TABLE 4
C
      CALL CXCYIN ( CX, CY, L2, L3 )                                    18JA8
C
C-----INPUT TABLE 5
```

```
C
      CALL QSIN ( Q, S, HX, HY, L2, L3, L4, L5 )                        18JA8
C
C-----INPUT TABLE 6
C
      CALL TXPXIN ( TX, PX, HY, L2, L3, L5 )                            18JA8
C
C-----INPUT TABLE 7
C
      CALL TYPYIN ( TY, PY, HX, L2, L3, L4 )                            18JA8
C
C-----PLACE SPRING AT STATIONS BEYOND BOUNDARIES OF REAL SLAB TO MAKE
C        SOLUTION OF NON-RECTANGULAR SLABS OR SLABS WITH HOLES POSSIBLE
C
      CALL BOUND ( DX, DY, S, L2, L3 )                                  18JA8
C
C-----RETURN TO SUBROUTINE V03SUB
C
      RETURN                                                            22JA8
C
      END                                                               22JA8
```

```
      SUBROUTINE ZERO ( AA, A1, BB, BB1, CC, CC1, CX, CY, DX, DY, HX,    03FE8
     1                  HY, PX, PY, Q, S, TX, TY, L1, L2, L3, L4, L5 )   03FE8
C
C-----THIS SUBROUTINE ZEROS OUT THE DATA FROM PREVIOUS PROBLEMS.
C
C
C-----NOTATION FOR SUBROUTINE ZERO
C
C          AA( )                RECURSION COEFFICIENT
C          A1( )                TEMPORARY AA( ) TERM
C          BB( )                RECURSION COEFFICIENT
C          BB1( )               TEMPORARY BB( ) TERMS
C          CC( )                RECURSION COEFFICIENT
C          CC1( )               TEMPORARY CC( ) TERMS
C          CX( )                TWISTING STIFFNESS X-DIRECTION
C          CY( )                TWISTING STIFFNESS Y-DIRECTION
C          DX( )                BENDING STIFFNESS X-DIRECTION
C          DY( )                BENDING STIFFNESS Y-DIRECTION
C          HT1, HUM, MXP,       POSITIONING VARIABLES IN COMMON FIELD
C          NT1, NUM
C          HX( )                INCREMENT LENGTH X-BEAM
C          HY( )                INCREMENT LENGTH Y-BEAM
C          I, II, J             COUNTERS IN DO LOOPS
C          KERR                 ERROR FLAG
C          L1 --- L5            CONSTANTS FOR TRANSFERRING DIMENSION STATEMENT
C          MXP3                 MX PLUS THREE
C          MXP5                 MX PLUS FIVE
C          MYP5                 MY PLUS FIVE
C          PX( )                AXIAL LOAD X-DIRECTION
C          PY( )                AXIAL LOAD Y-DIRECTION
C          Q( )                 LATERAL LOAD
C          S( )                 SUPPORT SPRING
C          TX( )                EXTERNAL COUPLE X-DIRECTION
C          TY( )                EXTERNAL COUPLE Y-DIRECTION
C
C
      DIMENSION      AA(L1,1),       A1(L1,1),       BB(L1,L1),            03FE8
     1               BB1(L1,L1),     CC(L1,L1),      CC1(L1,L1),           03FE8
     2               CX(L2,L3),      CY(L2,L3),      DX(L2,L3),            03FE8
     3               DY(L2,L3),      HX(L4),         HY(L5),               03FE8
     4               PX(L2,L3),      PY(L2,L3),      Q(L2,L3),             03FE8
     5               S(L2,L3),       TX(L2,L3),      TY(L2,L3)             03FE8
C
      COMMON    KERR, I, II, J, NUM(12), HUM(36), NT1(11), HT1(2), MYP2,  18JA8
     1          MXP3, MXP(3), MXP5, MYP5                                  16DE7
C
C
           DO  200  J = 1, MYP5                                           18JA8
C
                HY(J) = 0.0                                               18AG7
C
           DO  100  I = 1, MXP5                                           11SE7
C
                CX(I,J) = 0.0                                             01NO6
                CY(I,J) = 0.0                                             01NO6
```

```
                DX(I,J) = 0.0                                          01NO6
                DY(I,J) = 0.0                                          01NO6
                PX(I,J) = 0.0                                          23JE7
                PY(I,J) = 0.0                                          23JE7
                 Q(I,J) = 0.0                                          01NO6
                 S(I,J) = 0.0                                          01NO6
                TX(I,J) = 0.0                                          23JE7
                TY(I,J) = 0.0                                          23JE7
C
  100       CONTINUE                                                   04MY7
C
  200       CONTINUE                                                   18JA8
C
            DO  400  II = 1, MXP3                                      18JA8
C
                HX(II)    =  0.0                                       09NO7
                AA(II,1)  =  0.0                                       16JA8
                A1(II,1)  =  0.0                                       03FE8
C
            DO  300  I  =  1, MXP3                                     18JA8
C
                BB(I,II)  =  0.0                                       03FE8
                CC(I,II)  =  0.0                                       03FE8
                BB1(I,II)  =  0.0                                      03FE8
                CC1(I,II)  =  0.0                                      03FE8
C
  300       CONTINUE                                                   18JA8
C
  400       CONTINUE                                                   18JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
       RETURN                                                          09NO7
C
       END                                                             09NO7
```

```
      SUBROUTINE INCRI ( HX, HY, L1, L2 )                                16JA8
C
C-----THIS SUBROUTINE INPUTS THE VARIBLE INCREMENT LENGTHS
C
C
C-----NOTATION FOR SUBROUTINE INCRI
C
C          HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C          HX( )               INCREMENT LENGTH X-BEAM
C          HXHX                RATIO OF HX(I) AND HX(I+1)
C          HXN                 TEMPORARY INPUT CONSTANT
C          HY( )               INCREMENT LENGTH Y-BEAM
C          HYHY                RATIO OF HY(J) AND HY(J+1)
C          HYN                 TEMPORARY INPUT CONSTANT
C          I, I1, I2, J1,      COUNTERS AND INDICES IN DO LOOPS
C          J2, N
C          IN1, JN1, IN2, JN2  INPUT BAR NUMBERS
C          KERR                ERROR FLAG
C          L1, L2              TRANSFER CONSTANTS
C          MXP3                MX PLUS THREE
C          MYP3                MY PLUS THREE
C          NCT2                NUMBER OF CARDS IN TABLE 2
C
C
      DIMENSION     HX(L1),        HY(L2)                                16JA8
C
      COMMON   KERR, I, IN1, IN2, I1, I2, JN1, JN2, J1, J2, N, NUM(5),   21MR8
     1         HXHX, HXN, HYHY, HYN, HUM(32), NCT2, NT1(10), HT1(2),     21MR8
     2         MYP2, MXP3, MYP3                                          21MR8
C
   10 FORMAT( 2I5, E10.3, 10X, 2I5, E10.3 )                              21MR8
   20 FORMAT ( //35H     TABLE 2.  INCREMENT LENGTH    , //              09NO7
     1           40H       FROM   THRU      HX                          21MR8
     2           40H       FROM   THRU      HY              , /         21MR8
     3           40H      X-BAR  X-BAR     (L)                          21MR8
     4           40H      Y-BAR  Y-BAR     (L)              , / )       21MR8
   30 FORMAT ( 7X, I3, 4X, I3, 2X, E11.3, 17X, I3, 4X, I3, 2X, E11.3 )   21MR8
   40 FORMAT ( //30X, 25H*****   WARNING   *****                          16JA8
     1          /50H     POSSIBLE ERROR IN SOLUTION DUE TO THE RATIO O   24JA8
     2           30HF ADJACENT INCREMENT LENGTHS                         24JA8
     3          /50H          AT SOME POINTS EXCESSING FIVE TO ONE ( 5   24JA8
     4           10H TO 1 )   ,                                          24JA8
     5          /30X, 25H*****              *****  , // )               24JA8
   50 FORMAT ( //40H     ******************************     ,            03FE8
     1          /40H     *                            *     ,            03FE8
     2          /40H     * ERROR --- NO TABLE 2 INPUT *     ,            03FE8
     3          /40H     *                            *     ,            03FE8
     4          /40H     ******************************     , // )      03FE8
C
C
C-----INPUT TABLE 2
C
      PRINT 20                                                           09NO7
C
          IF ( 1 .GT. NCT2 )          GO TO 900                          26JA8
```

```
C
      DO  300  N = 1, NCT2                                                  26JA8
C
      READ  10, IN1, IN2, HXN, JN1, JN2, HYN                                21MR8
      PRINT 30, IN1, IN2, HXN, JN1, JN2, HYN                                21MR8
C
            I1  =  IN1 + 3                                                  09NO7
            I2  =  IN2 + 3                                                  09NO7
            J1  =  JN1 + 3                                                  21MR8
            J2  =  JN2 + 3                                                  21MR8
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
      IF ( J1 .GT. J2 )             KERR  =  KERR + 1                       21MR8
      IF ( J1 .LT. 4 )              KERR  =  KERR + 1                       21MR8
      IF ( J2 .GT. MYP3 )           KERR  =  KERR + 1                       21MR8
C
      IF ( KERR .GT. 0 )            GO TO 210                               15DE7
C
      IF ( HYN .EQ. 0.0 )           GO TO 210                               17NO7
C
C-----ADD Y INCREMENT FROM THIS CARD TO PREVIOUSLY STORED DATA
C
      DO  200  I = J1, J2                                                   21MR8
C
           HY(I)  =  HY(I) + HYN                                            09NO7
C
  200 CONTINUE                                                              09NO7
C
  210 IF ( HXN .EQ. 0.0 )           GO TO 300                               17NO7
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
      IF ( I1 .GT. I2 )             KERR  =  KERR + 1                       15DE7
      IF ( I1 .LT. 4 )              KERR  =  KERR + 1                       18JA8
      IF ( I2 .GT. MXP3 )           KERR  =  KERR + 1                       15DE7
C
      IF ( KERR .GT. 0 )            GO TO 300                               15DE7
C
C-----ADD X INCREMENT FROM THIS CARD TO PREVIOUSLY STORED DATA
C
      DO  250  I = I1, I2                                                   17NO7
C
           HX(I)  =  HX(I) + HXN                                            09NO7
C
  250 CONTINUE                                                              17NO7
C
  300 CONTINUE                                                              09NO7
C
      IF ( KERR .GT. 0 )            GO TO 750                               02MR8
C
C-----SET INCREMENT LENGTH OUTSIDE THE REAL SLAB
C
  400 DO  500  I = 1, 3                                                     09NO7
C
```

```
                HX(I)         =  HX(4)                                    09NO7
                HY(I)         =  HY(4)                                    09NO7
C
                HX(MXP3+I)    =  HX(MXP3)                                 09NO7
                HY(MYP3+I)    =  HY(MYP3)                                 09NO7
C
  500      CONTINUE                                                       09NO7
C
C-----CHECK RATIO OF INCREMENT LENGTHS
C
           DO  600  I = 3, MXP3                                           24JA8
C
                HXHX  =  HX(I) / HX(I+1)                                  16JA8
C
           IF ( HXHX .LT. 0.2 )          GO TO 800                        16JA8
           IF ( HXHX .GT. 5.0 )          GO TO 800                        16JA8
C
  600      CONTINUE                                                       16JA8
C
           DO  700  I = 3, MYP3                                           26JA8
C
                HYHY  =  HY(I) / HY(I+1)                                  26JA8
C
           IF ( HYHY .LT. 0.2 )          GO TO 800                        16JA8
           IF ( HYHY .GT. 5.0 )          GO TO 800                        16JA8
C
  700      CONTINUE                                                       16JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
  750      RETURN                                                         02MR8
C
C-----PRINT WARNING STATEMENT
C
  800 PRINT 40                                                            16JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
       RETURN                                                             16JA8
C
C-----PRINT ERROR STATEMENT
C
  900 PRINT 50                                                            16JA8
C
                KERR  =  KERR + 1                                         16JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
       RETURN                                                             16JA8
C
       END                                                                09NO7
```

```
      SUBROUTINE DXDYIN ( DX, DY, L1, L2 )                                  24JA8
C
C-----THIS SUBROUTINE INPUTS BENDING STIFFNESS.                             16JA8
C
C
C-----NOTATION FOR SUBROUTINE DXDYIN
C
C         DX( )               BENDING STIFFNESS X-DIRECTION
C         DXN                 TEMPORARY INPUT CONSTANT
C         DXNH                HALF OF DXN
C         DXNQ                QUARTER OF DXN
C         DY( )               BENDING STIFFNESS Y-DIRECTION
C         DYN                 TEMPORARY INPUT CONSTANT
C         DYNH                HALF OF DYN
C         DYNQ                QUARTER OF DYN
C         HT1, HUM, NT1       POSITIONING VARIABLES IN COMMON FIELD
C         I, II1, II2, I1,    COUNTERS AND INDICES IN DO LOOPS
C         I2, J, JJ1, JJ2,
C         J1, J2, N
C         IN1, JN1, IN2, JN2  INPUT STATION NUMBERS
C         KERR                ERROR FLAG
C         L1, L2              TRANSFER CONSTANTS
C         MXP3                MX PLUS THREE
C         MYP3                MY PLUS THREE
C         NCT3                NUMBER OF CARDS IN TABLE 3
C
C
      DIMENSION     DX(L1,L2),          DY(L1,L2)                           24JA8
C
      COMMON   KERR, I, II1, II2, IN1, IN2, I1, I2, J, JJ1, JJ2, JN1,     11DE7
     1         JN2, J1, J2, N, DXN, DXNH, DXNQ, DYN, DYNH, DYNQ,          21JA8
     2         HUM(30), NT, NCT3, NT1(9), HT1(2), MYP2, MXP3, MYP3        23JA8
C
   10 FORMAT ( 4 ( 2X, I3 ), 2E10.3 )                                       16JA8
   20 FORMAT ( //40H     TABLE 3.  BENDING STIFFNESS        ,              16JA8
     1          //40H       FROM   THRU      DX          DY   ,            16JA8
     2           /40H        STA    STA     (FL)        (FL)  ,  / )       02FE8
   30 FORMAT ( 5X, 2 ( 1X, I2, 1X, I3 ), 2E11.3 )                           16JA8
   40 FORMAT ( //40H          NO DATA INPUT FOR THIS TABLE  , / )           24JA8
C
C
C-----INPUT TABLE 3
C
      PRINT 20                                                              04NO7
C
          IF ( 1 .GT. NCT3 )            GO TO 500                         04NO7
C
          DO  450  N = 1, NCT3                                              04NO7
C
      READ  10, IN1, JN1, IN2, JN2, DXN, DYN                                16JA8
      PRINT 30, IN1, JN1, IN2, JN2, DXN, DYN                                16JA8
C
                I1  =  IN1 + 3                                              16NO7
                I2  =  IN2 + 3                                              07NO7
                J1  =  JN1 + 3                                              07NO7
```

```
                J2  =  JN2 + 3                                          07NO7
C
                II1  =  I1 + 1                                          04NO7
                II2  =  I2 - 1                                          04NO7
                JJ1  =  J1 + 1                                          04NO7
                JJ2  =  J2 - 1                                          04NO7
C
                DXNH  =  DXN * 0.50                                     03NO7
                DYNH  =  DYN * 0.50                                     03NO7
                DXNQ  =  DXN * 0.25                                     15NO7
                DYNQ  =  DYN * 0.25                                     15NO7
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
          IF ( I1 .GT. I2 )             KERR  =  KERR + 1               15DE7
          IF ( J1 .GT. J2 )             KERR  =  KERR + 1               15DE7
          IF ( I1 .LT. 3 )              KERR  =  KERR + 1               15DE7
          IF ( J1 .LT. 3 )              KERR  =  KERR + 1               15DE7
          IF ( I2 .GT. MXP3 )           KERR  =  KERR + 1               15DE7
          IF ( J2 .GT. MYP3 )           KERR  =  KERR + 1               15DE7
C
          IF ( KERR .GT. 0 )            GO TO 450                       15DE7
C
C-----ADD INPUT DATA FROM THIS CARD TO PREVIOUSLY STORED DATA
C
          IF ( II1 .GT. II2 )           GO TO 210                       04NO7
          IF ( JJ1 .GT. JJ2 )           GO TO 310                       04NO7
C
          DO  200  I = II1, II2                                         04NO7
C
          DO  150  J = JJ1, JJ2                                         04NO7
C
               DX(I,J)  =  DX(I,J) + DXN                                03NO7
               DY(I,J)  =  DY(I,J) + DYN                                03NO7
C
  150     CONTINUE                                                      04NO7
C
  200     CONTINUE                                                      04NO7
C
  210     IF ( JJ1 .GT. JJ2 )           GO TO 410                       04NO7
C
          DO  300  J = JJ1, JJ2                                         04NO7
C
               DX(I1,J)  =  DX(I1,J) + DXNH                             03NO7
               DY(I1,J)  =  DY(I1,J) + DYNH                             03NO7
C
               DX(I2,J)  =  DX(I2,J) + DXNH                             04NO7
               DY(I2,J)  =  DY(I2,J) + DYNH                             04NO7
C
  300     CONTINUE                                                      04NO7
C
  310     IF ( II1 .GT. II2 )           GO TO 410                       13DE7
C
          DO  400  I = II1, II2                                         13DE7
C
```

```
                DX(I,J1)  =  DX(I,J1) + DXNH                          03NO7
                DY(I,J1)  =  DY(I,J1) + DYNH                          03NO7
C
                DX(I,J2)  =  DX(I,J2) + DXNH                          04NO7
                DY(I,J2)  =  DY(I,J2) + DYNH                          04NO7
C
  400      CONTINUE                                                   04NO7
C
  410           DX(I1,J1)  =  DX(I1,J1) + DXNQ                       04NO7
                DY(I1,J1)  =  DY(I1,J1) + DYNQ                       04NO7
C
                DX(I1,J2)  =  DX(I1,J2) + DXNQ                       04NO7
                DY(I1,J2)  =  DY(I1,J2) + DYNQ                       04NO7
C
                DX(I2,J1)  =  DX(I2,J1) + DXNQ                       04NO7
                DY(I2,J1)  =  DY(I2,J1) + DYNQ                       04NO7
C
                DX(I2,J2)  =  DX(I2,J2) + DXNQ                       04NO7
                DY(I2,J2)  =  DY(I2,J2) + DYNQ                       04NO7
C
  450      CONTINUE                                                   04NO7
C
C-----RETURN TO SUBROUTINE INPT2
C
      RETURN                                                         16JA8
C
C-----PRINT NO DATA STATEMENT
C
  500 PRINT 40                                                       16JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
      RETURN                                                         16JA8
C
      END                                                            16JA8
```

```
      SUBROUTINE CXCYIN ( CX, CY, L1, L2 )                                  16JA8
C
C-----THIS SUBROUTINE INPUTS TWISTING STIFFNESS.                            16JA8
C
C
C-----NOTATION FOR SUBROUTINE CXCYIN
C
C         CX( )               TWISTING STIFFNESS X-DIRECTION
C         CXN                 TEMPORARY INPUT CONSTANT
C         CY( )               TWISTING STIFFNESS Y-DIRECTION
C         CYN                 TEMPORARY INPUT CONSTANT
C         HT1, HUM, NT,       POSITIONING VARIABLES IN COMMON FIELD
C         NT1, NUM
C         I, I1, I2, J, J1,   COUNTERS AND INDICES IN DO LOOPS
C         J2, N
C         IN1, JN1, IN2, JN2  INPUT STATION NUMBERS
C         KERR                ERROR FLAG
C         L1, L2,             TRANSFER CONSTANTS
C         MXP3                MX PLUS THREE
C         MYP3                MY PLUS THREE
C         NCT4                NUMBER OF CARDS IN TABLE 4
C
C
      DIMENSION     CX(L1,L2),          CY(L1,L2)                           16JA8
C
      COMMON   KERR, I, IN1, IN2, I1, I2, J, JN1, JN2, J1, J2, N,           11DE7
     1         NUM(4), CXN, CYN, HUM(34), NT(2), NCT4, NT1(8), HT1(2),      21JA8
     2         MYP2, MXP3, MYP3                                             23JA8
C
   10 FORMAT ( 4 ( 2X, I3 ), 2E10.3 )                                       16JA8
   20 FORMAT ( //40H     TABLE 4.  TWISTING STIFFNESS       ,               24JA8
     1         //40H       FROM   THRU      CX          CY   ,               16JA8
     2          /40H        STA    STA     (FL)        (FL)  , / )          21MR8
   30 FORMAT ( 5X, 2 ( 1X, I2, 1X, I3 ), 2E11.3 )                           16JA8
   40 FORMAT ( //40H          NO DATA INPUT FOR THIS TABLE  , / )           24JA8
C
C
C-----INPUT TABLE 4
C
      PRINT 20                                                              04NO7
C
          IF ( 1 .GT. NCT4 )            GO TO 500                         16JA8
C
          DO  300  N = 1, NCT4                                            21JA8
C
      READ  10, IN1, JN1, IN2, JN2, CXN, CYN                              21JA8
      PRINT 30, IN1, JN1, IN2, JN2, CXN, CYN                              21JA8
C
                I1  =  IN1 + 3                                            16NO7
                I2  =  IN2 + 3                                            07NO7
                J1  =  JN1 + 3                                            07NO7
                J2  =  JN2 + 3                                            07NO7
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
```

```
           IF ( I1 .GT. I2 )              KERR  =  KERR + 1              15DE7
           IF ( J1 .GT. J2 )              KERR  =  KERR + 1              15DE7
           IF ( I1 .LT. 4 )               KERR  =  KERR + 1              21JA8
           IF ( J1 .LT. 4 )               KERR  =  KERR + 1              21JA8
           IF ( I2 .GT. MXP3 )            KERR  =  KERR + 1              15DE7
           IF ( J2 .GT. MYP3 )            KERR  =  KERR + 1              15DE7
C
           IF ( KERR .GT. 0 )             GO TO 300                      21JA8
C
C-----ADD INPUT DATA FROM THIS CARD TO PREVIOUSLY STORED DATA
C
           DO  200  J = J1, J2                                           04NO7
C
           DO  100  I = I1, I2                                           21JA8
C
                CX(I,J)  =  CX(I,J) + CXN                                04NO7
                CY(I,J)  =  CY(I,J) + CYN                                04NO7
C
  100      CONTINUE                                                      21JA8
C
  200      CONTINUE                                                      04NO7
C
  300      CONTINUE                                                      21JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
      RETURN                                                            16JA8
C
C-----PRINT NO DATA STATEMENT
C
  500 PRINT 40                                                          16JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
      RETURN                                                            22JA8
C
      END                                                               22JA8
```

```
      SUBROUTINE QSIN ( Q, S, HX, HY, L1, L2, L3, L4 )                  22JA8
C
C-----THIS SUBROUTINE INPUTS LOAD AND SUPPORT DATA.
C
C
C-----NOTATION FOR SUBROUTINE QSIN
C
C          AREA, AREAX, AREAY  AREA FUNCTIONS COMPUTED FOR CONVENIENCE
C          HT1, HUM, NT, NT1   POSITIONING VARIABLES IN COMMON FIELD
C          HX( )               INCREMENT LENGTH X-BEAM
C          HY( )               INCREMENT LENGTH Y-BEAM
C          I, II1, II2, I1,    COUNTERS AND INDICES IN DO LOOPS
C          I2, J, JJ1, JJ2,
C          J1, J2, N
C          IN1, JN1, IN2, JN2  INPUT STATION NUMBERS
C          KERR                ERROR FLAG
C          L1 --- L4           TRANSFER CONSTANTS
C          MXP3                MX PLUS THREE
C          MYP3                MY PLUS THREE
C          NCT5                NUMBER OF CARDS IN TABLE 5
C          Q( )                LATERAL LOAD
C          QN                  TEMPORARY INPUT CONSTANT
C          QNH                 HALF OF QN
C          QNQ                 QUARTER OF QN
C          QQN                 TEMPORARY INPUT CONSTANT
C          S( )                SUPPORT SPRING
C          SN                  TEMPORARY INPUT CONSTANT
C          SNH                 HALF OF SN
C          SNQ                 QUARTER OF QN
C          SSN                 TEMPORARY INPUT CONSTANT
C
C
      DIMENSION     Q(L1,L2),      S(L1,L2),      HX(L3),         HY(L4) 22JA8
C
      COMMON   KERR, I, II1, II2, IN1, IN2, I1, I2, J, JJ1, JJ2, JN1,   11DE7
     1         JN2, J1, J2, N, AREA, AREAX, AREAY, QN, QNH, QNQ, QQN,   03FE8
     2         SN, SNH, SNQ, SSN, HUM(25), NT(3), NCT5, NT1(7), HT1(2), 03FE8
     3         MYP2, MXP3, MYP3                                         03FE8
C
   10 FORMAT ( 4 ( 2X, I3 ), 4E10.3 )                                   31JA8
   20 FORMAT ( //40H     TABLE 5.  LOAD AND SUPPORT DATA                24JA8
     1          //40H       FROM   THRU       Q          QQ            05FE8
     2            25H       S         SS                               16JA8
     3           /40H        STA    STA      (F)      (F/LL)           05FE8
     4            25H     (F/L)    (F/LLL)    , /  )                   16JA8
   30 FORMAT ( 5X, 2 ( 1X, I2, 1X, I3, ), 4E11.3 )                      16JA8
   40 FORMAT ( //40H          NO DATA INPUT FOR THIS TABLE  , / )        24JA8
C
C
C-----INPUT TABLE 5
C
      PRINT 20                                                          04NO7
C
          IF ( 1 .GT. NCT5 )            GO TO 500                       22JA8
C
```

```
          DO  450  N = 1, NCT5                                                  22JA8
C
      READ  10, IN1, JN1, IN2, JN2, QN, QQN, SN, SSN                            16JA8
      PRINT 30, IN1, JN1, IN2, JN2, QN, QQN, SN, SSN                            16JA8
C
                I1  =  IN1 + 3                                                  16NO7
                I2  =  IN2 + 3                                                  07NO7
                J1  =  JN1 + 3                                                  07NO7
                J2  =  JN2 + 3                                                  07NO7
C
                II1  =  I1 + 1                                                  04NO7
                II2  =  I2 - 1                                                  04NO7
                JJ1  =  J1 + 1                                                  04NO7
                JJ2  =  J2 - 1                                                  04NO7
C
                QNH  =  QN * 0.50                                               03FE8
                SNH  =  SN * 0.50                                               03FE8
                QNQ  =  QN * 0.25                                               03FE8
                SNQ  =  SN * 0.25                                               03FE8
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
          IF ( I1 .GT. I2 )             KERR  =  KERR + 1                     15DE7
          IF ( J1 .GT. J2 )             KERR  =  KERR + 1                     15DE7
          IF ( I1 .LT. 3 )              KERR  =  KERR + 1                     15DE7
          IF ( J1 .LT. 3 )              KERR  =  KERR + 1                     15DE7
          IF ( I2 .GT. MXP3 )           KERR  =  KERR + 1                     15DE7
          IF ( J2 .GT. MYP3 )           KERR  =  KERR + 1                     15DE7
C
          IF ( KERR .GT. 0 )            GO TO 450                             15DE7
C
C-----ADD INPUT DATA FROM THIS CARD TO PREVIOUSLY STORED DATA
C
          IF ( II1 .GT. II2 )           GO TO 210                             04NO7
          IF ( JJ1 .GT. JJ2 )           GO TO 310                             04NO7
C
          DO  200  I = II1, II2                                                 04NO7
C
          DO  100  J = JJ1, JJ2                                                 22JA8
C
               AREA  =  ( HX(I) + HX(I+1) ) * ( HY(J) + HY(J+1) ) * 0.2503NO7
C
               Q(I,J)  =  Q(I,J) + QN + QQN * AREA                              16JA8
               S(I,J)  =  S(I,J) + SN + SSN * AREA                              16JA8
C
  100     CONTINUE                                                              22JA8
C
  200     CONTINUE                                                              04NO7
C
  210     IF ( JJ1 .GT. JJ2 )           GO TO 410                             04NO7
C
          DO  300  J = JJ1, JJ2                                                 04NO7
C
               AREAY  =  ( HY(J) + HY(J+1) ) * 0.25                             07NO7
C
```

```
                AREA   =  AREAY * HX(I1+1)                                  04NO7
C
                Q(I1,J)  =  Q(I1,J) + QNH + QQN * AREA                      03FE8
                S(I1,J)  =  S(I1,J) + SNH + SSN * AREA                      03FE8
C
                AREA   =  AREAY * HX(I2)                                    04NO7
C
                Q(I2,J)  =  Q(I2,J) + QNH + QQN * AREA                      03FE8
                S(I2,J)  =  S(I2,J) + SNH + SSN * AREA                      03FE8
C
  300       CONTINUE                                                        04NO7
C
  310       IF ( II1 .GT. II2 )            GO TO 410                        13DE7
C
            DO  400  I = II1, II2                                           13DE7
C
                AREAX  =  ( HX(I) + HX(I+1) ) * 0.25                        04NO7
C
                AREA   =  AREAX * HY(J1+1)                                  04NO7
C
                Q(I,J1)  =  Q(I,J1) + QNH + QQN * AREA                      03FE8
                S(I,J1)  =  S(I,J1) + SNH + SSN * AREA                      03FE8
C
                AREA   =  AREAX * HY(J2)                                    04NO7
C
                Q(I,J2)  =  Q(I,J2) + QNH + QQN * AREA                      03FE8
                S(I,J2)  =  S(I,J2) + SNH + SSN * AREA                      03FE8
C
  400       CONTINUE                                                        04NO7
C
  410           AREA   =  HX(I1+1) * HY(J1+1) * 0.25                        10NO7
C
                Q(I1,J1)  =  Q(I1,J1) + QNQ + QQN * AREA                    03FE8
                S(I1,J1)  =  S(I1,J1) + SNQ + SSN * AREA                    03FE8
C
                AREA   =  HX(I1+1) * HY(J2) * 0.25                          10NO7
C
                Q(I1,J2)  =  Q(I1,J2) + QNQ + QQN * AREA                    03FE8
                S(I1,J2)  =  S(I1,J2) + SNQ + SSN * AREA                    03FE8
C
                AREA   =  HX(I2) * HY(J1+1) * 0.25                          10NO7
C
                Q(I2,J1)  =  Q(I2,J1) + QNQ + QQN * AREA                    03FE8
                S(I2,J1)  =  S(I2,J1) + SNQ + SSN * AREA                    03FE8
C
                AREA   =  HX(I2) * HY(J2) * 0.25                            10NO7
C
                Q(I2,J2)  =  Q(I2,J2) + QNQ + QQN * AREA                    03FE8
                S(I2,J2)  =  S(I2,J2) + SNQ + SSN * AREA                    03FE8
C
  450       CONTINUE                                                        04NO7
C
C-----RETURN TO SUBROUTINE INPT2
C
            RETURN                                                          22JA8
C
C-----PRINT NO DATA STATEMENT
C
  500 PRINT 40                                                              16JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
            RETURN                                                          22JA8
C
            END                                                             22JA8
```

```
      SUBROUTINE TXPXIN ( TX, PX, HY, L1, L2, L3 )                          22JA8
C
C-----THIS SUBROUTINE INPUTS TORQUES AND AXIAL LOADS IN THE X-DIRECTION.
C
C
C-----NOTATION FOR SUBROUTINE TXPXIN
C
C         HT1, HUM, NT,        POSITIONING VARIABLES IN COMMON FIELD
C         NT1, NUM
C         HY( )                INCREMENT LENGTH Y-BEAM
C         HYN, HYN1            TEMPORARY FUNCTIONS OF HY
C         I, I1, I2, J, JJ1,   COUNTERS AND INDICES IN DO LOOPS
C         JJ2, J1, J2, N
C         IN1, JN1, IN2, JN2   INPUT BAR NUMBERS
C         KERR                 ERROR FLAG
C         L1, L2, L3           TRANSFER CONSTANTS
C         MXP3                 MX PLUS THREE
C         MYP3                 MY PLUS THREE
C         NCT6                 NUMBER OF CARDS IN TABLE 6
C         PPXN                 TEMPORARY INPUT CONSTANT
C         PX( )                AXIAL LOAD X-DIRECTION
C         PXN                  TEMPORARY INPUT CONSTANT
C         PXNH                 HALF OF PX
C         TTXN                 TEMPORARY INPUT CONSTANT
C         TX( )                EXTERNAL COUPLE X-DIRECTION
C         TXN                  TEMPORARY INPUT CONSTANT
C         TXNH                 HALF OF TX
C
C
      DIMENSION     TX(L1,L2),     PX(L1,L2),     HY(L3)                    22JA8
C
      COMMON    KERR, I, IN1, IN2, I1, I2, J, JJ1, JJ2, JN1, JN2, J1,     24JA8
     1          J2, N, NUM(2), TXN, TXNH, TTXN, PXN, PXNH, PPXN, HYN,     03FE8
     2          HYN1, HUM(28), NT(4), NCT6, NT1(6), HT1(2), MYP2, MXP3,   03FE8
     3          MYP3                                                      03FE8
C
   10 FORMAT ( 4 ( 2X, I3 ), 4E10.3 )                                      31JA8
   20 FORMAT ( //40H     TABLE 6.  X-DIRECTIONAL LOAD DATA                 21MR8
     1          //40H       FROM   THRU      TX          TTX               05FE8
     2            25H      PX          PPX                                05FE8
     3           /40H      X-BAR  X-BAR     (FL)       (FL/L)             21MR8
     4            25H     (F)         (F/L)    , // )                     05FE8
   30 FORMAT ( 5X, 2 ( 1X, I2, 1X, I3 ), 4E11.3 )                          22JA8
   40 FORMAT ( //40H           NO DATA INPUT FOR THIS TABLE   , / )        24JA8
C
C
C-----INPUT TABLE 6
C
      PRINT 20                                                             04NO7
C
          IF ( 1 .GT. NCT6 )            GO TO 600                        22JA8
C
          DO  500  N = 1, NCT6                                           22JA8
C
      READ  10, IN1, JN1, IN2, JN2, TXN, TTXN, PXN, PPXN                 22JA8
```

```
      PRINT 30, IN1, JN1, IN2, JN2, TXN, TTXN, PXN, PPXN                22JA8
C
               I1  =  IN1 + 3                                           16NO7
               I2  =  IN2 + 3                                           07NO7
               J1  =  JN1 + 3                                           07NO7
               J2  =  JN2 + 3                                           07NO7
C
               JJ1  =  J1 + 1                                           04NO7
               JJ2  =  J2 - 1                                           04NO7
C
               TXNH  =  TXN * 0.50                                      03FE8
               PXNH  =  PXN * 0.50                                      03FE8
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
          IF ( I1 .GT. I2 )             KERR  =  KERR + 1              15DE7
          IF ( J1 .GT. J2 )             KERR  =  KERR + 1              15DE7
          IF ( I1 .LT. 3 )              KERR  =  KERR + 1              15DE7
          IF ( J1 .LT. 3 )              KERR  =  KERR + 1              15DE7
          IF ( I2 .GT. MXP3 )           KERR  =  KERR + 1              15DE7
          IF ( J2 .GT. MYP3 )           KERR  =  KERR + 1              15DE7
C
          IF ( I1 .GT. 3 )              GO TO 100                      22JA8
C
          IF ( PXN .NE. 0 )             KERR  =  KERR + 1              22JA8
          IF ( PPXN .NE. 0 )            KERR  =  KERR + 1              22JA8
          IF ( I1 .EQ. I2 )             KERR  =  KERR + 1              22JA8
C
  100     IF ( KERR .GT. 0 )            GO TO 500                      07FE8
C
C-----ADD INPUT DATA FROM THIS CARD TO PREVIOUSLY STORED DATA
C
          IF ( JJ1 .GT. JJ2 )           GO TO 310                      07FE8
C
          DO  300  J = JJ1, JJ2                                        22JA8
C
               HYN  =  ( HY(J) + HY(J+1) ) * 0.50                      22JA8
C
          DO  200  I = I1, I2                                          22JA8
C
               TX(I,J)  =  TX(I,J) + TXN + TTXN * HYN                  22JA8
               PX(I,J)  =  PX(I,J) + PXN + PPXN * HYN                  22JA8
C
  200     CONTINUE                                                     22JA8
C
  300     CONTINUE                                                     22JA8
C
  310          HYN  =  HY(J1+1) * 0.50                                 13DE7
               HYN1  =  HY(J2) * 0.50                                  04NO7
C
          DO  400  I = I1, I2                                          22JA8
C
               TX(I,J1)  =  TX(I,J1) + TXNH + TTXN * HYN               03FE8
               PX(I,J1)  =  PX(I,J1) + PXNH + PPXN * HYN               03FE8
C
```

```
                TX(I,J2)  =  TX(I,J2) + TXNH + TTXN * HYN1                03FE8
                PX(I,J2)  =  PX(I,J2) + PXNH + PPXN * HYN1                03FE8
C
  400      CONTINUE                                                       22JA8
C
  500      CONTINUE                                                       22JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
       RETURN                                                            22JA8
C
C-----PRINT NO DATA STATEMENT
C
  600 PRINT 40                                                           22JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
       RETURN                                                            22JA8
C
       END                                                               22JA8
```

```
      SUBROUTINE TYPYIN ( TY, PY, HX, L1, L2, L3 )                      22JA8
C
C-----THIS SUBROUTINE INPUTS TORQUES AND AXIAL LOADS IN THE Y-DIRECTION.
C
C
C-----NOTATION FOR SUBROUTINE TYPYIN
C
C          HT1, HUM, NT,        POSITIONING VARIABLES IN COMMON FIELD
C          NT1, NUM
C          HX( )                INCREMENT LENGTH X-BEAM
C          HXN, HXN1            TEMPORARY FUNCTIONS OF HX
C          I, II1, II2, I1,     COUNTERS AND INDICES IN DO LOOPS
C          I2, J, J1, J2, N
C          IN1, JN1, IN2, JN2   INPUT BAR NUMBERS
C          KERR                 ERROR FLAG
C          L1, L2, L3           TRANSFER CONSTANTS
C          MXP3                 MX PLUS THREE
C          MYP3                 MY PLUS THREE
C          NCT7                 NUMBER OF CARDS IN TABLE 7
C          PPYN                 TEMPORARY INPUT CONSTANT
C          PY( )                AXIAL LOAD Y-DIRECTION
C          PYN                  TEMPORARY INPUT CONSTANT
C          PYNH                 HALF OF PY
C          TTYN                 TEMPORARY INPUT CONSTANT
C          TY( )                EXTERNAL COUPLE Y-DIRECTION
C          TYN                  TEMPORARY INPUT CONSTANT
C          TYNH                 HALF OF TY
C
C
      DIMENSION     TY(L1,L2),     PY(L1,L2),     HX(L3)                 22JA8
C
      COMMON   KERR, I, II1, II2, IN1, IN2, I1, I2, J, JN1, JN2, J1,    24JA8
     1         J2, N, NUM(2), TYN, TYNH, TTYN, PYN, PYNH, PPYN, HXN,    03FE8
     2         HXN1, HUM(28), NT(5), NCT7, NT1(5), HT1(2), MYP2, MXP3,  03FE8
     3         MYP3                                                     03FE8
C
   10 FORMAT ( 4 ( 2X, I3 ), 4E10.3 )                                   31JA8
   20 FORMAT ( //40H      TABLE 7.  Y-DIRECTIONAL LOAD DATA             21MR8
     1         //40H        FROM   THRU      TY          TTY            05FE8
     2           25H      PY          PPY                              05FE8
     3          /40H      Y-BAR  Y-BAR     (FL)      (FL/L)            21MR8
     4           25H     (F)        (F/L)    , // )                    05FE8
   30 FORMAT ( 5X, 2 ( 1X, I2, 1X, I3 ), 4E11.3 )                       22JA8
   40 FORMAT ( //40H           NO DATA INPUT FOR THIS TABLE  , / )       24JA8
C
C
C-----INPUT TABLE 7
C
      PRINT 20                                                          22JA8
C
          IF ( 1 .GT. NCT7 )            GO TO 600                       22JA8
C
          DO  500  N = 1, NCT7                                          22JA8
C
      READ  10, IN1, JN1, IN2, JN2, TYN, TTYN, PYN, PPYN                22JA8
```

```
      PRINT 30, IN1, JN1, IN2, JN2, TYN, TTYN, PYN, PPYN                22JA8
C
                I1  =  IN1 + 3                                          16NO7
                I2  =  IN2 + 3                                          07NO7
                J1  =  JN1 + 3                                          07NO7
                J2  =  JN2 + 3                                          07NO7
C
                III  =  I1 + 1                                          04NO7
                II2  =  I2 - 1                                          04NO7
C
                TYNH  =  TYN * 0.50                                     03FE8
                PYNH  =  PYN * 0.50                                     03FE8
C
C-----CHECK FOR IMPROPER PROBLEM INPUT
C
          IF ( I1 .GT. I2 )             KERR  =  KERR + 1               15DE7
          IF ( J1 .GT. J2 )             KERR  =  KERR + 1               15DE7
          IF ( I1 .LT. 3 )              KERR  =  KERR + 1               15DE7
          IF ( J1 .LT. 3 )              KERR  =  KERR + 1               15DE7
          IF ( I2 .GT. MXP3 )           KERR  =  KERR + 1               15DE7
          IF ( J2 .GT. MYP3 )           KERR  =  KERR + 1               15DE7
C
          IF ( J1 .GT. 3 )              GO TO 100                       22JA8
C
          IF ( PYN .NE. 0 )             KERR  =  KERR + 1               22JA8
          IF ( PPYN .NE. 0 )            KERR  =  KERR + 1               22JA8
          IF ( J1 .EQ. J2 )             KERR  =  KERR + 1               22JA8
C
  100     IF ( KERR .GT. 0 )            GO TO 500                       07FE8
C
C-----ADD INPUT DATA FROM THIS CARD TO PREVIOUSLY STORED DATA
C
          IF ( III .GT. II2 )           GO TO 310                       07FE8
C
          DO  300  I = III, II2                                         22JA8
C
               HXN  =  ( HX(I) + HX(I+1) ) * 0.50                       04NO7
C
          DO  200  J = J1, J2                                           22JA8
C
               TY(I,J)  =  TY(I,J) + TYN + TTYN * HXN                   22JA8
               PY(I,J)  =  PY(I,J) + PYN + PPYN * HXN                   22JA8
C
  200     CONTINUE                                                      22JA8
C
  300     CONTINUE                                                      22JA8
C
  310          HXN  =  HX(I1+1) * 0.50                                  04NO7
               HXN1  =  HX(I2) * 0.50                                   04NO7
C
          DO  400  J = J1, J2                                           22JA8
C
               TY(I1,J)  =  TY(I1,J) + TYNH + TTYN * HXN                03FE8
               PY(I1,J)  =  PY(I1,J) + PYNH + PPYN * HXN                03F58
C
```

```
                TY(I2,J)  =  TY(I2,J) + TYNH + TTYN * HXN1                03FE8
                PY(I2,J)  =  PY(I2,J) + PYNH + PPYN * HXN1                03FE8
C
  400      CONTINUE                                                       22JA8
C
  500      CONTINUE                                                       22JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
      RETURN                                                             22JA8
C
C-----PRINT NO DATA STATEMENT
C
  600 PRINT 40                                                           22JA8
C
C-----RETURN TO SUBROUTINE INPT2
C
      RETURN                                                             22JA8
C
      END                                                                22JA8
```

```
      SUBROUTINE BOUND ( DX, DY, S, L1, L2 )                                11DE7
C
C-----THIS SUBROUTINE PLACES SPRINGS AT POINTS WHERE AA3(I,3) WOULD
C          BE CALCULATED TO BE ZERO WITHOUT THE SPRING.  THIS ALLOWS
C          THE SOLUTION OF NON-RECTANGULAR SLABS AND SLABS WITH HOLES
C          IN THEM.
C
C
C-----NOTATION FOR SUBROUTINE BOUND
C
C          DX( )                 BENDING STIFFNESS X-DIRECTION
C          DY( )                 BENDING STIFFNESS Y-DIRECTION
C          HT1, HUM, MXP,        POSITIONING VARIABLES IN COMMON FIELD
C          NT1, NUM
C          I, J                  COUNTERS OF DO LOOPS
C          KERR                  ERROR FLAG
C          L1, L2                TRANSFER CONSTANTS
C          MXP4                  MX PLUS FOUR
C          MYP4                  MY PLUS FOUR
C          S( )                  SUPPORT SPRING
C          SUM                   SUM OF DX AND DY
C
C
      DIMENSION     DX(L1,L2),     DY(L1,L2),     S(L1,L2)                  22JA8
C
      COMMON   KERR, I, J, NUM(13), SUM, HUM(35), NT1(11), HT1(2),          24JA8
     1         MXP(3), MXP4, MYP4                                           22JA8
C
C
C-----RETURN IF KERR IS NOT 0
C
          IF ( KERR .GT. 0 )            RETURN                            22JA8
C
C-----FIND POINTS WHERE AA3(I,3) HAS ALL ZERO VALUES FOR DX AND DY
C
          DO  400  J = 2, MYP4                                            04NO7
C
          DO  300  I = 2, MXP4                                            04NO7
C
               SUM = DX(I-1,J) + DX(I,J) + DX(I+1,J)                      21JL7
     1             + DY(I,J-1) + DY(I,J) + DY(I,J+1)                      21JL7
C
          IF ( SUM .EQ. 0.0 )           S(I,J) = 1.0E+20                  04NO7
C
  300     CONTINUE                                                        04NO7
C
  400     CONTINUE                                                        04NO7
C
C-----RETURN TO SUBROUTINE INPT2
C
      RETURN                                                              04NO7
C
      END                                                                 04NO7
```

```
      SUBROUTINE STIFF ( DX, DY, Q, S, CX, CY, HX, HY, TX, TY, PX, PY,  10NO7
     1                   L1, L2, L3, L4 )                               22JA8
C
C-----THIS SUBROUTINE PRINTS OUT THE ARRAYS USED FOR INPUT VARIBLES.
C
C
C-----NOATION FOR SUBROUTINE STIFF
C
C          CX( )               TWISTING STIFFNESS X-DIRECTION
C          CY( )               TWISTING STIFFNESS Y-DIRECTION
C          DX( )               BENDING STIFFNESS X-DIRECTION
C          DY( )               BENDING STIFFNESS Y-DIRECTION
C          HX( )               INCREMENT LENGTH X-BEAM
C          HY( )               INCREMENT LENGTH Y-BEAM
C          I, J                COUNTERS OF DO LOOPS
C          ISTA, JSTA          EXTERNAL STATION NUMBER
C          L1 --- L4           TRANSFER CONSTANTS
C          PX( )               AXIAL LOAD X-DIRECTION
C          PY( )               AXIAL LOAD Y-DIRECTION
C          Q( )                LATERAL LOAD
C          S( )                SUPPORT SPRING
C          TX( )               EXTERNAL COUPLE X-DIRECTION
C          TY( )               EXTERNAL COUPLE Y-DIRECTION
C
C
      DIMENSION     DX(L1,L2),     DY(L1,L2),     Q(L1,L2),              22JA8
     1              S(L1,L2),      CX(L1,L2),     CY(L1,L2),             22JA8
     2              TX(L1,L2),     TY(L1,L2),     PX(L1,L2),             22JA8
     3              PY(L1,L2),     HX(L3),        HY(L4)                 22JA8
C
      COMMON   I, ISTA, J, JSTA                                           12DE7
C
  101 FORMAT  (//51H     TABLE XX. STIFFNESS AND LOAD VALUES USED DURIN,10NO7
     1          35HG CALCULATION OF RESULTS                      , /     30AG6
     2         / 50H         I       J        DX          DY          Q   05FE8
     3           45H          S          CX          CY                ,/) 05FE8
  102 FORMAT  (//51H     TABLE XX(CONTD). STIFFNESS AND LOAD VALUES US,10NO7
     1          35HED DURING CALCULATION OF RESULTS              , /     29AG6
     2         / 50H         I       J                              TX   05FE8
     3           45H         TY          PX          PY                ,/) 05FE8
  103 FORMAT ( )                                                          29AG6
  104 FORMAT  ( 4X,  2( 2X, I4, 2X ),      6E11.3)                        10NO7
  105 FORMAT  ( 4X,  2( 2X, I4, 2X ), 22X, 4E11.3)                        10NO7
  106 FORMAT  (//51H     TABLE YY.  INCREMENT LENGTH USED IN CALCULATIO,10NO7
     1          15HN OF RESULTS   , /                                    10NO7
     2         / 40H         I       J        HX          HY   , / )     05FE8
  107 FORMAT ( 6X, I4, 10X, E11.3 )                                       10NO7
  108 FORMAT ( 16X, I4, 11X, E11.3 )                                      10NO7
C
C
C-----PRINT DX, DY, Q, S, CX, AND CY
C
      PRINT 101                                                           10NO7
C
          DO  1360  J = 1, L2                                             24JA8
```

```
C
      PRINT 103                                                          10NO7
C
               JSTA = J - 3                                              10NO7
C
          DO  1350  I = 1, L1                                            24JA8
C
               ISTA = I - 3                                              10NO7
C
      PRINT 104, ISTA, JSTA, DX(I,J), DY(I,J), Q(I,J), S(I,J), CX(I,J),  10NO7
     1          CY(I,J)                                                  10NO7
C
 1350     CONTINUE                                                       10NO7
C
 1360     CONTINUE                                                       10NO7
C
      PRINT 103                                                          10NO7
      PRINT 103                                                          10NO7
C
C-----PRINT TX, TY, PX, AND PY
C
      PRINT 102                                                          10NO7
C
          DO  1380  J = 1, L2                                            24JA8
C
      PRINT 103                                                          10NO7
C
               JSTA = J - 3                                              10NO7
C
          DO  1370  I = 1, L1                                            24JA8
C
               ISTA = I - 3                                              10NO7
C
      PRINT 105, ISTA, JSTA,                TX(I,J), TY(I,J),            10NO7
     1          PX(I,J), PY(I,J)                                         10NO7
C
 1370     CONTINUE                                                       10NO7
C
 1380     CONTINUE                                                       10NO7
C
      PRINT 103                                                          10NO7
      PRINT 103                                                          10NO7
C
C-----PRINT HX AND HY
C
      PRINT 106                                                          10NO7
C
      PRINT 103                                                          13NO7
C
          DO  1500  I = 1, L3                                            24JA8
C
               ISTA  =  I - 3                                            10NO7
C
      PRINT 107, ISTA, HX(I)                                             10NO7
C
```

```
 1500     CONTINUE                                                  10NO7
C
      PRINT 103                                                     13NO7
      PRINT 103                                                     05FE8
C
          DO  1600  J = 1, L4                                       24JA8
C
               JSTA  =  J - 3                                       10NO7
C
      PRINT 108, JSTA, HY(J)                                        10NO7
C
 1600     CONTINUE                                                  10NO7
C
C-----RETURN TO SUBROUTINE V03SUB
C
      RETURN                                                        10NO7
C
      END                                                           10NO7
```

```
      SUBROUTINE QMAT ( AA1, AA6, HX, HY, DY, Q, TX, TY, L1, L2, L3,     12DE7
     1                  L4, L5 )                                         12DE7
C
C-----THIS SUBROUTINE DEFINES THE Q VECTOR AND THE AA1 MATRIX.
C
C
C-----NOTATION FOR SUBROUTINE QMAT
C
C         AA1( )              SUBMATRIX OF STIFFNESS MATRIX
C         DY( )               BENDING STIFFNESS Y-DIRECTION
C         AA6( )              SUBMATRIX OF LOAD MATRIX
C         HT1, HUM, MXP       POSITIONING VARIABLES IN COMMON FIELD
C         NT1, NUM
C         HX( )               INCREMENT LENGTH X-BEAM
C         HXAV, HXI, HX1I     TEMPORARY FUNCTIONS OF HX
C         HXOHY               TEMPORARY FUNCTION OF HX AND HY
C         HY( )               INCREMENT LENGTH Y-BEAM
C         HYAVM1, HYI, HYQ1,  TEMPORARY FUNCTIONS OF HY
C         HY1I
C         I, II, J            COUNTERS IN DO LOOPS
C         L1 --- L5           TRANSFER CONSTANTS
C         MXP4                MX PLUS FOUR
C         Q( )                LATERAL LOAD
C         TX( )               EXTERNAL COUPLE X-DIRECTION
C         TY( )               EXTERNAL COUPLE Y-DIRECTION
C
C
      DIMENSION     AA1(L1),        AA6(L1),        DY(L2,L3),           05DE7
     1              HX(L4),         HY(L5),         Q(L2,L3),            05DE7
     2              TX(L2,L3),      TY(L2,L3)                            05DE7
C
      COMMON   I, II, K, J, NUM(12), HXAV, HXI, HXOHY, HX1I, HYAVM1,     22JA8
     1         HYI, HYQ1, HY1I, HUM(28), NT1(11), HT1(2), MXP(3), MXP4   22JA8
C
C-----SET CONSTANTS
C
               HYI     =  1.0 / HY(J)                                    05DE7
               HY1I    =  1.0 / HY(J+1)                                  05DE7
               HYAVM1  =  0.5 * ( HY(J-1) + HY(J) )                      05DE7
               HYQ1    =  1.0 / ( HYAVM1 * HY(J-1) )                     13DE7
C
          DO  500  I = 2, MXP4                                           05DE7
C
               HXAV    =  0.5 * ( HX(I) + HX(I+1) )                      09JA8
               HXOHY   =  HXAV * HYI                                     05DE7
               HXI     =  1.0 / HX(I)                                    05DE7
               HX1I    =  1.0 / HX(I+1)                                  05DE7
C
C-----SET INDEX
C
               II  =  I - 1                                              05DE7
C
C-----DEFINE AN ELEMENT OF THE AA1 MATRIX
C
               AA1(II)  =  DY(I,J-1) * HXOHY * HYQ1                      05DE7
```

```
C
C-----DEFINE AN ELEMENT OF THE Q VECTOR                                    05DE7
C
                AA6(II)   =  Q(I,J) - TX(I,J) * HXI + TX(I+1,J) * HX1I     05DE7
     1                       - TY(I,J) * HYI + TY(I,J+1) * HY1I            05DE7
C
  500      CONTINUE                                                        05DE7
C
C-----RETURN TO SUBROUTINE V03SUB
C
       RETURN                                                             05DE7
C
       END                                                                05DE7
```

```
      SUBROUTINE SUBMAT ( AA2, AA3, AA4, AA5, HX, HY, DX, DY, S, CX,     05JA8
     1                    CY, PX, PY, L1, L2, L3, L4, L5 )               05JA8
C
C-----THIS SUBROUTINE DEFINES THE AA2, AA3, AA4, AND AA5 MATRICES.
C
C
C-----NOTATION FOR SUBROUTINE SUBMAT
C
C          AA2( ) --- AA5( )    SUBMATRICES OF STIFFNESS MATRIX
C          CX( )                TWISTING STIFFNESS X-DIRECTION
C          CY( )                TWISTING STIFFNESS Y-DIRECTION
C          DX( )                BENDING STIFFNESS X-DIRECTION
C          DY( )                BENDING STIFFNESS Y-DIRECTION
C          HX( )                INCREMENT LENGTH X-BEAM
C          HXAV, HXAVM1,        TEMPORARY FUNCTIONS OF HX
C          HXAVP1, HXI, HXQ1,
C          HXQ2, HXQ3, HXQ4,
C          HXQ5, HXQ6, HXQ7,
C          HXQ8, HXQ9, HX1I
C          HXOHY, HXOHY1,       TEMPORARY FUNCTIONS OF HX AND HY
C          HXYQ1, HXYQ2, HXYQ3,
C          HXYQ4, HYOHX, HYOHX1
C          HY( )                INCREMENT LENGTH Y-BEAM
C          HYAV, HYAVM1,        TEMPORARY FUNCTIONS OF HY
C          HYQ2, HYQ3, HYQ4,
C          HYQ5, HYQ6, HYQ7,
C          HYQ8, HYQ9, HY1I
C          I, II, J             COUNTERS IN DO LOOPS
C          L1 --- L5            TRANSFER CONSTANTS
C          MXP3                 MX PLUS THREE
C          MXP4                 MX PLUS FOUR
C          NT1, NUM             POSITIONING VARIABLES IN COMMON FIELD
C          PR                   POISSONS RATIO
C          PX( )                AXIAL LOAD X-DIRECTION
C          PY( )                AXIAL LOAD Y-DIRECTION
C          S( )                 SUPPORT SPRING
C
C
      DIMENSION     AA2(L1,3),          AA3(L1,5),          AA4(L1,3),   12DE7
     1              AA5(L1),            HX(L4),             HY(L5),      12DE7
     2              DX(L2,L3),          DY(L2,L3),          S(L2,L3),    05JA8
     3              CX(L2,L3),          CY(L2,L3),          PX(L2,L3),   05JA8
     4              PY(L2,L3)                                            05JA8
C
      COMMON   I, II, K, J, NUM(12), HXAV, HXAVM1, HXAVP1, HXI, HXOHY,  22JA8
     1         HXOHY1, HXQ1, HXQ2, HXQ3, HXQ4, HXQ5, HXQ6, HXQ7, HXQ8,  22JA8
     2         HXQ9, HXYQ1, HXYQ2, HXYQ3, HXYQ4, HX1I, HYAV, HYAVM1,    22JA8
     3         HYAVP1, HYI, HYOHX, HYOHX1, HYQ1, HYQ2, HYQ3, HYQ4, HYQ5,22JA8
     4         HYQ6, HYQ7, HYQ8, HYQ9, HY1I, NT1(11), PR, THK, MYP2,    22JA8
     5         MXP3, MYP3, MXP4, MYP4                                   22JA8
C
C
C-----SET CONSTANTS
C
               HYI     =  1.0 / HY(J)                                   18AG7
```

```
                HY1I    =  1.0 / HY(J+1)                                    18AG7
                HYAVM1  =  0.5 * ( HY(J-1) + HY(J) )                        16AG7
                HYAV    =  0.5 * ( HY(J) + HY(J+1) )                        16AG7
                HYAVP1  =  0.5 * ( HY(J+1) + HY(J+2) )                      16AG7
                HYQ1    =  1.0 / ( HYAVM1 * HY(J-1) )                       18AG7
                HYQ2    =  1.0 / ( HY(J-1) * HY(J) )                        18AG7
                HYQ3    =  1.0 / ( HYAVM1 * HY(J) )                         07SE7
                HYQ4    =  1.0 / ( HYAV * HY(J) )                           07SE7
                HYQ5    =  HYI * HY1I                                       07SE7
                HYQ6    =  1.0 / ( HYAV * HY(J+1) )                         07SE7
                HYQ7    =  1.0 / ( HYAVP1 * HY(J+1) )                       07SE7
                HYQ8    =  1.0 / ( HY(J+1) * HY(J+2) )                      18AG7
                HYQ9    =  1.0 / ( HYAVP1 * HY(J+2) )                       07SE7
C
          DO  500  I = 2, MXP4                                              24JA8
C
                HXI     =  1.0 / HX(I)                                      18AG7
                HX1I    =  1.0 / HX(I+1)                                    18AG7
                HXAVM1  =  0.5 * ( HX(I-1) + HX(I) )                        16AG7
                HXAV    =  0.5 * ( HX(I) + HX(I+1) )                        16AG7
                HXAVP1  =  0.5 * ( HX(I+1) + HX(I+2) )                      16AG7
                HXQ1    =  1.0 / ( HXAVM1 * HX(I-1) )                       07SE7
                HXQ2    =  1.0 / ( HX(I-1) * HX(I) )                        07SE7
                HXQ3    =  1.0 / ( HXAVM1 * HX(I) )                         07SE7
                HXQ4    =  1.0 / ( HXAV * HX(I) )                           07SE7
                HXQ5    =  HXI * HX1I                                       07SE7
                HXQ6    =  1.0 / ( HXAV * HX(I+1) )                         07SE7
                HXQ7    =  1.0 / ( HXAVP1 * HX(I+1) )                       07SE7
                HXQ8    =  1.0 / ( HX(I+1) * HX(I+2) )                      18AG7
                HXOHY   =  HXAV * HYI                                       17AG7
                HXQ9    =  1.0 / ( HXAVP1 * HX(I+2) )                       07SE7
                HXOHY1  =  HXAV * HY1I                                      17AG7
                HYOHX   =  HYAV * HXI                                       17AG7
                HYOHX1  =  HYAV * HX1I                                      17AG7
                HXYQ1   =  HXI * HYI                                        19AG7
                HXYQ2   =  HX1I * HYI                                       19AG7
                HXYQ3   =  HXI * HY1I                                       19AG7
                HXYQ4   =  HX1I * HY1I                                      19AG7
C
C-----SET INDEX
C
                II = I - 1                                                  12OC6
C
C-----DEFINE THE MAIN DIAGONAL OF THE AA2, AA3, AA4, AND AA5 MATRICES
C
                AA2(II,2)  =   - HXOHY * ( 2.0 * DY(I,J-1) * HYQ2 +         18AG7
     1                         DY(I,J) * ( HYQ4 + HYQ6 ) ) - PR * ( 2.0    07SE7
     2                         * DY(I,J-1) * HXOHY * HXQ5 + DX(I,J) * (    07SE7
     3                         HXYQ1 + HXYQ2 ) ) - ( CX(I,J) + CY(I,J) )   18AG7
     4                         * HXYQ1 - ( CX(I+1,J) + CY(I+1,J) ) *       18AG7
     5                         HXYQ2 - PY(I,J) * HYI                       18AG7
C
                AA3(II,3)  =   HYOHX * ( DX(I-1,J) * HXQ3 + DX(I,J) *       07SE7
     1                         2.0 * HXQ5 ) + HYOHX1 * ( DX(I,J) * 2.0     07SE7
     2                         * HXQ5 + DX(I+1,J) * HXQ7 ) + HXOHY * (     07SE7
```

```
     3                          DY(I,J-1) * HYQ3 + DY(I,J) * 2.0 * HYQ5 ) 07SE7
     4                          + HXOHY1 * ( DY(I,J) * 2.0 * HYQ5 +       07SE7
     5                          DY(I,J+1) * HYQ7 ) + PR * ( DX(I,J) * 2.0 07SE7
     6                          * HYQ5 * ( HYOHX + HYOHX1 ) + DY(I,J) *   07SE7
     7                          2.0 * HXQ5 * ( HXOHY + HXOHY1 ) ) + (     07SE7
     8                          CX(I,J) + CY(I,J) ) * HXYQ1 + ( CX(I,J+1) 18AG7
     9                          + CY(I,J+1) ) * HXYQ3 + ( CX(I+1,J) +     18AG7
     1                          CY(I+1,J) ) * HXYQ2 + ( CX(I+1,J+1) +     18AG7
     2                          CY(I+1,J+1) ) * HXYQ4  + PX(I,J) * HXI +  18AG7
     3                          PX(I+1,J) * HX1I + PY(I,J) * HYI +        18AG7
     4                          PY(I,J+1) * HY1I + S(I,J)                 18AG7
C
               AA4(II,2)  =    - HXOHY1 * ( DY(I,J) * ( HYQ4 + HYQ6 ) +  08SE7
     1                         DY(I,J+1) * 2.0 * HYQ8 ) - PR * ( DX(I,J) 11SE7
     2                         * ( HXYQ3 + HXYQ4 ) + 2.0 * DY(I,J+1) *   11SE7
     3                         HXOHY1 * HXQ5 ) - ( CX(I,J+1) + CY(I,J+1) 08SE7
     4                         ) * HXYQ3 - ( CX(I+1,J+1) + CY(I+1,J+1) ) 08SE7
     5                         * HXYQ4 - PY(I,J+1) * HY1I                08SE7
C
               AA5(II)    =    HXOHY1 * DY(I,J+1) * HYQ9                 02MR8
C
C-----CHECK FOR EXISTENCE OF THE FIRST LOWER DIAGONAL
C
          IF ( II - 1 )                 410, 410, 405                     24JA8
C
C-----DEFINE THE FIRST LOWER DIAGONAL OF THE AA2, AA3, AND AA4 MATRICES
C
  405          AA2(II,1)  =    HXYQ1 * ( PR * ( DX(I-1,J) + DY(I,J-1) )  18AG7
     1                         + CX(I,J) + CY(I,J) )                     18AG7
C
               AA3(II,2)  =    - HYOHX * ( DX(I-1,J) * 2.0 * HXQ2 +      07SE7
     1                         DX(I,J) * ( HXQ4 + HXQ6 ) ) - PR * (      07SE7
     2                         HYOHX * DX(I-1,J) * 2.0 * HYQ5 + DY(I,J)  07SE7
     3                         * ( HXYQ3 + HXYQ1 ) ) - ( CX(I,J+1) +     18AG7
     4                         CY(I,J+1) ) * HXYQ3 - ( CX(I,J) +         18AG7
     5                         CY(I,J) ) * HXYQ1 - PX(I,J) * HXI         18AG7
C
               AA4(II,1)  =    HXYQ3 * ( PR * ( DX(I-1,J) + DY(I,J+1) )  18AG7
     1                         + CX(I,J+1) + CY(I,J+1) )                 18AG7
C
C-----CHECK FOR EXISTENCE OF THE SECOND LOWER DIAGONAL
C
  410      IF ( II - 2 )                 430, 430, 420                     24JA8
C
C-----DEFINE THE SECOND LOWER DIAGONAL OF THE AA3 MATRIX
C
  420          AA3(II,1)  =    HYOHX * DX(I-1,J) * HXQ1                  07SE7
C
C-----CHECK FOR EXISTENCE OF THE FIRST UPPER DIAGONAL
C
  430      IF ( II - MXP3 )              440, 450, 450                     24JA8
C
C-----DEFINE THE FIRST UPPER DIAGONAL OF THE AA2, AA3, AND AA4 MATRICES
C
  440          AA2(II,3)  =    HXYQ2 * ( PR * ( DX(I+1,J) + DY(I,J-1) )  28AG7
```

```
     1                          + CX(I+1,J) + CY(I+1,J) )                  18AG7
C
               AA3(II,4)  =    - HYOHX1 * ( DX(I,J) * ( HXQ4 + HXQ6 ) +   08SE7
     1                         DX(I+1,J) * 2.0 * HXQ8 ) - PR * ( DY(I,J) 18AG7
     2                         * ( HXYQ2 + HXYQ4 ) + DX(I+1,J) * 2.0 *    11SE7
     3                         HYOHX1 * HYQ5 ) - ( CX(I+1,J+1) +          07SE7
     4                         CY(I+1,J+1) ) * HXYQ4 - ( CX(I+1,J) +      28AG7
     5                         CY(I+1,J) ) * HXYQ2 - PX(I+1,J) * HX1I     18AG7
C
               AA4(II,3)  =    HXYQ4 * ( PR * ( DX(I+1,J) + DY(I,J+1) )   18AG7
     1                         + CX(I+1,J+1) + CY(I+1,J+1) )              18AG7
C
C-----CHECK FOR EXISTENCE OF THE SECOND UPPER DIAGONAL
C
  450     IF ( I - MXP3 )               460, 500, 500                     05FE8
C
C-----DEFINE THE SECOND UPPER DIAGONAL OF THE AA3 MATRIX
C
  460           AA3(II,5)  =    HYOHX1 * DX(I+1,J) * HXQ9                  18AG7
C
C
  500     CONTINUE                                                        04MY7
C
C-----RETURN TO SUBROUTINF V03SUB
C
      RETURN                                                              07NO7
C
      END                                                                 07NO7
```

```
      SUBROUTINE MATRIX ( L1, AA, AA1, AA2, AA3, AA4, AA5, AA6, A1, A2, 31JA8
     1                    BB, BB1, BB2, CC, CC1, CC2, D, E )            31JA8
C
C-----THIS SUBROUTINE SOLVES FOR THE RECURSION COEFFICIENTS.
C
C
C-----NOTATION FOR SUBROUTINE MATRIX
C
C          AA( )                RECURSION COEFFICIENT
C          AA1( ) --- AA6( )    SUBMATRICES OF STIFFNESS MATRIX
C          A1( ), A2( )         TEMPORARY AA( ) TERMS
C          BB( )                RECURSION COEFFICIENT
C          BB1( ), BB2( )       TEMPORARY BB( ) TERMS
C          CC( )                RECURSION COEFFICIENT
C          CC1( ), CC2( )       TEMPORARY CC( ) TERMS
C          D( )                 RECURSION COEFFICIENT
C          E( )                 RECURSION COEFFICIENT
C          L1                   TRANSFER CONSTANT
C          HT1, HUM, NT         POSITIONING VARIABLES IN COMMON FIELD
C          NT1, NUM
C          I, K                 COUNTERS IN DO LOOPS
C          KERR                 ERROR FLAG
C          ML                   MULTIPLE LOAD OPTION
C          MXP3                 MX PLUS THREE
C
C
      DIMENSION      AA(L1,1),       AA1(L1),        AA2(L1,3),          31JA8
     1               AA3(L1,5),      AA4(L1,3),      AA5(L1),            16JA8
     2               AA6(L1),        A1(L1,1),       A2(L1,1),           16JA8
     3               BB(L1,L1),      BB1(L1,L1),     BB2(L1,L1),         16JA8
     4               CC(L1,L1),      CC1(L1,L1),     CC2(L1,L1),         16JA8
     5               D(L1,L1),       E(L1,L1)                            16JA8
C
      COMMON    KERR, I, K, NUM(13), HUM(36), NT1(8), ML, NT(2), HT1(2), 31JA8
     1          MYP2, MXP3                                               31JA8
C
C
C-----CHECK PROBLEM TYPE
C
          IF ( ML )                       500, 520, 520                 24JA8
C
C-----READ RECURSION COEFFICIENTS D AND E, THEN JUMP TO 550
C
  500 READ ( 1 ) ( ( D(I,K), E(I,K), I = 1, MXP3 ), K = 1, MXP3 )       05DE7
C
          GO TO 550                                                     05DE7
C
C-----MULTIPLY MATRIX AA1 BY MATRIX BB2 TO OBTAIN MATRIX E
C
  520 CALL MATMY1 ( L1, MXP3, AA1, BB2, E )                             12DE7
C
C-----ADD MATRIX AA2 TO MATRIX E
C
      CALL MATA1 ( L1, E, AA2 )                                         26JA8
C
```

```
C-----MULTIPLY MATRIX E BY MATRIX BB1 TO OBTAIN MATRIX D
C
      CALL MATMPY ( L1, MXP3, E, BB1, D )                                  12DE7
C
C-----MULTIPLY MATRIX AA1 BY MATRIX CC2 TO OBTAIN MATRIX CC
C
      CALL MATMY1 ( L1, MXP3, AA1, CC2, CC )                               12DE7
C
C-----ADD MATRIX CC TO MATRIX D, THEN RETURN THE NEGATIVE OF MATRIX D
C
      CALL MATNA1 ( L1, D, CC )                                            24JA8
C
C-----SUBTRACT MATRIX AA3 FROM MATRIX D
C
      CALL MATS2 ( L1, D, AA3 )                                            31JA8
C
C-----FIND THE INVERSE OF MATRIX D
C
      CALL INVR4 ( L1, D )                                                 16JA8
C
C-----CHECK FOR NO INVERSE
C
          IF ( KERR .GT. 0 )            GO TO 600                        22JA8
C
C-----MULTIPLY MATRIX D BY MATRIX AA5 TO OBTAIN MATRIX CC
C
      CALL MATM2 ( L1, D, AA5, CC )                                        05FE8
C
C-----MULTIPLY MATRIX E BY MATRIX CC1 TO OBTAIN MATRIX BB2
C
      CALL MATMPY ( L1, MXP3, E, CC1, BB2 )                                24JA8
C
C-----ADD MATRIX AA4 TO MATRIX BB2
C
      CALL MATA1 ( L1, BB2, AA4 )                                          24JA8
C
C-----MULTIPLY MATRIX D BY MATRIX BB2 TO OBTAIN MATRIX BB
C
      CALL MATMPY ( L1, MXP3, D, BB2, BB )                                 16JA8
C
C-----MULTIPLY MATRIX E BY MATRIX A1 TO OBTAIN MATRIX AA
C
  550 CALL MATMPY ( L1, 1, E, A1, AA )                                     12DE7
C
C-----MULTIPLY MATRIX AA1 BY MATRIX A2 TO OBTAIN A NEW MATRIX A2
C
      CALL MATMY1 ( L1, 1, AA1, A2, A2 )                                   03FE8
C
C-----ADD MATRIX AA TO MATRIX A2
C
      CALL MATA2 ( L1, A2, AA )                                            03FE8
C
C-----SUBTRACT MATRIX AA6 FROM MATRIX A2
C
      CALL MATS3 ( L1, A2, AA6 )                                           03FE8
```

```
C
C-----MULTIPLY MATRIX D BY MATRIX A2 TO OBTAIN MATRIX AA
C
      CALL MATMPY ( L1, 1, D, A2, AA )                                  03FE8
C
C-----SET ERROR CHECK
C
                KERR  =  0                                              23JA8
C
C-----RETURN TO SUBROUTINE V03SUB
C
  600 RETURN                                                            08AG7
C
      END                                                               08AG7
```

```
      SUBROUTINE MATMY1 ( M1, L, X, Y, Z )                                  12DE7
C
C-----THIS SUBROUTINE MULTIPLIES A DIAGONAL MATRIX BY A FULL MATRIX.
C
C
C-----NOTATION FOR SUBROUTINE MATMY1
C
C         HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C         I, J                COUNTERS IN DO LOOPS
C         L, M1               TRANSFER CONSTANTS
C         L2                  MX PLUS THREE
C         X( )                DIAGONAL MATRIX
C         Y( )                FULL MATRIX
C         Z( )                PRODUCT MATRIX
C
C
      DIMENSION      X(M1),          Z(M1,L),      Y(M1,L)                  22JA8
C
      COMMON    I, J, NUM(14), HUM(36), NT1(11), HT1(2), MYP2, L2           22JA8
C
C
          DO  200  I = 1, L2                                                24JA8
C
          DO  100  J = 1, L                                                 24JA8
C
               Z(I,J) = X(I) * Y(I,J)                                       13JL7
C
  100     CONTINUE                                                          13JL7
C
  200     CONTINUE                                                          14NO7
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                                13JL7
C
      END                                                                   13JL7
```

```
      SUBROUTINE MATA1 ( M1, Y, X3 )                                    24JA8
C
C-----THIS SUBROUTINE ADDS A FULL MATRIX AND A TRI-DIAGONAL MATRIX.
C
C
C-----NOTATION FOR SUBROUTINE MATA1
C
C          HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C          I                   COUNTER IN DO LOOP
C          L2                  MX PLUS THREE
C          MM1                 MX PLUS TWO
C          M1                  TRANSFER CONSTANT
C          X3( )               TRI-DIAGONAL MATRIX
C          Y( )                FULL MATRIX
C
C
      DIMENSION      X3(M1,3),       Y(M1,M1)                           24JA8
C
      COMMON    I, MM1, NUM(14), HUM(36), NT1(11), HT1(2), MYP2, L2     24JA8
C
C
           MM1 = L2 - 1                                                 25JL7
C
          DO  300  I = 2, MM1                                           22JA8
C
           Y(I,I-1) = Y(I,I-1) + X3(I,1)                                13JL7
           Y(I,I) = Y(I,I) + X3(I,2)                                    13JL7
           Y(I,I+1) = Y(I,I+1) + X3(I,3)                                13JL7
C
  300     CONTINUE                                                      22JA8
C
           Y(1,1) = Y(1,1) + X3(1,2)                                    13JL7
           Y(1,2) = Y(1,2) + X3(1,3)                                    13JL7
C
           Y(L2,L2-1) = Y(L2,L2-1) + X3(L2,1)                           25JL7
           Y(L2,L2) = Y(L2,L2) + X3(L2,2)                               25JL7
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                            13JL7
C
      END                                                               13JL7
```

```
      SUBROUTINE MATMPY ( M1, L, X, Y, Z )                                  12DE7
C
C-----THIS SUBROUTINE MULTIPLIES TWO MATRICES.
C
C
C-----NOTATION FOR SUBROUTINE MATMPY
C
C         HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C         I, K, M             COUNTERS IN DO LOOPS
C         L, M1               TRANSFER CONSTANTS
C         L2                  MX PLUS THREE
C         X( )                FULL MATRIX
C         Y( )                MULTIPLIER MATRIX
C         Z( )                PRODUCT MATRIX
C
C
      DIMENSION     X(M1,M1),       Y(M1,L),        Z(M1,L)                 22JA8
C
      COMMON   I, M, K, NUM(13), HUM(36), NT1(11), HT1(2), MYP2, L2         31JA8
C
C
          DO  300  I = 1, L2                                                24JA8
C
          DO  200  M = 1, L                                                 24JA8
C
               Z(I,M) = 0.0                                                 27JE7
C
          DO  100  K = 1, L2                                                24JA8
C
               Z(I,M) = X(I,K) * Y(K,M) + Z(I,M)                            27JE7
C
  100     CONTINUE                                                          17NO7
C
  200     CONTINUE                                                          14NO7
C
  300     CONTINUE                                                          14NO7
C
C-----RETURN TO CALLING SUBROUTINE
C
      RETURN                                                                27JE7
C
      END                                                                   27JE7
```

```
      SUBROUTINE MATNA1 ( L1, X, Y )                                    24JA8
C
C-----THIS SUBROUTINE ADDS A MATRIX TO A MATRIX AND RETURN THE NEGATIVE
C          OF THE SUM.
C
C
C-----NOTATION FOR SUBROUTINE MATNA1
C
C          HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C          I, J                COUNTERS IN DO LOOPS
C          L2                  MX PLUS THREE
C          X( )                MATRIX TO BE CHANGED
C          Y( )                MATRIX TO BE ADDED
C
C
      DIMENSION      X(L1,L1),       Y(L1,L1)                           24JA8
C
      COMMON    I, J, NUM(14), HUM(36), NT1(11), HT1(2), MYP2, L2       24JA8
C
C
           DO  200  J = 1, L2                                           24JA8
C
           DO  100  I = 1, L2                                           24JA8
C
                X(I,J)  =  - X(I,J) - Y(I,J)                            02FE8
C
  100      CONTINUE                                                     24JA8
C
  200      CONTINUE                                                     24JA8
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                            24JA8
C
      END                                                               24JA8
```

```
      SUBROUTINE MATS2 ( L1, Z, X3 )                                    31JA8
C
C-----THIS SUBROUTINE SUBTRACTS A FIVE-DIAGONAL MATRIX FROM A FULL MATRIX.
C
C
C-----NOTATION FOR SUBROUTINE MATS2
C
C          HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C          I                   COUNTER IN DO LOOP
C          MM1, MM2, MM3       M1 MINUS NUMBER IN NAME
C          M1                  MX PLUS THREE
C          X3( )               FIVE-DIAGONAL MATRIX
C          Z( )                FULL MATRIX
C
C
      DIMENSION      X3(L1,5),       Z(L1,L1)                           22JA8
C
      COMMON    I, MM1, MM2, J, MM3, NUM(11), HUM(36), NT1(11), HT1(2),  22JA8
     1          MYP2, M1                                                12DE7
C
C
                MM1  =  M1 - 1                                          01DE7
                MM2  =  M1 - 2                                          01DE7
                MM3  =  M1 - 3                                          01DE7
C
          DO  100  I = 3, MM2                                           24JA8
C
                Z(I,I-2)   =   Z(I,I-2) - X3(I,1)                       16JA8
                Z(I,I-1)   =   Z(I,I-1) - X3(I,2)                       16JA8
                Z(I,I)     =   Z(I,I) - X3(I,3)                         16JA8
                Z(I,I+1)   =   Z(I,I+1) - X3(I,4)                       16JA8
                Z(I,I+2)   =   Z(I,I+2) - X3(I,5)                       16JA8
C
  100     CONTINUE                                                      13JL7
C
                Z(1,1)   =   Z(1,1) - X3(1,3)                           16JA8
                Z(1,2)   =   Z(1,2) - X3(1,4)                           16JA8
                Z(1,3)   =   Z(1,3) - X3(1,5)                           16JA8
                Z(2,1)   =   Z(2,1) - X3(2,2)                           16JA8
                Z(2,2)   =   Z(2,2) - X3(2,3)                           16JA8
                Z(2,3)   =   Z(2,3) - X3(2,4)                           16JA8
                Z(2,4)   =   Z(2,4) - X3(2,5)                           16JA8
C
                Z(MM1,MM3)   =   Z(MM1,MM3) - X3(MM1,1)                 16JA8
                Z(MM1,MM2)   =   Z(MM1,MM2) - X3(MM1,2)                 16JA8
                Z(MM1,MM1)   =   Z(MM1,MM1) - X3(MM1,3)                 16JA8
                Z(MM1,M1)    =   Z(MM1,M1)  - X3(MM1,4)                 16JA8
                Z(M1,MM2)    =   Z(M1,MM2)  - X3(M1,1)                  16JA8
                Z(M1,MM1)    =   Z(M1,MM1)  - X3(M1,2)                  16JA8
                Z(M1,M1)     =   Z(M1,M1)   - X3(M1,3)                  16JA8
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                            13JL7
C
      END                                                               13JL7
```

```
      SUBROUTINE INVR4 ( L1, A )                                        22JA8
C
C-----THIS SUBROUTINE TAKES THE INVERSE A FULL MATRIX.
C
C
C-----NOTATION FOR SUBROUTINE INVR4
C
C          A( )                  MATRIX TO BE INVERTED
C          EP                    ZERO LIMIT
C          HT1, HUM, NT1, NUM    POSITIONING VARIABLES IN COMMON FIELD
C          I, II, J, K, KK, L    COUNTERS AND INDICES IN DO LOOPS
C          KERR                  ERROR FLAG
C          L1                    TRANSFER CONSTANT
C          L2                    MX PLUS THREE
C          S, S2                 TEMPORARY HOLDING VARIABLES
C
C
      DIMENSION     A(L1,L1)                                            28DE7
C
      COMMON   KERR, I, II, M, J, K, KK, L, NUM(8), EP, S, S2, HUM(33), 22JA8
     1         NT1(11), HT1(2), MYP2, L2                                24JA8
C
   20 FORMAT ( //25H     NO INVERSE EXISTS    , // )                    22JA8
   30 FORMAT ( 1X,10E10.3 )                                            100C7
C
C
               KERR  =  0                                               22JA8
               EP  =  1.0E-12                                           22JA8
C
          DO  185  I = 1, L2                                            24JA8
C
               KK  =  I + 1                                             24JA8
C
          IF ( ABS( A(I,I) ) - EP )      110, 110, 150                  24JA8
C
  110     DO  120  II = KK, L2                                          100C7
C
          IF ( ABS ( A(II,I) ) - EP )    120, 120, 130                  05FE8
C
  120     CONTINUE                                                      100C7
C
          GO TO 990                                                     100C7
C
  130     DO  140  L = 1, L2                                            24JA8
C
               S  =  A(I,L)                                             24JA8
               A(I,L)  =  A(II,L)                                       24JA8
               A(II,L)  =  S                                            24JA8
C
  140     CONTINUE                                                      100C7
C
  150          S  =  1.0 / A(I,I)                                       24JA8
C
          DO  160  J = 1, L2                                            24JA8
C
```

```
                A(I,J)  =  A(I,J) * S                                  24JA8
C
  160       CONTINUE                                                    100C7
C
                A(I,I)  =  S                                          24JA8
C
            DO  180  J = 1, L2                                        24JA8
C
            IF ( J - I )                    170, 180, 170             24JA8
C
  170           S2  =  A(J,I)                                         24JA8
C
                A(J,I)  =  0.0                                        24JA8
C
            DO  175  K = 1, L2                                        24JA8
C
                A(J,K)  =  A(J,K) - S2 * A(I,K)                       24JA8
C
  175       CONTINUE                                                    100C7
C
  180       CONTINUE                                                    100C7
C
  185       CONTINUE                                                    100C7
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                          100C7
C
C-----PRINT NO INVERSE EXISTS AND DISPLAY THE ARRAY
C
  990 PRINT 20                                                        100C7
C
      PRINT 30, ( ( A(I,J), I = 1, L2 ), J = 1, L2 )                  31JA8
C
C-----SET ERROR FLAG
C
                KERR  =  999                                          23JA8
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                          22JA8
C
      END                                                             100C7
```

```
      SUBROUTINE MATM2 ( M1, Y, X, Z )                                  12DE7
C
C-----THIS SUBROUTINE MULTIPLIES A FULL MATRIX BY A DIAGONAL MATRIX.
C
C
C-----NOTATION FOR SUBROUTINE MATM2
C
C          HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C          I, J                COUNTERS IN DO LOOPS
C          L2                  MX PLUS THREE
C          M1                  TRANSFER CONSTANT
C          X( )                DIAGONAL MATRIX
C          Y( )                FULL MATRIX
C          Z( )                PRODUCT MATRIX
C
C
      DIMENSION      X(M1),          Z(M1,M1),       Y(M1,M1)           22JA8
C
      COMMON    I, J, NUM(14), HUM(36), NT1(11), HT1(2), MYP2, L2       22JA8
C
C
          DO  200  I = 1, L2                                            24JA8
C
          DO  100  J = 1, L2                                            24JA8
C
               Z(I,J) = X(J) * Y(I,J)                                   13JL7
C
  100     CONTINUE                                                      13JL7
C
  200     CONTINUE                                                      14NO7
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                            13JL7
C
      END                                                               13JL7
```

```
      SUBROUTINE MATA2 ( L1, X, Y )                                    24JA8
C
C-----THIS SUBROUTINE ADDS A VECTOR TO A VECTOR.
C
C
C-----NOTATION FOR SUBROUTINE MATA2
C
C          HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C          I                   COUNTER IN DO LOOP
C          L1                  TRANSFER CONSTANT
C          L2                  MX PLUS THREE
C          X( )                PRINCIPAL VECTOR
C          Y( )                ADDED VECTOR
C
C
      DIMENSION      X(L1,1),        Y(L1,1)                           24JA8
C
      COMMON    I, NUM(15), HUM(36), NT1(11), HT1(2), MYP2, L2         24JA8
C
C
          DO  100  I = 1, L2                                           24JA8
C
               X(I,1)  =  X(I,1) + Y(I,1)                              24JA8
C
  100     CONTINUE                                                     24JA8
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                           24JA8
C
      END                                                              24JA8
```

```
      SUBROUTINE MATS3 ( L1, X, Y )                                      24JA8
C
C-----THIS SUBROUTINE SUBTRACT A VECTOR FROM A VECTOR.
C
C
C-----NOTATION FOR SUBROUTINE MATS3
C
C          HT1, HUM, NT1, NUM  POSITIONING VARIABLES IN COMMON FIELD
C          I                   COUNTER IN DO LOOP
C          L1                  TRANSFER CONSTANT
C          L2                  MX PLUS THREE
C          X( )                PRINCIPAL VECTOR
C          Y( )                SUBTRACTED VECTOR
C
C
      DIMENSION      X(L1,1),        Y(L1)                               24JA8
C
      COMMON    I, NUM(15), HUM(36), NT1(11), HT1(2), MYP2, L2           24JA8
C
C
           DO  100  I = 1, L2                                            24JA8
C
                X(I,1)  =  X(I,1) - Y(I)                               24JA8
C
  100      CONTINUE                                                      24JA8
C
C-----RETURN TO SUBROUTINE MATRIX
C
      RETURN                                                             24JA8
C
      END                                                                24JA8
```

```
      SUBROUTINE OUTPTL ( BETA, BMO, BMT, BMX, BMY, CX, CY, DX, DY, HX,  24JA8
     1                    HY, PX, PY, REACT, SHEAR, SIGO, SIGI, TMMAX,  24JA8
     2                    TMX, TMY, W, L1, L2, L3, L4 )                 24JA8
C
C-----THIS SUBROUTINE CALCULATES MOMENTS, STRESSES, AND REACTION FOR THE
C          LONG FORM OF OUTPUT.
C
C
C-----NOTATION FOR SUBOUTINE OUTPTL
C
C          ALF                   ANGLE IN RADIANS
C          BETA( )               ANGLE FROM X TO MAXIMUM PRINCIPAL MOMENT
C          BHXQ1, BHXQ2,         TEMPORARY FUNCTIONS OF HX
C          BHXQ3
C          BHYQ1, BHYQ2,         TEMPORARY FUNCTIONS OF HY
C          BHYQ3
C          BMA                   CENTER OF MOHRS CIRCLE
C          BMO( )                MAXIMUM PRINCIPAL MOMENT
C          BMP                   HORIZONTAL DISTANCE FROM X TO CENTER
C          BMT( )                MINIMUM PRINCIPAL MOMENT
C          BMX( )                X-BENDING MOMENT
C          BMY( )                Y-BENDING MOMENT
C          CX( )                 TWISTING STIFFNESS X-DIRECTION
C          CY( )                 TWISTING STIFFNESS Y-DIRECTION
C          DX( )                 BENDING STIFFNESS X-DIRECTION
C          DY( )                 BENDING STIFFNESS Y-DIRECTION
C          HUM, MXP, NT1, NUM    POSITIONING VARIABLES IN COMMON FIELD
C          HX( )                 INCREMENT LENGTH X-BEAM
C          HXI, HX1I             TEMPORARY FUNCTIONS OF HX
C          HXHY, HXOHY,          TEMPORARY FUNCTIONS OF HX AND HY
C          HXOHY1, HYOHX,
C          HYOHX1
C          HY( )                 INCREMENT LENGTH Y-BEAM
C          HYI, HY1I             TEMPORARY FUNCTIONS OF HY
C          I, J                  COUNTERS IN DO LOOPS
C          L1 --- L4             TRANSFER CONSTANT
C          MXP4                  MX PLUS FOUR
C          MXP5                  MX PLUS FIVE
C          MYP4                  MY PLUS FOUR
C          MYP5                  MY PLUS FIVE
C          PR                    POISSONS RATIO
C          PX( )                 AXIAL LOAD X-DIRECTION
C          PY( )                 AXIAL LOAD Y-DIRECTION
C          QBMX                  SECOND DERIVATIVE OF X-BENDING MOMENT
C          QBMY                  SECOND DERIVATIVE OF Y-BENDING MOMENT
C          QPX                   SECOND DERIVATIVE OF X-AXIAL FORCE
C          QPY                   SECOND DERIVATIVE OF Y-AXIAL FORCE
C          QTMY                  SECOND DERIVATIVE OF Y-TWISTING MOMENT
C          QTMX                  SECOND DERIVATIVE OF X-TWISTING MOMENT
C          REACT( )              SLAB REACTION
C          RSUM                  SUMMATION OF SLAB REACTION
C          SDT2                  TEMPORARY FUNCTION OF THICKNESS
C          SHEAR( )              SHEAR STRESS
C          SIGO( )               MAXIMUM PRINCIPAL STRESS
C          THETA                 BETA TIMES TWO
```

```
C          THK                    SLAB THICKNESS
C          TMA                    VERTICAL DISTANCE FROM X TO CENTER
C          TMMAX( )               MAXIMUM TORQUE
C          TMX( )                 X-TWISTING MOMENT
C          TMY( )                 Y-TWISTING MOMENT
C          W( )                   DEFLECTION OF SLAB
C          WSUM1, WSUM2,          TEMPORARY FUNCTIONS OF W
C          WSUM3
C
C
      DIMENSION      BETA(L1,L2),    BMO(L1,L2),     BMT(L1,L2),            22JA8
     1               BMX(L1,L2),     BMY(L1,L2),     CX(L1,L2),             22JA8
     2               CY(L1,L2),      DX(L1,L2),      DY(L1,L2),             02MR8
     3               HX(L3),         HY(L4),         PX(L1,L2),             22JA8
     4               PY(L1,L2),      REACT(L1,L2),   SHEAR(L1,L2),          22JA8
     5               SIGO(L1,L2),    SIGT(L1,L2),    TMMAX(L1,L2),          22JA8
     6               TMX(L1,L2),     TMY(L1,L2),     W(L1,L2)               22JA8
C
      COMMON    I, J, NUM(14), ALF, BHXQ1, BHXQ2, BHXQ3, BHYQ1, BHYQ2,      22JA8
     1          BHYQ3, BMA, BMP, HXHY, HXI, HXOHY, HXOHY1, HX1I, HYI,       22JA8
     2          HYOHX, HYOHX1, HY1I, QBMX, QBMY, QPX, QPY, QTMX, QTMY,      22JA8
     3          RSUM, SDT2, THETA, TMA, WSUM1, WSUM2, WSUM3, HUM(5),        24JA8
     4          NT1(11), PR, THK, MXP(3), MXP4, MYP4, MXP5, MYP5            24JA8
C
   10 FORMAT ( //35H          SUMMATION OF REACTIONS  =  F15.8, // )        24JA8
C
C
C-----COMPUTE BENDING MOMENTS
C
          DO  200  J = 1, MYP5                                              22JA8
C
               BHYQ1  =  2.0 / ( HY(J) * ( HY(J) + HY(J+1) ) )              17NO7
               BHYQ2  =  2.0 / ( HY(J) * HY(J+1) )                          17NO7
               BHYQ3  =  2.0 / ( HY(J+1) * ( HY(J) + HY(J+1) ) )            17NO7
C
          DO  150  I = 1, MXP5                                              22JA8
C
          IF ( DX(I,J) + DY(I,J) )      100, 125, 100                       22JA8
C
  100          BHXQ1  =  2.0 / ( HX(I) * ( HX(I) + HX(I+1) ) )             22JA8
               BHXQ2  =  2.0 / ( HX(I) * HX(I+1) )                          17NO7
               BHXQ3  =  2.0 / ( HX(I+1) * ( HX(I) + HX(I+1) ) )            17NO7
C
               WSUM1  =  W(I-1,J) * BHXQ1 - W(I,J) * BHXQ2 + W(I+1,J) *     09NO7
     1                    BHXQ3                                             09NO7
               WSUM2  =  W(I,J-1) * BHYQ1 - W(I,J) * BHYQ2 + W(I,J+1) *     17NO7
     1                    BHYQ3                                             09NO7
C
               BMX(I,J)  =  DX(I,J) * WSUM1 + DY(I,J) * PR * WSUM2          09NO7
               BMY(I,J)  =  DY(I,J) * WSUM2 + DX(I,J) * PR * WSUM1          09NO7
C
          GO TO 150                                                         22JA8
C
  125          BMX(I,J)   =   0.0                                           22JA8
               BMY(I,J)   =   0.0                                           08SE7
```

```
C
  150     CONTINUE                                                        22JA8
C
  200     CONTINUE                                                        22JA8
C
                RSUM  =  0.0                                            23JA8
C
          DO  400  J = 2, MYP4                                          22JA8
C
                HYI    =  1.0 / HY(J)                                   18AG7
                HY1I   =  1.0 / HY(J+1)                                 18AG7
C
          DO  300  I = 2, MXP4                                          22JA8
C
                HXI    =  1.0 / HX(I)                                   18AG7
                HX1I   =  1.0 / HX(I+1)                                 18AG7
                HYOHX  =  ( HY(J) + HY(J+1) ) * 0.50 * HXI              18AG7
                HYOHX1 =  ( HY(J) + HY(J+1) ) * 0.50 * HX1I             18AG7
                HXOHY  =  ( HX(I) + HX(I+1) ) * 0.50 * HYI              18AG7
                HXOHY1 =  ( HX(I) + HX(I+1) ) * 0.50 * HY1I             18AG7
C
C-----COMPUTE ABSORBED LOAD
C
                QBMX   =   HYOHX * ( BMX(I-1,J) - BMX(I,J) ) - HYOHX1 *  18AG7
     1                     ( BMX(I,J) - BMX(I+1,J) )                    18AG7
                QBMY   =   HXOHY * ( BMY(I,J-1) - BMY(I,J) ) - HXOHY1 *  18AG7
     1                     ( BMY(I,J) - BMY(I,J+1) )                    18AG7
C
                QTMX   =   HXI * ( CX(I,J) * HYI * ( W(I,J) + W(I-1,J-1)  18AG7
     1                     -W(I-1,J) - W(I,J-1) ) + CX(I,J+1) * HY1I * (  18AG7
     2                     W(I,J) + W(I-1,J+1) - W(I-1,J) - W(I,J+1) ) )  18AG7
     3                     - HX1I * ( CX(I+1,J) * HYI * ( W(I+1,J) +      18AG7
     4                     W(I,J-1) - W(I,J) - W(I+1,J-1) ) + CX(I+1,J+1) 19AG7
     5                     * HY1I * ( W(I+1,J) + W(I,J+1) - W(I,J) -      19AG7
     6                     W(I+1,J+1) ) )                                 18AG7
                QTMY   =   HYI * ( CY(I,J) * HXI * ( W(I,J) + W(I-1,J-1)  18AG7
     1                     - W(I-1,J) - W(I,J-1) ) + CY(I+1,J) * HX1I * (  18AG7
     2                     W(I,J) + W(I+1,J-1) -W(I+1,J) - W(I,J-1) ) )   30AG7
     3                     - HY1I * ( CX(I,J+1) * HXI * ( W(I-1,J) +      29AG7
     4                     W(I,J+1) - W(I,J) - W(I-1,J+1) ) +             29AG7
     5                     CX(I+1,J+1) * HX1I * ( W(I+1,J) + W(I,J+1) -   18AG7
     6                     W(I,J) - W(I+1,J+1) ) )                        18AG7
C
                QPX    =   PX(I,J) * HXI * ( W(I,J) - W(I-1,J) ) -        18AG7
     1                     PX(I+1,J) * HX1I * ( W(I+1,J) - W(I,J) )       18AG7
                QPY    =   PY(I,J) * HYI * ( W(I,J) - W(I,J-1) ) -        18AG7
     1                     PY(I,J+1) * HY1I * ( W(I,J+1) - W(I,J) )       02MR8
C
C-----COMPUTE REACTION
C
                REACT(I,J)  =  QBMX + QBMY + QTMX + QTMY + QPX + QPY     22JA8
C
C-----SUM THE REACTIONS
C
                RSUM  =  RSUM + REACT(I,J)                              23JA8
```

```
C
      IF ( ( REACT(I,J) * REACT(I,J) ) .LT. 1.00E-14 )                  31JA8
     1                                  REACT(I,J)  =  0.0              31JA8
C
C-----COMPUTE TWISTING MOMENTS
C
            HXHY  =   1.0 / ( HX(I) * HY(J) + HX(I+1) * HY(J) +         18AG7
     1                HX(I) * HY(J+1) + HX(I+1) * HY(J+1) )             18AG7
C
            WSUM3  =  0.25 * HXHY * ( W(I-1,J-1) - W(I-1,J+1) -         22JA8
     1                W(I+1,J-1) + W(I+1,J+1) )                         09NO7
C
            TMX(I,J)  =  ( CX(I,J) + CX(I,J+1) + CX(I+1,J) +            09NO7
     1                   CX(I+1,J+1) ) * WSUM3                          22JA8
            TMY(I,J)  =  - ( CY(I,J) + CY(I,J+1) + CY(I+1,J) +          09NO7
     1                   CY(I+1,J+1) ) * WSUM3                          22JA8
C
  300   CONTINUE                                                        22JA8
C
  400   CONTINUE                                                        22JA8
C
C-----PRINT PART 1 OF TABLE 8
C
      CALL OUTPT1 ( BMX, BMY, REACT, TMX, TMY, W, L1, L2 )              22JA8
C
C-----PRINT THE SUM OF THE REACTIONS
C
      PRINT 10, RSUM                                                    24JA8
C
C-----IF A THICKNESS IS INPUT SET STRESS CONSTANT
C
      IF ( THK .EQ. 0 )                GO TO 450                        22JA8
C
            SDT2  =  6.0 / ( THK * THK )                                18OC7
C
C-----COMPUTE PRINCIPAL MOMENTS AND BETA
C
  450   DO  800  J = 2, MYP4                                             22JA8
C
      DO  750  I = 2, MXP4                                              22JA8
C
            BMA  =  ( BMX(I,J) + BMY(I,J) ) * 0.50                      28SE7
            TMA  =  ( TMY(I,J) - TMX(I,J) ) * 0.50                      28SE7
            BMP = BMX(I,J) - BMA                                        02AG7
C
            TMMAX(I,J)  =  SQRT ( BMP * BMP + TMA * TMA )               02OC7
            BMO(I,J)  =  BMA + TMMAX(I,J)                               02OC7
            BMT(I,J)  =  BMA - TMMAX(I,J)                               02OC7
C
      IF ( BMP )                       500, 550, 650                    24JA8
C
  500       ALF = ATAN ( TMA / BMP )                                    22JA8
C
      IF ( TMA )                       520, 590, 540                    24JA8
C
```

```
  520           THETA = ALF * 57.29578 - 180.0                          22JA8
C
          GO TO 700                                                     22JA8
C
  540           THETA = ALF * 57.29578 + 180.0                          22JA8
C
          GO TO 700                                                     22JA8
C
  550     IF ( TMA )                    570, 590, 610                   24JA8
C
  570           BETA(I,J)  =  45.0                                      22JA8
C
          GO TO 730                                                     22JA8
C
  590           BETA(I,J)  =  0.0                                       24JA8
C
          GO TO 730                                                     22JA8
C
  610           BETA(I,J)  =  - 45.0                                    24JA8
C
          GO TO 730                                                     22JA8
C
  650           ALF = ATAN ( TMA / BMP )                                22JA8
                THETA = ALF * 57.29578                                  02AG7
C
  700           BETA(I,J)  =  THETA * 0.50                              24JA8
C
C-----IF A THICKNESS IS INPUT COMPUTE THE STRESSES
C
  730     IF ( THK )                    740, 750, 740                   24JA8
C
C-----COMPUTE THE PRINCIPAL STRESSES
C
  740           SIGO(I,J)   =  BMO(I,J) * SDT2                          22JA8
                SIGT(I,J)   =  BMT(I,J) * SDT2                          18OC7
                SHEAR(I,J)  =  ( SIGO(I,J) - SIGT(I,J) ) * 0.50         12SE7
C
  750     CONTINUE                                                      22JA8
C
  800     CONTINUE                                                      22JA8
C
C-----PRINT PART 2 OF TABLE 8
C
C-----CHECK FOR STRESSES
C
          IF ( THK .EQ. 0 )             GO TO 850                       22JA8
C
C-----PRINT PRINCIPAL MOMENTS AND STRESSES
C
      CALL OUTPT2 ( BMO, BMT, TMMAX, SIGO, SIGT, SHEAR, BETA, L1, L2 )  22JA8
C
          GO TO 900                                                     22JA8
C
C-----PRINT PRINCIPAL MOMENTS
C
  850 CALL OUTPT3 ( BMO, BMT, TMMAX, BETA, L1, L2 )                     22JA8
C
C-----RETURN TO SUBROUTINE V03SUB                                       22JA8
C
  900 RETURN                                                            09NO7
C
      END                                                               09NO7
```

```
      SUBROUTINE OUTPT1 ( BMX, BMY, REACT, TMX, TMY, W, L1, L2 )        22JA8
C
C-----THIS SUBROUTINE PRINT PART 1 OF TABLE 8
C
C
C-----NOTATION FOR SUBOUTINE OUTPT1
C
C          BMX( )               X-BENDING MOMENT
C          BMY( )               Y-BENDING MOMENT
C          HT1, HUM, NT1, NUM   POSITIONING VARIABLES IN COMMON FIELD
C          I, J                 COUNTERS IN DO LOOPS
C          ISTA, JSTA           EXTERNAL STATION NUMBER
C          L1, L2               TRANSFER CONSTANTS
C          MXP3                 MX PLUS THREE
C          MYP3                 MY PLUS THREE
C          REACT( )             SLAB REACTION
C          TMX( )               X-TWISTING MOMENT
C          TMY( )               Y-TWISTING MOMENT
C          W( )                 DEFLECTION OF SLAB
C
C
      DIMENSION     BMX(L1,L2),    BMY(L1,L2),    REACT(L1,L2),         22JA8
     1              TMX(L1,L2),    TMY(L1,L2),    W(L1,L2)              22JA8
C
      COMMON   I, ISTA, J, JSTA, NUM(12), HUM(36), NT1(11), HT1(2),     24JA8
     1        MYP2, MXP3, MYP3                                          24JA8
C
    6 FORMAT ( )                                                        09NO7
   39 FORMAT ( //35H     TABLE 8.  PART 1.   RESULTS    , ///          24JA8
     1           45H                                X          Y       24JA8
     2           35H      X           Y                    , /         05FE8
     3           45H                                BENDING   BENDING  24JA8
     4           35H  TWISTING   TWISTING      NET       , /          24JA8
     5           45H      I, J  DEFLECTION   MOMENT      MOMENT       24JA8
     6           35H   MOMENT     MOMENT    REACTION    , /           02FE8
     7           45H     (X) (Y)     (L)      (LF/L)     (LF/L)       02FE8
     8           35H   (LF/L)     (LF/L)       (F)      , // )        02FE8
   45 FORMAT  ( 6X, I2, 1X, I3, 6E11.3  )                               26SE7
C
C
C-----PRINT W, BMX, BMY, TMX, TMY, AND REACT
C
      PRINT 39                                                          02NO6
C
          DO  200  J = 3, MYP3                                          24JA8
C
               JSTA  =  J  -  3                                         08SE7
C
      PRINT  6                                                          21JL7
C
          DO  100  I = 3, MXP3                                          24JA8
C
               ISTA  =  I  -  3                                         08SE7
C
      PRINT 45, ISTA, JSTA, W(I,J), BMX(I,J), BMY(I,J), TMX(I,J),       12SE7
     1                      TMY(I,J), REACT(I,J)                       12SE7
C
  100     CONTINUE                                                      23JA8
C
  200     CONTINUE                                                      23JA8
C
C-----RETURN TO SUBROUTINE OUTPTL
C
      RETURN                                                            23JA8
C
      END                                                               22JA8
```

```
      SUBROUTINE OUTPT2 ( BMO, BMT, TMMAX, SIGO, SIGT, SHEAR, BETA, L1, 23JA8
     1                    L2 )                                          23JA8
C
C-----THIS SUBROUTINE PRINTS PART 2 OF TABLE 8 WITH STRESSES.
C
C
C-----NOTATION FOR SUBOUTINE OUTPT2
C
C          BETA( )              ANGLE FROM X TO MAXIMUM PRINCIPAL MOMENT
C          BMO( )               MAXIMUM PRINCIPAL MOMENT
C          BMT( )               MINIMUM PRINCIPAL MOMENT
C          HT1, HUM, NT1, NUM   POSITIONING VARIABLES IN COMMON FIELD
C          I, J                 COUNTERS IN DO LOOPS
C          ISTA, JSTA           EXTERNAL STATION NUMBER
C          L1, L2               TRANSFER CONSTANTS
C          MXP3                 MX PLUS THREE
C          MYP3                 MY PLUS THREE
C          SHEAR( )             SHEAR STRESS
C          SIGO( )              MAXIMUM PRINCIPAL STRESS
C          SIGT( )              MINIMUM PRINCIPAL STRESS
C          TMMAX( )             MAXIMUM TORQUE
C
C
      DIMENSION     BETA(L1,L2),   BMO(L1,L2),    BMT(L1,L2),           23JA8
     1              TMMAX(L1,L2),  SIGO(L1,L2),   SIGT(L1,L2),          23JA8
     2              SHEAR(L1,L2)                                        23JA8
C
      COMMON   I, ISTA, J, JSTA, NUM(12), HUM(36), NT1(11), HT1(2),     24JA8
     1         MYP2, MXP3, MYP3                                         24JA8
C
    6 FORMAT ( )                                                        09NO7
   40 FORMAT ( //35H     TABLE 8.  PART 2.   RESULTS   , ///            24JA8
     1           45H                 MAXIMUM    MINIMUM                 21MR8
     2           40H   MAXIMUM     MINIMUM                BETA , /      21MR8
     3           45H                PRINCIPAL  PRINCIPAL   MAXIMUM      21MR8
     4           40H  PRINCIPAL   PRINCIPAL    SHEAR     X TO , /      24JA8
     5           45H      I, J     MOMENT     MOMENT      TORQUE       24JA8
     6           40H   STRESS     STRESS     STRESS    MAX. , /       02FE8
     7           45H     (X) (Y)   (LF/L)     (LF/L)     (LF/L)       02FE8
     8           40H   (F/LL)     (F/LL)     (F/LL)    (DEG), // )     02FE8
   47 FORMAT  ( 6X, I2, 1X, I3, 6E11.3, F7.2  )                         26SE7
C
C
C-----PRINT BMO, BMT, TMMAX, SIGO, SIGT, SHEAR, AND BETA
C
      PRINT 40                                                          23JA8
C
          DO  200  J = 3, MYP3                                          24JA8
C
               JSTA  =  J  -  3                                         11SE7
C
      PRINT 6                                                           02AG7
C
          DO  100  I = 3, MXP3                                          24JA8
C
```

```
                ISTA  =  I  -  3                                          28SE7
C
      PRINT 47, ISTA, JSTA, BMO(I,J), BMT(I,J), TMMAX(I,J), SIGO(I,J),   12SE7
     1                      SIGT(I,J), SHEAR(I,J), BETA(I,J)             12SE7
C
  100      CONTINUE                                                      23JA8
C
  200      CONTINUE                                                      23JA8
C
C-----RETURN TO SUBROUTINE OUTPTL
C
      RETURN                                                            23JA8
C
      END                                                               23JA8
```

```
      SUBROUTINE OUTPT3 ( BMO, BMT, TMMAX, BETA, L1, L2 )                23JA8
C
C-----THIS SUBROUTINE PRINTS PART 2 OF TABLE 8 WITHOUT STRESSES.
C
C
C-----NOTATION FOR SUBOUTINE OUTPT3
C
C          BETA( )              ANGLE FROM X TO MAXIMUM PRINCIPAL MOMENT
C          BMO( )               MAXIMUM PRINCIPAL MOMENT
C          BMT( )               MINIMUM PRINCIPAL MOMENT
C          HT1, HUM, NT1, NUM   POSITIONING VARIABLES IN COMMON FIELD
C          I, J                 COUNTERS IN DO LOOPS
C          ISTA, JSTA           EXTERNAL STATION NUMBER
C          L1, L2               TRANSFER CONSTANTS
C          MXP3                 MX PLUS THREE
C          MYP3                 MY PLUS THREE
C          TMMAX( )             MAXIMUM TORQUE
C
C
      DIMENSION      BETA(L1,L2),    BMO(L1,L2),     BMT(L1,L2),          23JA8
     1               TMMAX(L1,L2)                                        23JA8
C
      COMMON    I, ISTA, J, JSTA, NUM(12), HUM(36), NT1(11), HT1(2),     24JA8
     1          MYP2, MXP3, MYP3                                         24JA8
C
    6 FORMAT ( )                                                         23JA8
   40 FORMAT ( //35H      TABLE 8.  PART 2.   RESULTS    , ///          24JA8
     1            45H                  MAXIMUM     MINIMUM              21MR8
     2            10H  BETA     , /                                     24JA8
     3            45H                PRINCIPAL   PRINCIPAL    MAXIMUM    21MR8
     4            10H  X TO     , /                                     24JA8
     5            45H       I,  J    MOMENT      MOMENT      TORQUE     24JA8
     6            10H  MAX.     , /                                     02FE8
     7            45H      (X) (Y)   (LF/L)      (LF/L)      (LF/L)     02FE8
     8            10H  (DEG)    , // )                                  02FE8
   46 FORMAT  ( 6X, I2, 1X, I3, 3E11.3, F7.2  )                          26SE7
C
C
C-----PRINT BMO, BMT, TMMAX, AND BETA
C
      PRINT 40                                                           24JA8
C
          DO  200  J = 3, MYP3                                           24JA8
C
               JSTA  =  J - 3                                            23JA8
C
      PRINT 6                                                            23JA8
C
          DO  100  I = 3, MXP3                                           24JA8
C
               ISTA  =  I - 3                                            23JA8
C
      PRINT 46, ISTA, JSTA, BMO(I,J), BMT(I,J), TMMAX(I,J), BETA(I,J)    22JA8
C
  100     CONTINUE                                                       23JA8
C
  200     CONTINUE                                                       23JA8
C
C-----RETURN TO SUBROUTINE OUTPTL
C
      RETURN                                                             23JA8

      END                                                                23JA8
```

```
      SUBROUTINE OUTPTS ( BETA, BMX, BMY, CX, CY, DX, DY, HX, HY, PX,    24JA8
     1                    PY, REACT, SIGT, TMX, W, L1, L2, L3, L4 )      24JA8
C
C-----THIS SUBROUTINE CALCULATES MOMENTS, STRESSES, AND REACTION FOR THE
C          SHORT FORM OF OUTPUT.
C
C
C-----NOTATION FOR SUBOUTINE OUTPTS
C
C          ALF                    ANGLE IN RADIANS
C          BETA( )                ANGLE FROM X TO LARGEST PRINCIPAL MOMENT
C          BHXQ1, BHXQ2,          TEMPORARY FUNCTIONS OF HX
C          BHXQ3
C          BHYQ1, BHYQ2,          TEMPORARY FUNCTIONS OF HY
C          BHYQ3
C          BMA                    CENTER OF MOHRS CIRCLE
C          BMO                    MAXIMUM PRINCIPAL MOMENT
C          BMP                    HORIZONTAL DISTANCE FROM X TO CENTER
C          BMT                    MINIMUM PRINCIPAL MOMENT
C          BMX( )                 X-BENDING MOMENT
C          BMY( )                 Y-BENDING MOMENT
C          CX( )                  TWISTING STIFFNESS X-DIRECTION
C          CY( )                  TWISTING STIFFNESS Y-DIRECTION
C          DX( )                  BENDING STIFFNESS X-DIRECTION
C          DY( )                  BENDING STIFFNESS Y-DIRECTION
C          HUM, MXP, NT1, NUM     POSITIONING VARIABLES IN COMMON FIELD
C          HX( )                  INCREMENT LENGTH X-BEAM
C          HXI, HX1I              TEMPORARY FUNCTIONS OF HX
C          HXHY, HXOHY,           TEMPORARY FUNCTIONS OF HX AND HY
C          HXOHY1, HYOHX,
C          HYOHX1
C          HY( )                  INCREMENT LENGTH Y-BEAM
C          HYI, HY1I              TEMPORARY FUNCTIONS OF HY
C          I, J                   COUNTERS IN DO LOOPS
C          L1 --- L4              TRANSFER CONSTANT
C          MXP4                   MX PLUS FOUR
C          MXP5                   MX PLUS FIVE
C          MYP4                   MY PLUS FOUR
C          MYP5                   MY PLUS FIVE
C          PR                     POISSONS RATIO
C          PX( )                  AXIAL LOAD X-DIRECTION
C          PY( )                  AXIAL LOAD Y-DIRECTION
C          QBMX                   SECOND DERIVATIVE OF X-BENDING MOMENT
C          QBMY                   SECOND DERIVATIVE OF Y-BENDING MOMENT
C          QPX                    SECOND DERIVATIVE OF X-AXIAL FORCE
C          QPY                    SECOND DERIVATIVE OF Y-AXIAL FORCE
C          QTMX                   SECOND DERIVATIVE OF X-TWISTING MOMENT
C          QTMY                   SECOND DERIVATIVE OF Y-TWISTING MOMENT
C          REACT( )               SLAB REACTION
C          RSUM                   SUMMATION OF SLAB REACTION
C          SDT2                   TEMPORARY FUNCTION OF THICKNESS
C          SIGT( )                LARGEST PRINCIPAL MOMENT OR STRESS
C          THETA                  BETA TIMES TWO
C          THK                    SLAB THICKNESS
C          TMA                    VERTICAL DISTANCE FROM X TO CENTER
```

```
C          TMMAX                MAXIMUM TORQUE
C          TMX( )               X-TWISTING MOMENT
C          TMY                  Y-TWISTING MOMENT
C          W( )                 DEFLECTION
C          WSUM1, WSUM2,        TEMPORARY FUNCTIONS OF W
C          WSUM3
C
C
      DIMENSION     BETA(L1,L2),    BMX(L1,L2),     BMY(L1,L2),            24JA8
     1              CX(L1,L2),      CY(L1,L2),      DX(L1,L2),             24JA8
     2              DY(L1,L2),      HX(L3),         HY(L4),                24JA8
     3              PX(L1,L2),      PY(L1,L2),      REACT(L1,L2),          24JA8
     4              SIGT(L1,L2),    TMX(L1,L2),     W(L1,L2)               24JA8
C
      COMMON    I, J, NUM(14), ALF, BHXQ1, BHXQ2, BHXQ3, BHYQ1, BHYQ2,    22JA8
     1          BHYQ3, BMA, BMO, BMP, BMT, HXHY, HXI, HXOHY, HXOHY1,      24JA8
     2          HX1I, HYI, HYOHX, HYOHX1, HY1I, QBMX, QBMY, QPX, QPY,     24JA8
     3          QTMX, QTMY, RSUM, SDT2, THETA, TMA, TMMAX, TMY, WSUM1,    24JA8
     4          WSUM2, WSUM3, HUM, NT1(11), PR, THK, MXP(3), MXP4, MYP4,  24JA8
     5          MXP5, MYP5                                                24JA8
C
   10 FORMAT ( //35H          SUMMATION OF REACTIONS  =  F15.8, // )      24JA8
C
C
C-----COMPUTE BENDING MOMENTS
C
          DO  200  J = 1, MYP5                                            22JA8
C
               BHYQ1  =  2.0 / ( HY(J) * ( HY(J) + HY(J+1) ) )            17NO7
               BHYQ2  =  2.0 / ( HY(J) * HY(J+1) )                        17NO7
               BHYQ3  =  2.0 / ( HY(J+1) * ( HY(J) + HY(J+1) ) )          17NO7
C
          DO  150  I = 1, MXP5                                            22JA8
C
          IF ( DX(I,J) + DY(I,J) )       100, 125, 100                    22JA8
C
  100          BHXQ1  =  2.0 / ( HX(I) * ( HX(I) + HX(I+1) ) )            22JA8
               BHXQ2  =  2.0 / ( HX(I) * HX(I+1) )                        17NO7
               BHXQ3  =  2.0 / ( HX(I+1) * ( HX(I) + HX(I+1) ) )          17NO7
C
               WSUM1  =  W(I-1,J) * BHXQ1 - W(I,J) * BHXQ2 + W(I+1,J) * 09NO7
     1                    BHXQ3                                           09NO7
               WSUM2  =  W(I,J-1) * BHYQ1 - W(I,J) * BHYQ2 + W(I,J+1) * 17NO7
     1                    BHYQ3                                           09NO7
C
               BMX(I,J)  =  DX(I,J) * WSUM1 + DY(I,J) * PR * WSUM2        09NO7
               BMY(I,J)  =  DY(I,J) * WSUM2 + DX(I,J) * PR * WSUM1        09NO7
C
          GO TO 150                                                       22JA8
C
  125          BMX(I,J)   =   0.0                                         22JA8
               BMY(I,J)   =   0.0                                         08SE7
C
  150     CONTINUE                                                        22JA8
C
```

```
  200     CONTINUE                                                        22JA8
C
              RSUM   =  0.0                                               23JA8
C
C-----IF A THICKNESS IS INPUT SET STRESS CONSTANT                         24JA8
C
          IF ( THK .EQ. 0 )              GO TO 450                        22JA8
C
              SDT2   =  6.0 / ( THK * THK )                               18OC7
C
  450     DO  900  J = 2, MYP4                                            24JA8
C
              HYI    =  1.0 / HY(J)                                       18AG7
              HY1I   =  1.0 / HY(J+1)                                     18AG7
C
          DO  850  I = 2, MXP4                                            24JA8
C
              HXI    =  1.0 / HX(I)                                       18AG7
              HX1I   =  1.0 / HX(I+1)                                     18AG7
              HYOHX  =  ( HY(J) + HY(J+1) ) * 0.50 * HXI                  18AG7
              HYOHX1 =  ( HY(J) + HY(J+1) ) * 0.50 * HX1I                 18AG7
              HXOHY  =  ( HX(I) + HX(I+1) ) * 0.50 * HYI                  18AG7
              HXOHY1 =  ( HX(I) + HX(I+1) ) * 0.50 * HY1I                 18AG7
C
C-----COMPUTE ABSORBED LOAD
C
              QBMX   =   HYOHX * ( BMX(I-1,J) - BMX(I,J) ) - HYOHX1 *     18AG7
     1                   ( BMX(I,J) - BMX(I+1,J) )                        18AG7
              QBMY   =   HXOHY * ( BMY(I,J-1) - BMY(I,J) ) - HXOHY1 *     18AG7
     1                   ( BMY(I,J) - BMY(I,J+1) )                        18AG7
C
              QTMX   =   HXI * ( CX(I,J) * HYI * ( W(I,J) + W(I-1,J-1)    18AG7
     1                   -W(I-1,J) - W(I,J-1) ) + CX(I,J+1) * HY1I * (    18AG7
     2                   W(I,J) + W(I-1,J+1) - W(I-1,J) - W(I,J+1) ) )    18AG7
     3                   - HX1I * ( CX(I+1,J) * HYI * ( W(I+1,J) +        18AG7
     4                   W(I,J-1) - W(I,J) - W(I+1,J-1) ) + CX(I+1,J+1)   19AG7
     5                   * HY1I * ( W(I+1,J) + W(I,J+1) - W(I,J) -        19AG7
     6                   W(I+1,J+1) ) )                                   18AG7
              QTMY   =   HYI * ( CY(I,J) * HXI * ( W(I,J) + W(I-1,J-1)    18AG7
     1                   - W(I-1,J) - W(I,J-1) ) + CY(I+1,J) * HX1I * (   18AG7
     2                   W(I,J) + W(I+1,J-1) -W(I+1,J) - W(I,J-1) ) )     30AG7
     3                   - HY1I * ( CX(I,J+1) * HXI * ( W(I-1,J) +        29AG7
     4                   W(I,J+1) - W(I,J) - W(I-1,J+1) ) +               29AG7
     5                   CX(I+1,J+1) * HX1I * ( W(I+1,J) + W(I,J+1) -     18AG7
     6                   W(I,J) - W(I+1,J+1) ) )                          18AG7
C
              QPX    =   PX(I,J) * HXI * ( W(I,J) - W(I-1,J) ) -          18AG7
     1                   PX(I+1,J) * HX1I * ( W(I+1,J) - W(I,J) )         18AG7
              QPY    =   PY(I,J) * HYI * ( W(I,J) - W(I,J-1) ) -          18AG7
     1                   PY(I,J+1) * HY1I * ( W(I,J+1) - W(I,J) )         02MR8
C
C-----COMPUTE REACTION
C
              REACT(I,J)  =  QBMX + QBMY + QTMX + QTMY + QPX + QPY        22JA8
C
```

```
C-----SUM THE REACTIONS
C
                RSUM   =  RSUM + REACT(I,J)                                    23JA8
C
           IF ( ( REACT(I,J) * REACT(I,J) ) .LT. 1.00E-14 )                    31JA8
     1                                    REACT(I,J)  =  0.0                  31JA8
C
C-----COMPUTE TWISTING MOMENTS
C
                HXHY   =   1.0 / ( HX(I) * HY(J) + HX(I+1) * HY(J) +          18AG7
     1                     HX(I) * HY(J+1) + HX(I+1) * HY(J+1) )               18AG7
C
                WSUM3  =  0.25 * HXHY * ( W(I-1,J-1) - W(I-1,J+1) -           22JA8
     1                     W(I+1,J-1) + W(I+1,J+1) )                           09NO7
C
                TMX(I,J)  =   ( CX(I,J) + CX(I,J+1) + CX(I+1,J) +              09NO7
     1                        CX(I+1,J+1) ) * WSUM3                            22JA8
                TMY       =  - ( CY(I,J) + CY(I,J+1) + CY(I+1,J) +           24JA8
     1                        CY(I+1,J+1) ) * WSUM3                            22JA8
C
C-----COMPUTE PRINCIPAL MOMENTS AND BETA
C
                BMA   =   ( BMX(I,J) + BMY(I,J) ) * 0.50                      28SE7
                TMA   =   ( TMY      - TMX(I,J) ) * 0.50                      24JA8
                BMP = BMX(I,J) - BMA                                           02AG7
C
                TMMAX   =  SQRT ( BMP * BMP + TMA * TMA )                      24JA8
                BMO   =  BMA + TMMAX                                           24JA8
                BMT   =  BMA - TMMAX                                           24JA8
C
           IF ( BMP )                        500, 550, 650                     24JA8
C
  500           ALF = ATAN ( TMA / BMP )                                       22JA8
C
           IF ( TMA )                        520, 590, 540                     24JA8
C
  520           THETA = ALF * 57.29578 - 180.0                                 22JA8
C
           GO TO 700                                                           22JA8
C
  540           THETA = ALF * 57.29578 + 180.0                                 22JA8
C
           GO TO 700                                                           22JA8
C
  550      IF ( TMA )                        570, 590, 610                     24JA8
C
  570           BETA(I,J)   =  45.0                                            22JA8
C
           GO TO 730                                                           22JA8
C
  590           BETA(I,J)   =  0.0                                             24JA8
C
           GO TO 730                                                           22JA8
C
  610           BETA(I,J)   =  - 45.0                                          24JA8
```

```
C
          GO TO 730                                                      22JA8
C
  650           ALF = ATAN ( TMA / BMP )                                 22JA8
                THETA = ALF * 57.29578                                   02AG7
C
  700           BETA(I,J)  =  THETA * 0.50                               24JA8
C
C-----CHOOSE LARGEST PRINCIPAL MOMENT
C
  730     IF ( BMA )                        750, 780, 780                05FE8
C
  750           SIGT(I,J)  =  BMT                                        24JA8
C
          IF ( BETA(I,J) )                  760, 770, 770                05FE8
C
  760           BETA(I,J)  =  - BETA(I,J) - 90.00                        05FE8
C
          GO TO 800                                                      05FE8
C
  770           BETA(I,J)  =  - BETA(I,J) + 90.00                        05FE8
C
          GO TO 800                                                      24JA8
C
  780           SIGT(I,J)  =  BMO                                        05FE8
C
C-----IF A THICKNESS IS INPUT COMPUTE THE STRESS
C
  800     IF ( THK )                        810, 850, 810                24JA8
C
  810           SIGT(I,J)  =  SIGT(I,J) * SDT2                           24JA8
C
  850     CONTINUE                                                       24JA8
C
  900     CONTINUE                                                       24JA8
C
C-----PRINT THE RESULTS
C
C-----CHECK FOR STRESSES
C
          IF ( THK .EQ. 0 )                 GO TO 950                    24JA8
C
C-----PRINT DEFLECTIONS, MOMENTS, AND PRINCIPAL STRESS
C
      CALL OUTPT4 ( BMX, BMY, TMX, REACT, SIGT, W, BETA, L1, L2 )        24JA8
C
                GO TO 970                                                24JA8
C
C-----PRINT DEFLECTIONS, MOMENTS, AND PRINCIPAL MOMENT
C
  950 CALL OUTPT5 ( BMX, BMY, TMX, REACT, SIGT, W, BETA, L1, L2 )        24JA8
C
C-----PRINT THE SUM OF THE REACTIONS
C
  970 PRINT 10, RSUM                                                     24JA8
C
C-----RETURN TO SUBROUTINE V03SUB                                        22JA8
C
      RETURN                                                             24JA8
C
      END                                                                09NO7
```

```
      SUBROUTINE OUTPT4 ( BMX, BMY, TMX, REACT, SIGT, W, BETA, L1, L2 ) 24JA8
C
C-----THIS SUBROUTINE PRINTS THE SHORT FORM OF TABLE 8 WITH PRIN STRESS.
C
C
C-----NOTATION FOR SUBOUTINE OUTPT4
C
C          BETA( )              ANGLE FROM X TO LARGEST PRINCIPAL MOMENT
C          BMX( )               X-BENDING MOMENT
C          BMY( )               Y-BENDING MOMENT
C          HT1, HUM, NT1, NUM   POSITIONING VARIABLES IN COMMON FIELD
C          I, J                 COUNTERS IN DO LOOPS
C          ISTA, JSTA           EXTERNAL STATION NUMBER
C          L1, L2               TRANSFER CONSTANTS
C          MXP3                 MX PLUS THREE
C          MYP3                 MY PLUS THREE
C          REACT( )             SLAB REACTION
C          SIGT( )              LARGEST PRINCIPAL STRESS
C          TMX( )               X-TWISTING MOMENT
C          W( )                 DEFLECTION
C
C
      DIMENSION     BETA(L1,L2),   BMX(L1,L2),    BMY(L1,L2),          24JA8
     1              REACT(L1,L2),  SIGT(L1,L2),   TMX(L1,L2),          24JA8
     2              W(L1,L2)                                            4JA8
C
      COMMON   I, ISTA, J, JSTA, NUM(12), HUM(36), NT1(11), HT1(2),    24JA8
     1         MYP2, MXP3, MYP3                                        24JA8
C
    6 FORMAT ( )                                                       09NO7
   39 FORMAT (// 25H     TABLE 8.  RESULTS     ///                     24JA8
     1           45H                                  X          Y     24JA8
     2           40H      X       LARGEST                BETA , /     24JA8
     3           45H                              BENDING    BENDING   24JA8
     4           40H  TWISTING   PRINCIPAL      NET      X TO , /     24JA8
     5           45H      I, J  DEFLECTION   MOMENT      MOMENT       24JA8
     6           40H   MOMENT     STRESS    REACTION   LARG , /       02FE8
     7           45H     (X) (Y)     (L)      (LF/L)     (LF/L)       02FE8
     8           40H   (LF/L)     (F/LL)       (F)     (DEG), // )    02FE8
   45 FORMAT ( 6X, I2, 1X, I3, 6E11.3, F7.2 )                          24JA8
C
C
C-----PRINT W, BMX, BMY, TMX, SIGT, REACT, AND BETA
C
      PRINT 39                                                         02NO6
C
          DO  200  J = 3, MYP3                                         24JA8
C
               JSTA  =  J  -  3                                        08SE7
C
      PRINT  6                                                         21JL7
C
          DO  100  I = 3, MXP3                                         24JA8
C
               ISTA  =  I  -  3                                        08SE7
```

```
C
      PRINT 45, ISTA, JSTA, W(I,J), BMX(I,J), BMY(I,J), TMX(I,J),          12SE7
     1                      SIGT(I,J), REACT(I,J), BETA(I,J)               24JA8
C
  100     CONTINUE                                                         23JA8
C
  200     CONTINUE                                                         23JA8
C
C-----RETURN TO SUBROUTINE OUTPTS
C
      RETURN                                                               23JA8
C
      END                                                                  22JA8
```

```
      SUBROUTINE OUTPT5 ( BMX, BMY, TMX, REACT, SIGT, W, BETA, L1, L2 ) 24JA8
C
C-----THIS SUBROUTINE PRINTS THE SHORT FORM OF TABLE 8 WITH PRIN MOMENT.
C
C
C-----NOTATION FOR SUBOUTINE OUTPT5
C
C         BETA( )              ANGLE FROM X TO LARGEST PRINCIPAL MOMENT
C         BMX( )               X-BENDING MOMENT
C         BMY( )               Y-BENDING MOMENT
C         HT1, HUM, NT1, NUM   POSITIONING VARIABLES IN COMMON FIELD
C         I, J                 COUNTERS IN DO LOOPS
C         ISTA, JSTA           EXTERNAL STATION NUMBER
C         L1, L2               TRANSFER CONSTANTS
C         MXP3                 MX PLUS THREE
C         MYP3                 MY PLUS THREE
C         REACT( )             SLAB REACTION
C         SIGT( )              LARGEST PRINCIPAL MOMENT
C         TMX( )               X-TWISTING MOMENT
C         W( )                 DEFLECTION
C
C
      DIMENSION     BETA(L1,L2),   BMX(L1,L2),    BMY(L1,L2),          24JA8
     1              REACT(L1,L2),  SIGT(L1,L2),   TMX(L1,L2),          24JA8
     2              W(L1,L2)                                           24JA8
C
      COMMON   I, ISTA, J, JSTA, NUM(12), HUM(36), NT1(11), HT1(2),    24JA8
     1         MYP2, MXP3, MYP3                                        24JA8
C
    6 FORMAT ( )                                                       09NO7
   40 FORMAT ( //30H     TABLE 8.  RESULTS           ///               24JA8
     1           45H                                  X          Y     24JA8
     2           40H      X       LARGEST                 BETA , /    24JA8
     3           45H                                BENDING   BENDING  24JA8
     4           40H  TWISTING   PRINCIPAL     NET      X TO , /      24JA8
     5           45H      I, J  DEFLECTION   MOMENT      MOMENT        24JA8
     6           40H   MOMENT     MOMENT    REACTION   LARG , /       02FE8
     7           45H     (X) (Y)     (L)      (LF/L)     (LF/L)       02FE8
     8           40H   (LF/L)     (LF/L)       (F)     (DEG), // )     02FE8
   45 FORMAT ( 6X, I2, 1X, I3, 6E11.3, F7.2 )                          24JA8
C
C
C-----PRINT W, BMX, BMY, TMX, SIGT, REACT, AND BETA
C
      PRINT 40                                                         23JA8
C
          DO  200  J = 3, MYP3                                         24JA8
C
               JSTA  =  J  -  3                                        11SE7
C
      PRINT 6                                                          02AG7
C
          DO  100  I = 3, MXP3                                         24JA8
C
               ISTA  =  I  -  3                                        28SE7
```

```
C
      PRINT 45, ISTA, JSTA, W(I,J), BMX(I,J), BMY(I,J), TMX(I,J),          24JA8
     1                      SIGT(I,J), REACT(I,J), BETA(I,J)               24JA8
C
  100     CONTINUE                                                         23JA8
C
  200     CONTINUE                                                         23JA8
C
C-----RETURN TO SUBROUTINE OUTPTS
C
      RETURN                                                               23JA8
C
      END                                                                  23JA8
```

```
      SUBROUTINE CHANGE
C
C-----THIS SUBROUTINE SEARCHES FOR A PROBLEM NUMBER CARD
C
C
C-----NOTATION FOR SUBROUTINE CHANGE
C
C          AN1( ), AN2( )       ALPHANUMERIC REMARKS, INFORMATION ONLY
C          HT1, HUM, MXP,       POSITIONING VARIABLES IN COMMON FIELD
C          NT1, NUM
C          ITEST                BLANK FIELD FOR TESTING PROBLEM NUMBER
C          KCHEK                PROBLEM NUMBER CARD IDENTIFIER
C          N                    COUNTER
C          NPROB                PROBLEM NUMBER
C
C
      COMMON / CON / L1, L2, ITEST                                          23JA8
C
      COMMON   N, NUM(15), HUM(36), NT1(11), HT1(2), MXP(8), AN1(32),       24JA8
     1         AN2(14), NPROB, KCHEK                                        23JA8
C
   10 FORMAT ( //30H     ********************** ,                           03FE8
     1          /30H     *                    * ,                           03FE8
     2          /30H     * INCORRECT CARD COUNT * ,                         03FE8
     3          /30H     *                    * ,                           03FE8
     4          /30H     ********************** , /                         03FE8
     5          //50H    THE FOLLOWING CARDS ARE NOT PROBLEM NUMBER CA,     03FE8
     6            3HRDS,     //     )                                       03FE8
   15 FORMAT ( 16A5 )                                                       23JA8
   20 FORMAT ( 5X, 16A5 )                                                   23JA8
C
C
C-----PRINT ERROR STATEMENT
C
      PRINT 10                                                              23JA8
C
C-----PRINT CARD JUST READ
C
  100 PRINT 20, NPROB, KCHEK, ( AN2(N), N = 1, 14 )                         23JA8
C
C-----READ NEXT CARD
C
      READ  15, NPROB, KCHEK, ( AN2(N), N = 1, 14 )                         23JA8
C
C-----CHECK CARD JUST READ
C
          IF ( ( KCHEK - ITEST ) .NE. 0 )    GO TO 100                      23JA8
C
C-----CALL TIC TOC WHEN A PROBLEM NUMBER CARD IS FOUND
C
      CALL TIC TOC ( 3 )                                                    23JA8
C
C-----RETURN TO MAIN PROBLEM
C
      RETURN                                                                23JA8
C
      END                                                                   23JA8
```

```
      SUBROUTINE ERROR                                                  23JA8
C
C-----THIS SUBROUTINE EJECTS IMPROPER PROBLEMS.
C
      COMMON   KERR                                                     15DE7
C
   90 FORMAT ( //30H     NUMBER OF DATA ERRORS    , I5 )               15DE7
   91 FORMAT ( //45H     ************************************ ,        03FE8
     1           /45H     *                                  * ,        03FE8
     2           /45H     * ERROR IN INPUT - PROBLEM TERMINATED * ,     03FE8
     3           /45H     *                                  * ,        03FE8
     4           /45H     ************************************ , / )    03FE8
C
C
C-----PRINT ERROR MESSAGE
C
      PRINT 91
C
      PRINT 90, KERR                                                    15DE7
C
C-----CALL TIC TOC
C
      CALL TIC TOC ( 3 )                                                15DE7
C
C-----RETURN TO MAIN PROBLEM
C
      RETURN
C
      END
```

```
      SUBROUTINE TIC TOC (J)                                                240C6
C
C-----THIS SUBROUTINE PRINTS THE CPU TIME.
C
C
C-----NOTATION FOR SUBROUTINE TICTOC
C
C          AN, HT1, HUM, MXP,  POSITIONING VARIABLES IN COMMON FIELD
C          NT1, NUM
C          F                   ELAPSED CPU TIME
C          FI2, FI3, FI4,      TEMPORARY TIME FUNCTIONS
C          II1, I1, I2
C          I                   J MINUS TWO
C          J                   TIME OPTION
C
C
      COMMON   KERR, I, II1, I1, I2, NUM(11), FI2, FI3, FI4, HUM(33),   07FE8
     1         NT1(11), HT1(2), MXP(8), AN(48), F                       23JA8
C
   10 FORMAT(///30X19HELAPSED CPU TIME = I5,8H MINUTESF9.3,8H SECONDS ) 25SE6
   11 FORMAT(///30X15HCOMPILE TIME = ,I5,8H MINUTES,F9.3,8H SECONDS   ) 25SE6
   12 FORMAT(///30X24HTIME FOR THIS PROBLEM = ,I5,8H MINUTES,F9.3,      25SE6
     1       8H SECONDS  )                                              25SE6
C
C-----IF  J  =  1   PRINT COMPILE TIME
C               2   PRINT ELAPSED CPU TIME
C               3   PRINT TIME FOR THIS PROBLEM
C               4   PRINT BOTH 2 AND 3
C
C
               I  =  J - 2                                              02FE8
C
          IF ( I - 1 )                  40, 30, 30                      02FE8
C
   30           FI4  =  F                                               02FE8
C
C-----CALL THE SYSTEM TIME ROUTINE
C
   40 CALL SECOND (F)                                                   25SE6
C
               II1  =  F                                                02FE8
               I1   =  II1 / 60                                         02FE8
               FI2  =  F - I1 * 60                                      02FE8
C
          IF ( I )                      50, 70, 60                      02FE8
C
C-----PRINT COMPILE TIME
C
   50 PRINT 11, I1, FI2                                                 21JY7
C
          GO TO 990                                                     25SE6
C
   60           FI3  =  F - FI4                                         02FE8
               I2   =  FI3 / 60                                         02FE8
               FI3  =  FI3 - I2 * 60                                    02FE8
```

```
C
C-----PRINT TIME FOR THIS PROBLEM
C                                                                      25SE6
      PRINT 12, I2, FI3
C                                                                      02FE8
          IF ( I - 1 )                    990, 990, 70
C
C-----PRINT ELAPSED CPU TIME
C                                                                      21JY7
   70 PRINT 1J, I1, FI2
C                                                                      25SE6
  990 CONTINUE
C
C-----RETURN TO CALLING ROUTINE
C                                                                      25SE6
      RETURN
C                                                                      25SE6
      END
```

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

APPENDIX 4

HAND AND MACHINE LISTINGS OF CODED DATA INPUT EXAMPLE PROGRAMS

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 1 OF 10

```
CE9740O5  HIGHWAY SLAB PROJECT   VISAB 3   C M PEARRE
SAMPLE INPUT FOR EXAMPLE PROBLEMS FOR VISAB 3 FOR USE IN APPENDIX 4

EX 1      BRIDGE APPROACH SLAB EXAMPLE
    1   3   1   6   1   0  12  16   0   1   0      2.500E-01 0.0
    1  12 2.400E+01                   1  16 2.400E+01
    0   0  12  16 3.556E+08 3.556E+08
    0   7  12   7          -3.559E+08
    6   0   6  16-3.559E+08
    1   1  12  16 2.667E+08 2.667E+08
    0   0  12  16          -1.042E+00
    0   7  12  16                      1.000E+02
    0   7  12   7                      5.000E+01
    4  15   4  15-1.000E+04
    1  16   1  15-1.000E+04
    0   0  12   1 1.000E+99
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 2 OF 10

```
    1   0  12  16                                  -5.000E+03

EX 2      SLAB ON FOUNDATION
    3   1   1   2   0   0  16  16   0   1   0      1.500E-01 1.000E+01
    1  16 2.560E+01                   1  16 2.560E+01
    6  11-1.760E+01                   6  11-1.760E+01
    7  10-4.000E+00                   7  10-4.000E+00
    0   0  16  16 2.557E+08 2.557E+08
    1   1  16  16 2.179E+08 2.179E+08
    0   0  16  16                      2.000E+02
    8   8   8   8-1.000E+05

EX 3      SIMPLY SUPPORTED SQUARE STEEL PLATE
    3   1   1   5   0   1   8   8   0   1   0      2.500E-01 0.0
    1   2 8.000E+00                   1   2 8.000E+00
    3   6 4.000E+00                   3   6 4.000E+00
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 3 OF 10

```
 7  8 8.000E+00                          7  8 8.000E+00
 0  0  8  8 2.600E+06 2.600E+06
 1  1  8  8 1.875E+06 1.875E+06
 4  4  4  4 1.000E+05
 0  0  0  8                    1.000E+99
 0  0  8  0                    1.000E+99
 8  8  8  8                    1.000E+99
 0  8  8  8                    1.000E+99
 0  1  8  8                                1.667E+04

EX 4    CORPS OF ENGINEERS MODEL TEST --- SLAB 5
 7  1  1  5  0  0 18 20  0  1  0     2.500E-01 1.310E-01
 1  5 1.000E+00                   1  6 1.000E+00
 6  6 1.110E+00                   7  8 7.450E-01
 7  8 6.000E+00                   9  9 3.600E-01
 9 10 3.800E-01                  10 11 3.950E-01
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 4 OF 10

```
11 12 6.000E-01                     12 12 3.600E-01
13 13 1.110E+00                     13 14 7.450E-01
14 15 1.000E+00                     15 20 1.000E+00
 0  0 18 20 7.199E+03 7.199E+03
 1  1 18 20 5.395E+03 5.395E+03
 0  0 18 20                                3.500E+01
 7  9  7  9 6.000E+00
 7 11  7 11 6.000E+00
11  9 11  9 6.000E+00
11 11 11 11 6.000E+00

EX 5    SIMPLY SUPPORT SLAB WITH COT
11  1  2  5  0  0 39 40  0  0  0     3.360E-01 2.500E-01
 1  1 1.000E+00                   1  1 1.000E+00
 2  5 2.500E-01                   2  5 2.500E-01
 6  6 6.000E-01                   6  6 5.000E-01
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 5 OF 10

```
 7  16 1.000E+00                    7  15 1.000E+00
16  23 2.500E-01                   16  16 5.000E-01
24  28 1.000E+00                   17  24 1.250E-01
29  31 2.500E-01                   25  25 6.000E-01
32  34 1.000E+00                   26  34 1.000E+00
35  36 2.500E-01                   35  35 5.000E-01
37  39 1.000E+00                   36  39 2.000E-01
                                   40  40 1.000E+00
 0   0 39  40 1.600E+04 1.600E+04
29  16 39  22-1.600E+04-1.600E+04
 1   1 39  40 1.061E+04 1.061E+04
30  17 39  22-1.061E+04-1.061E+04
 3   3  3   3                          1.000E+99
 3  37  3  37                          1.000E+99
36   3 36   3                          1.000E+99
36  37 36  37                          1.000E+99
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 6 OF 10

```
17  16  21  29                -4.000E+01

EX  6     CONTINUOUS BRIDGE SLAB
 7   3   3  21   0   0  50  73   1   1   0      2.000E-01 1.050E+01
 1  50 1.200E+01                1  73 1.200E+01
 5   6 1.200E+01               60  73 2.400E+01
10  11 1.200E+01
16  16 1.200E+01
35  35 1.200E+01
40  41 1.200E+01
46  96 1.200E+01
 0   0  50  73 3.015E+08 3.015E+08
 0   0   1  73 3.760E+08 3.760E+08
44   0  50  73 3.750E+08 3.750E+08
 1   1  50  73 2.412E+08 2.412E+08
 1   1   1  73 3.000E+08 3.000E+08
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 7 OF 10

```
50  1  50 73 3.000E+08 2.000E+08
 0  0  50 73          -9.115E-01
 0  0   1 73          -8.681E-01
99  0  50 73          -8.681E-01
 0  0  50  0                    1.000E+99
 0 18  50 18                    1.000E+99
 0 39  50 39                    1.000E+99
 0 60  50 60                    1.000E+99
 0 67  50 67                    1.000E+99
 0 73  50 73                    1.000E+99
13  6  13  6-1.560E+04
18  6  18  6-1.560E+04
22  6  22  6-1.560E+04
28  6  28  6-1.560E+04
32  6  32  6-1.560E+04
37  6  37  6-1.560E+04
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 8 OF 10

```
13 10  13 10-1.560E+04
18 10  18 10-1.560E+04
22 10  22 10-1.560E+04
28 10  28 10-1.560E+04
32 10  32 10-1.560E+04
37 10  37 10-1.560E+04

EX 6A     CONTINUOUS BRIDGE SLAB
 0  0   0 15  0  0 50 73 -1  1  0      2.000E-01 1.000E+01
 0  0  50 73          -9.115E-01
 0  0   1 73          -8.681E-01
99  0  50 73          -8.681E-01
13 11  13 11-2.080E+04
18 11  18 11-2.080E+04
22 11  22 11-2.080E+04
28 11  28 11-2.080E+04
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 9 OF 10

```
 32  11  32  11-2.080E+04
 37  11  37  11-2.080E+04
 13  26  13  26-2.080E+04
 18  26  18  25-2.080E+04
 22  25  22  25-2.080E+04
 28  25  28  25-2.080E+04
 32  25  32  25-2.080E+04
 37  26  37  25-2.080E+04

EX 6B    CONTINUOUS BRIDGE SLAB
  0   0   0   7   0   0  50  73  -1   1   0    2.000E-01 1.000E+01
  0   0  60  73          -9.115E-01
  0   0   1  73          -8.681E-01
 49   0  50  73          -8.681E-01
 43  24  43  26-1.560E+04
 48  24  48  26-1.560E+04
```

IDENTIFICATION EXAMPLE PROBLEMS CODED BY CMP DATE 3-20-68 PAGE 10 OF 10

```
 43  30  43  30-1.560E+04
 48  30  48  30-1.560E+04
```

```
CE994005   HIGHWAY SLAB PROJECT   VISAB 3   C M PEARRE
SAMPLE INPUT FOR EXAMPLE PROBLEMS FOR VISAB 3 FOR USE IN APPENDIX 4


EX  1      BRIDGE APPROACH SLAB EXAMPLE
    1     3    1    6    1    0   12   16    0    1    0       2.500E-01 0.0
    1    12 2.400E+01               1   16 2.400E+01
    0     0   12   16 3.556E+08 3.556E+08
    0     7   12    7          -3.554E+08
    6     0    6   16-3.554E+08
    1     1   12   16 2.667E+08 2.667E+08
    0     0   12   16          -1.042E+00
    0     7   12   16                                1.000E+02
    0     7   12    7                                5.000E+01
    4    15    4   15-1.000E+04
    1    15    1   15-1.000E+04
    0     0   12    1                     1.000E+99
    1     0   12   16                               -5.000E+03


EX  2      SLAB ON FOUNDATION
    3     1    1    2    0    0   16   16    0    1    0       1.500E-01 1.000E+01
    1    16 2.560E+01               1   16 2.560E+01
    6    11-1.760E+01               6   11-1.760E+01
    7    10-4.000E+00               7   10-4.000E+00
    0     0   16   16 2.557E+08 2.557E+08
    1     1   16   16 2.174E+08 2.174E+08
    0     0   16   16                                2.000E+02
    8     8    8    8-1.000E+05


EX  3      SIMPLY SUPPORTED SQUARE STEEL PLATE
    3     1    1    5    0    1    8    8    0    1    0       2.500E-01 0.0
    1     2 8.000E+00               1    2 8.000E+00
    3     6 4.000E+00               3    6 4.000E+00
    7     8 8.000E+00               7    8 4.000E+00
    0     0    8    8 2.500E+06 2.500E+06
    1     1    8    8 1.875E+06 1.875E+06
    4     4    4    4 1.000E+05
    0     0    0    8                     1.000E+99
    0     0    8                          1.000E+99
    8     0    8    8                     1.000E+99
    0     8    8    8                     1.000E+99
    0     1    8    8                                1.667E+04
```

```
EX  4     CORPS OF ENGINEERS MODEL TEST --- SLAB 5
    7     1     1     5     0     0    18    20     0     1     0       2.500E-01 1.310E-01
   14    18 1.000E+00                 15    20 1.000E+00
    1     5 1.000E+00                  1     6 1.000E+00
    6     6 1.110E+00                  7     8 7.450E-01
    7     8 6.000E-01                  9     9 3.600E-01
    9    10 3.800E-01                 10    11 3.950E-01
   11    12 6.000E-01                 12    12 3.600E-01
   13    13 1.110E+00                 13    14 7.450E-01
    0     0    18    20 7.194E+03 7.194E+03
    1     1    18    20 5.395E+03 5.395E+03
    0     0    18    20                                  3.500E+01
    7     9     7     9-6.000E+00
    7    11     7    11-6.000E+00
   11     9    11     9-6.000E+00
   11    11    11    11-6.000E+00


EX  5     SIMPLY SUPPORTED PLATE WITH CUT
   11     2     2     5     0     0    39    40     0     0     0       3.360E-01 2.500E-01
    1     1 1.000E+00                  1     1 1.000E+00
    2     5 2.500E-01                  2     5 2.500E-01
    6     6 5.000E-01                  6     6 5.000E-01
    7    15 1.000E+00                  7    15 1.000E+00
   16    23 2.500E-01                 16    16 5.000E-01
   24    28 1.000E+00                 17    24 1.250E-01
   29    31 2.500E-01                 25    25 5.000E-01
   32    34 1.000E+00                 26    34 1.000E+00
   35    38 2.500E-01                 35    35 5.000E-01
   39    39 1.000E+00                 36    39 2.500E-01
                                      40    40 1.000E+00
    0     0    39    40 1.600E+04 1.600E+04
   29    18    39    22-1.600E+04-1.600E+04
    1     1    39    40 1.061E+04 1.061E+04
   30    19    39    22-1.061E+04-1.061E+04
   17    16    21    24           -4.000E+01
    3     3     3     3                      1.000E+99
    3    37     3    37                      1.000E+99
   36     3    36     3                      1.000E+99
   36    37    36    37                      1.000E+99
```

```
EX  6      CONTINUOUS BRIDGE SLAB
    7    3    3   21    0    0   50   73    1    1    0       2.000E-01 1.050E+01
    1   50 1.200E+01               1   73 1.200E+01
    5    5 1.200E+01              60   73 2.400E+01
   10   11 1.200E+01
   16   16 1.200E+01
   35   35 1.200E+01
   40   41 1.200E+01
   46   46 1.200E+01
    0    0   50   73 3.015E+08 3.015E+08
    0    0    1   73 3.750E+08 3.750E+08
   49    0   50   73 3.750E+08 3.750E+08
    1    1   50   73 2.412E+08 2.412E+08
    1    1    1   73 3.000E+08 3.000E+08
   50    1   50   73 3.000E+08 3.000E+08
    0    0   50   73          -9.115E-01
    0    0    1   73          -8.681E-01
   49    0   50   73          -8.681E-01
    0    0   50    0                    1.000E+99
    0   18   50   18                    1.000E+99
    0   39   50   39                    1.000E+99
    0   60   50   60                    1.000E+99
    0   67   50   67                    1.000E+99
    0   73   50   73                    1.000E+99
   13    6   13    6-1.560E+04
   18    6   18    6-1.560E+04
   22    6   22    6-1.560E+04
   28    6   28    6-1.560E+04
   32    6   32    6-1.560E+04
   37    6   37    6-1.560E+04
   13   10   13   10-1.560E+04
   18   10   18   10-1.560E+04
   22   10   22   10-1.560E+04
   28   10   28   10-1.560E+04
   32   10   32   10-1.560E+04
   37   10   37   10-1.560E+04
```

```
EX 6A      CONTINUOUS BRIDGE SLAB
    0     0     0    15     0     0    50    73    -1     1     0       2.000E-01 1.050E+01
    0     0    50    73              -9.115E-01
    0     0     1    73              -8.681E-01
   49     0    50    73              -8.681E-01
   13    11    13    11-2.080E+04
   18    11    18    11-2.080E+04
   22    11    22    11-2.080E+04
   28    11    28    11-2.080E+04
   32    11    32    11-2.080E+04
   37    11    37    11-2.080E+04
   13    25    13    25-2.080E+04
   18    25    18    25-2.080E+04
   22    25    22    25-2.080E+04
   28    25    28    25-2.080E+04
   32    25    32    25-2.080E+04
   37    25    37    25-2.080E+04


EX 6B      CONTINUOUS BRIDGE SLAB
    0     0     0     7     0     0    50    73    -1     1     0       2.000E-01 1.050E+01
    0     0    50    73              -9.115E-01
    0     0     1    73              -8.681E-01
   49     0    50    73              -8.681E-01
   43    26    43    26-1.560E+04
   48    26    48    26-1.560E+04
   43    30    43    30-1.560E+04
   48    30    48    30-1.560E+04
```

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

APPENDIX 5

SAMPLE OUTPUT FORMS
Table 8 Only

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

```
PROGRAM VISAB 3 -   MASTER DECK    - PEARRE - PROGRAM REVISION DATE - 02 MAR 1968
CE994005  ---  RUN FOR EXAMPLE OF OUTPUT FORMS
               RUN ON 12 MARCH 1968


PROB (CONTD)
FORM1      LONG FORM OF OUTPUT WITH STRESSES

TABLE 8.  PART 1.   RESULTS

                                    X          Y          X          Y
 I,  J  DEFLECTION              BENDING    BENDING    TWISTING   TWISTING       NET
(X) (Y)      (L)               MOMENT     MOMENT     MOMENT     MOMENT     REACTION
                               (LF/L)     (LF/L)     (LF/L)     (LF/L)        (F)

  0   0 -1.376E-02  1.355E-11  2.575E-10 -5.246E+00  5.246E+00  3.440E+01
  1   0 -1.425E-02  5.455E+01  4.879E-10 -1.886E+01  1.886E+01  7.125E+01
  2   0 -1.469E-02  1.408E+02  8.131E-10 -2.612E+01  2.612E+01  7.346E+01
  3   0 -1.502E-02  2.361E+02  9.693E-10 -2.612E+01  2.108E+01  7.508E+01
  4   0 -1.513E-02  3.313E+02  9.693E-10 -2.612E+01  2.612E+01  7.346E+01
  7   7 -1.425E-02  5.455E+01  4.065E-10 -1.886E+01  1.886E+01  7.125E+01
  8   8 -1.376E-02 -1.220E-10 -8.132E-11 -5.246E+00  5.246E+00  3.440E+01

     SUMMATION OF REACTIONS  =       .00000000


TABLE 8.  PART 2.   RESULTS

                MAXIMUN    MINIMUM                MAXIMUN    MINIMUM                BETA
                PRINCIPAL  PRINCIPAL  MAXIMUN     PRINCIPAL  PRINCIPAL    SHEAR     X TO
 I,  J          MOMENT     MOMENT     TORQUE      STRESS     STRESS       STRESS    MAX.
(X) (Y)         (LF/L)     (LF/L)     (LF/L)      (F/LL)     (F/LL)       (F/LL)    (DEG)

  0   0  5.246E+00 -5.246E+00  5.246E+00  3.148E-01 -3.148E-01  3.148E-01  45.00
  1   0  6.043E+01 -5.883E+00  3.316E+01  3.626E+00 -3.530E-01  1.989E+00  17.33
  2   0  1.455E+02 -4.692E+00  7.508E+01  8.728E+00 -2.815E-01  4.505E+00  10.18
  3   0  2.379E+02 -1.869E+00  1.210E+02  8.728E+00 -1.121E-01  7.194E+00   5.06
  4   0  2.849E+02 -1.869E+00  1.479E+02  8.728E+00 -1.121E-01  7.194E+00    .00
  7   7  6.043E+01 -5.883E+00  3.316E+01  3.626E+00 -3.530E-01  1.989E+00  17.33
  8   8  5.246E+00 -5.246E+00  5.246E+00  3.148E-01 -3.148E-01  3.148E-01  45.00

                        TIME FOR THIS PROBLEM =      0 MINUTES     1.793 SECONDS
```

Fig A5.1. Long form of output with principal stresses.

```
PROGRAM VISAB 3 -   MASTER DECK    - PEARRE - PROGRAM REVISION DATE - 02 MAR 1968
CE994005  ---  RUN FOR EXAMPLE OF OUTPUT FORMS
               RUN ON 12 MARCH 1968


PROB (CONTD)
FORM2     LONG FORM OF OUTPUT WITHOUT STRESSES

TABLE 8.  PART 1.   RESULTS

                                    X          Y          X          Y
 I,  J  DEFLECTION              BENDING    BENDING    TWISTING   TWISTING      NET
(X) (Y)      (L)               MOMENT     MOMENT     MOMENT     MOMENT     REACTION
                                (LF/L)     (LF/L)     (LF/L)     (LF/L)       (F)

  0   0 -1.376E-02  1.355E-11  2.575E-10 -5.246E+00  5.246E+00  3.440E+01
  1   0 -1.425E-02  5.455E+01  4.879E-10 -1.886E+01  1.886E+01  7.125E+01
  2   0 -1.469E-02  1.408E+02  8.131E-10 -2.612E+01  2.612E+01  7.346E+01
  3   0 -1.502E-02  2.361E+02  1.093E-10 -2.612E+01  2.612E+01  7.346E+01
  4   0 -1.514E-02  2.361E+02  4.065E-10 -1.886E+01  1.886E+01  7.125E+01
  8   8 -1.376E-02 -1.220E-10 -8.132E-11 -5.246E+00  5.246E+00  3.440E+01

     SUMMATION OF REACTIONS  =       .00000000


TABLE 8.  PART 2.   RESULTS

                 MAXIMUN     MINIMUM                    BETA
                PRINCIPAL   PRINCIPAL    MAXIMUN       X TO
 I,  J           MOMENT      MOMENT      TORQUE        MAX.
(X) (Y)          (LF/L)      (LF/L)      (LF/L)        (DEG)

  0   0  5.246E+00 -5.246E+00  5.246E+00   45.00
  1   0  6.043E+01 -5.883E+00  3.316E+01   17.33
  2   0  1.455E+02 -4.692E+00  7.508E+01   10.18
  3   0  2.379E+02 -1.869E+00  7.508E+01   10.18
  4   0  2.379E+02 -5.883E+00  3.316E+01   17.33
  8   8  5.246E+00 -5.246E+00  5.246E+00   45.00

                          TIME FOR THIS PROBLEM =      0 MINUTES     1.741 SECONDS
```

Fig A5.2. Long form of output without principal stresses.

```
PROGRAM VISAB 3 -   MASTER DECK    - PEARRE - PROGRAM REVISION DATE - 02 MAR 1968
CE994005  ---  RUN FOR EXAMPLE OF OUTPUT FORMS
               RUN ON 12 MARCH 1968


PROB (CONTD)
FORM3      SHORT FORM OF OUTPUT WITH PRINCIPAL STRESS

TABLE 8.  RESULTS

                                    X          Y          X        LARGEST                    BETA
 I,  J  DEFLECTION              BENDING    BENDING    TWISTING      PRINCIPAL        NET      X TO
(X) (Y)       (L)           MOMENT     MOMENT     MOMENT        STRESS     REACTION      LARG
                            (LF/L)     (LF/L)     (LF/L)        (F/LL)         (F)      (DEG)

 0   0 -1.376E-02  1.355E-11  2.575E-10 -5.246E+00  8.744E+01  3.440E+01   45.00
 1   0 -1.425E-02  5.455E+01  4.879E-10 -1.886E+01  1.007E+03  7.125E+01
 2   0 -1.469E-02  1.408E+02  8.131E-10 -2.612E+01                          .00
 3   0 -1.502E-02  2.361E+02                        3.965E+03  7.508E+01    5.06
 4   0 -1.513E-02                       -2.612E+01  2.424E+03  7.346E+01   10.18
                   5.455E+01  4.065E-10 -1.886E+01  1.007E+03  7.125E+01   17.33
 8   8 -1.376E-02 -1.220E-10 -8.132E-11 -5.246E+00 -8.744E+01  3.440E+01   45.00

     SUMMATION OF REACTIONS  =       .00000000


                          TIME FOR THIS PROBLEM =      0 MINUTES     1.594 SECONDS
```

Fig A5.3. Short form of output with largest principal stress.

```
PROGRAM VISAB 3 -   MASTER DECK    - PEARRE - PROGRAM REVISION DATE - 02 MAR 1968
CE994005  ---  RUN FOR EXAMPLE OF OUTPUT FORMS
               RUN ON 12 MARCH 1968


PROB (CONTD)
FORM4      SHORT FORM OF OUTPUT WITH PRINCIPAL MOMENT

TABLE 8.  RESULTS
```

I, (X)	J (Y)	DEFLECTION (L)	X BENDING MOMENT (LF/L)	Y BENDING MOMENT (LF/L)	X TWISTING MOMENT (LF/L)	LARGEST PRINCIPAL MOMENT (LF/L)	NET REACTION (F)	BETA X TO LARG (DEG)
0	0	-1.376E-02	1.355E-11	2.575E-10	-5.246E+00	5.246E+00	3.440E+01	45.00
1	0	-1.425E-02	5.455E+01	4.879E-10	-1.886E+01	6.043E+01	7.125E+01	
2	0	-1.469E-02	1.408E+02	8.131E-10	-2.612E+01			.00
3	0	-1.502E-02	2.361E+02			2.379E+02	7.508E+01	5.06
4	0			9.693E-10	-2.612E+01	1.455E+02	7.346E+01	10.18
		-1.425E-02	5.455E+01	4.065E-10	-1.886E+01	6.043E+01	7.125E+01	17.33
8	8	-1.376E-02	-1.220E-10	-8.132E-11	-5.246E+00	-5.246E+00	3.440E+01	45.00

```
     SUMMATION OF REACTIONS  =       .00000000


                        TIME FOR THIS PROBLEM =      0 MINUTES     1.642 SECONDS
```

Fig A5.4. Short form of output with largest principal moment.

APPENDIX 6

SAMPLE COMPUTER SOLUTIONS - EXAMPLE PROBLEMS

This page replaces an intentionally blank page in the original.
-- CTR Library Digitization Team

```
PROGRAM VISAB 3 -   MASTER DECK    - PEARRE - DRIVER REVISION DATE - 21 MAR 1968
CE994005    HIGHWAY SLAB PROJECT   VISAB 3        C.M.PEARRE
EXAMPLE PROBLEMS RUN FOR USE IN APPENDIX 6


PROB
EX  1     BRIDGE APPROACH SLAB EXAMPLE


TABLE 1.  CONTROL DATA

     NUM CARDS TABLE 2                                                  1
     NUM CARDS TABLE 3                                                  3
     NUM CARDS TABLE 4                                                  1
     NUM CARDS TABLE 5                                                  6
     NUM CARDS TABLE 6                                                  1
     NUM CARDS TABLE 7                                                  0
     NUM INCREMENTS  MX                                                12
     NUM INCREMENTS  MY                                                16
     POISSONS RATIO                                                .2500
     SLAB THICKNESS                                        0.

     MUL LOAD OPTION ( IF + OR - 1 )                                    0

     OPTION ( IF = 1 ) FOR SHORT FORM OF OUTPUT                         1

     OPTION ( IF = 1 ) TO PRINT INPUT VARIABLE ARRAYS                   0


TABLE 2.  INCREMENT LENGTH

  FROM   THRU       HX                    FROM   THRU       HY
 X-BAR  X-BAR      (L)                   Y-BAR  Y-BAR      (L)

    1     12    2.400E+01                    1     16    2.400E+01


TABLE 3.  BENDING STIFFNESS

  FROM    THRU       DX          DY
   STA     STA      (FL)        (FL)

  0    0 12   16  3.556E+08  3.556E+08
  0    7 12    7 -0.         -3.554E+08
  6    0  6   16 -3.554E+08 -0.


TABLE 4.  TWISTING STIFFNESS

  FROM    THRU       CX          CY
   STA     STA      (FL)        (FL)

  1    1 12   16  2.667E+08  2.667E+08


TABLE 5.  LOAD AND SUPPORT DATA

  FROM    THRU        Q          QQ           S          SS
```

STA		STA		(F)	(F/LL)	(F/L)	(F/LLL)
0	0	12	16	-0.	-1.042E+00	-0.	-0.
0	7	12	16	-0.	-0.	-0.	1.000E+02
0	7	12	7	-0.	-0.	-0.	5.000E+01
4	15	4	15	-1.000E+04	-0.	-0.	-0.
1	15	1	15	-1.000E+04	-0.	-0.	-0.
0	0	12	1	-0.	-0.	1.000E+99	-0.

TABLE 6. X-DIRECTIONAL LOAD DATA

FROM X-BAR		THRU X-BAR		TX (FL)	TTX (FL/L)	PX (F)	PPX (F/L)
1	0	12	16	-0.	-0.	-0.	-5.000E+03

TABLE 7. Y-DIRECTIONAL LOAD DATA

FROM Y-BAR		THRU Y-BAR		TY (FL)	TTY (FL/L)	PY (F)	PPY (F/L)

NO DATA INPUT FOR THIS TABLE

PROGRAM VISAB 3 - MASTER DECK - PEARRE - PROGRAM REVISION DATE - 21 MAR 1968
CE994005 HIGHWAY SLAB PROJECT VISAB 3 C.M.PEARRE
EXAMPLE PROBLEMS RUN FOR USE IN APPENDIX 6

PROB (CONTD)
EX 1 BRIDGE APPROACH SLAB EXAMPLE

TABLE 8. RESULTS

I, (X)	J (Y)	DEFLECTION (L)	X BENDING MOMENT (LF/L)	Y BENDING MOMENT (LF/L)	X TWISTING MOMENT (LF/L)	LARGEST PRINCIPAL MOMENT (LF/L)	NET REACTION (F)	BETA X TO LARG (DEG)
0	0	4.942E-96	-1.271-104	-5.083-104	-6.928E+01	-6.928E+01	-1.386E+03	45.00
1	0	5.193E-96	-1.044E-91	-9.244-104	-3.366E-91	-3.928E-91	-2.896E+03	40.59
2	0	5.082E-96	1.961E-92	-6.933-104	2.416E-91	2.516E-91	-2.841E+03	-43.84
3	0	5.040E-96	1.898E-92	-9.009-104	5.683E-93	2.055E-92	-2.820E+03	-15.46
4	0	5.063E-96	1.817E-92	-9.012-104	-2.203E-91	2.296E-91	-2.832E+03	43.82
5	0	5.149E-96	3.728E-94	-1.224-103	2.498E-91	2.500E-91	-2.874E+03	-44.98
6	0	5.236E-96	2.833E-93	1.143E-92	-6.711E-93	1.510E-92	-2.918E+03	61.32
7	0	5.175E-96	3.790E-94	-1.409-103	-2.632E-91	2.634E-91	-2.887E+03	44.98
8	0	5.115E-96	1.829E-92	-9.472-104	2.068E-91	2.161E-91	-2.858E+03	-43.73
9	0	5.118E-96	1.932E-92	-9.012-104	-2.010E-92	3.196E-92	-2.859E+03	32.16
10	0	5.189E-96	2.005E-92	-8.549-104	-2.634E-91	2.736E-91	-2.894E+03	43.91
11	0	5.328E-96	-1.081E-91	-1.086-103	3.221E-91	-3.807E-91	-2.964E+03	-40.24
12	0	5.094E-96	-1.964-104	-4.390-104	7.118E+01	-7.118E+01	-1.424E+03	-45.00
0	1	-7.935E-96	-1.819E-12	-1.386E+03	-1.148E+01	-1.386E+03	1.684E+03	89.53
1	1	-1.251E-95	-7.241E+02	-2.896E+03	2.149E+01	-2.897E+03	5.655E+03	-89.43
2	1	-1.071E-95	-7.103E+02	-2.841E+03	1.433E+01	-2.841E+03	4.755E+03	-89.61
3	1	-1.063E-95	-7.050E+02	-2.820E+03	1.831E+00	-2.820E+03	4.715E+03	-89.95
4	1	-1.068E-95	-7.079E+02	-2.832E+03	-1.020E+01	-2.832E+03	4.740E+03	89.72
5	1	-1.242E-95	-7.186E+02	-2.874E+03	-1.621E+01	-2.875E+03	4.879E+03	89.57
6	1	-8.323E-96	-7.295E+02	-2.918E+03	-2.441E+00	-2.918E+03	5.001E+03	89.94
7	1	-1.245E-95	-7.219E+02	-2.887E+03	1.133E+01	-2.887E+03	4.895E+03	-89.70
8	1	-1.074E-95	-7.144E+02	-2.858E+03	5.281E+00	-2.858E+03	4.772E+03	-89.86
9	1	-1.073E-95	-7.148E+02	-2.859E+03	-6.903E+00	-2.859E+03	4.763E+03	89.82
10	1	-1.084E-95	-7.236E+02	-2.894E+03	-1.966E+01	-2.895E+03	4.822E+03	89.48
11	1	-1.274E-95	-7.410E+02	-2.964E+03	-2.672E+01	-2.964E+03	5.768E+03	89.31
12	1	-8.148E-96	-1.819E-12	-1.424E+03	1.027E+01	-1.424E+03	1.737E+03	-89.59
0	2	-4.788E-03	0.	-4.527E+02	1.226E+02	-4.838E+02	-3.001E+02	-75.78
1	2	-4.692E-03	-2.213E+02	-8.692E+02	6.288E+01	-8.753E+02	-6.002E+02	-84.51
2	2	-4.602E-03	-2.488E+02	-8.681E+02	3.021E+01	-8.696E+02	-6.002E+02	-87.21
3	2	-4.568E-03	-2.478E+02	-8.681E+02	3.560E+00	-8.681E+02	-6.002E+02	-89.67
4	2	-4.586E-03	-2.464E+02	-8.681E+02	-2.207E+01	-8.688E+02	-6.002E+02	87.97
5	2	-4.656E-03	-2.178E+02	-8.689E+02	-5.200E+01	-8.731E+02	-8.412E+02	85.46
6	2	-4.727E-03	-2.411E+02	-9.646E+02	-6.765E+00	-9.647E+02	-1.635E+02	89.46
7	2	-4.677E-03	-2.204E+02	-8.790E+02	3.852E+01	-8.812E+02	-8.412E+02	-86.66
8	2	-4.629E-03	-2.516E+02	-8.879E+02	8.667E+00	-8.880E+02	-6.002E+02	-89.22
9	2	-4.631E-03	-2.556E+02	-8.973E+02	-1.702E+01	-8.977E+02	-6.002E+02	88.48
10	2	-4.688E-03	-2.589E+02	-9.058E+02	-4.390E+01	-9.088E+02	-6.002E+02	86.13
11	2	-4.801E-03	-2.313E+02	-9.141E+02	-7.649E+01	-9.226E+02	-6.002E+02	83.69

12	2	-4.919E-03	7.731E-11	-4.782E+02	-1.304E+02	-5.114E+02	-3.001E+02	75.70
0	3	-1.114E-02	-7.731E-11	2.545E+02	1.106E+02	2.958E+02	-3.001E+02	-69.51
1	3	-1.079E-02	3.276E+01	4.977E+02	7.636E+01	5.099E+02	-6.002E+02	-80.91
2	3	-1.060E-02	5.071E+01	4.896E+02	2.763E+01	4.914E+02	-6.002E+02	-86.41
3	3	-1.053E-02	5.938E+01	4.837E+02	1.493E+00	4.837E+02	-6.002E+02	-89.80
4	3	-1.057E-02	5.353E+01	4.805E+02	-2.370E+01	4.818E+02	-6.002E+02	86.83
5	3	-1.072E-02	3.650E+01	4.782E+02	-6.973E+01	4.889E+02	-4.732E+02	81.24
6	3	-1.102E-02	1.272E+02	5.081E+02	-1.005E+01	5.084E+02	-8.483E+02	88.49
7	3	-1.078E-02	3.490E+01	4.708E+02	4.983E+01	4.764E+02	-4.732E+02	-83.56
8	3	-1.068E-02	5.007E+01	4.661E+02	4.299E+00	4.662E+02	-6.002E+02	-89.41
9	3	-1.070E-02	5.381E+01	4.631E+02	-2.035E+01	4.641E+02	-6.002E+02	87.16
10	3	-1.083E-02	4.353E+01	4.639E+02	-4.620E+01	4.689E+02	-6.002E+02	83.80
11	3	-1.108E-02	2.643E+01	4.677E+02	-9.510E+01	4.873E+02	-6.002E+02	78.34
12	3	-1.149E-02	1.062E-10	2.377E+02	-1.208E+02	2.883E+02	-3.001E+02	67.27
0	4	-1.661E-02	2.510E-10	6.584E+02	5.898E+01	6.636E+02	-3.001E+02	-84.92
1	4	-1.604E-02	1.449E+02	1.269E+03	4.635E+01	1.271E+03	-6.002E+02	-87.64
2	4	-1.577E-02	2.053E+02	1.250E+03	1.385E+01	1.250E+03	-6.002E+02	-89.24
3	4	-1.568E-02	2.240E+02	1.240E+03	-3.683E+00	1.240E+03	-6.002E+02	89.79
4	4	-1.574E-02	2.086E+02	1.235E+03	-2.067E+01	1.235E+03	-6.002E+02	88.85
5	4	-1.597E-02	1.490E+02	1.236E+03	-5.179E+01	1.239E+03	-2.635E+02	87.28
6	4	-1.648E-02	3.372E+02	1.347E+03	-1.248E+01	1.348E+03	-1.238E+03	89.29
7	4	-1.608E-02	1.487E+02	1.231E+03	2.724E+01	1.232E+03	-2.635E+02	-88.56
8	4	-1.595E-02	2.074E+02	1.225E+03	-2.781E+00	1.225E+03	-6.002E+02	89.84
9	4	-1.600E-02	2.212E+02	1.226E+03	-1.843E+01	1.227E+03	-6.002E+02	88.95
10	4	-1.619E-02	2.006E+02	1.233E+03	-3.517E+01	1.234E+03	-6.002E+02	88.05
11	4	-1.657E-02	1.403E+02	1.250E+03	-6.820E+01	1.254E+03	-6.002E+02	86.50
12	4	-1.724E-02	6.730E-11	6.477E+02	-7.077E+01	6.553E+02	-3.001E+02	83.84
0	5	-1.981E-02	3.083E-10	7.491E+02	-7.804E+00	7.492E+02	-3.001E+02	89.40
1	5	-1.916E-02	1.589E+02	1.445E+03	-4.084E+00	1.445E+03	-6.002E+02	89.82
2	5	-1.887E-02	2.320E+02	1.422E+03	-7.480E+00	1.422E+03	-6.002E+02	89.64
3	5	-1.878E-02	2.552E+02	1.411E+03	-1.047E+01	1.411E+03	-6.002E+02	89.48
4	5	-1.887E-02	2.357E+02	1.405E+03	-1.340E+01	1.405E+03	-6.002E+02	89.34
5	5	-1.915E-02	1.625E+02	1.409E+03	-1.737E+01	1.409E+03	-2.170E+02	89.20
6	5	-1.977E-02	3.838E+02	1.534E+03	-1.431E+01	1.534E+03	-1.328E+03	89.29
7	5	-1.932E-02	1.640E+02	1.405E+03	-1.057E+01	1.405E+03	-2.170E+02	89.51
8	5	-1.920E-02	2.376E+02	1.398E+03	-1.269E+01	1.399E+03	-6.002E+02	89.37
9	5	-1.927E-02	2.558E+02	1.402E+03	-1.325E+01	1.402E+03	-6.002E+02	89.34
10	5	-1.950E-02	2.295E+02	1.412E+03	-1.477E+01	1.412E+03	-6.002E+02	89.28
11	5	-1.995E-02	1.540E+02	1.435E+03	-1.954E+01	1.435E+03	-6.002E+02	89.13
12	5	-2.075E-02	4.820E-11	7.429E+02	-5.226E+00	7.429E+02	-3.001E+02	89.60
0	6	-2.042E-02	5.730E-11	5.266E+02	-3.055E+01	5.283E+02	-3.482E+02	86.69
1	6	-1.986E-02	7.577E+01	1.025E+03	-4.696E+01	1.027E+03	-6.389E+02	87.17
2	6	-1.961E-02	1.358E+02	1.007E+03	-2.870E+01	1.008E+03	-6.240E+02	88.11
3	6	-1.956E-02	1.572E+02	9.977E+02	-1.690E+01	9.981E+02	-6.195E+02	88.85
4	6	-1.967E-02	1.399E+02	9.936E+02	-5.647E+00	9.936E+02	-6.227E+02	89.62
5	6	-1.997E-02	7.722E+01	9.952E+02	7.348E+00	9.952E+02	-3.778E+02	-89.54
6	6	-2.056E-02	2.645E+02	1.057E+03	-1.568E+01	1.057E+03	-9.960E+02	88.87
7	6	-2.020E-02	8.108E+01	9.932E+02	-3.777E+01	9.947E+02	-3.759E+02	87.63
8	6	-2.013E-02	1.461E+02	9.859E+02	-2.212E+01	9.904E+02	-6.194E+02	88.50
9	6	-2.023E-02	1.626E+02	9.929E+02	-7.132E+00	9.930E+02	-6.161E+02	89.51
10	6	-2.048E-02	1.356E+02	1.002E+03	7.488E+00	1.002E+03	-6.229E+02	-89.50
11	6	-2.092E-02	6.679E+01	1.022E+03	2.246E+01	1.022E+03	-6.449E+02	-88.65
12	6	-2.170E-02	6.730E-11	5.236E+02	4.738E+00	5.236E+02	-3.631E+02	-89.48
0	7	-1.921E-02	-1.925E+02	-4.772E+01	-1.012E+02	-2.445E+02	3.768E+02	27.21
1	7	-1.882E-02	-1.546E+02	-3.789E+01	-8.179E+01	-1.967E+02	7.235E+02	27.25
2	7	-1.867E-02	-9.519E+01	-2.304E+01	-2.701E+01	-1.042E+02	5.129E+02	18.41

3	7	-1.868E-02	-7.711E+01	-1.849E+01	4.635E+00	-7.748E+01	5.072E+02	-4.49
4	7	-1.882E-02	-8.969E+01	-2.159E+01	3.606E+01	-1.052E+02	5.177E+02	-23.32
5	7	-1.910E-02	-1.662E+02	-4.065E+01	8.784E+01	-2.114E+02	1.038E+03	-27.23
6	7	-1.965E-02	5.545E-01	1.175E+00	3.862E+00	4.739E+00	-4.803E+02	-47.30
7	7	-1.940E-02	-1.587E+02	-3.872E+01	-8.065E+01	-1.992E+02	1.053E+03	26.68
8	7	-1.941E-02	-7.661E+01	-1.822E+01	-3.006E+01	-8.932E+01	5.488E+02	22.92
9	7	-1.954E-02	-6.343E+01	-1.492E+01	7.721E-02	-6.343E+01	5.540E+02	-.09
10	7	-1.978E-02	-9.076E+01	-2.174E+01	2.970E+01	-1.018E+02	5.780E+02	-20.36
11	7	-2.016E-02	-1.789E+02	-4.374E+01	7.925E+01	-2.154E+02	8.701E+02	-24.78
12	7	-2.083E-02	-2.521E+02	-6.250E+01	9.448E+01	-2.912E+02	4.696E+02	-22.45
0	8	-1.550E-02	2.169E-10	-3.127E+02	-1.227E+02	-3.551E+02	9.829E+01	70.93
1	8	-1.545E-02	-1.418E+02	-5.771E+02	-7.896E+01	-5.909E+02	2.513E+02	80.03
2	8	-1.540E-02	-1.647E+02	-5.586E+02	-7.615E+00	-5.587E+02	2.630E+02	88.89
3	8	-1.539E-02	-1.601E+02	-5.409E+02	3.585E+01	-5.442E+02	2.669E+02	-84.67
4	8	-1.542E-02	-1.489E+02	-5.205E+02	8.052E+01	-5.372E+02	2.656E+02	-78.29
5	8	-1.549E-02	-1.278E+02	-5.016E+02	1.536E+02	-5.567E+02	1.103E+02	-70.29
6	8	-1.556E-02	-1.400E+02	-5.600E+02	3.068E+01	-5.622E+02	6.608E+02	-85.84
7	8	-1.568E-02	-1.171E+02	-4.655E+02	-9.519E+01	-4.898E+02	1.236E+02	75.67
8	8	-1.582E-02	-1.281E+02	-4.526E+02	-2.949E+01	-4.552E+02	2.916E+02	84.85
9	8	-1.597E-02	-1.332E+02	-4.475E+02	6.448E+00	-4.477E+02	3.039E+02	-88.83
10	8	-1.616E-02	-1.425E+02	-4.455E+02	4.134E+01	-4.510E+02	3.081E+02	-82.37
11	8	-1.641E-02	-1.369E+02	-4.484E+02	1.039E+02	-4.798E+02	3.003E+02	-73.15
12	8	-1.670E-02	5.775E-11	-2.459E+02	1.355E+02	-3.059E+02	1.179E+02	-66.10
0	9	-1.288E-02	1.010E-10	-4.720E+02	-4.143E+01	-4.756E+02	7.080E+01	85.02
1	9	-1.303E-02	-1.316E+02	-8.884E+02	-2.883E+01	-8.895E+02	1.503E+02	87.82
2	9	-1.302E-02	-1.799E+02	-8.463E+02	1.967E+01	-8.469E+02	1.499E+02	-88.31
3	9	-1.296E-02	-1.773E+02	-8.042E+02	5.883E+01	-8.096E+02	1.464E+02	-84.69
4	9	-1.286E-02	-1.465E+02	-7.572E+02	1.017E+02	-7.737E+02	1.405E+02	-80.79
5	9	-1.268E-02	-9.486E+01	-7.088E+02	1.534E+02	-7.450E+02	-5.087E+01	-76.73
6	9	-1.237E-02	-1.816E+02	-7.256E+02	5.480E+01	-7.310E+02	4.540E+02	-84.30
7	9	-1.272E-02	-7.987E+01	-6.179E+02	-4.978E+01	-6.225E+02	-4.875E+01	84.76
8	9	-1.295E-02	-1.137E+02	-5.860E+02	-1.295E+01	-5.863E+02	1.455E+02	88.43
9	9	-1.312E-02	-1.272E+02	-5.690E+02	1.407E+01	-5.695E+02	1.553E+02	-88.18
10	9	-1.326E-02	-1.260E+02	-5.616E+02	4.031E+01	-5.653E+02	1.635E+02	-84.76
11	9	-1.338E-02	-9.657E+01	-5.666E+02	7.355E+01	-5.779E+02	1.703E+02	-81.31
12	9	-1.342E-02	8.185E-11	-3.000E+02	4.711E+01	-3.072E+02	8.632E+01	-81.28
0	10	-1.188E-02	1.637E-10	-5.696E+02	-1.233E+00	-5.696E+02	4.216E+01	89.88
1	10	-1.208E-02	-1.255E+02	-1.079E+03	3.346E+01	-1.080E+03	9.568E+01	-87.99
2	10	-1.203E-02	-1.780E+02	-1.015E+03	6.742E+01	-1.020E+03	9.270E+01	-85.42
3	10	-1.185E-02	-1.664E+02	-9.460E+02	1.012E+02	-9.590E+02	8.217E+01	-82.73
4	10	-1.154E-02	-1.165E+02	-8.685E+02	1.444E+02	-8.953E+02	6.467E+01	-79.49
5	10	-1.107E-02	-5.857E+01	-7.801E+02	1.894E+02	-8.268E+02	-1.492E+02	-76.15
6	10	-1.035E-02	-1.869E+02	-7.461E+02	1.042E+02	-7.649E+02	3.615E+02	-79.78
7	10	-1.079E-02	-4.629E+01	-6.171E+02	6.152E+00	-6.171E+02	-1.651E+02	-89.38
8	10	-1.104E-02	-8.004E+01	-5.623E+02	1.920E+01	-5.631E+02	3.579E+01	-87.72
9	10	-1.119E-02	-9.391E+01	-5.289E+02	3.007E+01	-5.310E+02	4.425E+01	-86.06
10	10	-1.127E-02	-8.751E+01	-5.116E+02	4.127E+01	-5.156E+02	4.890E+01	-84.49
11	10	-1.128E-02	-5.759E+01	-5.098E+02	5.222E+01	-5.158E+02	4.953E+01	-83.50
12	10	-1.117E-02	1.587E-10	-2.633E+02	2.336E+01	-2.654E+02	2.160E+01	-84.97
0	11	-1.286E-02	-1.783E-10	-6.452E+02	3.942E+01	-6.476E+02	7.020E+01	-86.52
1	11	-1.294E-02	-1.227E+02	-1.225E+03	1.067E+02	-1.235E+03	1.454E+02	-84.52
2	11	-1.271E-02	-1.759E+02	-1.137E+03	1.334E+02	-1.155E+03	1.321E+02	-82.24
3	11	-1.229E-02	-1.452E+02	-1.043E+03	1.617E+02	-1.072E+03	1.080E+02	-80.10
4	11	-1.168E-02	-6.721E+01	-9.383E+02	2.139E+02	-9.880E+02	7.235E+01	-76.92
5	11	-1.077E-02	-1.802E+01	-8.031E+02	2.682E+02	-8.859E+02	-1.580E+02	-72.83
6	11	-9.546E-03	-1.787E+02	-7.127E+02	1.856E+02	-7.709E+02	3.095E+02	-72.60
7	11	-9.905E-03	-2.902E+01	-5.588E+02	7.783E+01	-5.700E+02	-2.077E+02	-81.81

8	11	-1.007E-02	-5.836E+01	-4.813E+02	6.666E+01	-4.915E+02	-1.998E+01	-81.25
9	11	-1.013E-02	-6.746E+01	-4.312E+02	5.621E+01	-4.397E+02	-1.647E+01	-81.41
10	11	-1.013E-02	-5.552E+01	-4.025E+02	4.939E+01	-4.094E+02	-1.697E+01	-82.05
11	11	-1.004E-02	-2.862E+01	-3.909E+02	4.384E+01	-3.962E+02	-2.195E+01	-83.20
12	11	-9.833E-03	1.205E-10	-1.968E+02	1.381E+01	-1.978E+02	-1.691E+01	-86.01
0	12	-1.606E-02	4.438E-10	-6.817E+02	8.383E+01	-6.919E+02	1.624E+02	-83.09
1	12	-1.587E-02	-9.862E+01	-1.303E+03	1.894E+02	-1.332E+03	3.139E+02	-81.27
2	12	-1.528E-02	-1.706E+02	-1.189E+03	2.161E+02	-1.233E+03	2.802E+02	-78.51
3	12	-1.448E-02	-1.095E+02	-1.080E+03	2.302E+02	-1.132E+03	2.340E+02	-77.31
4	12	-1.340E-02	2.949E+01	-9.684E+02	3.066E+02	-1.055E+03	1.717E+02	-74.22
5	12	-1.185E-02	3.964E+01	-7.878E+02	3.922E+02	-9.441E+02	-7.981E+01	-68.26
6	12	-9.894E-03	-1.631E+02	-6.497E+02	3.005E+02	-7.931E+02	3.076E+02	-64.50
7	12	-9.973E-03	-3.809E+01	-4.798E+02	1.654E+02	-5.349E+02	-1.880E+02	-71.58
8	12	-9.911E-03	-6.640E+01	-3.888E+02	1.257E+02	-4.320E+02	-2.931E+01	-71.03
9	12	-9.796E-03	-6.641E+01	-3.289E+02	9.008E+01	-3.568E+02	-3.595E+01	-72.77
10	12	-9.653E-03	-4.419E+01	-2.920E+02	6.413E+01	-3.077E+02	-4.416E+01	-76.32
11	12	-9.461E-03	-1.451E+01	-2.728E+02	4.569E+01	-2.807E+02	-5.525E+01	-80.26
12	12	-9.176E-03	2.410E-11	-1.333E+02	1.328E+01	-1.347E+02	-3.584E+01	-84.37
0	13	-2.162E-02	-2.601E-10	-5.937E+02	1.228E+02	-6.181E+02	3.225E+02	-78.76
1	13	-2.100E-02	2.505E+01	-1.172E+03	2.708E+02	-1.230E+03	6.096E+02	-77.83
2	13	-1.984E-02	-1.515E+02	-1.035E+03	3.127E+02	-1.135E+03	5.425E+02	-72.36
3	13	-1.849E-02	-4.585E+01	-9.329E+02	2.782E+02	-1.013E+03	4.647E+02	-73.95
4	13	-1.681E-02	2.617E+02	-8.628E+02	4.033E+02	-9.924E+02	3.682E+02	-72.17
5	13	-1.431E-02	1.364E+02	-6.585E+02	5.544E+02	-9.432E+02	9.525E+01	-62.82
6	13	-1.129E-02	-1.300E+02	-5.166E+02	4.343E+02	-7.987E+02	3.285E+02	-57.00
7	13	-1.085E-02	-8.039E+01	-3.711E+02	2.565E+02	-5.206E+02	-1.040E+02	-59.77
8	13	-1.039E-02	-1.118E+02	-2.933E+02	1.845E+02	-4.082E+02	-1.591E+00	-58.09
9	13	-9.997E-03	-9.712E+01	-2.391E+02	1.238E+02	-3.108E+02	-2.435E+01	-59.91
10	13	-9.667E-03	-5.754E+01	-2.023E+02	8.043E+01	-2.381E+02	-4.340E+01	-65.99
11	13	-9.348E-03	-1.611E+01	-1.804E+02	5.183E+01	-1.954E+02	-6.175E+01	-73.88
12	13	-8.979E-03	7.708E-11	-8.576E+01	1.634E+01	-8.877E+01	-4.149E+01	-79.57
0	14	-2.923E-02	6.939E-10	-2.260E+02	1.226E+02	-2.797E+02	5.417E+02	-66.34
1	14	-2.817E-02	4.539E+02	-5.071E+02	3.150E+02	-6.011E+02	1.023E+03	-73.37
2	14	-2.611E-02	-1.306E+02	-4.537E+02	4.080E+02	-7.309E+02	9.040E+02	-55.80
3	14	-2.409E-02	3.355E+01	-3.927E+02	2.647E+02	-5.194E+02	7.871E+02	-64.42
4	14	-2.183E-02	8.385E+02	-3.481E+02	4.651E+02	9.991E+02	6.571E+02	-19.05
5	14	-1.797E-02	2.754E+02	-2.518E+02	7.103E+02	7.694E+02	3.757E+02	-34.82
6	14	-1.353E-02	-6.042E+01	-2.375E+02	5.326E+02	-6.889E+02	3.101E+02	-49.72
7	14	-1.234E-02	-1.569E+02	-2.098E+02	3.165E+02	-5.010E+02	5.140E+01	-47.39
8	14	-1.133E-02	-1.901E+02	-1.888E+02	2.215E+02	-4.110E+02	5.254E+01	-44.92
9	14	-1.057E-02	-1.542E+02	-1.604E+02	1.455E+02	-3.029E+02	8.624E+00	-45.61
10	14	-1.000E-02	-9.141E+01	-1.331E+02	9.120E+01	-2.058E+02	-2.393E+01	-51.44
11	14	-9.540E-03	-3.087E+01	-1.141E+02	5.559E+01	-1.419E+02	-5.070E+01	-63.41
12	14	-9.079E-03	-4.775E-12	-5.392E+01	1.820E+01	-5.949E+01	-3.862E+01	-72.99
0	15	-3.762E-02	5.011E-10	4.937E+02	9.249E+01	5.105E+02	7.834E+02	-79.73
1	15	-3.642E-02	1.617E+03	1.625E+03	2.468E+02	1.868E+03	-8.503E+03	-45.45
2	15	-3.312E-02	-1.784E+02	5.706E+02	3.206E+02	6.891E+02	1.307E+03	-69.72
3	15	-3.038E-02	4.327E+01	5.752E+02	3.174E+02	7.233E+02	1.149E+03	-64.98
4	15	-2.781E-02	2.156E+03	1.554E+03	4.544E+02	2.400E+03	-8.999E+03	-28.24
5	15	-2.218E-02	3.676E+02	5.222E+02	6.125E+02	1.062E+03	7.285E+02	-48.60
6	15	-1.615E-02	4.951E+01	2.033E+02	4.843E+02	6.168E+02	-1.501E+02	-49.51
7	15	-1.412E-02	-2.497E+02	-2.359E+01	2.916E+02	-4.494E+02	2.642E+02	-34.41
8	15	-1.252E-02	-2.780E+02	-8.392E+01	2.172E+02	-4.189E+02	1.207E+02	-32.96
9	15	-1.135E-02	-2.204E+02	-8.650E+01	1.468E+02	-3.149E+02	5.373E+01	-32.74
10	15	-1.053E-02	-1.352E+02	-7.268E+01	9.122E+01	-2.004E+02	6.511E+00	-35.54
11	15	-9.916E-03	-5.406E+01	-6.027E+01	5.191E+01	-1.092E+02	-2.906E+01	-46.71
12	15	-9.365E-03	-1.156E-10	-2.984E+01	1.543E+01	-3.638E+01	-3.037E+01	-67.02

0	16	-4.431E-02	2.892E-11	4.048E-10	2.671E+01	2.671E+01	4.880E+02	-45.00
1	16	-4.255E-02	5.013E+02	1.040E-09	8.792E+01	5.163E+02	9.253E+02	-9.66
2	16	-3.906E-02	-3.509E+01	5.012E-10	1.155E+02	-1.344E+02	8.248E+02	-40.68
3	16	-3.569E-02	8.440E+01	7.518E-10	1.576E+02	2.053E+02	7.278E+02	-37.50
4	16	-3.203E-02	7.867E+02	4.529E-10	2.060E+02	8.374E+02	6.224E+02	-13.82
5	16	-2.565E-02	2.406E+02	1.833E-10	3.298E+02	4.713E+02	5.275E+02	-34.98
6	16	-1.844E-02	8.805E+01	3.552E+02	2.222E+02	4.808E+02	-3.932E+01	-60.51
7	16	-1.584E-02	-1.421E+02	1.494E-10	6.703E+01	-1.688E+02	2.448E+02	-21.66
8	16	-1.372E-02	-1.675E+02	2.267E-10	1.013E+02	-2.152E+02	9.518E+01	-25.21
9	16	-1.219E-02	-1.370E+02	1.253E-10	6.701E+01	-1.643E+02	5.098E+01	-22.18
10	16	-1.113E-02	-8.816E+01	2.888E-11	3.924E+01	-1.031E+02	2.041E+01	-20.84
11	16	-1.037E-02	-4.034E+01	1.446E-10	1.947E+01	-4.821E+01	-1.381E+00	-21.99
12	16	-9.757E-03	-4.819E-12	1.687E-11	5.268E+00	5.268E+00	-9.578E+00	-45.00

SUMMATION OF REACTIONS = -.00000000

TIME FOR THIS PROBLEM = 0 MINUTES 5.110 SECONDS

PROGRAM VISAB 3 - MASTER DECK - PEARRE - DRIVER REVISION DATE - 21 MAR 1968
CE994005 HIGHWAY SLAB PROJECT VISAB 3 C.M.PEARRE
EXAMPLE PROBLEMS RUN FOR USE IN APPENDIX 6

PROB
EX 2 SLAB ON FOUNDATION

TABLE 1. CONTROL DATA

NUM CARDS TABLE 2	3
NUM CARDS TABLE 3	1
NUM CARDS TABLE 4	1
NUM CARDS TABLE 5	2
NUM CARDS TABLE 6	0
NUM CARDS TABLE 7	0
NUM INCREMENTS MX	16
NUM INCREMENTS MY	16
POISSONS RATIO	.1500
SLAB THICKNESS	1.000E+01
MUL LOAD OPTION (IF + OR - 1)	0
OPTION (IF = 1) FOR SHORT FORM OF OUTPUT	1
OPTION (IF = 1) TO PRINT INPUT VARIABLE ARRAYS	0

TABLE 2. INCREMENT LENGTH

FROM X-BAR	THRU X-BAR	HX (L)	FROM Y-BAR	THRU Y-BAR	HY (L)
1	16	2.560E+01	1	16	2.560E+01
6	11	-1.760E+01	6	11	-1.760E+01
7	10	-4.000E+00	7	10	-4.000E+00

TABLE 3. BENDING STIFFNESS

FROM STA		THRU STA		DX (FL)	DY (FL)
0	0	16	16	2.557E+08	2.557E+08

TABLE 4. TWISTING STIFFNESS

FROM STA		THRU STA		CX (FL)	CY (FL)
1	1	16	16	2.174E+08	2.174E+08

TABLE 5. LOAD AND SUPPORT DATA

FROM	THRU	Q	QQ	S	SS

STA	STA		(F)	(F/LL)	(F/L)	(F/LLL)
0	0	16	16	-0.	-0.	-0.	2.000E+02
8	8	8	8	-1.000E+05	-0.	-0.	-0.

TABLE 6. X-DIRECTIONAL LOAD DATA

FROM X-BAR	THRU X-BAR	TX (FL)	TTX (FL/L)	PX (F)	PPX (F/L)

NO DATA INPUT FOR THIS TABLE

TABLE 7. Y-DIRECTIONAL LOAD DATA

FROM Y-BAR	THRU Y-BAR	TY (FL)	TTY (FL/L)	PY (F)	PPY (F/L)

NO DATA INPUT FOR THIS TABLE

PROGRAM VISAB 3 - MASTER DECK - PEARRE - PROGRAM REVISION DATE - 21 MAR 1968
CE994005 HIGHWAY SLAB PROJECT VISAB 3 C.M.PEARRE
EXAMPLE PROBLEMS RUN FOR USE IN APPENDIX 6

PROB (CONTD)
EX 2 SLAB ON FOUNDATION

TABLE 8. RESULTS

I, (X)	J (Y)	DEFLECTION (L)	X BENDING MOMENT (LF/L)	Y BENDING MOMENT (LF/L)	X TWISTING MOMENT (LF/L)	LARGEST PRINCIPAL STRESS (F/LL)	NET REACTION (F)	BETA X TO LARG (DEG)
0	0	2.903E-03	-1.291E-11	-7.017E-12	-2.492E+01	-1.495E+00	-9.511E+01	45.00
1	0	2.646E-03	6.482E+01	-1.887E-11	-7.222E+01	6.694E+00	-1.734E+02	32.91
2	0	2.729E-03	3.320E+01	-1.575E-11	-1.103E+02	7.688E+00	-1.788E+02	40.72
3	0	2.986E-03	-1.136E+01	-1.594E-11	-1.343E+02	-8.405E+00	-1.957E+02	43.79
4	0	3.183E-03	-2.595E+01	-1.118E-11	-1.229E+02	-8.192E+00	-2.086E+02	41.99
5	0	3.245E-03	-1.160E+01	-1.680E-11	-9.123E+01	-5.833E+00	-1.395E+02	43.18
6	0	3.251E-03	-9.110E+00	-3.742E-11	-3.871E+01	-2.612E+00	-4.994E+01	41.64
7	0	3.253E-03	-8.454E+00	-1.533E-11	-1.565E+01	-1.226E+00	-3.331E+01	37.44
8	0	3.253E-03	-8.230E+00	-2.196E-11	-4.596E-09	-4.938E-01	-3.332E+01	.00
9	0	3.253E-03	-8.454E+00	5.187E-13	1.565E+01	-1.226E+00	-3.331E+01	-37.44
10	0	3.251E-03	-9.110E+00	-3.773E-11	3.871E+01	-2.612E+00	-4.994E+01	-41.64
11	0	3.245E-03	-1.160E+01	-1.906E-11	9.123E+01	-5.833E+00	-1.395E+02	-43.18
12	0	3.183E-03	-2.595E+01	-1.671E-11	1.229E+02	-8.192E+00	-2.086E+02	-41.99
13	0	2.986E-03	-1.136E+01	-2.166E-11	1.343E+02	-8.405E+00	-1.957E+02	-43.79
14	0	2.729E-03	3.320E+01	-1.074E-11	1.103E+02	7.688E+00	-1.788E+02	-40.72
15	0	2.646E-03	6.482E+01	-1.506E-11	7.222E+01	6.694E+00	-1.734E+02	-32.91
16	0	2.903E-03	-7.710E-12	-6.238E-12	2.492E+01	-1.495E+00	-9.511E+01	-45.00
0	1	2.646E-03	1.091E-11	6.482E+01	-7.222E+01	6.694E+00	-1.734E+02	57.09
1	1	2.050E-03	5.995E+01	5.995E+01	-1.800E+02	1.440E+01	-2.687E+02	45.00
2	1	1.588E-03	-1.758E+01	-2.679E+01	-2.606E+02	-1.697E+01	-2.081E+02	45.51
3	1	1.090E-03	-8.094E+01	-1.683E+02	-3.391E+02	-2.799E+01	-1.429E+02	48.67
4	1	4.464E-04	-3.088E+01	-3.709E+02	-3.378E+02	-3.474E+01	-5.850E+01	58.36
5	1	-1.325E-04	9.999E+01	-5.645E+02	-2.650E+02	-3.943E+01	1.140E+01	70.71
6	1	-2.141E-04	1.201E+02	-5.922E+02	-1.140E+02	-3.660E+01	6.576E+00	81.12
7	1	-2.348E-04	1.253E+02	-5.992E+02	-4.622E+01	-3.613E+01	4.809E+00	86.36
8	1	-2.418E-04	1.271E+02	-6.016E+02	-1.320E-08	-3.610E+01	4.951E+00	90.00
9	1	-2.348E-04	1.253E+02	-5.992E+02	4.622E+01	-3.613E+01	4.809E+00	-86.36
10	1	-2.141E-04	1.201E+02	-5.922E+02	1.140E+02	-3.660E+01	6.576E+00	-81.12
11	1	-1.325E-04	9.999E+01	-5.645E+02	2.650E+02	-3.943E+01	1.140E+01	-70.71
12	1	4.464E-04	-3.088E+01	-3.709E+02	3.378E+02	-3.474E+01	-5.850E+01	-58.36
13	1	1.090E-03	-8.094E+01	-1.683E+02	3.391E+02	-2.799E+01	-1.429E+02	-48.67
14	1	1.588E-03	-1.758E+01	-2.679E+01	2.606E+02	-1.697E+01	-2.081E+02	-45.51
15	1	2.050E-03	5.995E+01	5.995E+01	1.800E+02	1.440E+01	-2.687E+02	-45.00
16	1	2.646E-03	-1.785E-11	6.482E+01	7.222E+01	6.694E+00	-1.734E+02	-57.09
0	2	2.729E-03	-2.785E-12	3.320E+01	-1.103E+02	7.688E+00	-1.788E+02	49.28
1	2	1.588E-03	-2.679E+01	-1.758E+01	-2.606E+02	-1.697E+01	-2.081E+02	44.49
2	2	3.836E-04	-1.769E+02	-1.769E+02	-3.944E+02	-3.428E+01	-5.028E+01	45.00
3	2	-1.215E-03	-2.354E+02	-4.568E+02	-5.702E+02	-5.562E+01	1.592E+02	50.49

4	2	-3.251E-03	-3.302E+01	-8.758E+02	-6.363E+02	-7.306E+01	4.261E+02	61.76
5	2	-5.029E-03	3.705E+02	-1.315E+03	-5.362E+02	-8.828E+01	4.326E+02	73.77
6	2	-5.279E-03	4.342E+02	-1.380E+03	-2.369E+02	-8.465E+01	1.622E+02	82.68
7	2	-5.343E-03	4.507E+02	-1.397E+03	-9.658E+01	-8.413E+01	1.094E+02	87.02
8	2	-5.364E-03	4.564E+02	-1.403E+03	-2.622E-08	-8.417E+01	1.099E+02	90.00
9	2	-5.343E-03	4.507E+02	-1.397E+03	9.658E+01	-8.413E+01	1.094E+02	-87.02
10	2	-5.279E-03	4.342E+02	-1.380E+03	2.369E+02	-8.465E+01	1.622E+02	-82.68
11	2	-5.029E-03	3.705E+02	-1.315E+03	5.362E+02	-8.828E+01	4.326E+02	-73.77
12	2	-3.251E-03	-3.302E+01	-8.758E+02	6.363E+02	-7.306E+01	4.261E+02	-61.76
13	2	-1.215E-03	-2.354E+02	-4.568E+02	5.702E+02	-5.562E+01	1.592E+02	-50.49
14	2	3.836E-04	-1.769E+02	-1.769E+02	3.944E+02	-3.428E+01	-5.028E+01	-45.00
15	2	1.588E-03	-2.679E+01	-1.758E+01	2.606E+02	-1.697E+01	-2.081E+02	-44.49
16	2	2.729E-03	-2.635E-11	3.320E+01	1.103E+02	7.688E+00	-1.788E+02	-49.28
0	3	2.986E-03	2.027E-11	-1.136E+01	-1.343E+02	-8.405E+00	-1.957E+02	46.21
1	3	1.090E-03	-1.683E+02	-8.094E+01	-3.391E+02	-2.799E+01	-1.429E+02	41.33
2	3	-1.215E-03	-4.568E+02	-2.354E+02	-5.702E+02	-5.562E+01	1.592E+02	39.51
3	3	-4.625E-03	-5.369E+02	-5.369E+02	-9.215E+02	-8.750E+01	6.062E+02	45.00
4	3	-9.232E-03	-3.877E+01	-1.068E+03	-1.169E+03	-1.098E+02	1.210E+03	56.88
5	3	-1.352E-02	1.079E+03	-1.794E+03	-1.075E+03	-1.291E+02	1.163E+03	71.60
6	3	-1.413E-02	1.280E+03	-1.920E+03	-5.084E+02	-1.200E+02	4.342E+02	81.19
7	3	-1.429E-02	1.334E+03	-1.954E+03	-2.105E+02	-1.181E+02	2.927E+02	86.35
8	3	-1.434E-02	1.353E+03	-1.966E+03	-4.997E-08	-1.179E+02	2.938E+02	90.00
9	3	-1.429E-02	1.334E+03	-1.954E+03	2.105E+02	-1.181E+02	2.927E+02	-86.35
10	3	-1.413E-02	1.280E+03	-1.920E+03	5.084E+02	-1.200E+02	4.342E+02	-81.19
11	3	-1.352E-02	1.079E+03	-1.794E+03	1.075E+03	-1.291E+02	1.163E+03	-71.60
12	3	-9.232E-03	-3.877E+01	-1.068E+03	1.169E+03	-1.098E+02	1.210E+03	-56.88
13	3	-4.625E-03	-5.369E+02	-5.369E+02	9.215E+02	-8.750E+01	6.062E+02	-45.00
14	3	-1.215E-03	-4.568E+02	-2.354E+02	5.702E+02	-5.562E+01	1.592E+02	-39.51
15	3	1.090E-03	-1.683E+02	-8.094E+01	3.391E+02	-2.799E+01	-1.429E+02	-41.33
16	3	2.986E-03	-1.785E-11	-1.136E+01	1.343E+02	-8.405E+00	-1.957E+02	-46.21
0	4	3.183E-03	3.112E-12	-2.595E+01	-1.229E+02	-8.192E+00	-2.086E+02	48.01
1	4	4.464E-04	-3.709E+02	-3.088E+01	-3.378E+02	-3.474E+01	-5.850E+01	31.64
2	4	-3.251E-03	-8.758E+02	-3.302E+01	-6.363E+02	-7.306E+01	4.261E+02	28.24
3	4	-9.232E-03	-1.068E+03	-3.877E+01	-1.169E+03	-1.098E+02	1.210E+03	33.12
4	4	-1.800E-02	-1.689E+02	-1.689E+02	-1.783E+03	-1.171E+02	2.359E+03	45.00
5	4	-2.714E-02	2.604E+03	-8.158E+02	-1.895E+03	2.068E+02	2.334E+03	23.98
6	4	-2.853E-02	3.335E+03	-1.070E+03	-1.133E+03	2.166E+02	8.764E+02	13.61
7	4	-2.889E-02	3.548E+03	-1.149E+03	-4.996E+02	2.161E+02	5.917E+02	6.00
8	4	-2.901E-02	3.625E+03	-1.178E+03	-7.554E-08	2.175E+02	5.941E+02	.00
9	4	-2.889E-02	3.548E+03	-1.149E+03	4.996E+02	2.161E+02	5.917E+02	-6.00
10	4	-2.853E-02	3.335E+03	-1.070E+03	1.133E+03	2.166E+02	8.764E+02	-13.61
11	4	-2.714E-02	2.604E+03	-8.158E+02	1.895E+03	2.068E+02	2.334E+03	-23.98
12	4	-1.800E-02	-1.689E+02	-1.689E+02	1.783E+03	-1.171E+02	2.359E+03	-45.00
13	4	-9.232E-03	-1.068E+03	-3.877E+01	1.169E+03	-1.098E+02	1.210E+03	-33.12
14	4	-3.251E-03	-8.758E+02	-3.302E+01	6.363E+02	-7.306E+01	4.261E+02	-28.24
15	4	4.464E-04	-3.709E+02	-3.088E+01	3.378E+02	-3.474E+01	-5.850E+01	-31.64
16	4	3.183E-03	1.317E-11	-2.595E+01	1.229E+02	-8.192E+00	-2.086E+02	-48.01
0	5	3.245E-03	-1.547E-10	-1.160E+01	-9.123E+01	-5.833E+00	-1.395E+02	46.82
1	5	-1.325E-04	-5.645E+02	9.999E+01	-2.650E+02	-3.943E+01	1.140E+01	19.29
2	5	-5.029E-03	-1.315E+03	3.705E+02	-5.362E+02	-8.828E+01	4.326E+02	16.23
3	5	-1.352E-02	-1.794E+03	1.079E+03	-1.075E+03	-1.291E+02	1.163E+03	18.40
4	5	-2.714E-02	-8.158E+02	2.604E+03	-1.895E+03	2.068E+02	2.334E+03	66.02
5	5	-4.392E-02	4.646E+03	4.646E+03	-2.329E+03	4.185E+02	2.479E+03	45.00
6	5	-4.704E-02	8.007E+03	4.309E+03	-1.951E+03	5.307E+02	9.484E+02	23.27
7	5	-4.789E-02	9.301E+03	4.042E+03	-1.015E+03	5.694E+02	6.437E+02	10.55
8	5	-4.819E-02	9.867E+03	3.888E+03	-8.297E-08	5.920E+02	6.477E+02	.00
9	5	-4.789E-02	9.301E+03	4.042E+03	1.015E+03	5.694E+02	6.437E+02	-10.55
10	5	-4.704E-02	8.007E+03	4.309E+03	1.951E+03	5.307E+02	9.484E+02	-23.27

11	5	-4.392E-02	4.646E+03	4.646E+03	2.329E+03	4.185E+02	2.479E+03	-45.00
12	5	-2.714E-02	-8.158E+02	2.604E+03	1.895E+03	2.068E+02	2.334E+03	-66.02
13	5	-1.352E-02	-1.794E+03	1.079E+03	1.075E+03	-1.291E+02	1.163E+03	-18.40
14	5	-5.029E-03	-1.315E+03	3.705E+02	5.362E+02	-8.828E+01	4.326E+02	-16.23
15	5	-1.325E-04	-5.645E+02	9.999E+01	2.650E+02	-3.943E+01	1.140E+01	-19.29
16	5	3.245E-03	-2.773E-11	-1.160E+01	9.123E+01	-5.833E+00	-1.395E+02	-46.82
0	6	3.251E-03	-9.594E-11	-9.110E+00	-3.871E+01	-2.612E+00	-4.994E+01	48.36
1	6	-2.141E-04	-5.922E+02	1.201E+02	-1.140E+02	-3.660E+01	6.576E+00	8.88
2	6	-5.279E-03	-1.380E+03	4.342E+02	-2.369E+02	-8.465E+01	1.622E+02	7.32
3	6	-1.413E-02	-1.920E+03	1.280E+03	-5.084E+02	-1.200E+02	4.342E+02	8.81
4	6	-2.853E-02	-1.070E+03	3.335E+03	-1.133E+03	2.166E+02	8.764E+02	76.39
5	6	-4.704E-02	4.309E+03	8.007E+03	-1.951E+03	5.307E+02	9.484E+02	66.73
6	6	-5.115E-02	9.812E+03	9.812E+03	-3.097E+03	7.745E+02	3.683E+02	45.00
7	6	-5.241E-02	1.359E+04	9.830E+03	-2.320E+03	8.817E+02	2.516E+02	25.49
8	6	-5.289E-02	1.641E+04	9.281E+03	-5.833E-08	9.844E+02	2.539E+02	.00
9	6	-5.241E-02	1.359E+04	9.830E+03	2.320E+03	8.817E+02	2.516E+02	-25.49
10	6	-5.115E-02	9.812E+03	9.812E+03	3.097E+03	7.745E+02	3.683E+02	-45.00
11	6	-4.704E-02	4.309E+03	8.007E+03	1.951E+03	5.307E+02	9.484E+02	-66.73
12	6	-2.853E-02	-1.070E+03	3.335E+03	1.133E+03	2.166E+02	8.764E+02	-76.39
13	6	-1.413E-02	-1.920E+03	1.280E+03	5.084E+02	-1.200E+02	4.342E+02	-8.81
14	6	-5.279E-03	-1.380E+03	4.342E+02	2.369E+02	-8.465E+01	1.622E+02	-7.32
15	6	-2.141E-04	-5.922E+02	1.201E+02	1.140E+02	-3.660E+01	6.576E+00	-8.88
16	6	3.251E-03	-7.901E-11	-9.110E+00	3.871E+01	-2.612E+00	-4.994E+01	-48.36
0	7	3.253E-03	-1.613E-10	-8.454E+00	-1.565E+01	-1.226E+00	-3.331E+01	52.56
1	7	-2.348E-04	-5.992E+02	1.253E+02	-4.622E+01	-3.613E+01	4.809E+00	3.64
2	7	-5.343E-03	-1.397E+03	4.507E+02	-9.658E+01	-8.413E+01	1.094E+02	2.98
3	7	-1.429E-02	-1.954E+03	1.334E+03	-2.105E+02	-1.181E+02	2.927E+02	3.65
4	7	-2.889E-02	-1.149E+03	3.548E+03	-4.996E+02	2.161E+02	5.917E+02	84.00
5	7	-4.789E-02	4.042E+03	9.301E+03	-1.015E+03	5.694E+02	6.437E+02	79.45
6	7	-5.241E-02	9.830E+03	1.359E+04	-2.320E+03	8.817E+02	2.516E+02	64.51
7	7	-5.392E-02	1.552E+04	1.552E+04	-2.975E+03	1.110E+03	1.725E+02	45.00
8	7	-5.459E-02	2.314E+04	1.561E+04	-1.659E-08	1.388E+03	1.747E+02	.00
9	7	-5.392E-02	1.552E+04	1.552E+04	2.975E+03	1.110E+03	1.725E+02	-45.00
10	7	-5.241E-02	9.830E+03	1.359E+04	2.320E+03	8.817E+02	2.516E+02	-64.51
11	7	-4.789E-02	4.042E+03	9.301E+03	1.015E+03	5.694E+02	6.437E+02	-79.45
12	7	-2.889E-02	-1.149E+03	3.548E+03	4.996E+02	2.161E+02	5.917E+02	-84.00
13	7	-1.429E-02	-1.954E+03	1.334E+03	2.105E+02	-1.181E+02	2.927E+02	-3.65
14	7	-5.343E-03	-1.397E+03	4.507E+02	9.658E+01	-8.413E+01	1.094E+02	-2.98
15	7	-2.348E-04	-5.992E+02	1.253E+02	4.622E+01	-3.613E+01	4.809E+00	-3.64
16	7	3.253E-03	-3.188E-11	-8.454E+00	1.565E+01	-1.226E+00	-3.331E+01	-52.56
0	8	3.253E-03	5.792E-11	-8.230E+00	-2.927E-09	-4.938E-01	-3.332E+01	90.00
1	8	-2.418E-04	-6.016E+02	1.271E+02	-5.672E-09	-3.610E+01	4.951E+00	.00
2	8	-5.364E-03	-1.403E+03	4.564E+02	-6.394E-09	-8.417E+01	1.099E+02	.00
3	8	-1.434E-02	-1.966E+03	1.353E+03	-7.130E-09	-1.179E+02	2.938E+02	.00
4	8	-2.901E-02	-1.178E+03	3.625E+03	-4.007E-09	2.175E+02	5.941E+02	90.00
5	8	-4.819E-02	3.888E+03	9.867E+03	2.335E-09	5.920E+02	6.477E+02	-90.00
6	8	-5.289E-02	9.281E+03	1.641E+04	6.034E-09	9.844E+02	2.539E+02	-90.00
7	8	-5.459E-02	1.561E+04	2.314E+04	1.207E-08	1.388E+03	1.747E+02	-90.00
8	8	-5.550E-02	3.372E+04	3.372E+04	1.659E-08	2.023E+03	-9.982E+04	-2.42
9	8	-5.459E-02	1.561E+04	2.314E+04	1.961E-08	1.388E+03	1.747E+02	-90.00
10	8	-5.289E-02	9.281E+03	1.641E+04	1.609E-08	9.844E+02	2.539E+02	-90.00
11	8	-4.819E-02	3.888E+03	9.867E+03	7.363E-09	5.920E+02	6.477E+02	-90.00
12	8	-2.901E-02	-1.178E+03	3.625E+03	5.186E-09	2.175E+02	5.941E+02	-90.00
13	8	-1.434E-02	-1.966E+03	1.353E+03	1.959E-09	-1.179E+02	2.938E+02	-.00
14	8	-5.364E-03	-1.403E+03	4.564E+02	-3.757E-10	-8.417E+01	1.099E+02	.00
15	8	-2.418E-04	-6.016E+02	1.271E+02	-7.660E-10	-3.610E+01	4.951E+00	.00
16	8	3.253E-03	-9.148E-11	-8.230E+00	-5.745E-10	-4.938E-01	-3.332E+01	90.00

0	9	3.253E-03	1.807E-11	-8.454E+00	1.565E+01	-1.226E+00	-3.331E+01	-52.56
1	9	-2.348E-04	-5.992E+02	1.253E+02	4.622E+01	-3.613E+01	4.809E+00	-3.64
2	9	-5.343E-03	-1.397E+03	4.507E+02	9.658E+01	-8.413E+01	1.094E+02	-2.98
3	9	-1.429E-02	-1.954E+03	1.334E+03	2.105E+02	-1.181E+02	2.927E+02	-3.65
4	9	-2.889E-02	-1.149E+03	3.548E+03	4.996E+02	2.161E+02	5.917E+02	-84.00
5	9	-4.789E-02	4.042E+03	9.301E+03	1.015E+03	5.694E+02	6.437E+02	-79.45
6	9	-5.241E-02	9.830E+03	1.359E+04	2.320E+03	8.817E+02	2.516E+02	-64.51
7	9	-5.392E-02	1.552E+04	1.552E+04	2.975E+03	1.110E+03	1.725E+02	-45.00
8	9	-5.459E-02	2.314E+04	1.561E+04	4.601E-08	1.388E+03	1.747E+02	-.00
9	9	-5.392E-02	1.552E+04	1.552E+04	-2.975E+03	1.110E+03	1.725E+02	45.00
10	9	-5.241E-02	9.830E+03	1.359E+04	-2.320E+03	8.817E+02	2.516E+02	64.51
11	9	-4.789E-02	4.042E+03	9.301E+03	-1.015E+03	5.694E+02	6.437E+02	79.45
12	9	-2.889E-02	-1.149E+03	3.548E+03	-4.996E+02	2.161E+02	5.917E+02	84.00
13	9	-1.429E-02	-1.954E+03	1.334E+03	-2.105E+02	-1.181E+02	2.927E+02	3.65
14	9	-5.343E-03	-1.397E+03	4.507E+02	-9.658E+01	-8.413E+01	1.094E+02	2.98
15	9	-2.348E-04	-5.992E+02	1.253E+02	-4.622E+01	-3.613E+01	4.809E+00	3.64
16	9	3.253E-03	-3.605E-11	-8.454E+00	-1.565E+01	-1.226E+00	-3.331E+01	52.56
0	10	3.251E-03	-1.265E-10	-9.110E+00	3.871E+01	-2.612E+00	-4.994E+01	-48.36
1	10	-2.141E-04	-5.922E+02	1.201E+02	1.140E+02	-3.660E+01	6.576E+00	-8.88
2	10	-5.279E-03	-1.380E+03	4.342E+02	2.369E+02	-8.465E+01	1.622E+02	-7.32
3	10	-1.413E-02	-1.920E+03	1.280E+03	5.084E+02	-1.200E+02	4.342E+02	-8.81
4	10	-2.853E-02	-1.070E+03	3.335E+03	1.133E+03	2.166E+02	8.764E+02	-76.39
5	10	-4.704E-02	4.309E+03	8.007E+03	1.951E+03	5.307E+02	9.484E+02	-66.73
6	10	-5.115E-02	9.812E+03	9.812E+03	3.097E+03	7.745E+02	3.683E+02	-45.00
7	10	-5.241E-02	1.359E+04	9.830E+03	2.320E+03	8.817E+02	2.516E+02	-25.49
8	10	-5.289E-02	1.641E+04	9.281E+03	7.442E-08	9.844E+02	2.539E+02	-.00
9	10	-5.241E-02	1.359E+04	9.830E+03	-2.320E+03	8.817E+02	2.516E+02	25.49
10	10	-5.115E-02	9.812E+03	9.812E+03	-3.097E+03	7.745E+02	3.683E+02	45.00
11	10	-4.704E-02	4.309E+03	8.007E+03	-1.951E+03	5.307E+02	9.484E+02	66.73
12	10	-2.853E-02	-1.070E+03	3.335E+03	-1.133E+03	2.166E+02	8.764E+02	76.39
13	10	-1.413E-02	-1.920E+03	1.280E+03	-5.084E+02	-1.200E+02	4.342E+02	8.81
14	10	-5.279E-03	-1.380E+03	4.342E+02	-2.369E+02	-8.465E+01	1.622E+02	7.32
15	10	-2.141E-04	-5.922E+02	1.201E+02	-1.140E+02	-3.660E+01	6.576E+00	8.88
16	10	3.251E-03	-7.069E-11	-9.110E+00	-3.871E+01	-2.612E+00	-4.994E+01	48.36
0	11	3.245E-03	-4.157E-12	-1.160E+01	9.123E+01	-5.833E+00	-1.395E+02	-46.82
1	11	-1.325E-04	-5.645E+02	9.999E+01	2.650E+02	-3.943E+01	1.140E+01	-19.29
2	11	-5.029E-03	-1.315E+03	3.705E+02	5.362E+02	-8.828E+01	4.326E+02	-16.23
3	11	-1.352E-02	-1.794E+03	1.079E+03	1.075E+03	-1.291E+02	1.163E+03	-18.40
4	11	-2.714E-02	-8.158E+02	2.604E+03	1.895E+03	2.068E+02	2.334E+03	-66.02
5	11	-4.392E-02	4.646E+03	4.646E+03	2.329E+03	4.185E+02	2.479E+03	-45.00
6	11	-4.704E-02	8.007E+03	4.309E+03	1.951E+03	5.307E+02	9.484E+02	-23.27
7	11	-4.789E-02	9.301E+03	4.042E+03	1.015E+03	5.694E+02	6.437E+02	-10.55
8	11	-4.819E-02	9.867E+03	3.888E+03	9.042E-08	5.920E+02	6.477E+02	-.00
9	11	-4.789E-02	9.301E+03	4.042E+03	-1.015E+03	5.694E+02	6.437E+02	10.55
10	11	-4.704E-02	8.007E+03	4.309E+03	-1.951E+03	5.307E+02	9.484E+02	23.27
11	11	-4.392E-02	4.646E+03	4.646E+03	-2.329E+03	4.185E+02	2.479E+03	45.00
12	11	-2.714E-02	-8.158E+02	2.604E+03	-1.895E+03	2.068E+02	2.334E+03	66.02
13	11	-1.352E-02	-1.794E+03	1.079E+03	-1.075E+03	-1.291E+02	1.163E+03	18.40
14	11	-5.029E-03	-1.315E+03	3.705E+02	-5.362E+02	-8.828E+01	4.326E+02	16.23
15	11	-1.325E-04	-5.645E+02	9.999E+01	-2.650E+02	-3.943E+01	1.140E+01	19.29
16	11	3.245E-03	-4.124E-11	-1.160E+01	-9.123E+01	-5.833E+00	-1.395E+02	46.82
0	12	3.183E-03	3.214E-11	-2.595E+01	1.229E+02	-8.192E+00	-2.086E+02	-48.01
1	12	4.464E-04	-3.709E+02	-3.088E+01	3.378E+02	-3.474E+01	-5.850E+01	-31.64
2	12	-3.251E-03	-8.758E+02	-3.302E+01	6.363E+02	-7.306E+01	4.261E+02	-28.24
3	12	-9.232E-03	-1.068E+03	-3.877E+01	1.169E+03	-1.098E+02	1.210E+03	-33.12
4	12	-1.800E-02	-1.689E+02	-1.689E+02	1.783E+03	-1.171E+02	2.359E+03	-45.00
5	12	-2.714E-02	2.604E+03	-8.158E+02	1.895E+03	2.068E+02	2.334E+03	-23.98
6	12	-2.853E-02	3.335E+03	-1.070E+03	1.133E+03	2.166E+02	8.764E+02	-13.61

7	12	-2.889E-02	3.548E+03	-1.149E+03	4.996E+02	2.161E+02	5.917E+02	-6.00
8	12	-2.901E-02	3.625E+03	-1.178E+03	7.286E-08	2.175E+02	5.941E+02	-.00
9	12	-2.889E-02	3.548E+03	-1.149E+03	-4.996E+02	2.161E+02	5.917E+02	6.00
10	12	-2.853E-02	3.335E+03	-1.070E+03	-1.133E+03	2.166E+02	8.764E+02	13.61
11	12	-2.714E-02	2.604E+03	-8.158E+02	-1.895E+03	2.068E+02	2.334E+03	23.98
12	12	-1.800E-02	-1.689E+02	-1.689E+02	-1.783E+03	-1.171E+02	2.359E+03	45.00
13	12	-9.232E-03	-1.068E+03	-3.877E+01	-1.169E+03	-1.098E+02	1.210E+03	33.12
14	12	-3.251E-03	-8.758E+02	-3.302E+01	-6.363E+02	-7.306E+01	4.261E+02	28.24
15	12	4.464E-04	-3.709E+02	-3.088E+01	-3.378E+02	-3.474E+01	-5.850E+01	31.64
16	12	3.183E-03	2.075E-12	-2.595E+01	-1.229E+02	-8.192E+00	-2.086E+02	48.01
0	13	2.986E-03	1.022E-11	-1.136E+01	1.343E+02	-8.405E+00	-1.957E+02	-46.21
1	13	1.090E-03	-1.683E+02	-8.094E+01	3.391E+02	-2.799E+01	-1.429E+02	-41.33
2	13	-1.215E-03	-4.568E+02	-2.354E+02	5.702E+02	-5.562E+01	1.592E+02	-39.51
3	13	-4.625E-03	-5.369E+02	-5.369E+02	9.215E+02	-8.750E+01	6.062E+02	-45.00
4	13	-9.232E-03	-3.877E+01	-1.068E+03	1.169E+03	-1.098E+02	1.210E+03	-56.88
5	13	-1.352E-02	1.079E+03	-1.794E+03	1.075E+03	-1.291E+02	1.163E+03	-71.60
6	13	-1.413E-02	1.280E+03	-1.920E+03	5.084E+02	-1.200E+02	4.342E+02	-81.19
7	13	-1.429E-02	1.334E+03	-1.954E+03	2.105E+02	-1.181E+02	2.927E+02	-86.35
8	13	-1.434E-02	1.353E+03	-1.966E+03	4.222E-08	-1.179E+02	2.938E+02	-90.00
9	13	-1.429E-02	1.334E+03	-1.954E+03	-2.105E+02	-1.181E+02	2.927E+02	86.35
10	13	-1.413E-02	1.280E+03	-1.920E+03	-5.084E+02	-1.200E+02	4.342E+02	81.19
11	13	-1.352E-02	1.079E+03	-1.794E+03	-1.075E+03	-1.291E+02	1.163E+03	71.60
12	13	-9.232E-03	-3.877E+01	-1.068E+03	-1.169E+03	-1.098E+02	1.210E+03	56.88
13	13	-4.625E-03	-5.369E+02	-5.369E+02	-9.215E+02	-8.750E+01	6.062E+02	45.00
14	13	-1.215E-03	-4.568E+02	-2.354E+02	-5.702E+02	-5.562E+01	1.592E+02	39.51
15	13	1.090E-03	-1.683E+02	-8.094E+01	-3.391E+02	-2.799E+01	-1.429E+02	41.33
16	13	2.986E-03	6.892E-13	-1.136E+01	-1.343E+02	-8.405E+00	-1.957E+02	46.21
0	14	2.729E-03	2.075E-12	3.320E+01	1.103E+02	7.688E+00	-1.788E+02	-49.28
1	14	1.588E-03	-2.679E+01	-1.758E+01	2.606E+02	-1.697E+01	-2.081E+02	-44.49
2	14	3.836E-04	-1.769E+02	-1.769E+02	3.944E+02	-3.428E+01	-5.028E+01	-45.00
3	14	-1.215E-03	-2.354E+02	-4.568E+02	5.702E+02	-5.562E+01	1.592E+02	-50.49
4	14	-3.251E-03	-3.302E+01	-8.758E+02	6.363E+02	-7.306E+01	4.261E+02	-61.76
5	14	-5.029E-03	3.705E+02	-1.315E+03	5.362E+02	-8.828E+01	4.326E+02	-73.77
6	14	-5.279E-03	4.342E+02	-1.380E+03	2.369E+02	-8.465E+01	1.622E+02	-82.68
7	14	-5.343E-03	4.507E+02	-1.397E+03	9.658E+01	-8.413E+01	1.094E+02	-87.02
8	14	-5.364E-03	4.564E+02	-1.403E+03	2.331E-08	-8.417E+01	1.099E+02	-90.00
9	14	-5.343E-03	4.507E+02	-1.397E+03	-9.658E+01	-8.413E+01	1.094E+02	87.02
10	14	-5.279E-03	4.342E+02	-1.380E+03	-2.369E+02	-8.465E+01	1.622E+02	82.68
11	14	-5.029E-03	3.705E+02	-1.315E+03	-5.362E+02	-8.828E+01	4.326E+02	73.77
12	14	-3.251E-03	-3.302E+01	-8.758E+02	-6.363E+02	-7.306E+01	4.261E+02	61.76
13	14	-1.215E-03	-2.354E+02	-4.568E+02	-5.702E+02	-5.562E+01	1.592E+02	50.49
14	14	3.836E-04	-1.769E+02	-1.769E+02	-3.944E+02	-3.428E+01	-5.028E+01	45.00
15	14	1.588E-03	-2.679E+01	-1.758E+01	-2.606E+02	-1.697E+01	-2.081E+02	44.49
16	14	2.729E-03	-1.990E-13	3.320E+01	-1.103E+02	7.688E+00	-1.788E+02	49.28
0	15	2.646E-03	-2.166E-11	6.482E+01	7.222E+01	6.694E+00	-1.734E+02	-57.09
1	15	2.050E-03	5.995E+01	5.995E+01	1.800E+02	1.440E+01	-2.687E+02	-45.00
2	15	1.588E-03	-1.758E+01	-2.679E+01	2.606E+02	-1.697E+01	-2.081E+02	-45.51
3	15	1.090E-03	-8.094E+01	-1.683E+02	3.391E+02	-2.799E+01	-1.429E+02	-48.67
4	15	4.464E-04	-3.088E+01	-3.709E+02	3.378E+02	-3.474E+01	-5.850E+01	-58.36
5	15	-1.325E-04	9.999E+01	-5.645E+02	2.650E+02	-3.943E+01	1.140E+01	-70.71
6	15	-2.141E-04	1.201E+02	-5.922E+02	1.140E+02	-3.660E+01	6.576E+00	-81.12
7	15	-2.348E-04	1.253E+02	-5.992E+02	4.622E+01	-3.613E+01	4.809E+00	-86.36
8	15	-2.418E-04	1.271E+02	-6.016E+02	1.226E-08	-3.610E+01	4.951E+00	-90.00
9	15	-2.348E-04	1.253E+02	-5.992E+02	-4.622E+01	-3.613E+01	4.809E+00	86.36
10	15	-2.141E-04	1.201E+02	-5.922E+02	-1.140E+02	-3.660E+01	6.576E+00	81.12
11	15	-1.325E-04	9.999E+01	-5.645E+02	-2.650E+02	-3.943E+01	1.140E+01	70.71
12	15	4.464E-04	-3.088E+01	-3.709E+02	-3.378E+02	-3.474E+01	-5.850E+01	58.36
13	15	1.090E-03	-8.094E+01	-1.683E+02	-3.391E+02	-2.799E+01	-1.429E+02	48.67

14	15	1.588E-03	-1.758E+01	-2.679E+01	-2.606E+02	-1.697E+01	-2.081E+02	45.51
15	15	2.050E-03	5.995E+01	5.995E+01	-1.800E+02	1.440E+01	-2.687E+02	45.00
16	15	2.646E-03	-8.470E-12	6.482E+01	-7.222E+01	6.694E+00	-1.734E+02	57.09
0	16	2.903E-03	5.978E-12	-2.790E-11	2.492E+01	-1.495E+00	-9.511E+01	-45.00
1	16	2.646E-03	6.482E+01	-5.446E-11	7.222E+01	6.694E+00	-1.734E+02	-32.91
2	16	2.729E-03	3.320E+01	-1.975E-11	1.103E+02	7.688E+00	-1.788E+02	-40.72
3	16	2.986E-03	-1.136E+01	-4.360E-11	1.343E+02	-8.405E+00	-1.957E+02	-43.79
4	16	3.183E-03	-2.595E+01	-3.189E-11	1.229E+02	-8.192E+00	-2.086E+02	-41.99
5	16	3.245E-03	-1.160E+01	-4.643E-11	9.123E+01	-5.833E+00	-1.395E+02	-43.18
6	16	3.251E-03	-9.110E+00	-2.771E-12	3.871E+01	-2.612E+00	-4.994E+01	-41.64
7	16	3.253E-03	-8.454E+00	-1.802E-11	1.565E+01	-1.226E+00	-3.331E+01	-37.44
8	16	3.253E-03	-8.230E+00	-3.327E-11	5.230E-09	-4.938E-01	-3.332E+01	-.00
9	16	3.253E-03	-8.454E+00	4.851E-11	-1.565E+01	-1.226E+00	-3.331E+01	37.44
10	16	3.251E-03	-9.110E+00	-3.465E-11	-3.871E+01	-2.612E+00	-4.994E+01	41.64
11	16	3.245E-03	-1.160E+01	-6.239E-12	-9.123E+01	-5.833E+00	-1.395E+02	43.18
12	16	3.183E-03	-2.595E+01	-8.498E-12	-1.229E+02	-8.192E+00	-2.086E+02	41.99
13	16	2.986E-03	-1.136E+01	-2.391E-11	-1.343E+02	-8.405E+00	-1.957E+02	43.79
14	16	2.729E-03	3.320E+01	-9.521E-12	-1.103E+02	7.688E+00	-1.788E+02	40.72
15	16	2.646E-03	6.482E+01	-8.527E-13	-7.222E+01	6.694E+00	-1.734E+02	32.91
16	16	2.903E-03	1.473E-12	-1.473E-12	-2.492E+01	1.495E+00	-9.511E+01	45.00

SUMMATION OF REACTIONS = .00000001

TIME FOR THIS PROBLEM = 0 MINUTES 8.435 SECONDS

```
PROGRAM VISAB 3 -   MASTER DECK    - PEARRE - DRIVER REVISION DATE - 21 MAR 1968
CE994005    HIGHWAY SLAB PROJECT   VISAB 3        C.M.PEARRE
EXAMPLE PROBLEMS RUN FOR USE IN APPENDIX 6


PROB
EX  3      SIMPLY SUPPORTED SQUARE STEEL PLATE


TABLE 1.  CONTROL DATA

     NUM CARDS TABLE 2                                              3
     NUM CARDS TABLE 3                                              1
     NUM CARDS TABLE 4                                              1
     NUM CARDS TABLE 5                                              5
     NUM CARDS TABLE 6                                              0
     NUM CARDS TABLE 7                                              1
     NUM INCREMENTS  MX                                             8
     NUM INCREMENTS  MY                                             8
     POISSONS RATIO                                             .2500
     SLAB THICKNESS                                    0.

     MUL LOAD OPTION ( IF + OR - 1 )                                0

     OPTION ( IF = 1 ) FOR SHORT FORM OF OUTPUT                     1

     OPTION ( IF = 1 ) TO PRINT INPUT VARIABLE ARRAYS               0


TABLE 2.  INCREMENT LENGTH

  FROM    THRU       HX                    FROM    THRU       HY
 X-BAR   X-BAR      (L)                   Y-BAR   Y-BAR      (L)

    1       2    8.000E+00                   1       2    8.000E+00
    3       6    4.000E+00                   3       6    4.000E+00
    7       8    8.000E+00                   7       8    4.000E+00


TABLE 3.  BENDING STIFFNESS

  FROM    THRU       DX          DY
   STA     STA      (FL)        (FL)

  0   0  8   8  2.500E+06  2.500E+06


TABLE 4.  TWISTING STIFFNESS

  FROM    THRU       CX          CY
  STA     STA      (FL)        (FL)

  1   1  8   8  1.875E+06  1.875E+06


TABLE 5.  LOAD AND SUPPORT DATA

  FROM    THRU       Q          QQ          S          SS
```

STA		STA		(F)	(F/LL)	(F/L)	(F/LLL)
4	4	4	4	1.000E+05	-0.	-0.	-0.
0	0	0	8	-0.	-0.	1.000E+99	-0.
0	0	8	0	-0.	-0.	1.000E+99	-0.
8	0	8	8	-0.	-0.	1.000E+99	-0.
0	8	8	8	-0.	-0.	1.000E+99	-0.

TABLE 6. X-DIRECTIONAL LOAD DATA

FROM X-BAR	THRU X-BAR	TX (FL)	TTX (FL/L)	PX (F)	PPX (F/L)

NO DATA INPUT FOR THIS TABLE

TABLE 7. Y-DIRECTIONAL LOAD DATA

FROM Y-BAR	THRU Y-BAR	TY (FL)	TTY (FL/L)	PY (F)	PPY (F/L)
0	1 8	8 -0.	-0.	-0.	1.667E+04

PROGRAM VISAB 3 - MASTER DECK - PEARRE - PROGRAM REVISION DATE - 21 MAR 1968
CE994005 HIGHWAY SLAB PROJECT VISAB 3 C.M.PEARRE
EXAMPLE PROBLEMS RUN FOR USE IN APPENDIX 6

PROB (CONTD)
EX 3 SIMPLY SUPPORTED SQUARE STEEL PLATE

TABLE 8. RESULTS

I, (X)	J (Y)	DEFLECTION (L)	X BENDING MOMENT (LF/L)	Y BENDING MOMENT (LF/L)	X TWISTING MOMENT (LF/L)	LARGEST PRINCIPAL MOMENT (LF/L)	NET REACTION (F)	BETA X TO LARG (DEG)
0	0	-6.849E-96	3.638-105	2.859-105	8.561E+02	8.561E+02	6.849E+03	-45.00
1	0	3.785E-96	2.168E-12	8.674E-12	1.622E+03	1.622E+03	-3.785E+03	-45.00
2	0	6.477E-96	8.674E-12	3.469E-11	1.335E+03	1.335E+03	-6.477E+03	-45.00
3	0	5.387E-96	8.674E-12	3.469E-11	6.462E+02	6.462E+02	-5.387E+03	-45.00
4	0	5.893E-96	8.674E-12	3.469E-11	9.758E-10	9.976E-10	-5.893E+03	-45.38
5	0	5.387E-96	8.674E-12	3.469E-11	-6.462E+02	6.462E+02	-5.387E+03	45.00
6	0	6.477E-96	8.674E-12	3.469E-11	-1.335E+03	1.335E+03	-6.477E+03	45.00
7	0	3.785E-96	2.168E-12	8.674E-12	-1.622E+03	1.622E+03	-3.785E+03	45.00
8	0	-6.849E-96	2.079-105	4.418-105	-8.561E+02	8.561E+02	6.849E+03	45.00
0	1	2.218E-96	-3.692E-92	-1.477E-91	1.526E+03	-1.526E+03	-2.218E+03	-45.00
1	1	1.169E-01	-7.306E+02	-1.113E+03	3.004E+03	-3.932E+03	0.	-46.82
2	1	2.214E-01	-2.415E+03	-1.801E+03	2.660E+03	-4.786E+03	0.	-41.70
3	1	2.536E-01	-3.410E+03	-1.819E+03	1.466E+03	-4.282E+03	0.	-30.75
4	1	2.655E-01	-3.944E+03	-1.749E+03	1.926E-09	-3.944E+03	0.	-.00
5	1	2.536E-01	-3.410E+03	-1.819E+03	-1.466E+03	-4.282E+03	0.	30.75
6	1	2.214E-01	-2.415E+03	-1.801E+03	-2.660E+03	-4.786E+03	0.	41.70
7	1	1.169E-01	-7.306E+02	-1.113E+03	-3.004E+03	-3.932E+03	0.	46.82
8	1	2.218E-96	-3.692E-92	-1.477E-91	-1.526E+03	-1.526E+03	-2.218E+03	45.00
0	2	3.724E-96	-3.469E-11	-8.674E-12	1.088E+03	-1.088E+03	-3.724E+03	-45.00
1	2	2.084E-01	-9.307E+02	-2.755E+03	2.342E+03	-4.356E+03	0.	-55.64
2	2	4.102E-01	-4.283E+03	-5.295E+03	2.469E+03	-7.309E+03	0.	-50.79
3	2	4.807E-01	-7.456E+03	-5.745E+03	1.883E+03	-8.668E+03	0.	-32.78
4	2	5.102E-01	-1.002E+04	-5.513E+03	1.804E-09	-1.002E+04	0.	-.00
5	2	4.807E-01	-7.456E+03	-5.745E+03	-1.883E+03	-8.668E+03	0.	32.78
6	2	4.102E-01	-4.283E+03	-5.295E+03	-2.469E+03	-7.309E+03	0.	50.79
7	2	2.084E-01	-9.307E+02	-2.755E+03	-2.342E+03	-4.356E+03	0.	55.64
8	2	3.724E-96	6.939E-11	1.735E-11	-1.088E+03	1.088E+03	-3.724E+03	45.00
0	3	3.090E-96	1.716E-92	6.863E-92	2.429E+02	2.429E+02	-3.090E+03	-45.00
1	3	2.283E-01	-7.270E+02	-3.590E+03	7.874E+02	-3.792E+03	0.	-75.59
2	3	4.613E-01	-4.306E+03	-8.177E+03	1.466E+03	-8.670E+03	0.	-71.43
3	3	5.546E-01	-9.398E+03	-1.056E+04	2.342E+03	-1.239E+04	0.	-51.95
4	3	6.018E-01	-1.667E+04	-1.140E+04	9.368E-10	-1.667E+04	0.	-.00
5	3	5.546E-01	-9.398E+03	-1.056E+04	-2.342E+03	-1.239E+04	0.	51.95
6	3	4.613E-01	-4.306E+03	-8.177E+03	-1.466E+03	-8.670E+03	0.	71.43
7	3	2.283E-01	-7.270E+02	-3.590E+03	-7.874E+02	-3.792E+03	0.	75.59
8	3	3.090E-96	6.939E-11	1.735E-11	-2.429E+02	2.429E+02	-3.090E+03	45.00

0	4	3.334E-96	-1.378E-92	-5.511E-92	-4.843E+02	-4.843E+02	-3.334E+03	45.00
1	4	2.250E-01	-4.847E+02	-3.986E+03	-8.760E+02	-4.193E+03	0.	76.71
2	4	4.639E-01	-3.685E+03	-1.045E+04	-6.593E+02	-1.052E+04	0.	84.49
3	4	5.724E-01	-9.780E+03	-1.745E+04	-2.717E+02	-1.746E+04	0.	87.97
4	4	6.439E-01	-2.834E+04	-2.955E+04	0.	-2.955E+04	1.000E+05	90.00
5	4	5.724E-01	-9.780E+03	-1.745E+04	2.717E+02	-1.746E+04	0.	-87.97
6	4	4.639E-01	-3.685E+03	-1.045E+04	6.593E+02	-1.052E+04	0.	-84.49
7	4	2.250E-01	-4.847E+02	-3.986E+03	8.760E+02	-4.193E+03	0.	-76.71
8	4	3.334E-96	3.469E-11	8.674E-12	4.843E+02	4.843E+02	-3.334E+03	-45.00
0	5	2.873E-96	5.204E-11	1.301E-11	-1.202E+03	1.202E+03	-2.873E+03	45.00
1	5	1.952E-01	-4.532E+02	-3.421E+03	-2.539E+03	-4.878E+03	0.	60.15
2	5	4.015E-01	-3.643E+03	-8.004E+03	-2.803E+03	-9.375E+03	0.	63.94
3	5	4.878E-01	-8.663E+03	-1.041E+04	-2.891E+03	-1.256E+04	0.	53.41
4	5	5.327E-01	-1.598E+04	-1.125E+04	-1.197E-09	-1.598E+04	0.	.00
5	5	4.878E-01	-8.663E+03	-1.041E+04	2.891E+03	-1.256E+04	0.	-53.41
6	5	4.015E-01	-3.643E+03	-8.004E+03	2.803E+03	-9.375E+03	0.	-63.94
7	5	1.952E-01	-4.532E+02	-3.421E+03	2.539E+03	-4.878E+03	0.	-60.15
8	5	2.873E-96	-7.825E-93	-3.130E-92	1.202E+03	-1.202E+03	-2.873E+03	-45.00
0	6	2.011E-96	3.469E-11	8.674E-12	-1.758E+03	1.758E+03	-2.011E+03	45.00
1	6	1.429E-01	-4.165E+02	-2.357E+03	-3.661E+03	-5.174E+03	0.	52.42
2	6	2.906E-01	-2.935E+03	-4.866E+03	-3.690E+03	-7.715E+03	0.	52.33
3	6	3.468E-01	-5.924E+03	-5.443E+03	-2.713E+03	-8.407E+03	0.	42.47
4	6	3.719E-01	-8.658E+03	-5.238E+03	-2.576E-09	-8.658E+03	0.	.00
5	6	3.468E-01	-5.924E+03	-5.443E+03	2.713E+03	-8.407E+03	0.	-42.47
6	6	2.906E-01	-2.935E+03	-4.866E+03	3.690E+03	-7.715E+03	0.	-52.33
7	6	1.429E-01	-4.165E+02	-2.357E+03	3.661E+03	-5.174E+03	0.	-52.42
8	6	2.011E-96	-2.590E-93	-1.036E-92	1.758E+03	-1.758E+03	-2.011E+03	-45.00
0	7	1.016E-96	8.674E-12	2.168E-12	-2.094E+03	2.094E+03	-1.016E+03	45.00
1	7	7.526E-02	-2.563E+02	-1.174E+03	-4.257E+03	-4.997E+03	0.	48.08
2	7	1.515E-01	-1.612E+03	-2.229E+03	-3.982E+03	-5.915E+03	0.	47.21
3	7	1.789E-01	-2.931E+03	-2.330E+03	-2.382E+03	-5.032E+03	0.	41.40
4	7	1.902E-01	-3.861E+03	-2.199E+03	-3.591E-09	-3.861E+03	0.	.00
5	7	1.789E-01	-2.931E+03	-2.330E+03	2.382E+03	-5.032E+03	0.	-41.40
6	7	1.515E-01	-1.612E+03	-2.229E+03	3.982E+03	-5.915E+03	0.	-47.21
7	7	7.526E-02	-2.563E+02	-1.174E+03	4.257E+03	-4.997E+03	0.	-48.08
8	7	1.016E-96	-1.727E-91	-6.907E-91	2.094E+03	-2.094E+03	-1.016E+03	-45.00
0	8	-8.820E-96	-1.040-105	1.143-104	-1.102E+03	1.102E+03	8.82CE+03	45.00
1	8	4.739E-96	5.204E-11	2.082E-10	-2.220E+03	2.220E+03	-4.739E+03	45.00
2	8	9.665E-96	5.204E-11	2.082E-10	-2.024E+03	2.024E+03	-9.665E+03	45.00
3	8	9.069E-96	1.735E-11	6.939E-11	-1.132E+03	1.132E+03	-9.069E+03	45.00
4	8	1.067E-95	-2.498E-91	-6.244E-92	-2.082E-09	-2.082E-09	-1.067E+04	45.00
5	8	9.069E-96	-1.735E-11	-6.939E-11	1.132E+03	-1.132E+03	-9.069E+03	-45.00
6	8	9.665E-96	-1.735E-11	-6.939E-11	2.024E+03	-2.024E+03	-9.665E+03	-45.00
7	8	4.739E-96	-1.686E-91	-4.215E-92	2.220E+03	-2.220E+03	-4.739E+03	-45.00
8	8	-8.820E-96	4.158-105	8.836-105	1.102E+03	1.102E+03	8.820E+03	-45.00

SUMMATION OF REACTIONS = -.0000C000

TIME FOR THIS PROBLEM = 0 MINUTES 1.689 SECONDS