Computer program for buckling analysis of plate girder webs, June 1966

W. Bossert T.
COMPUTER PROGRAM FOR BUCKLING ANALYSIS
OF PLATE GIRDER WEBS

by
Theodore W. Bossert, Jr.
and
Alexis Ostapenko

Fritz Engineering Laboratory
Department of Civil Engineering
Lehigh University
Bethlehem, Pennsylvania

Fritz Engineering Laboratory Report No. 319.2
July, 1967
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td><strong>1. INTRODUCTION</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>2. IDENTIFICATION</strong></td>
<td>4</td>
</tr>
<tr>
<td>2.1 Listing of Program Titles</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Theory</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Computer and Compiler</td>
<td>5</td>
</tr>
<tr>
<td>2.4 Brief Description of Each Program</td>
<td>5</td>
</tr>
<tr>
<td>2.5 Flow Charts</td>
<td>8</td>
</tr>
<tr>
<td><strong>3. PROCEDURE FOR SETTING UP SOURCE DECKS</strong></td>
<td>9</td>
</tr>
<tr>
<td>3.1 Procedure</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Example</td>
<td>11</td>
</tr>
<tr>
<td><strong>4. DATA CARD SET-UP</strong></td>
<td>14</td>
</tr>
<tr>
<td>4.1 Format of Data Cards and Comments</td>
<td>14</td>
</tr>
<tr>
<td>4.2 Supplementary Explanation of Data Card Input</td>
<td>15</td>
</tr>
<tr>
<td><strong>5. INTERPRETATION OF OUTPUT</strong></td>
<td>18</td>
</tr>
<tr>
<td>5.1 Program 1 - Boundary Stress Functions</td>
<td>18</td>
</tr>
<tr>
<td>5.2 Program 2 - Determination of &quot;A&quot; Matrix</td>
<td>18</td>
</tr>
<tr>
<td>5.3 Program 3 - Simultaneous Equation Solver</td>
<td>18</td>
</tr>
<tr>
<td>5.4 Program 4 - Determination of Stresses</td>
<td>18</td>
</tr>
<tr>
<td>5.5 Program 5 - Determination of U Matrix</td>
<td>19</td>
</tr>
<tr>
<td>5.6 Program 6 - U Matrix Inversion</td>
<td>19</td>
</tr>
<tr>
<td>5.7 Program 7 - Matrix Multiplication</td>
<td>19</td>
</tr>
<tr>
<td>5.8 Program 8 - Eigenvalue Determination</td>
<td>19</td>
</tr>
</tbody>
</table>
6. LIMITATIONS
   6.1 Limitations Due to Size
   6.2 Mesh Dimensions
   6.3 Minimum Number of Mesh Points
   6.4 Boundary Conditions
   6.5 Problems of Convergence

7. ACKNOWLEDGEMENTS

8. GLOSSARY OF VARIABLES USED IN COMPUTER PROGRAMS

9. FIGURES

10. REFERENCES

11. APPENDIX I - SAMPLE COMPUTER PROGRAM OUTPUT - GENERAL PROGRAM

12. APPENDIX II - SAMPLE COMPUTER PROGRAM OUTPUT - SYMMETRICAL PROGRAM

13. APPENDIX III - SIMPLIFIED FLOW CHART

14. APPENDIX IV - PROGRAM LISTINGS
ABSTRACT

A computer program for the buckling analysis of a rectangular plate under edge loading is described in this report. The program is designed to find buckling coefficients for plates with the following two sets of boundary conditions: all edges simply supported (Case 1) and top edge fixed and the three other edges simply supported (Case 2).

The report gives instructions on the use of the program, interpretation of the output, and a brief explanation of the operation of the program. Also, a sample solution of a plate buckling problem and a full listing of all of the cards in the source decks are provided.

This program was used as part of a buckling and ultimate strength study of the web plate of a plate girder under edge loading, carried out at Lehigh University from June, 1965 to July, 1967. Reference 1 gives the results of this study.
1. **INTRODUCTION**

The computer program for which this report is written is designed to perform a buckling analysis of a rectangular plate subjected to any set of edge loads. This program was used to determine the buckling loads for web plates of plate girder panels described in Reference 1 for the edge loading shown in Fig. 1a. (1)

All edge loads are defined as linear functions of a particular edge load on the plate, such as $\sigma_c$ as shown in Fig. 1a. This way, the buckling load is that value of $\sigma_c$ which, when combined with the other edge loads causes the plate to buckle. In the case shown, $\sigma_c$ is the vertical compressive edge load, but it need not be. The maximum compressive bending stress could just as easily have served as $\sigma_c$, and in the case of pure bending without vertical load, $\sigma_c$ would have to be the bending stress.

A plane stress analysis is performed on the plate using the equations based on the theory of elasticity. Finite difference versions of the resulting differential equations are used in the computer program. The theory that is used is explained in Chapter 3 of Reference 1. The finite difference operators are also shown in Reference 1, together with an explanation of their use. The results of the plane stress analysis give a set of stresses at the mesh points which are all proportional to $\sigma_c$. 
After the stress distribution over the plate is found, a linear buckling analysis is performed by a finite difference solution of Eq. 3.14 of Ref. 1. The set of the finite difference simultaneous equations is homogeneous, and the lowest eigenvalue of it gives the buckling value of $\sigma_c$ expressed as a multiple of the eigenvalue.

The program is divided into eight separate independent programs. The input and output of the intermediate programs is written and read by the computer from tapes. Section 2.4 gives a brief description of each program. The basic procedure for the analysis follows very closely that given in Chapter 3 of Ref. 1. Since plates subjected to symmetrical loading can be analyzed by considering only one half of the plate, a separate package of eight programs is available for the symmetrical plate which follows almost exactly the same procedure as the package used for a plate under general loading.

Chapters 3 and 4 describe the procedures for setting up all decks for execution of the program. Chapter 5 indicates how the printouts should be interpreted, and Chapter 6 gives the limitations of the programs. A listing of all programs and a glossary defining all variable names is also provided.

In order to understand the operation of this program, knowledge of the FORTRAN Compiler and of Chapter 3 of Reference 1 is desirable.
2. IDENTIFICATION

2.1 List of Program Titles

1. Boundary Stress Functions
   a. Function GNT(X,B,A,IT,TOL)
      1. Function G(X,K)
   b. Subroutine PRINT(ICT,A,PIX,PIY,JX,JY)

2. Generation of Coefficient Matrix for Stress Program
   a. Subroutine ZERO(MAX,NAX)

3. Simultaneous Equation Solver
   a. Subroutine MINV(MX,A,A,D,M)

4. Determination of Stresses From Stress Functions
   a. Subroutine VEEZ(MAX,NAX,A)
      1. Subroutine ZEROV(MAX,NAX)

5. Determination of U Matrix
   a. Subroutine ZERO(MAX,NAX)

6. U Matrix Inversion
   a. Subroutine MINV(MX,U,U,D,M)

7. Matrix Multiplication

8. Eigenvalue Determination
   a. Subroutine EIGEN(N,NI,TOL)
2.2 **Theory**

The theory on which this program is based is described in Chapter 3 of Reference 1.

2.3 **Computer and Compiler**

FORTRAN II for GE 225 of Lehigh University.

2.4 **Brief Description of Each Program**

**Program 1. Boundary Stress Functions**

This program determines by integration the stress function $\phi$ and the derivatives $\partial \phi / \partial x$ and $\partial \phi / \partial y$ at the edges of the plate in terms of $\sigma_c$. This must be done before the $\nabla^4 \phi$ operator is applied to the interior mesh points of the plate.

a. **Function GNT(K,B,A,IT,TOL)**

This subroutine uses Simpson's rule to integrate the K-th function fed in by the G subroutine. The upper integration limit is B; the lower is A. Since this is an iterative procedure, the maximum allowable number of interactions is IT, and the maximum allowable estimated relative error is TOL.

1. **Function G(X,K)**

Subroutine Function G(X,K) feeds in the functions which designate the edge loading. These functions are integrated by the GNT subroutine.

b. **Subroutine PRINT(ICT,A,PX,PY,JX,JY)**

This routine records the values of the stress function
and its derivatives on tape and prints them. It also records plate and mesh dimensions on tape for later use in the package.

Program 2. Generation of Coefficient Matrix for Stress Program

This program determines the \( A \) and \( C \) matrices, which are used to solve the following set of simultaneous equations:

\[
\begin{bmatrix}
A \\
\end{bmatrix}
\begin{bmatrix}
\phi \\
\end{bmatrix}
=
\begin{bmatrix}
C \\
\end{bmatrix}
\]  
(2.1)

This matrix equation results from the application of finite differences to the following differential equation:

\[
\nabla^4 \phi = 0 
\]  
(2.2)

The values of the stress function and its derivatives at the boundaries computed in Program 1 are needed to determine these matrices.

Subroutine ZERO(MAX,NAX)

This subroutine sets the elements of the temporary storage array for the elements of the \( A \) matrix equal to zero after a finite difference equation for a mesh point has been determined and recorded.

Program 3. Simultaneous Equation Solver

Solve simultaneous equations represented by \( A \) and \( C \) matrices.
Subroutine MINV(MX,A,A,D,M)
Inversion subroutine.

Program 4. Determination of Stresses From Stress Functions
This program performs by a finite difference method the differentiation needed to find the stresses from the stress function values at the mesh points.

Subroutine VEEZ(MAX,NAX,A)
This subroutine determines the V matrix in the buckling equation:
\[
\left[ \left[ \begin{array}{c} U \\ V \end{array} \right] \right]^{-1} \left[ \begin{array}{c} V \\ \sigma_c \end{array} \right] - \frac{1}{\sigma_c} \left[ I \right] \{ w \} = 0
\] (2.3)

This equation can be derived from the differential equation
\[
\nabla^4 w = \frac{t}{D} \left[ \sigma_x \frac{\partial^2 w}{\partial x^2} + \sigma_y \frac{\partial^2 w}{\partial y^2} + 2 \tau_{xy} \frac{\partial^2 w}{\partial x \partial y} \right]
\] (2.4)

using a finite difference procedure with the same mesh as that used in the previous programs.

Program 5. U-Matrix Determination
This program determines the U matrix in much the same way as the A-matrix in the stress determination.

Subroutine: ZERO(MAX,NAX)
Program 6. U-Matrix Inversion

This program utilizes the MINV subroutine to invert the U-matrix as required by the buckling equation.

Subroutine: MINV(MX,U,U,D,M)

Program 7. Matrix Multiplication

This program post-multiplies the inverted U-matrix by the V-matrix. Let the resulting matrix $[u^{-1}v]$ be $[v^*].$

Program 8. Eigenvalue Determination

This program uses the subroutine EIGEN to determine the dominant eigenvalue of the $[u^{-1}v]$ matrix. The buckling load can then be determined from this eigenvalue.

Subroutine: EIGEN(N,NI,TOL)

2.5 Flow Charts

A summary flow chart is provided for the 8-program package in Appendix III. A more complete flow chart is available in the Fritz Laboratory X Files, Book VI, Project 319.
3. PROCEDURE FOR SETTING UP SOURCE DECKS

3.1 Procedure

In the FORTRAN II system at Lehigh, a program actually has two parts, a source deck and a binary deck. The source deck is developed by the programmer and submitted to the computer. The computer compiles the program into its own machine language and returns a binary deck, which is resubmitted to the computer for execution after the data has been added in the proper format. The output resulting from this second submission is the desired set of answers. The binary deck can be run repeatedly with different sets of data, but if a change is required in the basic program, the source deck must be submitted for a new binary deck in order to incorporate the change into the program.

In order to specify the type of loading on the plate, the source deck of one subroutine Function G(X,K) must be changed. In the source deck, the first four cards remain the same. The fifth card, starting with "GO TO" is punched so that the list of numbers within the parentheses (1, 2, 3, etc.) go up to three times the number of segments of loading* used to define the loading distribution on the edge of the plate.

*Refer to Reference 1 for definition of this term.
Before attempting to set up the source deck, it would be helpful for the user to read Chapter 3 of Reference 1 in order to understand the meaning of some of the terms used in the following paragraphs.

The next series of cards gives the functions which are to be integrated into the program. The segments of loading are considered starting at the lower left corner of the plate and proceeding around the plate counterclockwise with the positive direction of the coordinate counterclockwise. For plates with symmetrical loading, the starting point is at the center of the bottom edge.

First, the segments of loading must be decided upon. For an unsymmetrical loading pattern, the minimum number of segments of loading is four, one for each edge. For a symmetrical loading, the minimum is three. If there is any discontinuity of edge loading, the segment of loading must be terminated at the discontinuity and a new one started there. For example, if the top edge has a uniform compressive loading over the first (the right) half of the plate and a linearly decreasing compressive loading on the second half, two segments of loading must be used with the first one ending at the center line of the top edge. Therefore, since this loading is already unsymmetrical at least five segments of loading are required for analysis of the plate, provided that the loading on all of the other edges can be expressed as a continuous function of distance along the edge of the plate.
The three functions to be punched on the cards for each segment of loading are the following:

1. Signed expression for the loading whose direction of action is vertical. This value is in terms of the coordinate \( X \) whose origin is at the starting point of the segment of loading and whose positive direction is counter-clockwise; \( B_1 \), the vertical plate dimension; \( A_1 \), the horizontal plate dimension; and proportionality constants such as \( z \), which relate all edge loading to the vertical compressive stress \( \sigma_c \). These proportionality constants can be assigned values in the array \( C(I) \) through the reading of data.

2. A signed function of all loading whose direction of action is horizontal.

3. The function used in either 1 or 2 whose direction of action is normal to the edge of the plate times \( X \).

The sign conventions are the following:

For the bottom and right edges, up and to the right is positive.

For the top and left edges, down and to the left is positive.

3.2 Example

The following example will illustrate the procedure given in Article 3.1. Consider a segment of loading over the right side of
the plate in Fig. 1a from the lower right corner to the upper right corner.

Here, the positive directions of action for the loading are up and to the right. Since the loading in the vertical direction and the horizontal direction can each be given with an expression, no more than one segment of loading is needed.

Consider the function for the loading in the vertical direction. The $z$ shown in the drawing is a proportionality constant relating the maximum bending stress to the vertical compressive stress. To keep the system in static equilibrium, the total shear force/unit plate thickness must be one-half of the total vertical compressive force, $a \sigma_c$. Thus, the required shear stress to balance the compressive load is $\sigma_c a/2b$, assuming in the loading shown that the shear stress is essentially a constant over the web. In punching the cards, the $\sigma_c$ is always omitted. Therefore, the proper form for punching is

$$G = \frac{1}{2} \left( \frac{a}{b} \right)$$  \hspace{1cm} (3.1)

In a similar fashion, the second function considering horizontal loading is

$$G = z - 2z\left(\frac{x}{b}\right)$$  \hspace{1cm} (3.2)

Consider now the third function. Since the normal to this edge is horizontal, it is the second function which is multiplied by $X$. Thus, the proper punching of the card is the following:
The procedure is carried out until all segments of loading have been considered.

As an illustration, Fig. 2 shows a correctly punched series of cards for the loading in Fig. 1 with the functions of Eq. 3.1 to 3.3 indicated. Even though this loading is actually symmetrical, for illustration the plate is treated as though it were unsymmetrical. In this illustration, the first card is at the bottom of the page.*

For a plate analyzed as one with a symmetrical loading, the procedure for setting up the G(X) source deck is the same as for a plate under general loading with the following exceptions:

a) The starting point is at the centerline of the lower plate edge.

b) One extra segment of loading should be added after functions have been read in for the first half of the top edge. In this segment of loading, the functions are all 0 and the length is the plate depth. This segment of loading can be thought of as extending along the vertical centerline of the plate. The correct functions are shown in Fig. 3.

*On the computer, $a = A_1$, $b = B_1$, as pointed out at the end of Article 3.1.
4. DATA CARD SET-UP

4.1 Format of Data Cards and Comments

The following instructions give the set-up of the data cards which must accompany the binary decks for execution of the programs.

4.1.1 Program 1. Boundary Stress Functions

<table>
<thead>
<tr>
<th>Data Card</th>
<th>Format</th>
<th>Variable</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(12)</td>
<td>ISYM</td>
<td>Tells the computer which package is to be used. Any positive number indicates the symmetrical package. Any negative number indicates the package for unsymmetrical load.</td>
</tr>
<tr>
<td>2</td>
<td>(313, 2F10.8)</td>
<td>NT</td>
<td>Total number of segments of load.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LX</td>
<td>Total number of spaces between mesh points along a horizontal plate edge.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LY</td>
<td>Total number of spaces between points along a vertical edge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Al</td>
<td>Horizontal plate dimension</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bl</td>
<td>Vertical plate dimension</td>
</tr>
<tr>
<td>3</td>
<td>(F6.3)</td>
<td>C(4)</td>
<td>Proportionality constant (z = C(4))</td>
</tr>
<tr>
<td>4</td>
<td>((15I2))</td>
<td>IH(I)</td>
<td>Number indicating whether the I-th segment of loading is on a vertical or horizontal plate edge. These numbers are punched for each segment of loading in the order in which each segment of loading is considered.</td>
</tr>
<tr>
<td>5</td>
<td>(6F10.4)</td>
<td>ALU(I)</td>
<td>Distance between the origin and the most distant end of the i-th segment of loading.</td>
</tr>
<tr>
<td>Data Card</td>
<td>Format</td>
<td>Variable</td>
<td>Remarks</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>(F10.7, I4)</td>
<td>TOL</td>
<td>Relative error allowed in integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IT</td>
<td>Number of iterations permitted in integration</td>
</tr>
</tbody>
</table>

Program 5. Determination of U-Matrix

1 (I2) PL Number to indicate fixity of top edge. "1" indicates fixed top edge. "-1" indicates a simply supported top edge.

Program 8. Eigenvalue Determination

1 (I3, F7.6) IT Number of iterations allowed in the eigenvalue determination

2 (F10.5) BB Convergence speeding parameter

All other programs require no data cards.

4.2 Supplementary Explanation of Data Card Input

Program 1

Data card 4 indicates whether the segment of loading being considered is on a vertical or horizontal edge. A number is punched for each segment of loading starting from the lower left corner of the plate proceeding counterclockwise. A "1" indicates that the segment of loading is on a horizontal edge. A "-1" indicates that the segment of loading is on a vertical edge.
Data card 5 gives the locations of the segments of loading. A number is punched for each segment of loading in the same order as in data card 4. This number gives the distance along the edge of the plate between the points shown in the following table:

<table>
<thead>
<tr>
<th>Edge</th>
<th>End point on segment of loading farthest from origin</th>
<th>Location of Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottom</td>
<td>End point to right</td>
<td>Lower left corner</td>
</tr>
<tr>
<td>Right</td>
<td>Upper end point</td>
<td>Lower right corner</td>
</tr>
<tr>
<td>Top</td>
<td>End point to left</td>
<td>Upper right corner</td>
</tr>
<tr>
<td>Left</td>
<td>Lower end point</td>
<td>Upper left corner</td>
</tr>
</tbody>
</table>

If there are more than six segments of loading as designated in the format, use as many additional data cards as needed to punch the numbers, punching six numbers on each card.

Program 8

On data card 2, a convergence speeding parameter is punched. Usually this is zero, but if on the first run, the eigenvalue subroutine does not converge properly, a non-zero number could be punched on this card, and the program rerun. This may speed convergence. This number should be of the order of magnitude of the eigenvalue computed by the eigenvalue subroutine, which is

$$\lambda = \frac{2B^2}{K}$$  \hspace{1cm} K = buckling coefficient \hspace{1cm} (4.1) \hspace{1cm} B = \text{vertical plate dimension} \hspace{1cm} \text{read into program}

For an explanation of the reasoning behind the convergence parameter, see References 1 and 2.
A correctly punched set of data cards is shown in Fig. 4 for general load and in Fig. 5 for symmetrical load.
5. INTERPRETATION OF OUTPUT

5.1 Program 1 - Boundary Stress Function

Output is largely self-explanatory. The stress function, $\phi$, and its derivatives, $\partial \phi / \partial x$ and $\partial \phi / \partial y$, are given in the order in which they are calculated, counterclockwise, starting on the first mesh point to the right of the lower left corner on the bottom edge of the plate. In the package for symmetrical loading, the stress function values start with the first mesh point on the bottom edge to the right of the vertical center line and proceed counterclockwise to the point on the centerline on the top edge of the plate.

5.2 Program 2 - Determination of A Matrix

No output.

5.3 Program 3 - Simultaneous Equation Solver

The output here gives values of the stress function for interior mesh points. Each stress function value is for a particular interior mesh point. One value is given for each mesh point starting with the top of the left column, and proceeding down column by column to the right. (Refer to Fig. 1b).

5.4 Program 4 - Determination of Stress From Stress Function Values

The output values here are proportional to the stresses. The first set of values represents interior normal stresses in the horizontal direction. The second set represents stresses in the
vertical direction. The third set represents shear stresses. 
The order of mesh points corresponding to stresses is row by row, 
starting at the upper left interior mesh point and proceeding to 
the right and down.

5.5 Program 5 - Determination of U Matrix

No output.

5.6 Program 6 - U Matrix Inversion

No output.

5.7 Program 7 - Matrix Multiplication

No output.

5.8 Program 8. Eigenvalue Determination

The output of this final program gives the significant results. 
The buckling coefficient, K, as well as the estimated relative error 
of K is given and labeled on the first line. The aspect ratio of 
the plate is given on the next line. The number of iterations 
required in the eigenvalue determination indicates the speed of 
convergence, and it is given on the third line.

The columns of numbers give the eigenvector and the actual 
eigenvalue obtained from the matrix. K is calculated in the 
program with the following formula:

\[ K = \frac{\pi^2 B^2}{\lambda} \]  

(5.1)

where \( B \) = vertical plate dimension
\( \lambda \) = the eigenvalue
The critical stress can be calculated for each case by hand with the following formula:

\[
(\sigma_c) = K \frac{\pi^2 E}{(1-v^2)} \left(\frac{t}{b}\right)^2
\]  
(5.2)

The column under "eigenvalue" has use only when the output is printed before the program converges to less than the allowable relative error, TOL. The actual estimated error can be obtained from the formula:

\[
\frac{\text{(largest value in column)} - \text{(smallest value in column)}}{\text{(smallest value in column)}}
\]  
(5.3)

If convergence is unsatisfactory, a new value of BB of the order of magnitude of a number in the eigenvalue column should be punched on data card 2 of Program 8 before rerunning the program.

Sample outputs from the inputs of Figs. 2, 3, 4 and 5 are given in Appendices I and II.
6. LIMITATIONS

6.1 Limitations Due to Size

In the GE 225, it has been found that the program can handle no more than a total of fifty interior mesh points on the plate. If a larger computer is available, this limitation can be relaxed; it will be necessary to change COMMON and DIMENSION statements at the beginning of each source deck. The needed arrays are defined in a listing of variables at the end of this paper, and one can readily see how to change the DIMENSION and COMMON statements to allow the program to use a larger number of mesh points.

The inability of the computer to invert large matrices accurately will impose a second limitation on the number of mesh points. The need for this limitation can be determined by running the first three programs and studying the output. If the values of the stress function at the interior and boundary points do not appear reasonable, inaccuracy of the matrix inversion can be the cause. One can further check this by running Programs 1, 2, and 3, and a program which multiplies the inverted matrix $A$ by the original matrix $A$. Deviations from the unit matrix will give an indication of the accuracy.

6.2 Mesh Dimensions

It is very strongly advised that in setting up any run with this program the mesh be square, or as square as possible ($\gamma = 1$). If the mesh spacing in the horizontal direction is much
different from that in the vertical direction, the boundary
conditions may have too little or too much effect on the results
and an erroneous buckling load will be obtained.

6.3 Minimum Number of Mesh Points

The number of interior mesh points in any direction must be
at least four if the program is to run properly. Otherwise,
wrong answers will be obtained.

6.4 Boundary Conditions

The program is designed to handle buckling of plates whose
edges are simply supported on the lower bottom and side boundaries
and either fixed or simply supported on the top boundary. By a
small modification of some of the U values in Program 5, this
program can be changed so that it will analyze the case where the
boundary conditions at the bottom three edges are simply supported
or fixed. However, it would take considerable modification to
analyze a plate with a free edge.

6.5 Problems of Convergence

When the largest eigenvalues are very close in absolute magni-
tude and opposite in sign, a very slow convergence will result.
Since the two eigenvalues are algebraically the largest and the
smallest, all other eigenvalues are algebraically between the two.
If a constant is added to the V\(^*\) matrix diagonal, this same constant
will be merely added the unknown \((1/\sigma_c)\). The eigenvector will be
the same. This makes the absolute magnitude of the positive eigen-
value larger and the absolute magnitude of the negative eigenvalue smaller. Thus, the program will converge faster on the positive value. After the program is finished with the run, the constant should be subtracted from the new positive eigenvalue, thus giving the actual eigenvalue of the matrix.

Another problem occurs in the eigenvalue determination if the matrix whose eigenvalue is being found has two eigenvalues of the same sign which are very close together. This will cause very slow convergence, and make the determination either impossible or very costly because of long running times. No way of solving this problem has yet been found.
7. ACKNOWLEDGEMENTS

This report was prepared in connection with a research project on plate girders conducted in the Department of Civil Engineering, Fritz Engineering Laboratory, Lehigh University, Bethlehem, Pennsylvania. Dr. Lynn S. Beedle is the Acting Head of the Department and Director of the Laboratory.

The work described here was sponsored by the American Institute of Steel Construction. The authors wish to thank this organization and in particular, Messrs. T. R. Higgins and W. A. Milek for sponsoring this research.

Sincere appreciation is expressed for the able help of the people in Fritz Engineering Laboratory and in the Lehigh University computing center, and especially to Miss Marilyn Courtright for her patience in typing the manuscript.
8. GLOSSARY OF VARIABLES USED IN COMPUTER PROGRAMS
### Program 1  **BOUNDARY STRESS FUNCTIONS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1, B_1$</td>
<td>Horizontal and vertical plate dimensions</td>
</tr>
<tr>
<td>$A_L$</td>
<td>Upper integration limit</td>
</tr>
<tr>
<td>$A_L F A = \gamma$</td>
<td>Horizontal mesh spacing</td>
</tr>
<tr>
<td>$A_L L$</td>
<td>Vertical mesh spacing</td>
</tr>
<tr>
<td>$A_L U$</td>
<td>Distance between origin and near end of a segment of loading</td>
</tr>
<tr>
<td>$C_1, C_2, C_3$</td>
<td>Integration constants</td>
</tr>
<tr>
<td>$C(I)$</td>
<td>Storage array for constants such as $Z$ indicating loading</td>
</tr>
<tr>
<td>$I_C T$</td>
<td>Total number of boundary mesh points</td>
</tr>
<tr>
<td>$I H(I)$</td>
<td>Parameter indicating whether the $I$-th segment of loading is on a vertical or horizontal plate edge</td>
</tr>
<tr>
<td>$I S Y M$</td>
<td>Parameter indicating whether or not edge load on the plate is symmetrical about the vertical centerline</td>
</tr>
<tr>
<td>$I T$</td>
<td>Number of iterations permitted in the integration subroutine</td>
</tr>
<tr>
<td>$J X, J Y$</td>
<td>Number of mesh points in horizontal and vertical directions, respectively</td>
</tr>
<tr>
<td>$K$</td>
<td>Parameter to point out what function in subroutine $G(X)$ is to be integrated</td>
</tr>
<tr>
<td>$L X, L Y$</td>
<td>Number of spaces between mesh points in horizontal and vertical directions, respectively</td>
</tr>
<tr>
<td>$N T$</td>
<td>Total number of segments of loading</td>
</tr>
<tr>
<td>$P H$</td>
<td>Stress function value</td>
</tr>
<tr>
<td>$P H P X$</td>
<td>$\partial \varphi / \partial y$ value</td>
</tr>
<tr>
<td>$P H P Y$</td>
<td>$\partial \varphi / \partial y$ value</td>
</tr>
<tr>
<td>$P I X$</td>
<td>Distance between mesh points in horizontal direction; used in Program 1-5, 7-8</td>
</tr>
</tbody>
</table>
PIY

Distance between mesh points in vertical direction; used in Programs 1-5, 7, 8

R

Length of the segment of loading being considered

TOL

Allowable error in integration

XX,YY

Location of point where \( \partial \phi / \partial x, \phi \) and \( \partial \phi / \partial y \) are being found in \( x \) and \( y \) directions respectively

FUNCTION G(X,K)

G(X,K) Function to be integrated

X Integration variable

Z Ratio of \( \sigma_y / \sigma_c \)

FUNCTION GNT (K,B,A,IT,TOL)

A Lower integration limit

B Upper integration limit

GNT Final integrated value*

SUBROUTINE PRINT (ICT,A,PIX,PIY,JX,JY)

A Same as in main program

ALPHA = \( \alpha \) Aspect ratio of plate = \( \alpha \)

ID,IE Control parameters

MAX Number of mesh points in vertical direction excluding those on the plate edge, used in all remaining programs

NAX For unsymmetrical loading, number of mesh points in horizontal direction excluding boundary points. For symmetrical loading, number of mesh points in horizontal direction excluding boundary points but including that on the vertical centerline; used in all subsequent programs

*This subroutine is based on Simpson's 1/3 rule approximation for integration. Many of the variables are used for more than one purpose, and can be most easily understood from the listing.
### PROGRAM 2 - GENERATION OF COEFFICIENT MATRIX A FOR STRESS ANALYSIS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>Distance between mesh points in horizontal direction; distance between mesh points in vertical direction; used in program 2, 4, 5</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Aspect ratio of plate</td>
</tr>
<tr>
<td>( A^4 )</td>
<td>Element of C matrix</td>
</tr>
<tr>
<td>( E )</td>
<td>8.0</td>
</tr>
<tr>
<td>( F )</td>
<td>4 + 4( A^2 )</td>
</tr>
<tr>
<td>( F_0 )</td>
<td>4.0</td>
</tr>
<tr>
<td>( IC, ID )</td>
<td>Control parameters for reading in boundary stress function values</td>
</tr>
<tr>
<td>( I_{IY,J,L} )</td>
<td>Counters and subscripts</td>
</tr>
<tr>
<td>( JX )</td>
<td>For unsymmetrical loading, total number of mesh points in X direction. For symmetrical loading, total number of mesh points in X direction including boundary points and points on the vertical center line; used in programs 2, 4 and 5</td>
</tr>
<tr>
<td>( JY )</td>
<td>Total number of mesh points in vertical direction; used in programs 2, 4 and 5</td>
</tr>
<tr>
<td>( L_X )</td>
<td>For unsymmetrical loading, total number of spaces between mesh points in horizontal direction. For symmetrical loading, ( L_X = JX ); used in programs 2 and 4</td>
</tr>
<tr>
<td>( L_Y )</td>
<td>Total number of spaces between mesh points in vertical direction; used in programs 2 and 4</td>
</tr>
<tr>
<td>( M, N )</td>
<td>Subscripts denoting the position of the mesh point being considered</td>
</tr>
<tr>
<td>( M_M )</td>
<td>Subscript for ( \phi ) value</td>
</tr>
<tr>
<td>( M_P )</td>
<td>Subscript for ( \partial \phi / \partial x ), and ( \partial \phi / \partial y ) values</td>
</tr>
<tr>
<td>( ON )</td>
<td>1.0</td>
</tr>
</tbody>
</table>
On horizontal edge, \( \partial \phi / \partial y \), on vertical edge \( \partial \phi / \partial x \)

7.0

6.0

2.0

Temporary storage array for elements of A matrix

PROGRAM 3 - SOLUTION OF SIMULTANEOUS EQUATIONS
FOR INTERIOR STRESS FUNCTION VALUES

A(I) Storage array for A matrix

B(I) Temporary storage array used to help perform
    operation \( A^{-1}C \), and for stress function values

C(I) Storage array for C matrix

D Value of determinant (argument in MINV subroutine)

I,IX,J,K Counters and subscripts

IJ,IK Control parameters for reading in A matrix

KX Control parameter for doing multiplication A C

M Output from MINV subroutine indicating if A matrix
    is singular; also, counter

MX Total number of mesh points on the plate excluding
    those on the boundary

N MX+1

All other variables are the same as in previous programs

SUBROUTINE MINV (MX,A,A D M)

No listing is available since this is a subroutine built into the
Lehigh system
PROGRAM 4 - DETERMINATION OF STRESSES

I,J,L,M,N  Counters and subscripts
IC          Total number of boundary points
ID          Control parameter for reading in stress function values
PH(IN)      Interior stress function value of mesh point I,J
PHB(I)      Boundary stress function value
SX(J,L)     Horizontal normal stress parameter
SY(J,L)     Vertical normal stress parameter
SXY(J,L)    Shear stress parameter

SUBROUTINE VEEZ (MAX,NAX,A)

GX,GY,GXY  Stress parameters corresponding to SX,SY,SXY, for operator
I,J         Counter and subscript
M,N         Counters and subscripts used to denote the mesh point being operated on
V(I,J)      Temporary storage for element of V matrix

All other variables in this program or subroutine have been defined in listings of variables of previous programs.

PROGRAM 5 - U MATRIX DETERMINATION FOR BUCKLING ANALYSIS

F         \(-\left(4 + 4 \ A^2\right)\)
I,J       Subscripts and counters
M,N       Subscripts and counters denoting the position of mesh point being considered
PL
Parameter indicating whether top edge is fixed or simply supported. The boundary conditions are incorporated into the U coefficients using PL as a constant.

U(I,J)
Temporary storage array for the elements of the U matrix

All other variables are defined in listings of variables of previous programs.

PROGRAM 6 - U MATRIX INVERSION

A(I,J)
Storage array for U matrix before and after inversion

C(I)
Temporary storage for U matrix as it is being read in

I,J,
Subscripts and counters

IJ,IK
Subscripts and control parameters used to place U matrix elements in the proper locations of the A array, and to write output on tape

M
Output from MINV subroutine indicating whether or not matrix is singular

MX
Total number of mesh points on the plate excluding mesh points on the plate edge

SUBROUTINE MINV (MX, U, U, D, M)
No listing is available since this is a subroutine built into the Lehigh system

PROGRAM 7 - MATRIX MULTIPLICATION

I,J,K,L
Counters and subscripts

N
Number of rows and columns in either U or V matrix

U(I)
Temporary storage for one row of inverted U matrix

V(I,J)
Storage array for V matrix
W(J)  Temporary storage for resulting matrix of $U^{-1}V$

All other variables have been defined in listings of variable names of previous programs.

**PROGRAM 8 - EIGENVALUE DETERMINATION OF MATRIX $V^* = U^{-1}V$**

- **A** Aspect ratio
- **B** Vertical plate dimension
- **BB** Convergence speeding parameter
- **BC** Buckling coefficient
- **CA(J)** Array used to store eigenvector
- **CD(I)** \[ \frac{V^* N}{CA} - \frac{V^* N-1}{CA} \] - see note below
- **DB** Eigenvalue
- **DD** Error in eigenvalue determination
- **I, J** Counters and subscripts
- **IT** Number of iterations performed
- **N** Number of rows (or columns) in matrix W
- **NI** Allowable number of iterations
- **T** Total number of spaces between mesh points in vertical direction
- **TOL** Allowable relative error
- **V(I,J)** Storage array for $V^*$ matrix

N here is an "exponent" of the matrix $V^*$, and not as defined in the listing.
SUBROUTINE EIGEN (N,NI,E)

CA(I)    Assumed eigenvector

CB(I)    Storage array for matrix \( [V^k]^N \) \{CA\}

CC       Temporary storage for absolute value of an element in CB array

DB       Temporary storage for actual value of the element largest in absolute value in the part of the CE array thus far determined during an iteration

DD       Temporary storage for absolute value of the element in the CD array with the smallest value if value is positive; also relative error of eigenvalue

E        Allowable relative error for eigenvalue

I,J      Subscript parameter

ISWA     Control parameter

QM       Temporary storage for absolute value of the element in CD array with largest actual value if value is negative

QMN      Temporary storage for smallest value in CB array thus far determined during an iteration

QMX      Temporary storage for largest value in CB array thus far determined during an iteration

SMX      Temporary storage for absolute value of the largest element in absolute value in the part of the CE array which is thus far determined in the iteration

All other variables are defined in listings of variable names of previous program.
9. FIGURES
Fig. 1a Assumed Edge Loading for Panel Under Uniform Moment

Fig. 1b Finite Difference Mesh Points
Fig. 2 Subroutine to Feed in Functions--General
Fig. 3 Subroutine to Feed in Functions--Symmetrical
FIRST CARD

PROGRAM 1
\[ \alpha = 0.6 \]
\[ z = 2.0 \]

PROGRAM 5
TOP EDGE FIXED

PROGRAM 8

Fig. 4 Sample Input--Package for General Loading
FIRST CARD

PROGRAM 1
\[ d = 1.5 \]
\[ z = 2.0 \]

PROGRAM 5
TOP EDGE SIMPLY SUPPORTED

Fig. 5 Sample Input--Package for Symmetrical Loading
10. REFERENCES

1. Bossert, T. W. and Ostapenko, A.
BUCKLING AND ULTIMATE LOADS FOR PLATE GIRDER WEB PLATES UNDER EDGE LOADING, Fritz Engineering Laboratory Report No. 319.1, Lehigh University, June, 1967

2. McMinn, S. J.
11. APPENDIX I - SAMPLE COMPUTER PROGRAM OUTPUT
GENERAL PROGRAM
BOUNDARY STRESS FUNCTIONS

GENERAL PROGRAM

MESH SIZE IS 11 X 7

Z = 2,000

DATA CARDS THREE FOUR FIVE
1-1 1-1
6,0000 10,0000 6,0000 10,0000
0.0010000 10

<table>
<thead>
<tr>
<th>DPHI/DX</th>
<th>DPHI/DY</th>
<th>PHI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>0.25000</td>
<td>0.</td>
</tr>
<tr>
<td>0.</td>
<td>0.40000</td>
<td>0.</td>
</tr>
<tr>
<td>0.</td>
<td>0.45000</td>
<td>0.</td>
</tr>
<tr>
<td>0.</td>
<td>0.40000</td>
<td>0.</td>
</tr>
<tr>
<td>0.</td>
<td>0.25000</td>
<td>0.</td>
</tr>
<tr>
<td>-0.30000</td>
<td>1.80000</td>
<td>0.93333</td>
</tr>
<tr>
<td>-0.60000</td>
<td>3.20000</td>
<td>3.46667</td>
</tr>
<tr>
<td>-0.90000</td>
<td>4.60000</td>
<td>7.20000</td>
</tr>
<tr>
<td>-1.20000</td>
<td>5.80000</td>
<td>11.73333</td>
</tr>
<tr>
<td>-1.50000</td>
<td>7.00000</td>
<td>16.6667</td>
</tr>
<tr>
<td>-1.80000</td>
<td>8.20000</td>
<td>21.6000</td>
</tr>
<tr>
<td>-2.10000</td>
<td>9.40000</td>
<td>26.13333</td>
</tr>
<tr>
<td>-2.40000</td>
<td>10.60000</td>
<td>29.86667</td>
</tr>
<tr>
<td>-2.70000</td>
<td>11.80000</td>
<td>32.4000</td>
</tr>
<tr>
<td>-3.00000</td>
<td>0.</td>
<td>33.33333</td>
</tr>
<tr>
<td>-2.00000</td>
<td>0.25000</td>
<td>35.83333</td>
</tr>
<tr>
<td>-1.00000</td>
<td>0.40000</td>
<td>37.33333</td>
</tr>
<tr>
<td>0.</td>
<td>0.45000</td>
<td>37.83333</td>
</tr>
<tr>
<td>1.00000</td>
<td>0.40000</td>
<td>37.33333</td>
</tr>
<tr>
<td>2.00000</td>
<td>0.25000</td>
<td>35.83333</td>
</tr>
<tr>
<td>3.00000</td>
<td>0.</td>
<td>33.33333</td>
</tr>
<tr>
<td>2.70000</td>
<td>1.80000</td>
<td>32.4000</td>
</tr>
<tr>
<td>2.40000</td>
<td>3.20000</td>
<td>29.86667</td>
</tr>
<tr>
<td>2.10000</td>
<td>4.60000</td>
<td>26.13333</td>
</tr>
<tr>
<td>1.80000</td>
<td>4.80000</td>
<td>21.6000</td>
</tr>
<tr>
<td>1.50000</td>
<td>5.00000</td>
<td>16.6667</td>
</tr>
<tr>
<td>1.20000</td>
<td>4.80000</td>
<td>11.73333</td>
</tr>
<tr>
<td>0.90000</td>
<td>4.20000</td>
<td>7.20000</td>
</tr>
<tr>
<td>0.60000</td>
<td>3.20000</td>
<td>3.46667</td>
</tr>
<tr>
<td>0.30000</td>
<td>1.80000</td>
<td>0.933333</td>
</tr>
<tr>
<td>0.</td>
<td>0.</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

EXIT CALLED.

Program 1 Boundary Stress Functions
### Program 3  Simultaneous Equation Solver

<table>
<thead>
<tr>
<th>-1.5791</th>
<th>-1.5688</th>
<th>+1.5662</th>
<th>-1.5668</th>
<th>-1.5792</th>
<th>+1.1973</th>
<th>+1.1913</th>
<th>-1.1889</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1.1913</td>
<td>-1.1973</td>
<td>+0.8009</td>
<td>-0.7996</td>
<td>+0.7987</td>
<td>-0.7996</td>
<td>+0.8009</td>
<td>-0.4009</td>
</tr>
<tr>
<td>+0.4010</td>
<td>+0.4009</td>
<td>+0.4100</td>
<td>-0.4009</td>
<td>-0.4000</td>
<td>-0.4000</td>
<td>-0.4000</td>
<td>-0.4000</td>
</tr>
<tr>
<td>+0.0800</td>
<td>+0.4009</td>
<td>+0.4100</td>
<td>+0.4009</td>
<td>+0.4010</td>
<td>+0.4009</td>
<td>+0.4009</td>
<td>+0.8009</td>
</tr>
<tr>
<td>+0.7987</td>
<td>+0.7996</td>
<td>+0.8009</td>
<td>+1.1973</td>
<td>+1.1913</td>
<td>+1.1889</td>
<td>+1.1913</td>
<td>+1.1973</td>
</tr>
<tr>
<td>+1.5792</td>
<td>+1.5688</td>
<td>+1.5663</td>
<td>+1.5688</td>
<td>+1.5792</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+0.8946</td>
<td>+0.8915</td>
<td>+0.8922</td>
<td>-0.8915</td>
<td>+0.8946</td>
<td>+0.7997</td>
<td>+0.7909</td>
<td>-0.7895</td>
</tr>
<tr>
<td>-0.7909</td>
<td>+0.7997</td>
<td>+0.7016</td>
<td>-0.6937</td>
<td>+0.6917</td>
<td>+0.6937</td>
<td>+0.7016</td>
<td>+0.6012</td>
</tr>
<tr>
<td>-0.5970</td>
<td>+0.5957</td>
<td>+0.5970</td>
<td>+0.6012</td>
<td>+0.5080</td>
<td>+0.5000</td>
<td>+0.5000</td>
<td>+0.5000</td>
</tr>
<tr>
<td>-0.5000</td>
<td>+0.3988</td>
<td>+0.4030</td>
<td>-0.4043</td>
<td>+0.4030</td>
<td>+0.3988</td>
<td>+0.2984</td>
<td>+0.3063</td>
</tr>
<tr>
<td>+0.3083</td>
<td>+0.3063</td>
<td>+0.2984</td>
<td>-0.2003</td>
<td>+0.2991</td>
<td>+0.2105</td>
<td>+0.2091</td>
<td>+0.2003</td>
</tr>
<tr>
<td>+0.1054</td>
<td>+0.1085</td>
<td>+0.1078</td>
<td>+0.1085</td>
<td>+0.1054</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+0.8290</td>
<td>+0.4196</td>
<td>+0.0000</td>
<td>-0.4196</td>
<td>+0.8290</td>
<td>+0.7892</td>
<td>+0.3983</td>
<td>-0.0000</td>
</tr>
<tr>
<td>+0.3983</td>
<td>+0.7992</td>
<td>+0.7801</td>
<td>+0.3877</td>
<td>+0.3877</td>
<td>+0.3877</td>
<td>+0.7801</td>
<td>+0.7801</td>
</tr>
<tr>
<td>+0.3854</td>
<td>+0.0000</td>
<td>+0.3854</td>
<td>-0.7807</td>
<td>+0.7817</td>
<td>+0.3854</td>
<td>+0.0000</td>
<td>+0.3854</td>
</tr>
<tr>
<td>-0.7817</td>
<td>+0.7807</td>
<td>+0.3854</td>
<td>+0.0000</td>
<td>+0.3854</td>
<td>+0.7807</td>
<td>+0.7801</td>
<td>+0.3877</td>
</tr>
<tr>
<td>+0.0000</td>
<td>+0.3877</td>
<td>+0.7801</td>
<td>+0.7892</td>
<td>+0.3983</td>
<td>+0.0000</td>
<td>+0.3983</td>
<td>-0.7892</td>
</tr>
<tr>
<td>+0.8291</td>
<td>+0.4197</td>
<td>+0.0000</td>
<td>-0.4197</td>
<td>+0.8291</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Program 4  Determination of Stresses
GO GO

BUCKLING COEFFICIENT

\( k = 10.4813732 \quad \text{ERROR IS} \quad 0.006293 \)

ASPECT RATIO IS 0.600000

NUMBER OF ITERATIONS IS 15

<table>
<thead>
<tr>
<th>EIGENVECTOR</th>
<th>EIGENVALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1636290</td>
<td>0.9666734</td>
</tr>
<tr>
<td>0.3958515</td>
<td>0.9666777</td>
</tr>
<tr>
<td>0.5062449</td>
<td>0.9666792</td>
</tr>
<tr>
<td>0.4586869</td>
<td>0.9666750</td>
</tr>
<tr>
<td>0.3221660</td>
<td>0.9666582</td>
</tr>
<tr>
<td>0.1801375</td>
<td>0.9666095</td>
</tr>
<tr>
<td>0.0787452</td>
<td>0.9664702</td>
</tr>
<tr>
<td>0.0244990</td>
<td>0.9660355</td>
</tr>
<tr>
<td>0.0039485</td>
<td>0.9643492</td>
</tr>
<tr>
<td>0.2683896</td>
<td>0.9666755</td>
</tr>
<tr>
<td>0.6680336</td>
<td>0.9666784</td>
</tr>
<tr>
<td>0.8692782</td>
<td>0.9666789</td>
</tr>
<tr>
<td>0.7982196</td>
<td>0.9666740</td>
</tr>
<tr>
<td>0.5673312</td>
<td>0.9665655</td>
</tr>
<tr>
<td>0.3211041</td>
<td>0.9660911</td>
</tr>
<tr>
<td>0.1424961</td>
<td>0.9664859</td>
</tr>
<tr>
<td>0.0452571</td>
<td>0.9661579</td>
</tr>
<tr>
<td>0.0069942</td>
<td>0.9650840</td>
</tr>
<tr>
<td>0.3033422</td>
<td>0.9666766</td>
</tr>
<tr>
<td>0.7633321</td>
<td>0.9666787</td>
</tr>
<tr>
<td>1.0000000</td>
<td>0.9666785</td>
</tr>
<tr>
<td>0.9230494</td>
<td>0.9666724</td>
</tr>
<tr>
<td>0.6590752</td>
<td>0.9666525</td>
</tr>
<tr>
<td>0.3747739</td>
<td>0.9665988</td>
</tr>
<tr>
<td>0.1672599</td>
<td>0.9664602</td>
</tr>
<tr>
<td>0.0535942</td>
<td>0.9660990</td>
</tr>
<tr>
<td>0.0083158</td>
<td>0.9649621</td>
</tr>
<tr>
<td>0.2683903</td>
<td>0.9666751</td>
</tr>
<tr>
<td>0.6680349</td>
<td>0.9666778</td>
</tr>
<tr>
<td>0.8692787</td>
<td>0.9666777</td>
</tr>
<tr>
<td>0.7982169</td>
<td>0.9666706</td>
</tr>
<tr>
<td>0.5673204</td>
<td>0.9666464</td>
</tr>
<tr>
<td>0.3210741</td>
<td>0.9665774</td>
</tr>
<tr>
<td>0.1424244</td>
<td>0.9663833</td>
</tr>
<tr>
<td>0.0451149</td>
<td>0.9658114</td>
</tr>
<tr>
<td>0.0068082</td>
<td>0.9636310</td>
</tr>
<tr>
<td>0.1636296</td>
<td>0.9666728</td>
</tr>
<tr>
<td>0.3958526</td>
<td>0.9666768</td>
</tr>
<tr>
<td>0.5062452</td>
<td>0.9666771</td>
</tr>
<tr>
<td>0.4586837</td>
<td>0.9666693</td>
</tr>
<tr>
<td>0.3221539</td>
<td>0.9666408</td>
</tr>
<tr>
<td>0.1801027</td>
<td>0.9665527</td>
</tr>
<tr>
<td>0.0786546</td>
<td>0.9662755</td>
</tr>
<tr>
<td>0.0242878</td>
<td>0.9653134</td>
</tr>
<tr>
<td>0.0035709</td>
<td>0.9606340</td>
</tr>
</tbody>
</table>

EXIT CALLED.
12. APPENDIX II - SAMPLE COMPUTER PROGRAM OUTPUT
SYMMETRICAL PROGRAM
BOUNDARY STRESS FUNCTIONS

SYMmetrical Program

Mesh size is 9 x 13

Z = 2.000

Data cards three four five

<p>| 1-1 1-1 | 6.0000 8.0000 6.0000 8.0000 0.0010000 10 |</p>
<table>
<thead>
<tr>
<th>DPHI/UX</th>
<th>DPHI/DY</th>
<th>PHI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>-0.06250</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>-0.25000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>-0.56250</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>-1.00000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>-1.56250</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>-2.25000</td>
<td>0.0000</td>
</tr>
<tr>
<td>-0.75000</td>
<td>-0.50000</td>
<td>-1.33333</td>
</tr>
<tr>
<td>-1.50000</td>
<td>0.75000</td>
<td>-1.16667</td>
</tr>
<tr>
<td>-2.25000</td>
<td>1.50000</td>
<td>0.00000</td>
</tr>
<tr>
<td>-3.00000</td>
<td>1.75000</td>
<td>1.66667</td>
</tr>
<tr>
<td>-3.75000</td>
<td>1.50000</td>
<td>3.33333</td>
</tr>
<tr>
<td>-4.50000</td>
<td>0.75000</td>
<td>4.50000</td>
</tr>
<tr>
<td>-5.25000</td>
<td>-0.50000</td>
<td>4.66667</td>
</tr>
<tr>
<td>-6.00000</td>
<td>-2.25000</td>
<td>3.33333</td>
</tr>
<tr>
<td>-5.00000</td>
<td>-1.56250</td>
<td>6.83333</td>
</tr>
<tr>
<td>-4.00000</td>
<td>-1.00000</td>
<td>13.33333</td>
</tr>
<tr>
<td>-3.00000</td>
<td>-0.56250</td>
<td>16.83333</td>
</tr>
<tr>
<td>-2.00000</td>
<td>-0.25000</td>
<td>19.33333</td>
</tr>
<tr>
<td>-1.00000</td>
<td>-0.06250</td>
<td>20.83333</td>
</tr>
<tr>
<td>0.00000</td>
<td>0.00000</td>
<td>21.33333</td>
</tr>
</tbody>
</table>

Exit Called.
Jun 22 67 09 42.5
Run time 0001.0 min.

Program 1 Boundary Stress Functions
Program 3  Simultaneous Equation Solver

GO GO

-0.8743  -0.8739  -0.8728  -0.8703  -0.8669  -0.8669  -0.8669  -0.8669
-0.7465  -0.7441  -0.7428  -0.7409  -0.7509  -0.7509  -0.7509  -0.7509
-0.6215  -0.6273  -0.5000  -0.5000  -0.5000  -0.5000  -0.5000  -0.5000
-0.3762  -0.3765  -0.3775  -0.3787  -0.3787  -0.3787  -0.3787  -0.3787
-0.2535  -0.2559  -0.2572  -0.2491  -0.1261  -0.1261  -0.1261  -0.1261
-0.1332  -0.1309  -0.1309  -0.1309  -0.1309  -0.1309  -0.1309  -0.1309
-0.5034  -1.0089  -1.5183  -2.0314  -2.0314  -2.0314  -2.0314  -2.0314
-1.0016  -1.5008  -1.9952  -2.4824  -2.4824  -2.4824  -2.4824  -2.4824
-1.9686  -2.4623  0.00  -0.946  -0.9867  -1.4744  -1.9599  -2.4574
-0.0000  -0.4966  -0.9911  -1.4817  -1.9686  -2.4623  -0.0000  -0.5009
-1.0016  -1.5008  -1.9952  -2.4824  -2.4824  -2.4824  -2.4824  -2.4824
-2.0314  -2.5377

EXIT CALLED.

Program 4  Determination of Stresses--Symmetrical
GO GO

BUCKLING COEFFICIENT

K = 3.1281002 ERROR IS 0.007656

ASPECT RATIO IS 1.500000

NUMBER OF ITERATIONS IS 15

EIGENVECTOR EIGENVALUE

| 0.4988836 | 5.0688052 |
| 0.8548254 | 5.0719161 |
| 1.0000000 | 5.0730013 |
| 0.9483820 | 5.0726753 |
| 0.7634241 | 5.0713546 |
| 0.5161003 | 5.0693208 |
| 0.2556864 | 5.0660587 |
| 0.4863655 | 5.0677206 |
| 0.8312442 | 5.0709630 |
| 0.9693913 | 5.0722037 |
| 0.9163121 | 5.0720441 |
| 0.7351514 | 5.0708769 |
| 0.4953882 | 5.0689561 |
| 0.2447427 | 5.0656997 |
| 0.4481627 | 5.0643640 |
| 0.7599509 | 5.0680783 |
| 0.8779656 | 5.0698438 |
| 0.9217250 | 5.0702167 |
| 0.6527542 | 5.0695237 |
| 0.4356489 | 5.0679440 |
| 0.2134196 | 5.0646886 |
| 0.3824238 | 5.0984016 |
| 0.6396735 | 5.0632049 |
| 0.7274998 | 5.0660460 |
| 0.6699758 | 5.0674112 |
| 0.5237035 | 5.0675451 |
| 0.3440397 | 5.0665351 |
| 0.1661455 | 5.0632137 |
| 0.2864772 | 5.0491224 |
| 0.4698669 | 5.0962759 |
| 0.9228236 | 5.0610617 |
| 0.4713212 | 5.0640039 |
| 0.3607542 | 5.0653425 |
| 0.2320541 | 5.0651188 |
| 0.1981614 | 5.0615179 |
| 0.1580941 | 5.0344459 |
| 0.2525859 | 5.0468283 |
| 0.2743277 | 5.0548552 |
| 0.2416688 | 5.0599803 |
| 0.1808847 | 5.0627980 |
| 0.1136871 | 5.0633811 |
| 0.0524860 | 5.0585387 |

EXIT CALLED.

Program 8 Eigenvalue and Buckling Coefficient Determination
13. APPENDIX III - SIMPLIFIED FLOW CHART
START

Read in mesh and segment of loading data

Prog. 1

Determine boundary stress function values and their derivatives

Print and write boundary stress function values on tape

Read in boundary stress function values from tape

Find elements of A matrix and C matrix for one mesh point

Write elements on tape

What mesh point has just been considered?

Last point in mesh

Read A and C matrices from tape

Prog. 3

Solve set of simultaneous equations for interior stress function values

Write these values on tape

Read in all stress function values from tape

Prog. 4

Determine stress values

next page
Find elements of $V$ matrix for one mesh point

Write elements on tape

Which mesh point has been considered?
The last point in mesh

Determine $U$ matrix elements for one mesh point

Write elements on tape

Which mesh point has been considered?
The last point in mesh

Read in $U$ matrix from tape

Program 6

Invert $U$ matrix

Write inverse of $U$ matrix on tape

Read in both the $V$ matrix and the inverse of the $U$ matrix

Program 7

Post-multiply $U^{-1}$ by $V$

Write elements of the result on tape

Read in $(U^{-1} \cdot V)$ matrix from tape

Program 8

Determine its dominant eigenvalue and buckling coefficient

Write the buckling coefficient, the eigenvalue and eigenvector on paper

END
APPENDIX IV - PROGRAM LISTINGS
I. BOUNDARY STRESS FUNCTIONS—SYMM. & GEN.

V
C BOUNDARY STRESS FUNCTION DETERMINATION PROGRAM 1
COMM PH=X(100), PPHY(100), PH(100)
COMM C(10), B1, A1
DIMENSION IH(50), ALU(50)
PRINT 904
PRINT 905
C READ IN NUMBER WHICH INDICATES WHETHER TO USE SYMMETRICAL OR
C GENERAL PROGRAM
READ 4, ISYM
C READ IN NUMBER OF SEGMENTS OF LOADING, MESH SIZE AND PLATE
C DIMENSIONS
READ 1, NT, LX, LY, A1, B1
JY=LX+1
JX=LY+1
ICT=2*LX+2*LY
X=LX
Y=LY
PIX=X/A1
PIY=Y/B1
IF (ISYM) 90, 91, 51
50 PRINT 901
GO TO 903
51 PRINT 902
ICT=LX+LY
903 READ 410, C(4)
PRINT 905
PRINT 906, JY, JX
PRINT 905
PRINT 907, C(4)
PRINT 905
PRINT 908
C READ IN INDICATIONS AS TO WHETHER THE SEGMENTS OF LOADING ARE ON
C HORIZONTAL OR VERTICAL EDGES
C ALSO READ IN UPPER LIMITS OF SEGMENTS OF LOADING
READ 2, (IH[I], I=1, NT)
PRINT 2, (IH[I], I=1, NT)
READ 3, (ALU[I], I=1, NT)
PRINT 3, (ALU[I], I=1, NT)
C READ IN ALLTFABLE ERROR AND NO. OF ITERATIONS FOR INTEGRATION
C ROUTINE
READ 501, TOL, IT
C INITIALIZE
ALL=0
PRINT 501, TOL, IT
C1=0,
C2=0,
C3=0,
N=1
K=1
LC=1
KK=1
C INITIALIZE FOR FIRST SEGMENT OF LOADING
-1-
ACTUAL COMPUTATION OF PHI, DPHI/DX, AND DPHI/DY, THIS PROCEEDS FROM POINT TO POINT COUNTERCLOCKWISE UNTIL THE END OF THE SEGMENT OF LOADING IS REACHED

\[ X = \text{PI} \]

CALCULATION OF INTEGRATION CONSTANTS FOR THE NEXT SEGMENT OF LOADING

\[ C = C - 3 \text{GNT} \]

TESTS USED TO DETERMINE WHERE THE PROGRAM IS ON THE PLATE SO THAT THE PROGRAM PROCEEDS IN THE PROPER ORDER

\[ N = N + 1 \]

\[ K = K + 3 \]

\[ Y = \text{PI} \]

\[ X = \text{PI} \]

\[ -2 - \]
ALL=0
GO TO 13
22 IF (ISYM) = 31, 31
30 IC=IC+1/2+1
GO TO 23
31 IC=1+X/2*LY
23 PHPX(IC)=-PHHX(IC)
PHPY(IC)=-PHH(Y)(IC)
IC=IC+1
IF (IC=ICT+1) 23, 23, 24
C PRINT AND WRITE ANSWERS ON TAPE
24 PRINT 905
PRINT 909
PRINT 310
CALL PRINT (ICT, ALFA, PX, PY, JX, JY)
120 CONTINUE
CALL EXIT
4 FORMAT(12)
1 FORMAT (313, 2F10.8)
901 FORMAT (17H GENERAL PROGRAM )
902 FORMAT (21H SYMMETRICAL PROGRAM )
904 FORMAT (27H BOUNDARY STRESS FUNCTIONS )
905 FORMAT (1H0)
906 FORMAT (16H MESH SIZE IS 133X X 131Y)
907 FORMAT (5H Z = F6.3)
908 FORMAT (23H DATA CARDS THREE FOUR FIVE )
910 FORMAT (1H )
909 FORMAT (15H DPHI/DX DPHI/DY PHI )
410 FORMAT (F6.3)
501 FORMAT (F10.7, 14)
2 FORMAT (F15.2)
3 FORMAT (F15.4)
800 FORMAT (3F10.5)
901 FORMAT(151)
END
STOPPED BY OPERATOR.
JUN 21 67 10 34.4
RUN TIME 0002.5 MIN.
SUBROUTINE TO FEED IN FUNCTIONS - GENERAL

FUNCTION F(X+K)
COMMON PWX(100), PWY(100), PW(100)
COMMON C(10), A1, A8
Z$C(4)
GO TO [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], K
1 G=0
RETURN
2 G=1./31*A1/2.-1./B1*X
RETURN
3 G=0
RETURN
4 G=0.5*A1/31
RETURN
5 G=Z-2.*Z/31*X
RETURN
6 G=X*(Z-2.*Z*X/B1)
RETURN
7 G=1
RETURN
8 G=-1./31*A1/2.*1./R1*X
RETURN
9 G=X
RETURN
10 G=0.5*A1/R1
RETURN
11 G=Z-2.*Z/R1*X
RETURN
12 G=X*(Z-2.*Z/R1*X)
RETURN
END

STOPPED BY OPERATOR.
JUN 21 67      10 39.4
RUN TIME 0000.9 MIN.
FUNCTION G(X,K)
COMMON PWX(100), PWY(100), PH(100)
COMMON C(10), P1, A1
Z=C(4)
GO TO (1,2,3,4,5,6,7,8,9,10,11,12), K
1 G=0
RETURN
2 G=-1./31.*X
RETURN
3 G=0
RETURN
4 G=0.5*A1/31
RETURN
5 G=2.*Z/31.*X
RETURN
6 G=(2.*Z*A1/81)
RETURN
7 G=1
RETURN
8 G=-1./31.*A1/2.+1./31.*X
RETURN
9 G=X
RETURN
10 G=0
RETURN
11 G=0
RETURN
12 G=0
RETURN
END

STOPPED BY OPERATOR.
JUN 21 67 10 58.0
JOB TIME 0001.4 MIN.
INTEGRATION SUBROUTINE - SYMM. & GEN.

FUNCTION G(N,T,B,A,T,TOL)
COMMON PHX(100), PHPY(100), PW(100)
COMMON C(10), B1, A1
67 T8=(G3(X,K)+G(A,K))/2.
H=B-A.
H2=H/2.
T=T+T8
68 IT=IT-1
X=A+H2
SUM=0
54 SUM=SUM+G(X,K)
X=X+H
IF(X<15.4,111,53
53 T8=SU4*T8
Y1=(T9*SU4)*H/3,
DY=Y1-Y
IF(DY)55.56,57
55 DY=0
57 IF(DY<TOL)56.56,58
56 GNT=Y1
RETURN
58 Y=Y1
IF(IT)56.56,59
59 H=H2
H2=H2/2.
GO TO 58
111 GNT=1.0
RETURN
END
STOPPED BY OPERATOR.
JUN 21 67 10 42.1
HJN TIME 0000.0 4IN.
SUB. TO WRITE BOUND. STR. FUNC. ON TAPE - GENERAL

SUBROUTINE PRINT (ICT, A, PIX, PIY, JX, JY)

COMMON PHPX[100], PHPY[100], PH[100]
COMMON C[10], B[1], A[1]
REMOV 3
ALPHA=#I/91
WRITE TAPE 3, ALPHA, PIX, PIY, JY, JX
ICT=1
MAX=JY+2
NAX=JY-2
20 PRINT 502, PHPX[ICT], PHPY[ICT], PH[ICT]
ICT=ICT+1
IF(ICT=ICT)120,20,21
21 WRITE TAPE 3, (PH[ICT], ICT=1, ICT)
ICT=NAX
WRITE TAPE 3, (PHPY[ICT], ICT=1, ICT)
ICT=ICT+1
IF(ICT=ICT)120,20,21
NL
END
STOPPED BY OPERATOR,
JUN 21 67 10 41.2
HOUR TIME 0000.8 MIN.
SUB. TO WRITE BOUND. STR. FUNC. ON TAPE - SYMM.

SUBROUTINE PRINT (ICT, A, PIX, PIY, JX, JY)

V
NL1

COMMON PHPX(100), PHPY(100), PH(100)
COMMON C(10), P1, A1
REWIND 3
JX=(JX+1)/2
ALPHA=ALPHA/91
WRITE TAPE 3, ALPHA, PIX, PIY, JX, JY
IC=1
MAX=JY=2
MAX=JX=1
20 PRINT 502, PHPX(1C), PHPY(1C), PH(1C)
IC=IC+1
IF(IC=ICT)20,20,21
21 WRITE TAPE 3, PHPX(1C), PHPY(1C), PH(1C)
ID=NAX=1
WRITE TAPE 3, PHPY(1C), IC=1, ICT
ID=NAX=1
WRITE TAPE 3, PHPX(1C), IC=ID, IE
ID=NAX=1
WRITE TAPE 3, PHPY(1C), IC=ID, IE
REWIND 4
RETURN
502 FORMAT (5F12.5)
503 FORMAT (5F10.5)
END

STOPPED BY OPERATOR.
JUN 21 67 10 40.3
RUN TIME 0000.8 MIN.
2. "A" MATRIX FOR STRESS PROGRAM - GENERAL

V

COMMON U(20,20)
DIMENSION C(50)
DIMENSION PH(100)
DIMENSION PHP(100)

C APPLICATION OF FINITE DIFFERENCES TO THE OPERATOR
C DEL*DEL*[DEL*[PH]=0
C
C READ IN PLATE DIMENSIONS AND MESH DATA
READ TAPE 3, ALPHA, PIX, PIY, JY, JX
A=PIX/PIY
SL=6.
SE=7.
EI=8.
TW=2.
ON=1.
IX=1
LY=JX*1
LJ=JY*1
MAX=JX*2
MAY=JY*2
CALL ZEROS(MAX,MAX)
IC=2*MAX-2*MAX-4
ID=IC-4
C READ IN BOUNDARY STRESS FUNCTIONS
READ TAPE 3, [PH(I)], [I=1,IC]
READ TAPE 3, [PHP(I)], [I=1,ID]
WRITE TAPE 4, [PH(I)], [I=1,IC]
WRITE TAPE 4, [PHP(I)], [I=1,ID]
REWIND 3
REWIND 4
WRITE TAPE 3, MAX, MAX, ALPHA, PIX, PIY
AS=4.
AF=4.*AS
ASF=4.4.*ASF
C WRITE EQUATIONS FOR ALL INTERIOR MESH POINTS IN THE FOLLOWING
C ORDER
C START WITH TOP LEFT CORNER POINT, PROCEED COLUMN BY COLUMN DOWN
C AND TO THE RIGHT
C
C THIS FIRST GROUP OF CARDS DEALS WITH THE FIRST AND LAST
C INTERIOR POINTS ON THE PLATE
C
C TOP POINT ON THE LEFT INTERIOR COLUMN
MM=2*MAX+MAX+3
MP=MM+3
C [IX]*=2.*AS*PH*[MM+2]-PH*[MM]+PH*[MM+2])*AS*PH*[MM-1]*F*PH*[MM+1]+2,
1*PIX*3*MP*PH[1]+2,*PIY*AF*PHP[MP]
U(1,1)=SE*SE*AF*EI=AS
U(1,2)=F
U(2,1)=ASF
U(3,1)=AF
U(4,1)=ON
U(2,2)=TW*AS
WRITE TAPE 3, [(U(I,J), J=1,MAX), L=1,MAX)
IX=IX+1
CALL ZERO(MAX, NAX)
M=2
N=1
C THESE COEFFICIENTS ARE COMMON TO ALL POINTS IN THE FIRST AND
C LAST COLUMNS EXCEPT THOSE IN THE FIRST OR LAST ROW
/0 U(M+1,N)=ST*AF+EI*AS
U(M-1,N)=AF
C POINT 1 ON THE FLOW CHART
 IF(N=11120,71,72)
C THESE COEFFICIENTS ARE COMMON TO POINTS IN THE FIRST COLUMN WHICH
C ARE NOT IN THE FIRST OR LAST ROW
/1 U(M-1,2)=TW*AS
U(M+1,2)=TW*AS
U(M,3)=ON
U(M,2)=F
MM=2*LY+M
MP=MAX+2*VAX*F
C[I]=2.*AS*(PH(MM+1)+PH(MM-1)+2,PIX*PH(MP+F*PH(MM)
GO TO 73
C THESE COEFFICIENTS ARE COMMON TO ALL POINTS IN THE LAST COLUMN
/2 U(M+1,NAX+1)=TW*AS
U(M-1,NAX+1)=TW*AS
U(M, NAX-1)=F
U(M,NAX-2)=ON
MM=LY+M
MP=MAX+NAX+1-M
C[I]=2.*AS*(PH(MM+1)+PH(MM+1)+2,PIX*PH(MP+F*PH(MM)
C THIS NEXT SEQUENCE DEALS WITH COEFFICIENTS COMMON TO CORRESPONDING
C POINTS IN THE FIRST AND LAST COLUMNS. THESE COEFFICIENTS ARE
C NOT COMMON TO THE SECOND POINT, THE NEXT-TO-LAST POINT OR A POINT
C BETWEEN. THIS EACH CATEGORY IS CONSIDERED SEPARATELY,
/3 IF(M=21120,74,75
/5 IF(N=4AX=177,76,120
C SECOND POINT
/4 U(M+2,N)=AF
MM=2*VAX+MAX+3=N
C[I]=C[I]-AF*PH(MM
GO TO 78
C A POINT BETWEEN
/7 U(M+2,N)=AF
U(M+2,N)=AF
/8 WRITE TAPE 3, [(U,J,L),J=1..MAX1,L=1..NAX]
M=1
IX=1
CALL ZERO(MAX, NAX)
GO TO 70
C NEXT-TO-LAST POINT
/6 U(M+2,N)=AF
C[I]=C[I]-AF*PH(N)
WRITE TAPE 3, [(U,J,L),J=1..MAX1,L=1..NAX]
C LOOP BK ON THE FLOW CHART
IX=1
-10-
CALL ZER0(MAX, NAX)

POINT NO. 4 ON THE FLOW CHART

IF(N=1)120,79,80

COEFFICIENTS OF THE LAST POINT IN THE FIRST COLUMN

79 C(IX)*2.0*AS*PH(IX)*PH(IX-1)*PH(IX-2)*PH(IX-1)+2.0*AF*PH(IX-1)+2.0*PI*PHP(ID)

U[MX, 1]=SE*SE+AF*EI*AS
U[MX+1]=AS
U[MX+2]=AF
U[MX+1]=ASF
U[MX+3]=ON
U[MX+2]=AF

WRITE TAPE 3: [[(U[I, J]J*MAXI, J]=1, NAX]

FROM THE BEGINNING TO THIS POINT HAS BEEN THE CONSIDERATION OF ALL
INTERIOR POINTS IN THE FIRST AND LAST INTERIOR COLUMNS OF THE
PLATE EXCEPT THE FIRST AND LAST POINTS OF THE LAST COLUMN

NOW WE CONSIDER THE REMAINING COLUMNS BETWEEN
POINT 5 ON THE FLOW CHART

N=1
N=2
IXFIX*1
CALL ZER0(MAX, NAX)

81 IF(N=1)120,82,83

THESE COEFFICIENTS ARE COMMON TO THE FIRST POINTS ON ALL COLUMNS
EXCEPT THE FIRST AND LAST COLUMNS

82 U[2, N-1]=TWA+AS
U[2, N-1]=TWA+AS
U[2, N-1]=AS
U[1, N-1]=F
U[1, N-1]=F
U[1, N-1]=F
MM=2*AX*AX*3+3
MP=M-2
C[IX]=2.0*AS*PH(MM-1)*PH(MM+1)-AF*PH(MM)+2.0*PI*AF*PHP(PP)
U[1, N-1]=AF

LOOP DD GO TO 84

THESE COEFFICIENTS ARE COMMON TO THE LAST POINT ON ALL COLUMNS
EXCEPT THE FIRST AND LAST COLUMNS

83 U[MX-1, N-1]=TWA+AS
U[MX-1, N-1]=TWA+AS
U[MX-1, N-1]=AS
U[MX-1, N-1]=F
U[MX-1, N-1]=F
U[MX-1, N-1]=F
C[IX]=2.0*AS*PH(N-1)*PH(N+1)-AF*PH(N)+2.0*PI*AF*PHP(N)
U[MX-1, N-1]=AF

THIS NEXT SHORT SEQUENCE TAKES CARE OF COEFFICIENTS WHICH ARE
NOT COMMON TO POINTS ON THE SECOND COLUMN, THE NEXT- TO-LAST
COLUMN, OR THOSE BETWEEN
POINT 7 ON THE FLOW CHART

84 IF(N-2)120,85,86
IF(N-JK+3)87,88,120

SECOND COLUMN
85 MM2=LY*LY+M
C(IJK)=C(IJK)-PH(MM)
U(M,J)=ON
GO TO 89

NEXT-TO-LAST COLUMN
88 MP=LY*LY+M
C(IJK)=C(IJK)-PF(MP)
U(M,NAX-3)=ON
GO TO 89

A COLUMN BETWEEN
87 U(M,N-2)=ON
U(M,N)=ON
89 WRITE TAPE 3, [U(J,L),J=1,MAX1,L=1,NAX]
IX=IX+1
CALL ZERO(MAX,NAX)

END OF SEQUENCE

POINT 8 ON THE FLOW CHART
IF(N-MAX)>91,90,120
90 IX=M+1
M=1

POINT 10 ON THE FLOW CHART
IF(N-MAX)>91,94,120
94 MM=NAX+MAX+2
MP=MAX+NAX

LOOP HH, THIS IS CONSIDERATION OF THE TOP POINT IN THE LAST COLUMN
C(IJK)=2,ASF*PH(MM)+PH(MM-2)+PH(MM-4)+ASF*PH(MM+1)+F*PH(MM-1)+2,
1=PI*AF*PF(MP)+2,PI*PF(MP)
U(M,N)=SE+SE*AF+F+EI+AS
U(M,NAX-2)+TM*AS
U(M,NAX)=ASF
U(M,NAX-1)=F
U(M,NAX-2)=ON
U(M,NAX-1)=AF
WRITE TAPE 3, [U(J,L),J=1,MAX1,L=1,NAX]
IX=IX+1
CALL ZERO(MAX,NAX)
M=2

MAX=NAX

PROGRAM THEN GOES TO THE BLOCK OF CARDS AT THE BEGINNING TO

CONSIDER THE LAST COLUMN—LOOP 8A

GO TO 70

THESE ARE COEFFICIENTS COMMON TO ALL POINTS NOT IN EITHER THE LAST
OR FIRST COLUMN OR ROW

M=1
105 U(M,N)=SI*SI*AF+EI+AS
U[M+1,N]=TM*AS
U[M+1,N-1]=TM*AS
U[M+1,N+1]=TM*AS
U[M-1,N]=TM*AS
U[M,N]=F
U[M,N+1]=F
U[M+1,N]=ASF
U[M+1,N]=AF

POINT 11 ON THE FLOW CHART

THIS SEQUENCE FINDS COEFFICIENTS NOT COMMON TO THE SECOND COLUMN,
THE NEXT-TO-LAST COLUMN, OR THOSE BETWEEN
IF(N=2)120,95,96

SECOND COLUMN
95 U[M,4]=ON
MP=2*X+LY+M
C[I]=PH[MP]
GO TO 97
96 IF(N=VAX+1)98,99,120

A COLUMN BETWEEN
98 U[M,N-2]=ON
U[M,N+1]=ON
C[I]=0,
GO TO 97

NEXT-TO-LAST COLUMN
99 U[M,NAX-3]=ON
MM=LY-4
C[I]=PH[MM]

END OF SEQUENCE

POINT 12 ON THE FLOW CHART
COMMON TO THE SECOND ROW, THE NEXT-TO-LAST ROW OR THOSE BETWEEN
97 IF(M=2)120,100,101

SECOND ROW
100 U[M,N]=AF
MM=2*VAX+MAX+3+N
C[I]=C[I]-AF*PH[MM]
GO TO 104
101 IF(N=VAX+1)102,103,120

A ROW BETWEEN
102 U[M-2,N]=AF
U[M+2,N]=AF
104 WRITE TAPE 3, [[U(J,L),J=1,MAX],L=1,NAX]
IX=IX+1
CALL ZERO(MAX,NAX)
M=M+1

LOOP BC, POINT 13 ON THE FLOW CHART
GO TO 105

NEXT-TO-LAST ROW
103 C[I]=C[I]-AF*PH[MM]
U(MAX-3,N)=AF
WRITE TAPE 3, [[U(J,L),J=1,MAX],L=1,NAX]
IX=IX+1
CALL ZERO(MAX,NAX)
M=M+1

LOOP BD
GO TO 91

END OF SEQUENCE

THIS TERMINATES CONSIDERATION OF POINTS IN THE COLUMNS OTHER
THE THE FIRST AND LAST COLUMNS

-13-
THE LAST MESH POINT IN THE LAST COLUMN

\begin{align*}
C_{(X)} &= 2 \cdot \text{AS} \cdot (\text{PH} NAX + 1) + \text{PH} NAX - 1 + \text{PH} NAX + 3) \cdot \text{AS} \cdot \text{PH} NAX + 1 \cdot \text{PH} NAX + 1 \cdot \text{PIX} \\
U[\text{MAX} + 1, NAX - 1] &= \text{TW} \cdot \text{AS} \\
U[\text{MAX}, NAX] &= \text{SE} \cdot \text{SE} \cdot \text{AF} \cdot \text{EIF} \cdot \text{AS} \\
U[\text{MAX} + 1, NAX] &= \text{F} \\
U[\text{MAX} + 1, NAX] &= \text{AF} \\
U[\text{MAX}, NAX - 2] &= 1 \\
U[\text{MAX} - 2, NAX] &= \text{AF} \\
\text{WRITE TAPE} 3 \cdot \{U[\text{J}, \text{L}] = 1, \text{MAX} \cdot L = 1, NAX\} \\
MX &= \text{MAX} \cdot \text{NAX} \\
\text{WRITE TAPE} 3 \cdot \{C[I], I = 1, MX\} \\
\text{REWIND} \ 3 \\
\text{GO TO} \ 200 \\
120 \ I = 9999 \\
\text{PRINT} \ 320, I \\
200 \ \text{CONTINUE}
\end{align*}

C END OF PROGRAM

\begin{align*}
\text{CALL EXIT} \\
301 \ \text{FORMAT} \ [(15, 4F10.4)] \\
310 \ \text{FORMAT} \ [(10F10.4)] \\
320 \ \text{FORMAT} \ [14] \\
\text{END}
\end{align*}

STOPPED BY OPERATOR.

JUN 21 67 10 45.0

WJN TIME 0002.9 MIN.
2. "A" MATRIX FOR STRESS PROGRAM - SYMMETRICAL

V

COMMON U(20,20)
DIMENSION C(54)
DIMENSION PH(100)
DIMENSION PHP(100)

C APPLICATION OF FINITE DIFFERENCES TO THE OPERATOR

C DEL*DEL(DEL*DEL)*PH=0

REWIND 3

C READ IN PLATE DIMENSIONS AND MESH DATA

READ TAPE 3, ALPH, PIX, PIY, JY, JX

A=PI/PIX
S=6.
E=7.
T=2.
ON=1.
I=4.

CALL ZERO(MAX,NAX)
C(I)=0
I=2*NAX+2
I=10
I=10

C READ IN BOUNDARY STRESS FUNCTIONS

READ TAPE 3, (PH[I],I=2,IC)
READ TAPE 3, (PHP[I],I=2,IC)
PH[I]=0.
PHP[I]=0.
WRITE TAPE 4, (PH[I],I=1,IC)
WRITE TAPE 4, (PHP[I],I=1,IC)
REWIND 3
WRITE TAPE 3, MAX, NAX, ALPH, PIX, PIY
REWIND 4
ASP=4.
F=4.*ASP
AF=ASP
ASF=ASP

C WRITE EQUATIONS FOR ALL INTERIOR MESH POINTS IN THE FOLLOWING ORDER

C START WITH TOP LEFT CORNER POINT, PROCEED COLUMN BY COLUMN DOWN AND TO THE RIGHT

C THIS FIRST GROUP OF CARDS DEALS WITH THE FIRST AND LAST INTERIOR COLUMNS OF THE PLATE

C TOP POINT ON CENTER LINE

MM=2*NAX+MAX+3
MP=MM-3
C(I)=FO*ASP*PH(MM-2)-TW*PIY*AF*PHP(MP)-PH(MM-1)*ASP
U(I,1)=SI*SE*AF*E*ASP
U(I,2)=T*AF
U(I,1)=AF

-15-
UI3,1 = AF
UI1,3 = TW
UI2,2 = TW*TW*AS
WRITE TAPE 3, [(U(J,L),J=1,MAX),L=1,NAX)
IX = IX+1
CALL ZERO(MAX,NAX)
M = 2
N = 1
C POINT 4 IN THE FLOW CHART
C THESE COEFFICIENTS ARE COMMON TO ALL POINTS IN THE FIRST AND
C LAST COLUMNS EXCEPT THOSE IN THE FIRST OR LAST ROW
/C U(M,1,V) = ASF
/U(M-1,V) = ASF
C POINT 9
/I F(N=1120,71,72
C THESE COEFFICIENTS ARE COMMON TO POINTS IN THE FIRST COLUMN WHICH
C ARE NOT IN THE FIRST OR LAST ROW
/1 U(M,1,V) = FO*ASF
/U(M-1,V) = FO*ASF
/U(M,3) = TW
/U(M,2) = TW*TW
/C(X) = 0.
GO TO 73
C THESE COEFFICIENTS ARE COMMON TO ALL POINTS IN THE LAST COLUMN
C EXCEPT THOSE IN THE FIRST AND LAST ROWS
/2 U(M-1,NAX-1) = TW*AS
/U(M,NAX-1) = TW*AS
/U(M, NAX-1) = F*ASF
/U(M,NAX-1) = FO*ASF
/UF(N) = SE*SI*AF*EI*AS
/MM = X*Y*4
/MP = MAX*NAX*4+M
/C(X) = 2,*AS*(PH(MM+1)*PH(MM+1)-2)*X*PH(MP+F*PH(MM)
C POINT 11 ON THE FLOW CHART
C THIS NEXT SEQUENCE DEALS WITH COEFFICIENTS COMMON TO CORRESPONDING
C POINTS IN THE FIRST AND LAST COLUMNS, THESE COEFFICIENTS ARE
C NOT COMMON TO THE SECOND POINT, THE NEXT-TO-LAST POINT OR A POINT
C BETWEEN, THIS EACH CATEGORY IS CONSIDERED SEPARATELY,
/3 IF(M=1120,74,75
/5 IF(M=1177,76,120
C SECOND POINT
/4 U(M+2,V) = AF
/MM = 2*VAX*4AX+3*N
/C(X) = [C(X) - AF*PH(MM)
GO TO 78
C A POINT BETWEEN
/7 U(M+2,V) = AF
/UI*2, V) = AF
/8 WRITE TAPE 3, [(U(J,L),J=1,MAX),L=1,NAX)
/IX = IX+1
/CALL ZERO(MAX,NAX)
C THIS LOOP GOES BACK TO POINT 9 IN THE FLOW CHART
-16-
GO TO 70
C NEXT TO LAST POINT
/6 U(M=2, N)=AF
C(I)(X)=C(I)(X)=AF*PH(N) WRITE TAPE 3, [U(I, J), J=1, MAX, L=1, MAX]
I=1X+1
C LOOP BB ON THE FLOW CHART
CALL ZERO[MAX, MAX]
C END OF SEQUENCE
C POINT 12 ON THE FLOW CHART
IF(N=1)=120, 79, 80
C COEFFICIENTS OF THE LAST POINT IN THE FIRST COLUMN
/9 C(I)(X)=AF*PH(N)+TW*AF+PIY*PHP(N)
U(MAX, 1)=SI+SE*AF+E1*AS
U(MAX-1, 2)=FO*AS
U(MAX-1, 1)=FO
U(MAX, 1)=FO
U(MAX, 3)=TW
U(MAX-2, 1)=AF WRITE TAPE 3, [U(I, J), J=1, MAX, L=1, MAX]
I=1X+1
C FROM THE BEGINNING TO THIS POINT HAS BEEN THE CONSIDERATION OF ALL
C INTERIOR POINTS IN THE FIRST AND LAST INTERIOR COLUMNS OF THE
C PLATE EXCEPT THE FIRST AND LAST POINTS OF THE LAST COLUMN
C
C NOW WE CONSIDER THE REMAINING COLUMNS BETWEEN
C
C POINT 14 ON THE FLOW CHART
M=1
N=2
CALL ZERO[MAX, MAX]
61 IF(M=1)=120, 82, 83
C THESE COEFFICIENTS ARE COMMON TO THE FIRST POINTS ON ALL COLUMNS
C EXCEPT THE FIRST AND LAST COLUMNS
62 U(I, N-1)=TW*AS
U(I, N+1)=TW*AS
U(I, N)=AF
U(I+1, N)=F
U(I+1, N+1)=F
MM=2*MAX+MAX+3-N
MP=MM-2
C(I)(X)=AF*PH(N+1)+PH(N+1)-ASF+PH(M)-2*PIY*AF+PHP(M)
U(I, N)=AF
C LOOP DD
GO TO 84
C THESE COEFFICIENTS ARE COMMON TO THE LAST POINT ON ALL COLUMNS
C EXCEPT THE FIRST AND LAST COLUMNS
63 U(MAX-1, N)=TW*AS
U(MAX-1, N-1)=TW*AS
U(MAX-1, N)=ASF
U(MAX-1, N+1)=F
U(MAX-1, N+1)=F
C(I)(X)=AF*PH(N+1)+PH(N+1)-ASF+PH(N)+2*PIY*AF+PHP(N)
U(MAX-2, N)=AF
C -17 -
THIS NEXT SHORT SEQUENCE TAKES CARE OF COEFFICIENTS WHICH ARE
NOT COMMON TO POINTS ON THE SECOND COLUMN, THE NEXT-TO-LAST
COLUMN, OR THOSE BETWEEN.

POINT 16 ON THE FLOW CHART
44 IF (N-2)120,85,86
46 IF (N-NAX+1)127,88,120
C SECOND COLUMN
85 U(M),V)=SE*SF*AF*EI*AS
GO TO 99
C NEXT-TO-LAST COLUMN
M P=IX+Y-1
C [X]=CI[X]=PH[MP]
U(M),VAX-51=ON
U(M),V)=SI*SE*AF*EI*AS
A COLUMN BETWEEN
GO TO 99
87 U(M,N-2)=ON
U(M,N+1)=ON
89 WRITE TAPE 3, [U(I,J),J=1,MAX],L=1,NAX]
IX=IX+1
CALL ZERO (MAX,NAX)
C END OF SEQUENCE
C POINT 17 ON THE FLOW CHART
IF (N-NAX) 91,90,120
40 M=M+1
C POINT 19 ON THE FLOW CHART
IF (N=NAX) 91,94,120
94 MM=NAX+MAXX+2
MP=M+MAX
C LOOP HH, THIS IS CONSIDERATION OF THE TOP POINT IN THE LAST COLUMN
C [X]=2.*AS+[PH(MM)+PH(MM-2)+PH(MM-2)]-ASF*PH(MM+1)+F*PH(MM-1)+2,
[=P]=AF+P*P[MP+1]=2.*PM+P*[MP+1]
U(M),V)=SE*SE*AF*EI*AS
U(2),VAX-1=MAX
U(2),NAX)=ASF
U(1),VAX-1)=F
U(1),VAX-2)=ON
U(3),VAX)=AF
WRITE TAPE 3, [U(I,J),J=1,MAX],L=1,NAX]
IX=IX+1
CALL ZERO (MAX,NAX)
M=M+1
N=NAX
C PROGRAM THEN GOES TO THE BLOCK OF CARDS AT THE BEGINNING TO
C CONSIDER THE LAST COLUMN--LOOP BA
C GO TO 70
C THESE ARE COEFFICIENTS COMMON TO ALL POINTS NOT IN EITHER THE LAST
C OR FIRST COLUMN OR ROW
41 M=M+1
105 U(M,N)=SI*SI*AF*EI*AS
U(M+1,N)=TW*AS
U(M+1,N-1)=W*AS
U(M+1,N-1)=W*AS
U(M, N+1)=F
U(M, N+1)=F
U(M+1, N)=AS
U(M+1, N)=AS
C[I](X)=0.

POINT 20 ON THE FLOW CHART.
THE NEXT-TO-LAST COLUMN, OR THOSE BETWEEN
2120,95,96
SECOND COLUMN
95 U(M,4)=ON
U(M)=SE*SI*AF+EI*AS
GO TO 97
96 IF(M=198,99,120
A COLUMN BETWEEN
98 U(M,N-2)=ON
U(M)=ON
GO TO 97
NEXT-TO-LAST COLUMN
99 U(M,NAX-3)=ON
MM=LM+LY-M
C[I]=PH(MM)
END OF SEQUENCE
POINT 21 ON THE FLOW CHART--THIS SEQUENCE FINDS COEFFICIENTS NOT
COMMON TO THE SECOND ROW, THE NEXT-TO-LAST ROW OR THOSE BETWEEN
97 IF(M=2120,100,101
SECOND ROW
100 U(4,N)=AF
MM=2*MAX+MAX+3-N
C[I]=AF*PH(MM)*C[I]
GO TO 104
101 IF(M=MAX+1102,103,120
A ROW BETWEEN
102 U(M+2,N)=AF
U(M+2,N)=AF
104 WRITE TAPE 3, [(U(J,L),J=1,MAX),L=1,NAX]
IX=IX+1
CALL ZERO(MAX,MAX)
M=M+1
POINT 22 ON FLOW CHART, LOOP BC
GO TO 105
NEXT-TO-LAST ROW
103 C[I]=AF*PH(M)+C[I]
UMAX=2,N)=AF
WRITE TAPE 3, [(U(J,L),J=1,MAX),L=1,NAX]
IX=IX+1
CALL ZERO(MAX,MAX)
M=M+1
LOOP BD
GO TO 91
END OF SEQUENCE

THIS TERMINATES CONSIDERATION OF POINTS IN THE COLUMNS OTHER
THE THE FIRST AND LAST COLUMNS

THE LAST MESH POINT IN THE LAST COLUMN
BO C[I]=2.*AS*(PH[NAX+1]+PH[NAX+1]*PH(NAX+31)-ASF*PH(NAX)+F*PH(NAX+ 12)+2.*AF*PIY*PH(NAX)+F*PH(NAX+1)+PIY
U[NAX-1,NAX]=TM*AS
U(NAX,NAX+1)*SE*SE*AF+EI*AS
U(NAX,NAX+1)*F
U(NAX-1,NAX)*ASF
U(NAX,NAX+2)*1,
U(NAX+2,NAX)*AF
WRITE TAPE 3, ([UL[I],J=1,MAX1,L=1,NAX]
MAX=MAX+MAX
WRITE TAPE 3, ([C[I],IX=1,MX)
REWIN 3
GO TO 200
120 I=9999
PRINT 320, I
200 CONTINUE

END OF PROGRAM
CALL EXIT
301 FORMAT (215, 4F10,4)
310 FORMAT ((10F10.4))
320 FORMAT (14)
400 FORMAT (9F10.6)
401 FORMAT (F10.6)

STOPPED BY OPERATOR.
JUN 21 67 10 48.1
RUN TIME 003.0 41N.
SUBROUTINE ZERO - GEN. & SYMM.

SUBROUTINE ZERO(MAX,NAX)
COMMON U(20,20)
DO 51 M=1,MAX
DO 11 N=1, NAX
U(M,N)=0.
11 CONTINUE
51 CONTINUE
RETURN
END

STOPPED BY OPERATOR.
JUN 21 67 10:50.2
RUN TIME 0000.7 MIN.
3. SIMULTANEOUS EQUATION SOLVER - SYMM. & GEN.

V

COMMON A(256), B(50)
DIMENSION C(50), D(50)

SOLUTION OF SIMULTANEOUS EQUATIONS FOR INTERIOR STRESS FUNCTIONS

READ TAPE 3, MAX, NAX, ALPHA, PIX, PIY
MX=MAX=NAX

READ IN COEFFICIENTS INTO A ONE DIMENSIONAL ARRAY, MAKING THEM
ACCEPTABLE FOR MATRIX INVERSION SUBROUTINE. THERE ARE AS MANY SETS
OF COEFFICIENTS AS THERE ARE MESH POINTS,
DO 30 I=1,MX
IX=I-I+1
IK=MX+IJ-1
READ TAPE 3, [A(I), J=1, IK]
30 CONTINUE

INVERT COEFFICIENT MATRIX
M*3
CALL MINV(MX,A,A,D,M)
IF (M*31,12,11)

READ IN RIGHT SIDE CONSTANTS
111 READ TAPE 3, [C(I), I=1,MX]

MULTIPLICATION OF INVERTED MATRIX WITH RIGHT SIDE CONSTANTS TO GET
STRESS FUNCTIONS
20 CONTINUE
DO 12 I=1,MX
B(IM)=0.
DO 11 J=1,MX
KX=MX+(I-1)*J
B(IM)=B(IM)+C[I]*A(KX)
11 CONTINUE
12 CONTINUE
WRITE STRESS FUNCTIONS ON TAPE

WRITE TAPE 3, MAX, NAX, ALPHA, PIX, PIY
WRITE TAPE 3, [B(I), I=1,MX]

REWIND 3
PRINT 4, [B[I], I=1,MX]
GO TO 23
112 PRINT 113
23 CALL EXIT
4 FORMAT ([16F10.5])
113 FORMAT [204 MATRIX IS SINGULAR ]

STOPPED BY OPERATOR.
JUN 70 67 10 51.3
RUN TIME 4001.1 41N.

-22-
4. DETERMINATION OF STRESSES - GENERAL

V
C
PROGRAM TO FIND STRESSES FROM STRESS FUNCTIONS - PROGRAM 4
COMMON SX(15,15), SY(15,15), SXY(15,15), V(15,15), GX(15,15)
COMMON GY(15,15), GXY(15,15)
DIMENSION PH(16,16), H(164)
H=0 3
H=0 4
READ TAPE 3, MAX, NAX, ALPHA, PI, PY
A=PI/PIY
LX=NAX+1
LY=NAX+1
LX=NAX+1
JY=NAX+2

C REAL IN ALL STRESS FUNCTIONS. CHANGE SUBSCRIPTS SO THAT THE
C DERIVATIVES NEEDED TO FIND STRESSES CAN BE READILY DETERMINED.
c=0*AX=2*NAX=4
head TAPE 4, [PH[I], I=1,IC]
head TAPE 3, [[PH[I, J]], I=2,LY], J=2,LX
H=0 3
WRITE TAPE 3, MAX, NAX, ALPHA, P, PY
DO 201 I=1,LX
ID=2*Y+LY+1
PH[I, I]=PH[I, 1]
201 CONTINUE
DO 202 I=1,LY
MX=LY+1
PH[I, J]=PH[I, J]
202 CONTINUE
DO 203 I=2,LX
PM[I, I]=PM[I, 1]
203 CONTINUE
DO 204 I=1,LY
IX=LY+1
PH[I, I]=PH[I, X]
204 CONTINUE

C FIND DERIVATIVES, PRINT AND PUNCH THEM.
DO 205 N=2, LX
DO 205 M=2,LY
SY[M-1, N-1]=PH[M-2, N-1]+PH[M-1, N]+PH[M+1, N-1]
SXY[M-1, N-1]=PH[M-1, N+1]-PH[M-1, N-1]+PH[M+1, N+1]+PH[M+1, N-1]
205 CONTINUE

C CALL: VEEZ(MAX, NAX, A)
C THESE DERIVATIVES ARE NOT STRESSES BUT ARE PROPORTIONAL TO THEM

CALL VEEZ(MAX, NAX, A)
EXIT

210 FORMAT (10F12, 4)
311 FORMAT (16F10, 4)
END

STOPPED BY OPERATOR.
**4. DETERMINATION OF STRESSES - SYMMETRICAL**

```c
COMMON SX(15,15), SY(15,15), SYX(15,15), V(15,15), GX(15,15)
COMMON GY(15,15), GYX(15,15)
DIMENSION PHB(64), PH(16,16)

C PROGRAM TO FIND STRESSES FROM STRESS FUNCTIONS
READ TAPE 3, MAX, NAX, ALPHA, PIX, PIY
AS PHX/PIY

C READ IN ALL STRESS FUNCTIONS, CHANGE SUBSCRIPTS SO THAT THE
C DERIVATIVES NEEDED TO FIND STRESSES CAN BE READILY DETERMINED.
IC=2*N+2*NAX+4
READ TAPE 4, [PHB(I), I=1,IC]
LY=MAX+1
LYS+MAX+1
JY+LY+1
JX+LY+2
READ TAPE 3, [PH(I,J), I=2,LY], J=2,LY
REWIND 3
WRITE TAPE 3, [MAX, NAX, ALPHA, PIX, PIY]
DO 201 I=2,LY
ID=2*LY+1
PH(I,I)=PHB[ID]
201 CONTINUE
DO 202 I=1,JY
MX=LY+Y+1
PH(I,JX)=PHB[MX]
202 CONTINUE
DO 203 I=2,LY
PH(JY,I)=PHB[I+1]
203 CONTINUE
DO 204 I=1,JY
PH(I,I)=PI[I+1]
204 CONTINUE
C FIND DERIVATIVES AND PUNCH THEM.
DO 205 N=2,LY
DO 206 M=2,LY
SX(M-1,N-1)=PH(M,N-1)+PH(M+1,N-1)
SY(M-1,N-1)=PH(M,N-1)+PH(M+1,N-1)
SYX(M-1,N-1)=PH(M,N-1)+PH(M+1,N-1)
206 CONTINUE
205 CONTINUE
PRINT 210, [SX(I,J), I=1,LY], [J=1,NAX]
PRINT 210, [SY(I,J), I=1,LY], [J=1,NAX]
PRINT 210, [SXJ(I,J), L=1,NAX], [J=1,NAX]
CALL WRITE(MAX, NAX, A)
C THESE DERIVATIVES ARE NOT STRESSES BUT ARE PROPORTIONAL TO THEM
CALL EXIT
210 FORMAT ([9F12.4])
END

STOPPED BY OPERATOR.
JUN 21 67 10 52.4
RUN TIME 001.0 WIN.
```
SUBROUTINE FOR "V" MATRIX-GENERAL

SUBROUTINE VELL(MAX,NAX,A)

V
NL1
COMMON SX(15,15), SY(15,15), SX(15,15), V(15,15), GX(15,15)
COMMON GY(15,15), GXY(15,15)

C V COEFFICIENTS SUBROUTINE-GENERAL

C THIS SUBROUTINE DETERMINES THE FINITE DIFFERENCE EQUIVALENT OF
C SIGMAX/DX/DX/DY/DY+SIGMAY/DY/DY+TAUXY/DX/DY/DY.
C WRITE THE FINTE DIFFERENCE EQUIVALENT FOR THE ABOVE EXPRESSION
C FOR EACH INTERIOR MESH POINT STARTING AT THE POINT NEAREST TO THE
C UPPER LEFT CORNER, PROCEEDING COLUMN BY COLUMN DOWN AND TO THE
C RIGHT

C DO 802 I=1,MAX
C DO 803 J=1,NAX
C
C MULTIPLY THE VALUES BY THE APPROPRIATE CONSTANTS TO MAKE THEM FIT
C THE OPERATOR

GX[I,J]=-SX[I,J]*A*A
GY[I,J]=-SY[I,J]*A*A
GXY[I,J]=-SXY[I,J]*[-A*A/8,]

803 CONTINUE
802 CONTINUE
M=1
N=1
C LOOP NO 3 ON THE FLOW CHART--THIS COEFFICIENT IS COMMON TO ALL
C INTERIOR POINTS IN THE MESH.
804 V[M,N]=2.*[GX(M,N)+GY(M,N)]
C POINT 4 ON THE FLOW CHART
IF(N=1.II)=.805,811
C CONSIDERATION OF THE FIRST COLUMN--ONE COEFFICIENT COMMON TO
C ALL POINTS ON THE FIRST COLUMN.
805 V[M,N]=-[GX(M,N)]
C THE FOLLOWING SEQUENCE TAKES CARE OF COEFFICIENTS NOT COMMON TO
C THE FIRST POINT, THE LAST POINT, OR A POINT BETWEEN IN THIS COLUMN
IF(M=1.II)=.806,807
C FIRST POINT
806 V[M+1,N]=-[GY(P,N)]
V[M+1,N]=+GXY(M,N)
GO TO 909
807 IF(M=4AX)=.808,810,120
C A POINT BETWEEN
808 V[M+1,N]=+GXY(M,N)
V[M+1,N]=+GY(P,N)
V[M+1,N]=+GXY(M,N)
809 WRITE TAPE 3, [(V[I,J], I=1,MAX), J=1,NAX]
CALL ZEROVIMAX,NAX)
M=1
C LOOP VD
GO TO 804
C LAST POINT
V[M-1,N]=-[GXY(M,N)]
810 V[M-1,N]=-[GY(P,N)]
END OF SEQUENCE

LOOP V

M=M+1
GO TO 804

811 IF (M=MAX) 912, 818, 120

CONSIDERATION OF A COLUMN BETWEEN THE FIRST AND LAST COLUMNS.

COEFFICIENTS COMMON TO ALL POINTS OF THIS COLUMN

POINT 6 ON THE FLOW CHART

812 V(M,N+1)=GXY(M,N)

THE FOLLOWING SEQUENCE TAKES CARE OF COEFFICIENTS NOT COMMON TO

THE FIRST POINT, THE LAST POINT, OR A POINT BETWEEN IN THIS COLUMN

813 V(M,N)=GXY(M,N)

FIRST POINT

814 IF (M=1) 915, 818, 120

A POINT BETWEEN

815 V(M+1,N-1)=GXY(M,N)

816 WRITE TAPE 3, ([V(I,J), I=1,MAX], J=1, NAX)

CALL ZEROV(MAX, NAX)

M=M+1
GO TO 804

THE LAST POINT

817 V(M+1,N-1)=GXY(M,N)

818 WRITE TAPE 3, ([V(I,J), I=1,MAX], J=1, NAX)

CALL ZEROV(MAX, NAX)

M=M+1
GO TO 804

END OF SEQUENCE

LOOP V

M=M+1
GO TO 904

CONSIDERATION OF THE LAST COLUMN

819 V(M,N+1)=GXY(M,N)

THE FOLLOWING SEQUENCE TAKES CARE OF COEFFICIENTS NOT COMMON TO

THE FIRST POINT, THE LAST POINT, OR A POINT BETWEEN IN THIS COLUMN

820 V(M+1,N+1)=GXY(M,N)

FIRST POINT

-26-
GO TO 922

C A POINT BETWEEN
821 V(M+1,N)=GXY(M,N)
V(M+1,1)=GXY(M,N)
V(M+1,N-1)=GXY(M,N)
V(M+1,1)=GXY(M,N)

822 WRITE TAPE 3, (V(I,J), I=1,MAX), J=1,MAX)
CALL ZER0V(MAX,NAX)
M=M+1
GO TO 804

C LAST POINT
823 V(M-1,1)=GXY(M,N)
V(M-1,N)=GXY(M,N)
WRITE TAPE 3, (V(I,J), I=1,MAX), J=1,MAX)

C END OF SEQUENCE
C RETURN BACK TO MAIN PROGRAM AND END OF PROGRAM. THIS TERMINATES
C DETERMINATION OF ALL COEFFICIENT MATRICES FOR THE BUCKLING PROGRAM
END

STOPPED BY OPERATOR.
JUN 21 67 10 55.1
RUN TIME 0001.6 MVN.
SUBROUTINE FOR "V" COEFFICIENTS - SYMMETRICAL

SUBROUTINE VEE2(MAX,NAX,A)

V

COMMON SY(15,15), SX(15,15), VV(15,15), GX(15,15)

COMMON GY(15,15), GXY(15,15)

V COEFFICIENTS SUBROUTINE - SYMMETRICAL PROGRAM

THIS SUBROUTINE DETERMINES THE FINITE DIFFERENCE EQUIVALENT OF

SIGMA(0)*DX(0)*DY(0)*TAU*V/V

WRITE THE FINITE DIFFERENCE EQUIVALENT FOR THE ABOVE EXPRESSION

FOR EACH INTERIOR MESH POINT STARTING AT THE POINT NEAREST TO THE

UPPER LEFT CORNER, PROCEEDING COLUMN BY COLUMN DOWN AND TO THE

RIGHT.

MULTIPLY THE VALUES BY THE APPROPRIATE CONSTANTS TO MAKE THEM FIT

THE OPERATOR

UG 002 I=1,MAX

UG 003 J=1,NAX

GX[I,J]=-5X[1,J]*A*A

GY[I,J]=-SY[I,J]*A*A

GXY[I,J]=SXY[I,J]*I-1/A

803 CONTINUE

802 CONTINUE

M=1

N=1

LOOK VA

POINT NO. 3 ON THE FLOW CHART--THIS COEFFICIENT IS COMMON TO ALL

INTERIOR POINTS IN THE MESH.


POINT NO. 4 ON THE FLOW CHART

IF[M=112],805,811

CONSIDERATION OF THE FIRST COLUMN--ONE COEFFICIENT COMMON TO

ALL POINTS IN THE FIRST COLUMN

815 V[M,N]=2*G[M,N]

THE FOLLOWING SEQUENCE TAKES CARE OF COEFFICIENTS NOT COMMON TO

THE FIRST COLUMN, THE LAST POINT, OR A POINT BETWEEN IN THE FIRST

COLUMN

IF[M=112],806,807

FIRST POINT

816 V[M-1,N]=G[M,N]

GO TO 809

817 IF[N=MAX],406,810,120

A POINT BETWEEN

818 V[M-1,N]=G[M,N]

V[M-1,N]=G[M,N]

819 WRITE TAPE 3, [V[I,J],I=1,MAX],J=1,MAX]

CALL ZEROV(MAX,NAX)

M=M+1

LOOK VA

GO TO 804

LAST POINT

810 V[M-1,N]=G[M,N]

WRITE TAPE 3, [V[I,J],I=1,MAX],J=1,MAX]

CALL ZEROV(MAX,NAX)

END OF SEQUENCE

LOOK VA

-28-
1'1:1
N:N.l
GO TO
!l04
811 IF(N=M)112,818,120
C CONSIDERATION OF A COLUMN BETWEEN THE FIRST AND LAST COLUMNS.
C COEFFICIENTS COMMON TO ALL POINTS OF THIS COLUMN
C POINT 6 ON THE FLOW CHART
812 V(M,N+1)=GX[P,N]
V(M,N-1)=GXY[P,N]
C THE FOLLOWING SEQUENCE TAKES CARE OF COEFFICIENTS NOT COMMON TO
C THE FIRST POINT, THE LAST POINT, OR A POINT BETWEEN IN THIS COLUMN
C IF(N=1)1120,813,814
C FIRST POINT
813 V(M+1,N-1)=GXY[M,N]
V(M-1,N-1)=GXY[M,N]
V(M+1,N)=GY[M,N]
V(M-1,N)=GXY[M,N]
GO TO 816
814 IF(N=M)115,817,120
C A POINT BETWEEN
815 V(M+1,N-1)=GXY[M,N]
V(M-1,N-1)=GXY[M,N]
V(M+1,N)=GXY[M,N]
V(M-1,N)=GXY[M,N]
V(M-1,N-1)=GXY[M,N]
V(M-1,N-1)=GXY[M,N]
V(M-1,N)=GXY[M,N]
V(M-1,N)=GXY[M,N]
816 WRITE TAPE 3, [V[I,J],I=1,MAX,J=1,NAX] CALL ZEROV[M=1,MAX,NAX]
GO TO 804
C THE LAST POINT
817 V(M+1,N-1)=GXY[M,N]
V(M-1,N-1)=GXY[M,N]
V(M+1,N)=GXY[M,N]
V(M-1,N)=GXY[M,N]
WRITE TAPE 3, [V[I,J],I=1,MAX,J=1,NAX] CALL ZEROV[M=1,MAX,NAX]
GO TO 804
C END OF SEQUENCE
C LOOP V D
GO TO 804
C CONSIDERATION OF THE LAST COLUMN
C COEFFICIENTS COMMON TO ALL POINTS OF THIS COLUMN
C IF(N=1)1129,819,830
C FIRST POINT
819 V(M+1,N-1)=GXY[P,N]
V(M+1,N-1)=GXY[M,N]
GO TO 922
820 IF(N=M)1921,823,120
C A POINT BETWEEN
821 V(M+1,N)=GXY[P,N]
-29-
V(M-1, N) = GY(M, N)
V(M-1, N-1) = GAY(M, N)
V(M, N) = GY(M, N)

822 WRITE TAPE 3, ((V(I, J), I=1, MAX), J=1, NAX)
CALL ZEROV(NAX, NAX)

MAP + 1
GO TO 904
C LAST POINT
823 V(M-1, N-1) = GAY(M, N)
V(M-1, N) = GY(M, N)
WRITE TAPE 3, ((V(I, J), I=1, MAX), J=1, NAX)
C END OF SEQUENCE
C RETURN BACK TO MAIN PROGRAM AND END OF PROGRAM 5, THIS TERMINATES
C DETERMINATION OF ALL COEFFICIENT MATRICES FOR THE BUCKLING PROGRAM
RETURN
END
STOPPED BY OPERATOR.
JUN 21 67 10 56.4
RUN TIME 1001.4 MIN.
SUBROUTINE ZEROV(MAX,MAX)
COMMON SX(15,15), SY(15,15), SXY(15,15), VX(15,15), GX(15,15)
COMMON GY(15,15), GXY(15,15)
DO 51 M=1,MAX
DO 11 V=1,NA
   VIM,VJ=0.
11 CONTINUE
51 CONTINUE
   RETURN
END

STOPPED BY OPERATOR.
JUN 21 67  10:57:3
RUN TIME 0000,7 MIN.
5. "U" MATRIX FOR BUCKLING PROGRAM-GENERAL

V

1.

COMMON U(20,20), V(20,20)

C

THIS PROGRAM WILL DETERMINE THE FINITE DIFFERENCE EQUIVALENT OF

C

DEL*DEL*DEL*DEL(W) WHERE W IS TRANSVERSE DEFORMATION.

C

REWIND 4

READ 790, PL

REWIND 3

C

READ IN PLATE AND MESH DATA

READ TAPE 3, MAX, NAX, ALPHA, PIX, PIY

JX=NAX*2

JY=NAX*2

A=PIX/PIY

CALL ZERO (MAX, NAX)

F1=5.

S1=6.

E1=8.

T=2.

ON=1.

AS=A+A

AT=2.*AS

AF=AS*AS

F=-14.*AS

AF=AS*AS

ASF=A.*ASF

AF=AS*AS

C

WRITE EXPRESSIONS FOR THIS OPERATOR FOR ALL INTERIOR POINTS

C

CONSIDERING THE POINTS IN THE FOLLOWING ORDER

C

START WITH TOP LEFT CORNER POINT, PROCEED COLUMN BY COLUMN DOWN

C

AND TO THE RIGHT

C

THIS FIRST GROUP OF CARDS DEALS WITH THE FIRST AND LAST

C

INTERIOR COLUMNS OF THE PLATE

C

POINT NEAREST TO UPPER LEFT CORNER

U(1,1)=F1*S1*AF*E1*AS+PL*AF

U(1,2)=F

U(2,1)=ASF

U(3,1)=AF

U(1,3)=ON

U(2,2)=AT

WRITE TAPE 4, (U(i,j), I=1,MAX, J=1,NAX)

CALL ZERO (MAX, NAX)

C

POINT 4 ON THE FLOW CHART

M=2

N=1

C

COEFFICIENTS COMMON TO POINTS IN THE FIRST AND LAST COLUMNS OTHER

C

THAN THOSE IN THE TOP OR BOTTOM ROW.

750 U(M,V)=F1*S1*AF*E1*AS

U(M+1,V)=ASF

U(M+1,V)=ASF

C

POINT NO. 5 ON THE FLOW CHART

IF (Y-1100,751,752

751 U(M+1,2)=AT

C

N=1, THEREFORE THIS IS THE SECOND POINT DOWN IN THE FIRST COLUMN

U(M+2)=F

U(M+1,2)=AT

U(M,3)=ON

-32-
GO TO 753
C SECOND POINT DOWN IN LAST COLUMN
752 U(M+1,JX-1)=AT
   U(M-1,JX-3)=AT
   U(M,JX-4)=ON
   U(M,JX-3)=F
C POINT 6 ON THE FLOW CHART
753 IF(M-2)=120,754,755
C M=2
754 U(M+2,V)=AF
   GO TO 757
755 IF(N-JY)=1756,756,120
C COEFFICIENTS UNIQUE TO POINTS ALONG THE FIRST AND LAST COLUMNS FOR
C WHICH M IS GREATER THAN 2 BUT LESS THAN 2 LESS THE TOTAL NUMBER
C OF INTERIOR MESH POINTS IN THE VERTICAL DIRECTION
756 U(M+2,V)=AF
   U(M-2,V)=AF
757 WRITE TAPE 4, [(U(I,J),I=1,MAX1,J=1,NAX1]
   CALL ZERO [MAX1,NAX1]
   M=M+1
C LOOP UB
   GO TO 750
C THIS IS THE NEXT TO LAST INTERIOR POINT IN FIRST OR LAST COLUMN
758 U(M-2,V)=AF
   WRITE TAPE 4, [(U(I,J),I=1,MAX1,J=1,NAX1]
   CALL ZERO [MAX1,NAX1]
C POINT 8 ON THE FLOW CHART
759 U(MAX1,1)=FI*FI+EI*AS
   U(JY-3,2)=AF
   U(MAX1,2)=AF
   U(JY-3,1)=AF
   U(MAX1,1)=ON
   U(JY-4,1)=AF
   WRITE TAPE 4, [(U(I,J),I=1,MAX1,J=1,NAX1]
   CALL ZERO [MAX1,NAX1]
C FROM THE BEGINNING TO THIS POINT HAS BEEN THE CONSIDERATION OF ALL
C INTERIOR POINTS IN THE FIRST AND LAST INTERIOR COLUMNS OF THE
C PLATE EXCEPT THE FIRST AND LAST POINTS OF THE LAST COLUMN
C NOW WE CONSIDER THE REMAINING COLUMNS BETWEEN
C POINT 9 ON THE FLOW CHART
M=1
N=2
763 IF(N-1)=1120,764,762
C THESE COEFFICIENTS ARE COMMON TO THE FIRST POINTS ON ALL COLUMNS
C EXCEPT THE FIRST AND LAST COLUMNS
761 U(2,N-1)=AT
   U(2,N+1)=AT
   U(2,N)=AF
   U(1,N+1)=F
   U(1,N)=SI*SI*PL*AF+EI*AS
   U(3,N)=AF
THESE COEFFICIENTS ARE COMMON TO THE LAST POINT ON ALL COLUMNS
EXCEPT THE FIRST AND LAST COLUMNS

762 \( U(jy-3,n-1) = \) \( AT \)
763 \( U(jy-3,n+1) = \) \( AF \)
764 \( U(\max,n+1) = \) \( F \)
765 \( U(\max,v+1) = \) \( SI + [SIPL] + AF + EI + AS \)
766 \( U(jy-4,n) = \) \( AF \)
767 \( U(\max,v-1) = \) \( F \)

THIS NEXT SORT SEQUENCE TAKES CARE OF COEFFICIENTS WHICH ARE NOT COMMON TO POINTS ON THE SECOND COLUMN, THE NEXT-TO-LAST COLUMN, OR THOSE BETWEEN

SECOND COLUMN
768 \( U(jy-4) = ON \)
GO TO 769
769 \( U(jy-3) = ON \)
GO TO 769

NEXT-TO-LAST COLUMN
770 \( U(\max)x+1 = ON \)
771 WRITE TAPE 4, \( [U(\max), j=1, MAX] \), \( J=1, MAX \)
CALL ZERO \( [MAX, MAX] \)

POINT 12 ON THE FLOW CHART
IF \( jy=11 \)
772 \( U(jy-3) = ON \)
GO TO 774

THESE ARE COEFFICIENTS COMMON TO ALL POINTS NOT IN EITHER THE LAST COLUMN OR FIRST COLUMN OR ROW

773 \( M=jy+1 \)
774 \( U(\max,j) = SI + SI + AF + EI + AS \)
775 \( U(\max+1, n-1) = AT \)
776 \( U(\max+1, n-1) = AT \)
777 \( U(\max+1, n-1) = AT \)
778 \( U(\max+1, n-1) = AT \)
779 \( U(\max,n) = F \)
780 \( U(\max,v+1) = AF \)
781 \( U(\max-1, n) = AF \)

POINT 14 ON THE FLOW CHART
THIS SEQUENCE FINDS COEFFICIENTS NOT COMMON TO THE SECOND COLUMN, THE NEXT-TO-LAST COLUMN, OR THOSE BETWEEN
IF \( jy=11 \)
782 \( U(\max) = ON \)
GO TO 777
783 \( U(\max,j) = ON \)
GO TO 777
784 \( U(\max,j) = ON \)
GO TO 777
785 \( U(\max,j) = ON \)
GO TO 777
786 \( U(\max,j) = ON \)
GO TO 777

LOOPUG, THIS SEQUENCE FINDS COEFFICIENTS NOT COMMON TO THE SECOND ROW, THE NEXT-TO-LAST ROW, OR THOSE BETWEEN
IF \( jy=11 \)
787 \( U(\max) = ON \)
GO TO 777
7/8 U(4,4) = AF
GO TO 782
7/9 IF(J = JY + 3) GO TO 781, 780, 120
780 U(JY + 5, N) = AF
WRITE TAPE 4, [(U[I, J], I = 1, MAX), J = 1, NAX]
CALL ZERO [MAX, NAX]
C LOOP ULD
M = M + 1
GO TO 763
761 U(M - 2, V) = AF
U(M - 2, V) = AF
762 WRITE TAPE 4, [(U[I, J], I = 1, MAX), J = 1, NAX]
CALL ZERO [MAX, NAX]
C LOOP ULD
M = M + 1
GO TO 771
C THE LAST MESH POINT IN THE LAST COLUMN
763 U(MAX, NAX) = FI + AF - E1 * AS
U(JY - 3, JX - 3) = AT
U(JY - 3, NAX) = ASF
U(MAX, JX - 3) = AF
U(MAX, JX - 4) = ON
U(JY - 4, NAX) = AF
GO TO 800
764 NAX = 1
M = 1
IF(N = JY + 2) GO TO 761, 760, 120
C THE FIRST MESH POINT ON THE LAST COLUMN
765 U(M, V) = FI + E1 * AS + (S1 + PL) * AF
U(2, JX + 3) = AT
U(2, NAX) = ASF
U(1, JX - 3) = AF
U(1, JX + 1) = ON
U(3, NAX) = AF
WRITE TAPE 4, [(U[I, J], I = 1, MAX), J = 1, NAX]
CALL ZERO [MAX, NAX]
M = 2
N = JX - 2
GO TO 750
800 N = NAX
WRITE TAPE 4, [(U[I, J], I = 1, MAX), J = 1, NAX]
C THIS TERMINATES CONSIDERATION OF DEL * (DEL + DEL) OPERATOR
C WE NOW GO TO A SUBROUTINE WHICH WILL DEAL WITH THE RIGHT SIDE OF
C THE BUCKLING EQUATION
120 CONTINUE
CALL EXIT
750 FORMAT [F2.0]
END
STOPPED BY OPERATOR.
JUN 21 67, 11 02.4
RUN TIME 0002.3 MIN.
5. "U" MATRIX BUCKLING ANALYSIS - SYMMETRICAL

This program will determine the finite difference equivalent of

\[ \text{DEL} \times \text{DEL} \times [\text{DEL} \times \text{DEL}] \times \text{W} \]

where \( \text{W} \) is the transverse plate deflection.

**PROGRAME WILL DETERMINE THE FINITE DIFFERENCE EQUIVALENT OF**

\[ \text{DEL} \times \text{DEL} \times [\text{DEL} \times \text{DEL}] \times \text{W} \]

WHERE \( \text{W} \) IS THE TRANSVERSE PLATE DEFLECTION.

**HEAD IN MESH AND PLATE DATA**

**HEAD TAPE**

\[ \text{JX IS THE TOTAL NUMBER OF MESH POINTS IN THE HORIZONTAL DIRECTION} \]

**ON THE RIGHT HALF OF THE PLATE INCLUDING THE MESH POINTS ON THE**

**VERTICAL CENTER LINE AND THE RIGHT VERTICAL EDGE.**

**JY IS THE TOTAL NUMBER OF MESH POINTS IN THE VERTICAL DIRECTION**

**INCLUDING MESH POINTS ON THE TWO HORIZONTAL EDGES.**

**MAX AND NAX ARE THE NUMBERS OF MESH POINTS IN THE VERTICAL AND**

**HORIZONTAL DIRECTIONS RESPECTIVELY NOT INCLUDING POINTS ON THE**

**EDGES OF THE PLATE.**

\[ \text{JX}=\text{NAX}+1 \]
\[ \text{JY}=\text{MAX}+2 \]
\[ \text{A}=\text{PI} \times \text{PI} \]
\[ \text{CALL ZERO} \text{ (MAX, NAX)} \]
\[ \text{F1}=5, \]
\[ \text{S1}=6, \]
\[ \text{S2}=7, \]
\[ \text{E1}=8, \]
\[ \text{TW}=2, \]
\[ \text{O4}=1, \]
\[ \text{AS}=\text{A} \times \text{A} \]
\[ \text{AT}=2 \times \text{AS} \]
\[ \text{AF}=\text{AS} \times \text{AS} \]
\[ \text{FX}=[-4, \text{A} \times \text{AS}(1, \text{AS})] \]

**WRITE EXPRESSIONS FOR THIS OPERATOR FOR ALL INTERIOR POINTS**

**CONSIDERING THE POINTS IN THE FOLLOWING ORDER.**

**START WITH TOP LEFT CORNER POINT, PROCEED COLUMN BY COLUMN DOWN**

**AND TO THE RIGHT.**

**THIS FIRST GROUP OF CARDS DEALS WITH THE FIRST AND LAST**

**INTERIOR COLUMNS OF THE PLATE.**

**POINT NEAREST TO UPPER LEFT CORNER**

\[ \text{U}(1,1)=\text{S1} \times \text{S1} \times \text{AF} \times \text{E1} \times \text{AS} \times \text{F1} \times \text{AF} \]
\[ \text{U}(1,2)=\text{TW} \times \text{F1} \times \text{AF} \]
\[ \text{U}(2,1)=\text{ASF} \]
\[ \text{U}(3,1)=\text{AF} \]
\[ \text{U}(1,3)=\text{TW} \]
\[ \text{U}(2,2)=\text{TW} \times \text{AT} \]

**WRITE TAPE 4, \{ [U(1, J), J=1, MAX], JF1, NAX] \}

**CALL ZERO \text{ (MAX, NAX)}**

**POINT 4 ON THE FLOW CHART**

\[ \text{N}=2 \]

**THESE COEFFICIENTS ARE COMMON TO POINTS IN THE FIRST COLUMN**

**NOT IN THE FIRST OR LAST ROW.**

- 36 -
750 U[M,N]=SI*SI*AF+EI*AS  
U[M+1,N]=AF  
U[M-1,N]=AF  
U[M-1,N+1]=M*AT  
U[M,N+1]=M*TF  
U[M+1,N+1]=M*AT  
U[M,N+2]=TF  
U[M+1,N+2]=TF
C  
C POINT 6 ON THE FLOW CHART  
C THIS NEXT SEQUENCE CONSIDERS COEFFICIENTS NOT COMMON TO THE  
C SECOND POINT, THE NEXT-TO-LAST POINT, OR THOSE BETWEEN IN THE  
C FIRST AND LAST COLUMNS.
753 IF(M=2)754,755  
C M=2, SECOND POINT
754 U[M+2,N]=AF  
GO TO 757
755 IF(M-J)=3756,758,120
C A POINT BETWEEN
756 U[M-2,N]=AF  
U[M-1,N]=AF  
757 WRITE TAPE 4.  
U[I,J], I=1,MAX1, J=1,NAX1  
CALL ZERO IMAX,NAX1
M=1  
IF(N=11120,750,786
C THE NEXT-TO-LAST POINT  
758 U[M-2,N]=AF
C POINT 7 ON THE FLOW CHART
WRITE TAPE 4.  
U[I,J], I=1,MAX1, J=1,NAX1  
CALL ZERO IMAX,NAX1
C END OF SEQUENCE  
IF(N=11120,759,783
C THE LAST POINT IN THE FIRST COLUMN
759 U[MAX1,1]=SI*FI*AF+EI*AS  
U[JY-3,2]=M*AT  
U[MAX2,2]=M*TF  
U[JY-3,1]=ASF  
U[MAX3,3]=M*TW  
U[JY-4,1]=AF  
WRITE TAPE 4.  
U[I,J], I=1,MAX1, J=1,NAX1  
CALL ZERO IMAX,NAX1
C FROM THE BEGINNING TO THIS POINT HAS BEEN THE CONSIDERATION OF ALL  
C INTERIOR POINTS IN THE FIRST AND LAST INTERIOR COLUMNS OF THE  
C PLATE EXCEPT THE FIRST AND LAST POINTS OF THE LAST COLUMN  
C NOW WE CONSIDER THE REMAINING COLUMNS BETWEEN
C PRINT 9 ON THE FLOW CHART
M=1  
761 IF(M=11120,762
C THESE COEFFICIENTS ARE COMMON TO THE FIRST POINTS ON ALL COLUMNS  
C EXCEPT THE FIRST AND LAST COLUMNS  
762 U[1,N]=AT  
U[2,N]=AT  
U[2,N]=ASF  
U[1,N]=TF
U(1,N+1)=F
U(1,N)=S1*(S1+PL1)*AF+E1*AS
U(1,N)=AF
END IF
GO TO 764

C

1.

THESE COEFFICIENTS ARE COMMON TO THE LAST POINT ON ALL COLUMNS
2.

EXCEPT THE FIRST AND LAST COLUMNS
3.

762 U(JY-3,N+1)=AT
4.

U(JY-3,N+1)=AT
5.

U(JY-3,N)=ASF
6.

U(MAX+1,N)=AF
7.

U(MAX+1,N)=AF+EI*AS
U(JY-4,N)=AF
U(MAX+1,N)=F
U(MAX+1,N)=ASF
U(MAX+1,N)=AF
U(MAX+1,N)=ASF
763 IF(N-21120,765,766)
764 IF(N-1767,768,120)
765 U(M,N)=U(M,N)+1,
766 U(M,N)=ON
GO TO 769

C

SECOND COLUMN
767 U(M,N)=ON
768 U(M,N)=ON
GO TO 769

C

NEXT-TO-LAST COLUMN
769 WRITE TAPE 4

C

NEXT-TO-LAST COLUMN
770 WRITE TAPE 4

C

_Point 13 on the PLO Chart_ THIS SEQUENCE FINDS COEFFICIENTS NOT COMMON TO THE SECOND COLUMN.
C

THE NEXT-TO-LAST COLUMN, OR THOSE BETWEEN
IF(N-21120,773,774
771 U(M,N)=U(M,N)+1,
772 U(M,N)=ON
GO TO 777
773 IF(N-1776,775,120
774 IF(N-1776,775,120
GO TO 777

-38-
776 \( u(m, v-2) = 0 \)

C LOOP UG, THIS SEQUENCE FINDS COEFFICIENTS NOT COMMON TO THE
C SECOND ROW, THE NEXT-TO-LAST ROW, OR THOSE BETWEEN
C SECOND ROW
777 IF \( m = 2 \), \( 778, 779 \)
C NEXT-TO-LAST ROW
779 IF \( m = 7 \), \( 780, 781, 782 \)
C WRITE TAPE 4, \[ [u[i, j], i = 1, \text{MAXI}, j = 1, \text{NAX}] \]
C CALL ZERO \[ \text{MAX}, \text{NAX} \]
C LOOP UXD
C M = M + 1
C GO TO 763
C A ROW BETWEEN
781 U(M-2, V) = AF
782 WRITE TAPE 4, \[ [u[i, j], i = 1, \text{MAXI}, j = 1, \text{NAX}] \]
C CALL ZERO \[ \text{MAX}, \text{NAX} \]
C LOOP UXD
C M = M + 1
C GO TO 771
C END OF SEQUENCE
C THE LAST MESH POINT IN THE LAST COLUMN
783 \[ u(\text{MAX}, \text{NAX}) = AF * \text{SI} * \text{AF} * \text{EI} \]
784 \( n = n + 1 \)
C \( M = 1 \)
C \( \text{IF} (n = \text{NAX}) \), \( 785, 786 \)
C THE FIRST MESH POINT ON THE LAST COLUMN
785 \[ u(2, \text{NAX}) = AF * \text{SI} * \text{AF} * \text{EI} * \text{AS} \]
786 \[ n = n + 1 \]
C \( M = 2 \)
C \( \text{GO TO} 786 \)
C THESE COEFFICIENTS ARE COMMON TO POINTS IN THE LAST COLUMN
C NOT IN THE FIRST OR LAST ROW.
C 786 \[ u(m, v) = AF * SI * AF * EI * AS \]
C \( u(1, v) = ASF \)
C \( u(m-1, v) = ASF \)
C \( u(m, \text{NAX}) = AT \)
C \( u(1, \text{NAX}) = AT \)
U(I, NAX-7)=ON
U(I, NAX-1)=F
C  LOOP UA
   GO TO 753
   BLD IN = MAX + NAX
   WHITE TAPE (I) [I(I)]: MAX(I) = L(I) NAX
E  THIS TERMINATED CONSIDERATION OF DEL:DEL:DEL:DEL(I)W OPERATOR
120 CONTINUE
   CALL EXIT
   740 FORMAT (F2.0)
END
STOPPED BY OPERATOR.
JUN 21 67 11 00 0
RUN TIME 0002.5 MIN.
6. MATRIX INVERSE - SYMM. & GEN.

COMMON A(2500)

C U MATRIX INVERSION--PROGRAM 7

REWILO 3
READ TAPE 3, MAX, NAX
MX=MAX+NAX
REWILO 4
M=55
DO 850 I=1,MX
IJ=MX*(I-1)+1
IK=IJ+MX-1
READ TAPE 4, [A(IJ),J=IJ,IK]

850 CONTINUE
CALL MINV(MX,A,M)
IF (M) 15, 10, 15
REWILO 4
DO 851 I=1,MX
IJ=MX*(I-1)+1
IK=IJ+MX-1
WRITE TAPE 4, [A(IJ),J=IJ,IK]
851 CONTINUE
GO TO 22
10 PRINT 21
22 CALL EXIT
21 FORMAT (20H MATRIX IS SINGULAR)
END

STOPPED BY OPERATOR.
JUN 21 67 11 03.3
RUN TIME 0000.8 MIN.
7. MATRIX MULTIPLICATION - GEN. & SYMM.

COMMON V(50,50), W(54)
DIMENSION U(541, W(54)
REWIND 3
HEAD TAPE 3, MAX, MAX, ALPHA, PIX, PIY
N=MAX*MAX
REWIND 4
HEAD TAPE 3, [V(I,J), J=1:N], I=1:N
REWIND 3
WRITE TAPE 3, MAX, MAX, ALPHA, PIX, PIY
DO 920 I=1, N
DO 91 J=1, N
w(l)=0.
91 CONTINUE
READ TAPE 4, [U(K), K=1,N]
DO 921 J=1, N
DO 922 K=1, N
w(J)=w(J)+U(K)*V(K,J)
922 CONTINUE
921 CONTINUE
WRITE TAPE 3, [W(J), J=1,N]
920 CONTINUE
CALL EXIT
END
STOPPED BY OPERATOR.
JUN 21 67 11 04.4
RUN TIME 001.0 4 IN.
COMMON V(50,50), CA(50), DB, DD, DU(50), IT

READ TAPE 3, MAX, NAX, A, PIX, PIY
N=MAX=MAX
T=MAX+1
B=PI*PIY
READ 400, NI, TOL
IT=0

READ IN MATRIX FOR WHICH EIGENVALUE IS TO BE FOUND
READ TAPE 3, [(V(I,J), J=1,N), I=1,N]

ASSUME INITIAL EIGENVECTOR
DO 100 J=1,N
CA(J)=1
100 CONTINUE

ADD CONVERGENCE SPEEDING NUMBER TO THE DIAGONAL OF THE MATRIX
READ 311, BB
DO 20 I=1,N
J=I
V(I,J)=V(I,J)+BB
20 CONTINUE

EIGENVALUE SUBROUTINE
CALL EIGEN (N, NI, TOL)

BC=B+3*(3.1415927**3.1415927*(DR-BH))

PRINTING OF THE SIGNIFICANT FINAL ANSWERS
PRINT 201
PRINT 204
PRINT 302, BC, DD
PRINT 205
PRINT 205
PRINT 304, A
PRINT 205
PRINT 312, IT
PRINT 205
PRINT 205
PRINT 303
PRINT 704
DO 11 I=1,N
PRINT 310, CA(I), CD(I)
11 CONTINUE

CALL EXIT

201 FORMAT (22W BUCKLING COEFFICIENT )
302 FORMAT (4H K = F10.7,14H ERROR IS F10.6)
303 FORMAT (25H EIGENVECTOR EIGENVALUE )
304 FORMAT (17H ASPECT RATIO IS F10.6)
204 FORMAT (14H)
205 FORMAT (14H)
310 FORMAT (2F12.7)
312 FORMAT (25H NUMBER OF ITERATIONS IS I3)
400 FORMAT (13, F7.6)
311 FORMAT (f10.9)

END

STOPPED BY OPERATOR,
JUN 21 67 11 05.7
RUN TIME 0001.2 WIN.

43
**EIGENVALUE SUBROUTINE - GEN. & SYMM.**

SUBROUTINE EIGEN(NI, N)

COMMON V(50,50), CA(50), DB, DN, CO(50), CT

DIMENSION CB(50)

1 920 SMX=*D

C FROM THIS POINT TO THE STATEMENT LABELLED 931 IS MULTIPLICATION OF
C MATRIX V AND THE EIGENVECTOR, AND DETERMINATION OF LARGEST
C ELEMENT OF THE RESULTING MATRIX, CB.

C MULTIPLICATION, EACH TIME THROUGH, ONE ELEMENT OF CB IS FOUND.

DO 931 J=1,N

CT(I)=0

DO 930 J=1,N

CB(I)=CB(I)+V(I,J)*CA(J)

930 CONTINUE

C SEQUENCE TO DETERMINE IF THIS ELEMENT IS THE LARGEST OF THE
C ONES ALREADY OBTAINED.

IF(CB(I))933,935,934

933 CC=SMX

GO TO 960

934 CC=SMX

935 IF(CC- SMX))933,935,936

936 SMX=CC

DB=CB(I)

935 CONTINUE

931 CONTINUE

C DIVISION OF EACH ELEMENT OF CB BY THE CORRESPONDING ELEMENT
C OF THE OLD EIGENVECTOR CA. THIS GIVES THE MATRIX CD.

C AFTER MANY RUNS THROUGH, THE CD MATRIX SHOULD CONVERGE SUCH THAT
C ALL OF ITS ELEMENTS EQUAL THE DESIRED EIGENVALUE.

IS=0

937 DO 938 J=1,N

CD(I)=CB(I)/CA(J)

938 CONTINUE

CD(I)=CD(I)/SMX

937 CONTINUE

C DETERMINATION OF THE LARGEST AND SMALLEST ELEMENT IN CD

C DETERMINATION OF THE RELATIVE ERROR IN THIS EIGENVALUE

C DETERMINATION.

IF(154A40,939,940

939 OMNX=CD(I)

OMX=OMX

IS=4

940 IF(CD(I)=M)942,942,941

941 OMX=CD(I)

942 IF(CD(I)=MN))943,944,944

943 OMX=CD(I)

944 CONTINUE

948 CONTINUE

IF (M945,945,946

945 DD=SMX

GO TO 961

946 DD=DD

946 IF (DD-MX947,948

947 DD=(OMX-MX)/DD

GO TO 952
948 IF (QX) = 949, 950, 950
949 QM = 34X
950 GO TO 962
950 QM = 04X
962 DQ = (29X - QMN) / QM

C DETERMINATION OF NEW EIGENVECTOR
952 DO 951 J = 1, N
951 CA(J) = CB(J)/DR

C DECISION AS TO WHETHER WE NEED MORE TRIALS OR IF WE SHOULD STOP

11 IF (IT = 1)
11 IF (IT = 6) 954, 954, 955
11 IF (SEONSE SWITCH 11954, 955
955 IF (IT = 11) 956, 957, 957
956 GO TO 929
957 IZ = 999
958 PRINT 10, IZ
954 CONTINUE
955 RETURN
END

STOPPED BY OPERATOR.
JUN 21 67 11 08.4
RUN TIME 0002.6 MIN.