DIGITAL COMPUTER PROGRAMS FOR
GENERATING OBLIQUE ORTHOGRAPHIC
PROJECTIONS AND CONTOUR PLOTS

by Gary L. Giles
Langley Research Center
Hampton, Va. 23665
**Title and Subtitle**
DIGITAL COMPUTER PROGRAMS FOR GENERATING OBLIQUE ORTHOGRAPHIC PROJECTIONS AND CONTOUR PLOTS

**Abstract**
User and programmer documentation is presented for two programs for automatic plotting of digital data. One of the programs generates oblique orthographic projections of three-dimensional numerical models and the other program generates contour plots of data distributed in an arbitrary planar region. The user documentation gives a general description of the computational algorithms, user instructions, and complete listings of the programs. Several plots are included to illustrate various program options, and a single example is described in detail to facilitate learning the use of the programs.

**Key Words (Suggested by Author(s))**
- Computer graphics
- Contour plots
- Orthographic projections

**Distribution Statement**
Unclassified - Unlimited

**For sale by the National Technical Information Service, Springfield, Virginia 22151**
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td>1</td>
</tr>
<tr>
<td>GENERAL PROGRAM FEATURES</td>
<td>3</td>
</tr>
<tr>
<td>OBLIQUE ORTHOGRAPHIC PROJECTIONS</td>
<td>4</td>
</tr>
<tr>
<td>Specification of View</td>
<td>4</td>
</tr>
<tr>
<td>Exploded Plots</td>
<td>6</td>
</tr>
<tr>
<td>Sectioning or Cutting Plan</td>
<td>7</td>
</tr>
<tr>
<td>CONTOUR PLOTS</td>
<td>7</td>
</tr>
<tr>
<td>General Description</td>
<td>7</td>
</tr>
<tr>
<td>Definition of Contour Surface</td>
<td>7</td>
</tr>
<tr>
<td>Constructing Contour Lines</td>
<td>9</td>
</tr>
<tr>
<td>DESCRIPTION OF COMPUTER PROGRAMS</td>
<td>11</td>
</tr>
<tr>
<td>Computer Hardware and System Requirements</td>
<td>11</td>
</tr>
<tr>
<td>Storage Allocation</td>
<td>12</td>
</tr>
<tr>
<td>Programs and Subroutines</td>
<td>12</td>
</tr>
<tr>
<td>COMPUTER PROGRAM USE</td>
<td>13</td>
</tr>
<tr>
<td>General Setup of Input Deck</td>
<td>13</td>
</tr>
<tr>
<td>Input for Oblique Orthographic Projections</td>
<td>14</td>
</tr>
<tr>
<td>Input for Contour Plots</td>
<td>19</td>
</tr>
<tr>
<td>EXAMPLES OF DATA PREPARATION</td>
<td>25</td>
</tr>
<tr>
<td>Oblique Orthographic Projections</td>
<td>25</td>
</tr>
<tr>
<td>Contour Plot</td>
<td>26</td>
</tr>
<tr>
<td>CONCLUDING REMARKS</td>
<td>27</td>
</tr>
<tr>
<td>APPENDIX A – INCENTER OF A GENERAL TRIANGLE</td>
<td>28</td>
</tr>
<tr>
<td>APPENDIX B – LISTING OF COMPUTER PROGRAM FOR OBLIQUE ORTHOGRAPHIC PROJECTIONS</td>
<td>30</td>
</tr>
<tr>
<td>APPENDIX C – LISTING OF COMPUTER PROGRAM FOR CONTOUR PLOTS</td>
<td>68</td>
</tr>
<tr>
<td>APPENDIX D – INPUT DATA FOR OBLIQUE ORTHOGRAPHIC PROJECTION OF EXAMPLE PROBLEM</td>
<td>115</td>
</tr>
</tbody>
</table>
DIGITAL COMPUTER PROGRAMS FOR GENERATING
OBLIQUE ORTHOGRAPHIC PROJECTIONS
AND CONTOUR PLOTS

By Gary L. Giles
Langley Research Center

SUMMARY

User and programmer documentation is presented for two programs for automatic plotting of digital data. One of the programs generates oblique orthographic projections of three-dimensional numerical models and the other program generates contour plots of data distributed in an arbitrary planar region. The user documentation gives a general description of the computational algorithms, user instructions, and complete listings of the programs. Several plots are included to illustrate various program options, and a single example is described in detail to facilitate learning the use of the programs.

INTRODUCTION

Studies based on mathematical models of structural configurations using a digital computer yield large amounts of output data which must be comprehensively analyzed. Graphical presentation of this data is often needed for effective evaluation by an analyst. The present paper contains a description of computational algorithms as well as user and programmer documentation for two computer programs which generate graphical displays of data. These programs generate (1) oblique orthographic projections of three-dimensional models and (2) contour plots of data distributed in an arbitrary planar region. The programs were developed initially for the graphical presentation of cumbersome output data resulting from structural analyses by finite-element methods. However, the programs have been extended for display of a general class of data. Several plots are included in this paper to illustrate various program options, and a single example is described in detail to facilitate learning the use of the programs.

SYMBOLS

\( \bar{A}, A_i, A_j, A_k \) areas of triangles shown in figures 7 and 11

\( A_\theta, A_\phi, A_\psi \) transformation matrices for rotation about Y-, X-, and Z-axis, respectively (see eqs. (2) to (4))
a, b, c  length of sides of triangles shown in figure 11

h  radius of circle inscribed in triangle

$L_1, L_j, L_k$  triangular area coordinates defined in equation (8)

$l_i$  distance between a particular grid point and centroid of ith surrounding element

M  number of grid points contained by a finite element

N  number of elements which connect at a particular grid point (eq. (7))

s  perimeter of a triangle

w  control variable represented by contour plots

$w_c$  constant value of a desired contour line

$X, Y, Z$  coordinate system fixed in model

$X_0, Y_0, Z_0$  coordinate system containing viewing planes

$x, y, z$  coordinates of a point in $X, Y, Z$ coordinate system

$\bar{x}, \bar{y}, \bar{z}$  coordinates of element centroid

$x_0, y_0, z_0$  coordinates of a point in $X_0, Y_0, Z_0$ coordinate system

$x_p, y_p, z_p$  coordinates of generic point p

$\beta_i$  weighting function to interpolate data values from ith element centroid to a particular grid point

$\theta$  Euler angle rotation about Y-axis, performed second

$\phi$  Euler angle rotation about X-axis, performed third

$\psi$  Euler angle rotation about Z-axis, performed first

2
GENERAL PROGRAM FEATURES

Programs to generate graphical displays of large amounts of output data are an important aspect of any analysis system. Many of the programs which have been developed for this purpose are reviewed in references 1 and 2. The two programs which are described in detail in this paper are distinguished by provision for generality and ease of use, as well as numerous display options, computational speed, and efficient use of core storage. These programs were developed initially for the graphical presentation of cumbersome output data resulting from structural analyses by finite-element methods. However, the programs have been extended for display of a general class of data. A general discussion of the programs is given in reference 3 and their incorporation into a computer-aided design system is discussed in reference 4.

In structural analysis, finite-element or finite-difference methods are capable of providing displacements, stresses, and vibration or buckling mode shapes at many discrete points on a structural model. Manual reduction of the data is a time-consuming task which sometimes necessitates reducing only that data associated with particular regions of the overall model considered to be critical. Errors in the analyst's judgment can prevent the detection of other critical regions. Graphical display of analytical information by oblique orthographic projections and contour plots over large portions of the model, on the other hand, allow the analyst to assimilate accurately and evaluate the data with a consequent reduction in time and manpower cost. For computer-aided design programs as described in references 4 and 5, verification of data between steps in the design process is often needed. Visual display of oblique orthographic projections or contour plots by means of a cathode-ray tube (CRT) allows timely assessment of design information needed to make intelligent changes in the design variables during program execution.

Oblique orthographic projections are useful in checking input data describing a numerical model. The topology of a finite-element model for structural analysis is described in a user-prepared input deck which includes a set of cards containing grid point identification numbers and corresponding spatial coordinates \((x,y,z)\) and a set of cards containing grid point connections for rod, beam, and plate elements. The program described herein includes options for plots of numerical models annotated with grid point or element numbers. One option of the program allows boundaries of an isolated portion of the model to be specified by cutting planes, and detailed inspection can be made of that selected region. Also, exploded views can be generated which separate the elements in a model to provide clarity in detecting the absence or presence of model elements. Structural deformations calculated from a particular analysis can be superimposed on the grid point coordinates of the undeformed structure. These displacements can also be represented as vectors extending from the grid points. The program accommodates
Another method of displaying output data is in the form of a contour plot. A computer program is described herein that generates contour plots of data relative to a planar mathematical reference surface which can have irregular boundaries or internal cutouts. The program input data require a set of grid points and connecting triangular and/or quadrilateral elements defining a planar region as well as the data to be represented by contour lines which can be specified at either the grid points or centroids of the connecting elements. The contour lines are labeled to provide quantitative information in a graphical form. For the most general case, the grid points and finite elements of a model are located by three-dimensional coordinates. As developed for the plotting routine herein, the reference surface is merely a planar projection of the three-dimensional surface. This arrangement is adequate for nearly flat surfaces; however, the presentation of contour plots on the two-dimensional projection of a three-dimensional surface having a large curvature may be difficult to interpret. In this case, a two-dimensional planar surface must be developed from the three-dimensional surface by the user outside the program.

The input decks for the two programs are similar and thus allow for the programs to be used as a set. Both programs contain options for selecting various plotting equipment including CalComp, Varian, and CRT displays. Finally, this paper includes complete listings of the programs which contain comment cards to aid the user.

OBLIQUE ORTHOGRAPHIC PROJECTIONS

Specification of View

A three-dimensional analytical model consists of a user-prepared set of grid points with given spatial coordinates (x,y,z) and a set of elements (e.g., rod, beam, triangular or quadrilateral elements) connected at the grid points. An example of an oblique orthographic projection of a finite-element structural model of an airplane is shown in figure 1. This model was used in preliminary design studies focused on the wing structure. The fuselage is represented by a structural model having a simplified rectangular geometric cross section to reduce model complexity, but the model does provide proper overall stiffness and mass distributions to represent symmetric behavior of the entire airplane. Oblique orthotropic projections allow a model to be viewed in any selected orientation. Euler angle transformations are used to specify orientation of the model to be projected. This transformation resolves the coordinate system of the model to a principal plane (i.e., viewing plane) on which the display is to be plotted. The model coordinate system is coincident with the coordinate system containing the viewing plane when all the rotation angles are zero.
Various approaches can be taken in formulating the Euler angle transformations. Herein, the viewing planes are fixed in space and the model rotated about its body axes. In this approach the principal plane is the plane normal to the user's line of vision. The model can then be rotated in space until the Euler angles giving the desired view are determined. The rotations \((\phi, \theta, \psi)\) of the body about the model axes \((X,Y,Z)\) for this approach are shown in figure 2. The order of the Euler angle rotations is taken as \(\psi, \theta, \phi\) in the program and transform coordinates of a point on the model \((x,y,z)\) to viewing plane coordinates \((x_0,y_0,z_0)\) as

\[
\begin{bmatrix}
x
y
z
\end{bmatrix} = \begin{bmatrix}
A_\psi
A_\theta
A_\phi
\end{bmatrix}
\begin{bmatrix}
x
y
z
\end{bmatrix}
\]

(1)

where

\[
A_\psi = \begin{bmatrix}
\cos \psi & -\sin \psi & 0 \\
\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(2)

\[
A_\theta = \begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}
\]

(3)

\[
A_\phi = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi & \cos \phi
\end{bmatrix}
\]

(4)

These transformations are equivalent to those used in reference 6. The option to specify which of the principal planes is to be used as the viewing plane determines which two rows of equation (1) are used. The model can be projected on either the \(X_0-Y_0\), \(X_0-Z_0\), or \(Y_0-Z_0\) planes and is shown projected on the \(X_0-Z_0\) plane in figure 2.
In addition to plots of the undeformed structure, oblique orthographic projections can be used to display the deformed structure by adding given displacements to the coordinates of the grid points before transformation and subsequent plotting of elements. An option is also available to plot the undeformed model and represent displacements by vectors extending from the grid points in the undeformed structure to the location of the grid points in the deformed structure.

Exploded Plots

Oblique orthographic projections may be used to check the topology of an analytical model. Grid points with erroneous coordinates are easily detected. Often, however, absence or presence of elements in the model cannot be determined from a conventional oblique orthographic projection. For example, a line element coincident with an edge of a triangular or quadrilateral element could not be detected. To show clearly each element, an algorithm for generating exploded oblique orthographic projections was developed. Such a plot is shown in figure 3 for a selected portion of the model shown in figure 1.

The algorithm for generating exploded oblique orthographic projections follows:

For each element in the model —

(a) Calculate the centroid of the element by using

\[
\bar{x} = \frac{\sum_{i=1}^{M} x_i}{M}, \quad \bar{y} = \frac{\sum_{i=1}^{M} y_i}{M}, \quad \text{and} \quad \bar{z} = \frac{\sum_{i=1}^{M} z_i}{M}
\]

where \(M\) is the number of grid points contained by a finite element

(b) Multiply coordinates of the element by a reduction factor to give the location of the reduced element centroid and vertices

(c) Translate the reduced element until its centroid matches the location of the centroid of the element before reduction.

Although this procedure provides exploded plots, the resulting sides of reduced triangular elements generally are not equidistant from the sides of the original element. To correct this behavior, the incenter (center of a circle inscribed in a triangle) is used for the reference point of the triangles during translation. The equations for the incenter of a general triangle are given in appendix A.
Sectioning or Cutting Planes

It is often desirable when checking input for a model to be able to isolate a portion of the model for detailed examination. The program for generating oblique orthographic projections described herein contains the option to specify sectioning or cutting planes to isolate such a portion. For example, in the case of a finite-element model of an aircraft structure, a particular fuselage frame could be generated by placing cutting planes immediately in front of and behind the frame. Annotation of grid point and element numbers show clearly on such a plot but might be obscured on a plot of the entire model. This sectioning option was used to select the structural members of a fuselage frame-wing spar cross section shown in figure 4 from the complete model shown in figure 1.

CONTOUR PLOTS

General Description

A contour plot is a graphic representation of a scalar variable as function of two independent variables, \( w = f(x,y) \). Here \( w \) is a quantity such as deformation, pressure, or stress which is distributed over some specified area of the X-Y plane. The set of values of \( w \) are herein referred to as control variables and have been measured or calculated at given locations called control points on a reference or datum surface. The locus of lines of constant values of \( w \) (contour lines) over the reference surface can be determined from the given values of \( w \) at the discrete control points. This section describes a method for generating contour plots which is particularly adaptable for displaying the results of a structural analysis by finite-element techniques. The data to be represented by contour plots can be referenced to either the grid points or to the element centroids. These alternatives are particularly convenient since finite-element analyses generally evaluate displacements at grid points and stresses at the element centroids. A contour plot of a selected vibration mode shape for the model presented in figure 1 is shown in figure 5(a). Oblique orthotropic projections of the same mode shape are also included in figures 5(b) and 5(c).

Definition of Contour Surface

The data to be represented by contour lines can be visualized as a set of points located perpendicular to and at a distance equal to the magnitude of the data value from each control point. A surface which passes through this set of data points will be referred to as a contour surface as shown in figures 5(b) and 5(c). Contour lines (lines connecting constant values of \( w \)) are projections onto the datum surface of intersections of regularly spaced planes, parallel to the datum surface, with the contour surface. The regular spacing of the planes, hence contour interval, occurs in units of 10 in figure 5(a).
The primary difference between various contour plotting programs is the method of representing the contour surface. Options for this representation include least-squares fitting of the data with polynomial surfaces and surface splines (ref. 7). Such representations result in smooth contour lines, however, considerable computational time is often required for the surface fitting, and difficulties are sometimes encountered with irregular boundaries or cutouts. Herein, the contour surface is idealized as a series of triangular planes with their vertices connected at points where the control variables are located. The control variables may be located at either the grid points or element centroids of the model. A computationally efficient algorithm which uses linear interpolation is used to determine the location of contour lines on each of the triangular planes.

Prior to generating contour lines, a triangular mesh must be generated over the datum surface and the magnitude of the control variable at each of the vertices in the mesh determined. The triangular and/or quadrilateral elements in the model are used to triangulate the datum surface. When control variables are given at the grid points, the quadrilateral elements are divided into two triangles. Such a triangular mesh is shown in figure 6(a).

Control variables located at the centroid of the elements, on the other hand, require a different method of triangulating the datum surface since the control points (element centroids) do not coincide with the grid points. For this case, a triangular mesh with control points at both the centroid of the two-dimensional elements and the grid points of the structure is generated. The centroid of each element can be readily calculated from the data describing its corner points. Since the control variables are given only at the centroidal control points, a suitable method is required to interpolate this data to each of the grid control points. The following steps describe the method of interpolation used in this algorithm and is illustrated in figure 6(b):

(a) Determine the elements that are connected to a particular grid point \( G \) (five elements are connected to the grid point \( G \) in fig. 6(b))

(b) For each element connected to the grid point, calculate the centroid which is the location of the given control values

(c) Calculate a weighted value of the contour value at the grid point \( W_G \) based on the contour values at the surrounding element centroids as follows:

\[
W_G = \sum_{i=1}^{N} w_i \beta_i
\]

where \( w_i \) is the given data value at the centroid of the \( i \)th element, \( \beta_i \) is a weighting function for that value, and \( N \) is the total number of elements con-
ected to grid point G (N = 5 in fig. 6(b)). The weighting function selected for use herein is

\[ \beta_1 = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{l_j} \]  

(7)

where \( l_j \) is the distance between the grid point G and the centroid of the ith element surrounding the grid point. (See fig. 6(b).)

Constructing Contour Lines

The construction of contour lines which are projections onto the datum surface of the intersections of planes parallel to the datum surface and the contour surface is described in this section. For convenience of discussion, the datum plane will be assumed to lie in the X-Y plane although it could lie in another plane defined by two independent variables. The contour lines are generated by considering one triangular element of the contour surface mesh at a time. Area coordinates of a triangular plane, as described in reference 8, are used to give simple expressions for the location of a point on such a plane.

The area coordinates \( L_i, L_j, \) and \( L_k \) of a point \( p \) on the triangle shown in figure 7 are given by

\[
\begin{align*}
L_i &= \frac{A_i}{A} \\
L_j &= \frac{A_j}{A} \\
L_k &= \frac{A_k}{A}
\end{align*}
\]

(8)

and

\[ L_i + L_j + L_k = 1 \]

(9)

The coordinates of a generic point \( p \) on the triangular plane (one facet of the contour surface) with vertices \( (x_i, y_i, w_i), (x_j, y_j, w_j), \) and \( (x_k, y_k, w_k) \) can be expressed as
where \( x \) and \( y \) are measured in the reference surface and \( w \) is the magnitude of the control value measured perpendicular to the reference surface.

For a contour line, \( w \) is equal to \( w_c \) where \( w_c \) is the constant value of a desired contour line. Procedures contained in the computer program require that the data be scaled so that the contour lines have positive or negative integer values. The number of contour lines, if any, and their associated magnitudes that are contained in a particular triangular plane are determined from the minimum and maximum values at the vertices and the prescribed contour interval. For those triangles which contain contour lines, the intersection of a particular contour line \( w_c \) with an edge of the triangular plane can be determined through the use of equation (12) and the definition of area coordinates given in equations (8). For example, along the edge \( i-j \), \( L_k \) is equal to zero and \( L_j \) equals \( 1 - L_i \) from equation (9). Solving equation (12) for \( L_i \) along the edge \( i-j \) yields

\[
L_i = \frac{w_c - w_j}{w_i - w_j}
\]  

Similarly along the edge \( j-k \),

\[
L_i = 0 \quad \quad L_k = 1 - L_j
\]

Thus,

\[
L_j = \frac{w_c - w_k}{w_j - w_k}
\]  

Along the edge \( k-i \),

\[
L_j = 0 \quad \quad L_k = 1 - L_i
\]

which yields

\[
L_k = \frac{w_c - w_i}{w_k - w_i}
\]
Two of the intersections given by equations (13) to (15) of the contour line and triangle edges will be between the vertices of the triangle and the third will be outside the triangle. The point outside the triangle is indicated by values of area coordinates which are less than zero or greater than unity and this point is disregarded. The x-y coordinates at the intersection of the contour line and an edge are solved from equations (10) and (11); for example, the Cartesian coordinates at an intersection along the edge $i-j$ are given as

$$x = x_i + (x_j - x_i)L_j \quad \text{(16)}$$

$$y = y_i + (y_j - y_i)L_j \quad \text{(17)}$$

The end points (x-y coordinates at intersections) of each contour line within the triangular facet are then connected by straight lines. This procedure is repeated for each triangular facet over the contour surface. Segments of the contour lines on adjacent elements meet at the common element sides and result in all contour lines being connected properly over the entire surface.

The plot is completed by drawing the boundary of the surface and labeling the magnitude of the contour lines. Surfaces with irregular boundaries or cutouts are allowed. The boundary lines are determined by summing the corner angles of each triangular facet connected to a common grid point. If the total angle is less than $360^\circ$ (point not completely surrounded by elements), the point is obviously on a boundary. All elements in the datum plane are then tested and adjacent vertices which are boundary points are connected to form the complete boundary. Each contour line which intersects the boundary is labeled with its corresponding magnitude. An option is also available to label the contour plot at specified locations within the boundary of the contour plot. This option is necessary to allow labeling of contour lines which do not intersect the boundary of the datum surface.

**DESCRIPTION OF COMPUTER PROGRAMS**

Computer Hardware and System Requirements

The plotting programs described herein are written in FORTRAN IV for the CDC 6000 series computers. The programs are operational on the CDC SCOPE 3.2 system (NASA Langley Research Center version) and make use of subroutines from its graphic output system. This system includes CalComp, Varian, and CRT capabilities. These programs have been utilized on all the above equipment. These programs may not be operational directly on other computer systems without some modification. Therefore, a description of the graphics subroutines that are used is given on comment cards in the
subroutine called DOCMNT. This information is adequate for conversion to another graphics system.

Storage Allocation

Dynamic storage allocation is used in both programs for efficient accommodation of models of varying complexity. All large arrays used by the programs are stacked in blank COMMON designated in the MAIN program as ZZZ. The starting location of each array is calculated in the MAIN programs and the total blank COMMON requirement is printed out. The amount of blank COMMON is determined in general by the number of grid points, number of elements, and number of displacements considered at the grid points. The total required field length is the address of the first location in blank COMMON plus the blank COMMON length requirement.

Since the field length is problem dependent, the user must estimate the required field length for any new problem. An approximate formula (in decimal) for the field length required for the oblique-orthographic-projection program is

$$FL_{10} = 18000 + (4 + \text{NUDISP} + \text{NVDISP} + \text{NWDISP})\text{NNDEST}$$

where NNDEST is equal to the number of grid points and NUDISP, NVDISP, and NWDISP are the indicators for displacements in the x-, y-, and z-direction, respectively (must be 0 or 1). The field length for this program is independent of the number of elements which are stored on an auxiliary storage unit. A corresponding formula for the contour-plot program is

$$FL_{10} = 17000 + 5(\text{NNDEST}) + 6(\text{NELEST})$$

where NNDEST is equal to the number of grid points and NELEST is the number of elements. For operation of either of the programs on the CRT display at Langley Research Center an additional storage allocation of 3500 decimal words is required.

Programs and Subroutines

The two programs for oblique orthographic projections and contour plots are separate but intended to be used as companion programs. The input decks are designed to be similar, and many of the variables controlling options in the programs have identical names and purposes.

Complete listings of the programs are given in appendixes B and C as well as information on flow logic for the programs. Comment cards are incorporated in the
computer programs to give the purpose of each subroutine and to indicate the operations that are performed.

COMPONENT PROGRAM USE

General Setup of Input Deck

In general the input data decks for both the oblique-orthographic-projection program and the contour-plotting program consist of six separate groups of data as shown schematically in figure 8. These groups are as follows:

1. A single card containing any desired title information
2. NAMELIST OPTION containing values to allocate storage in blank COMMON and control values specifying various program options
3. a geometry deck containing the specification of grid points and connecting elements in the model
4. an optional single card used to identify the deck of data to be plotted
5. the deck of data to be plotted
6. NAMELIST PICT containing values to specify the type of plot desired and what information is to be included on the plots.

Selected groups of this basic input deck can be repeated to make different plots of the same data or to input additional groups of data to be plotted during a single run. Details of the data contained in the input decks for both programs are described in the two subsequent sections of this report.

Options are provided to input geometry data and data to be plotted in several different forms by selection of appropriate input subroutines. Generality is provided in subroutines GEOM1 and DATA1 by user specification of the format by which the data is to be read directly in the input deck. A specialized subroutine GEOM2 provides for direct input of bulk data decks for the NASTRAN (ref. 9) program. The plotting programs herein handle only one- or two-dimensional elements (no solid elements) containing up to four grid points. In addition, only a single coordinate system can be used and therefore models defined using alternate coordinate systems must be transformed to the basic coordinate system of the model before plotting. Grid point numbers and element numbers need not be sequential since they are reordered internally in the programs. Data to be plotted can be read from magnetic tape by using subroutine DATA5. In addition, dummy subroutines GEOM9 and DATA9 are included for user-prepared routines which read geometry (grid points and connecting elements) information and data to be plotted.
Input for Oblique Orthographic Projections

The input data deck is illustrated in the sequence shown schematically in figure 8. The data must be constructed in the order shown in figure 8 and is described in detail in this section. For clarity, no zeros appear in variable names described in this section.

Title card.- This single card contains any desired alphanumeric information in columns 1 to 80. The title will appear at the beginning of the plots.

NAMELIST OPTION.- This NAMELIST contains values to allocate storage in blank COMMON and control values specifying various program options.

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNDEST</td>
<td>200</td>
<td>Estimated number of grid points, must be equal to or greater than the actual number of grid points</td>
</tr>
<tr>
<td>NUDISP</td>
<td>0</td>
<td>0 no displacement data in x-direction. 1 data including displacements in x-direction</td>
</tr>
<tr>
<td>NVDISP</td>
<td>0</td>
<td>0 no displacement data in y-direction. 1 data including displacements in y-direction</td>
</tr>
<tr>
<td>NWDISP</td>
<td>0</td>
<td>0 no displacement data in z-direction. 1 data including displacements in z-direction</td>
</tr>
<tr>
<td>KGEOM</td>
<td>1</td>
<td>Specifies the subroutine and corresponding method of input for model geometry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 subroutine GEOM1 to read in grid points and elements from cards with user-specified/format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 subroutine GEOM2 to read in NASTRAN bulk data deck with data in column widths of 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9 subroutine GEOM9, a user-supplied subroutine</td>
</tr>
<tr>
<td>KDATA</td>
<td>1</td>
<td>Specifies the subroutine and corresponding method of input for displacement data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 subroutine DATA1 to read in displacement data from cards with user-specified format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 subroutine DATA5 to read in displacement data from TAPE20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9 subroutine DATA9, a user-supplied subroutine</td>
</tr>
<tr>
<td>NVALUS</td>
<td>0</td>
<td>Used if KDATA = 5; specifies the number of grid points at which displacement data is to be read from TAPE20 (must be less than or equal to NNDEST)</td>
</tr>
<tr>
<td>IRESEQ</td>
<td>1</td>
<td>Grid point numbers are stored in the program from 1 to the total number of grid points</td>
</tr>
</tbody>
</table>
FORTRAN Default Description

name value description

IRESEQ 1 0 no internal resequencing of grid points necessary; they are already ordered in ascending order starting with 1
1 resequence grid points in the same order as they are input

KPLT 1 Specifies the type of output device to be used
1 CalComp
2 CalComp with plotting speed reduced for Leroy pens
3 VARIAN
4 cathode-ray-tube (CRT) console (set up for CDC 250 scopes at Langley Research Center)

XSPACE 10.0 Space between plots in x-direction, in inches

FSIZE 25.0 Paper size in y-direction, in inches (used in scaling of plots to insure this dimension is not exceeded)

IDCASE 0 0 no identification card preceding decks of displacement values
1 identification card preceding decks of displacement values

Geometry input data deck.- This portion of the input deck contains the geometry definition of grid points and connecting elements. The deck has one of the following forms, depending on the value of KGEOM specified in NAMELIST OPTION.

KGEOM = 1

(a) A single card containing the word FORMAT in columns 1 to 6 and a variable format corresponding to the format of the grid point cards with the left parenthesis starting in column 11 and up to column 80 may be used.

(b) Deck of grid point cards. Each card contains 4 values: grid point number (integer), x-coordinate (real), y-coordinate (real), and z-coordinate (real). The format is specified in (a) above.

(c) A single card containing the word ENDGRID in columns 1 to 7.

(d) A single card containing the word FORMAT in columns 1 to 6 and a variable format corresponding to the format of the element cards with left parenthesis starting in column 11 and up to column 80 may be used.
(e) Deck of element cards. Each card contains 5 integer fields which are the element number and grid point numbers at the vertices of the elements. For triangular elements the last integer field must be blank or zero. For rod or beam elements the last two integer fields must be blank or zero. The format is specified in (d) on the preceding page.

(f) A single card containing the word ENDGEOM in columns 1 to 7.

KGEOM = 2

(a) A single card containing the word LINEEL in columns 1 to 6 and up to 9 NASTRAN lineal element connection names, which are adjusted to the left in field widths of 8, starting in column 9 (cols. 9 to 16, 17 to 24, . . . , 73 to 80). This card can be omitted if lineal elements are not used for the plot.

(b) A single card containing the word TRIAEL in columns 1 to 6 and up to 9 NASTRAN triangular element connection names, which are adjusted to the left in field widths of 8, starting in column 9 (cols. 9 to 16, 17 to 24, . . . , 73 to 80). This card can be omitted if triangular elements are not used for the plot.

(c) A single card containing the word QUADEL in columns 1 to 6 and up to 9 NASTRAN quadrilateral connection names, which are adjusted to the left in field widths of 8, starting in column 9 (cols. 9 to 16, 17 to 24, . . . , 73 to 80). This card can be omitted if quadrilateral elements are not used for the plot.

(d) A NASTRAN bulk data deck. Only the GRID cards and the element connection cards with names matching those given on the LINEEL, TRIAEL, and QUADEL cards will be used for the plot. All other cards in the NASTRAN bulk data deck will be ignored.

(e) A single card containing the word ENDGEOM in columns 1 to 7.

KGEOM = 9

Calls subroutine GEOM9 which is prepared by the user to read geometry data.

Case identification card.- This single card is omitted if IDCASE = 0 is specified in NAMELIST OPTION. If present, this card contains any desired alphanumeric information in columns 1 to 80. The identification will appear on all plots of the case.

Deck of data to be plotted.- This deck contains displacement sets at grid points for the oblique-orthographic-projection program. A displacement set for each grid point is defined to contain from 2 to 4 values (i.e., a grid point number and displacements corresponding to NUDISP, NVDISP, or NWDISP equal to 1). The deck has one of the following forms, depending on the value of KDATA specified in NAMELIST OPTION.
KDATA = 1

(a) A single card containing the word FORMAT in columns 1 to 6 and a variable format for the data cards with the left parenthesis starting in column 11 and up to column 80 may be used. If displacements are included for more than one grid point per card, the number of grid points per card must be entered as an integer in column 8.

(b) Deck of displacement sets. There can be multiple displacement sets per card or the set can extend to more than one card (often the case with NASTRAN punched output) which can be handled with a format for reading multiple cards.

(c) Blank card or cards to end the data deck. The number of blank cards must correspond to the number of cards read at one time by the specified variable format.

KDATA = 5

Reads NVALUS (from NAMELIST OPTION) displacement sets from TAPE20. Each displacement set must have been written on TAPE20 as an unformatted record.

KDATA = 9

Calls subroutine DATA9 which is prepared by the user to read displacement data.

NAMELIST PICT. - This NAMELIST contains values to specify the type of plot desired and the information that is to be included on the plots.

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KHORZ</td>
<td>1</td>
<td>Integer designating the horizontal axis of the viewing plane where 1 = X₀, 2 = Y₀, and 3 = Z₀</td>
</tr>
<tr>
<td>KVERT</td>
<td>2</td>
<td>Integer designating the vertical axis of the viewing plane where 1 = X₀, 2 = Y₀, and 3 = Z₀</td>
</tr>
<tr>
<td>PHI</td>
<td>0.0</td>
<td>Angular rotation of model about its X-axis in degrees (must be performed third)</td>
</tr>
<tr>
<td>THETA</td>
<td>0.0</td>
<td>Angular rotation of model about its Y-axis in degrees (must be performed second)</td>
</tr>
<tr>
<td>PSI</td>
<td>0.0</td>
<td>Angular rotation of model about its Z-axis in degrees (must be performed first)</td>
</tr>
<tr>
<td>NEWFR</td>
<td>1</td>
<td>1 frame change before plotting (a frame change resets the x-origin past previous plot by XSPACE given in NAMELIST OPTION and resets the y-origin at 0.0)</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NEWFR</td>
<td>1</td>
<td>0 no frame change before plotting</td>
</tr>
</tbody>
</table>
| ISCALE       | 1             | 1 automatic computation of proper origin location and scaling of plot  
|              |               | 2 user-specified origin and scaling |
| PLOTSZ       | 10.0          | Maximum dimension desired on completed plot, in inches (used for scaling if ISCALE = 1) |
| XORGN        | 0.0           | x-location of plot origin (used if ISCALE = 2) |
| YORGN        | 0.0           | y-location of plot origin (used if ISCALE = 2) |
| PSCALE       | 1.0           | Model size reduction factor (i.e., PScale is equal to actual model size divided by desired plot size (used if ISCALE = 2)) |
| NOTAT        | 0             | 0 no numbering on plots  
|              |               | 1 numbering of grid points  
|              |               | 2 numbering of elements |
| XLHT         | 0.15          | Height of integers specified by NOTAT, in inches (must be \(\geq 0.07\)) |
| KDISP        | 0             | 0 plot of undeformed structure  
|              |               | 1 plot of deformed structure  
|              |               | 2 exploded plot  
|              |               | 3 displacements represented by vectors |
| IDMAG        | 2             | 1 direct magnification of displacement data by DMAGS  
|              |               | 2 scaling of displacement data to a maximum value of DMAGS |
| DMAGS        | 1.0           | Magnification of displacements (if KDISP = 1 or 3)  
|              |               | Reduction factor of elements (if KDISP = 2) |
| KSYMXY       | 0             | 1 symmetry about X-Y plane |
| KSYMZX       | 0             | 1 symmetry about X-Z plane |
| KSYMYZ       | 0             | 1 symmetry about Y-Z plane |

Symmetries are performed consecutively (i.e., a plate quadrant with KSYMZX and KSYMYZ equal to one would yield a complete plate).
FORTRAN name | Default value | Description
---|---|---
XXMAX, YYMAX, ZZMAX | 1.0 E+20 | Locate cutting planes parallel to principal planes (X-Y, X-Z, Y-Z) to limit plot
XXMIN, YYMIN, ZZMIN | -1.0 E+20 | Maximum grid point identification number to be included in plot
NDMAX | 9999999999 | Minimum grid point identification number to be included in plot
NDMIN | 0 | Maximum element identification number to be included in plot
NELMAX | 9999999999 | Minimum element identification number to be included in plot
NELMIN | 0 | Specifies control option after plot is complete
KODE | 0 | 0 last plot, exit from program
 | 1 | read another NAMELIST PICT
 | 2 | read a new set of displacement data, including a case identification card if present
 | 3 | read a complete new set of input data, including a title card.

This section describes a complete basic set of input data if KODE = 0 in NAMELIST PICT. For KODE = 1, 2, or 3, additional sections of the basic deck must be repeated. The deck must end with NAMELIST PICT having a value of KODE = 0.

**Input for Contour Plots**

The input data deck is illustrated in the sequence shown schematically in figure 8. The data must be constructed in the order shown in figure 8 and is described in detail in this section. For clarity no zeros appear in the variable names described in this section.

**Title card.**- This single card contains any desired alphanumeric information in columns 1 to 80. The title will appear at the beginning of the plots.

**NAMELIST OPTION.**- This NAMELIST contains values to allocate storage in blank COMMON and control values specifying various program options.

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| NNDEST | 200 | Estimated number of grid points, must be equal to or greater than the actual number of grid points
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NELEST</td>
<td>200</td>
<td>Estimated number of elements, must be equal to or greater than the actual number of elements.</td>
</tr>
</tbody>
</table>
| KGEOM        | 1             | Specifies the subroutine and corresponding method of input for model geometry:  
1 subroutine GEOM1 to read in grid points and elements from cards with user-specified format  
2 subroutine GEOM2 to read in NASTRAN bulk data deck with data in column widths of 8  
9 subroutine GEOM9, a user-supplied subroutine. |
| KDATA        | 1             | Specifies the subroutine and corresponding method of input for control variable data to be represented by contour lines:  
1 subroutine DATA1 to read in data to be plotted from cards with user-specified format  
5 subroutine DATA5 to read in data to be plotted from TAPE20  
9 subroutine DATA9, a user-supplied subroutine. |
| NVALUS       | 0             | Used if KDATA = 5; specifies the number of given control variable points at which data to be plotted is to be read from TAPE20 (can be either the number of grid points or the number of elements). |
| IRESEQ       | 1             | Grid point numbers are stored in the program from 1 to the total number of grid points:  
0 no internal resequencing of grid points necessary; they are already ordered in ascending order starting with 1  
1 to resequence grid points in the same order as they are input. |
FORTRAN Default Description

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KPLOT</td>
<td>1</td>
<td>Specifies the type of output device to be used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 CalComp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 CalComp with plotting speed reduced for Leroy pens</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 VARIAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 cathode-ray-tube (CRT) console (set up for CDC 250 scopes at Langley Research Center)</td>
</tr>
<tr>
<td>INFOR</td>
<td>1</td>
<td>1 data to be plotted is specified at grid points</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 data to be plotted is specified at element centroids</td>
</tr>
<tr>
<td>XSPACE</td>
<td>10.0</td>
<td>Space between plots in x-direction, in inches</td>
</tr>
<tr>
<td>KSIGN</td>
<td>1</td>
<td>-1 change signs of y-coordinates</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 do not change signs of y-coordinates</td>
</tr>
<tr>
<td>IDCASE</td>
<td>0</td>
<td>0 no identification card preceding decks of data to be plotted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 identification card preceding decks of data to be plotted</td>
</tr>
</tbody>
</table>

Geometry input data deck. This portion of the input deck contains specifications of grid points and elements describing the reference surface. The deck has one of the following forms, depending on the value of KGEOM specified in NAMELIST OPTION.

KGEOM = 1

(a) A single card containing the word FORMAT in columns 1 to 6 and a variable format corresponding to the format of the grid point cards with the left parenthesis starting in column 11 and up to column 80 may be used.

(b) Deck of grid cards. Each card contains 3 values: grid point number (integer), x-coordinate (real), and y-coordinate (real). The format is specified by (a) above.

(c) A single card containing the word ENDGRID in columns 1 to 7.

(d) A single card containing the word FORMAT in columns 1 to 6 and a variable format corresponding to the format of the element cards with left parenthesis starting in column 11 and up to column 80 may be used.
(e) Deck of element cards. Each card contains 5 integer fields which are the element number and grid points at the vertices of the element. For triangular elements the last integer field must be blank or zero. The format is specified in (d) from the preceding page.

(f) A single card containing the word ENDGEOM in columns 1 to 7.

KGEOM = 2

(a) A single card containing the word TRIAEL in columns 1 to 6 and up to 9 NASTRAN triangular element connection names, which are adjusted to the left in field widths of 8, starting in column 9 (cols. 9 to 16, 17 to 24, . . . , 73 to 80). This card can be omitted if triangular elements are not used for the plot.

(b) A single card containing the word QUADEL in columns 1 to 6 and up to 9 NASTRAN quadrilateral connection names, which are adjusted to the left in field widths of 8, starting in column 9 (cols. 9 to 16, 17 to 24, . . . , 73 to 80). This card can be omitted if quadrilateral elements are not used for the plot.

(c) A NASTRAN bulk data deck. Only the GRID cards and the element connection cards with names matching those given on the TRIAEL and QUADEL cards will be used for the plot. All other cards in the NASTRAN bulk data deck will be ignored.

(d) A single card containing the word ENDGEOM in columns 1 to 7.

KGEOM = 9

Calls a subroutine GEOM9 which is prepared by the user to read geometry data.

Case identification card.- This single card is omitted if IDCASE = 0 is specified in NAMELIST OPTION. If present, this card contains any desired alphanumeric information in columns 1 to 80. The identification will appear on all plots of the case.

Deck of data to be plotted.- This deck contains sets of a control variable value corresponding to each grid point or element (depending on INFOR in NAMELIST OPTION) to be represented by contour lines. The deck has one of the following forms, depending on the value of KDATA specified in NAMELIST OPTION.

KDATA = 1

(a) A single card containing the word FORMAT in columns 1 to 6 and the variable format for the data cards with the left parenthesis starting in column 11 and up to column 80 may be used. If control variables are included for more than one grid point or element per card, the number of grid points or elements per card must be entered as an integer in column 8.
(b) Deck of data to be plotted. There can be multiple data value sets per card or
the set can extend to more than one card (often the case with NASTRAN punched
output) which can be handled with a format for reading multiple cards.

(c) Blank card or cards to end the data deck. The number of blank cards must cor-
respond to the number of cards read at one time by the specified variable format.

KDATA = 5

Reads NVALUS (from NAMELIST OPTION) sets of control variable values from
TAPE20. Each set of control variables must have been written on TAPE20 as an
unformatted record.

KDATA = 9

Calls subroutine DATA9 which is prepared by the user to read data to be plotted.

NAMELIST PICT.- This NAMELIST contains values to specify the type of plot
desired and what information is to be included on the plots.

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
</table>
| N PLOT       | 4            | Specifies the type of plot to be generated
|              |              | 1 layout of elements in reference surface
|              |              | without grid point or element labels
|              |              | 2 element layout with grid point labels
|              |              | 3 element layout with element labels
|              |              | 4 contour plots without symbols at grid
|              |              | points
|              |              | 5 contour plots with symbols at grid points
| XORGN        | 0.0          | x-location of origin of first plot, in inches
| YORGN        | 0.0          | y-location of origin of first plot, in inches
| PSSCALE      | 1.0          | Model size reduction factor (i.e., PSSCALE is
equal to actual model size divided by desired
plot size)
| ISCALE       | 3            | Method of scaling control variable data to be
plotted; contour lines have only integer val-
ues annotated on the plot and, thus, the data
must be scaled such that these integers will
contain the desired number of significant
digits; the definitions of WMAGS and ICNTRS
depend on the value of ISCALE
|              |              | 1 user specification of scale factors
<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISCALE</td>
<td>3</td>
<td>2 program calculation of scale factors to give the user-specified number of significant digits in annotation of the maximum absolute contour line.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 program calculation of scale factors to give WMAGS as the maximum value of data.</td>
</tr>
<tr>
<td>WMAGS</td>
<td>100.0</td>
<td>If ISCALE = 1, magnification of control variables for annotation of contour lines on plot.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ISCALE = 2, number of significant digits in annotation of maximum absolute contour line (WMAGS = 1.0, 2.0, 3.0, etc.).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ISCALE = 3, maximum value of scaled data, WMAGS ≥ 2.0 (maximum contour line is integer truncation of WMAGS).</td>
</tr>
<tr>
<td>ICNTRS</td>
<td>10</td>
<td>If ISCALE = 1, user-specified contour interval (difference in integer values of adjacent contour lines).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ISCALE = 2 or 3, approximate number of different contour line values; the contour interval is calculated by the program.</td>
</tr>
<tr>
<td>XLHT</td>
<td>0.15</td>
<td>Height of integers to be annotated on plots, in inches (must be ≥0.07).</td>
</tr>
<tr>
<td>NXLAB</td>
<td>0</td>
<td>Total number of lines parallel to Y-axis along which contour lines are labeled, must be ≤10 (all contour lines are labeled where they intersect with these selected lines); these labels are in addition to those automatically provided at boundaries of the contour surface.</td>
</tr>
<tr>
<td>XLAB</td>
<td>all zeros</td>
<td>Array of distances in x-direction from the origin to lines parallel to the Y-axis along which contour lines are labeled; there must be NXLAB of these values and they must be in units of the original (unscaled) model.</td>
</tr>
<tr>
<td>FORTRAN name</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>NYLAB</td>
<td>0</td>
<td>Same as NXLAB for label locations parallel to X-axis</td>
</tr>
<tr>
<td>YLAB</td>
<td>all zeros</td>
<td>Same as XLAB for label locations parallel to X-axis</td>
</tr>
<tr>
<td>KODE</td>
<td>0</td>
<td>Specifies control option after plot is complete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0  last plot, exit from program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1  read another NAMELIST PICT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2  read a new set of control variable values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to be plotted including a case identification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>card if present</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3  read a complete new set of input data,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>including a title card</td>
</tr>
</tbody>
</table>

The above NAMELIST comprises a complete basic set of input data if KODE = 0 in NAMELIST PICT. For KODE = 1, 2, or 3, additional sections of the basic deck must be repeated. The deck must end with NAMELIST PICT having a value of KODE = 0.

EXAMPLES OF DATA PREPARATION

An example problem is presented to illustrate preparation of input data for both the oblique-orthographic-projection program and contour-plotting program as well as printed and plotted output. The problem presented herein is designed to exercise and illustrate use of various options of the programs. The example selected is the graphical display of the normal displacements of a square flat plate under uniform lateral pressure. A finite-element structural model of the flat plate with 36 grid points and 25 quadrilateral plate elements is used to reduce the amount of input data required to illustrate the use of the plotting programs. The input data for the programs are presented in appendixes D and E. Differences between these input decks occur primarily in the quantities contained in the NAMELIST OPTION and NAMELIST PICT. Output listings are similar for both programs, therefore, only the output for the contour program is included herein.

Oblique Orthographic Projections

The input deck for oblique orthographic projections of the flat plate example is given in appendix D. The geometry portion of the input deck includes the entire NASTRAN case control and bulk data deck used for analysis of the plate. Option KGEOM = 2 is used to read this data; the TRIAEL card is omitted since no triangles are used, and the CQUAD1 element type is the only one specified on the QUADEL card. Therefore, in this case, all
cards in the NASTRAN deck are ignored except GRID cards and CQUAD1 cards. The
displacement data are punched output from a NASTRAN analysis. Two cards are punched
for each grid point containing displacements in the x-, y-, and z-directions and rotations
about the X-, Y-, and Z-axes. The specified format reads two cards at a time but retains
only the displacements in the x-, y-, and z-directions which are given on the first card
(x- and y-displacements are zero at all grid points for this example). Two blank cards
are needed to indicate the termination of the displacement data for this particular format.

Two NAMELISTS PICT are used to generate oblique orthographic projections. The
first NAMELIST contains data for generating the deformed shape shown in figure 9(a),
and contains KODE = 1 which causes the second NAMELIST to be read. In the second
NAMELIST, the value of KDISP is changed from 1 (as specified in the first NAMELIST)
to 3 which results in a plot with the displacements represented by vectors as shown in
figure 9(b). The magnitude of the displacements are arbitrarily exaggerated for visual
clarity. The second NAMELIST PICT terminates the run with KODE = 0.

Contour Plot

The input deck for a contour plot of the flat plate example is given in appendix E.
Default values are used for most of the NAMELIST parameters. Use of the option,
KGEOM = 1, to input grid point and element data with a user-specified format is illus-
trated. GRID cards and CQUAD1 cards from a NASTRAN deck are used, however, any
specified card formats could have been used. In the case of the GRID cards, only the x-
and y-coordinates are read. Thus, if nonzero z-coordinates were used to locate the grid
points, they would be neglected and only the projection would be used. The data to be
represented by contour lines are the displacements normal to the plate and are given at
four grid points per card. In NAMELIST PICT, the option to specify labels on contour
lines on the interior of the plotting surface (see values of NXLAB, XLAB, NYLAB, and
YLAB) is used since the contour lines do not intersect the boundary.

The contour plot which was generated by using this input is shown in figure 10. The
displacements given in the data are symmetric about the two orthogonal principal planes
through the center of the plate. However, the contour lines generated by the program are
slightly unsymmetric as shown in figure 10. This slight nonsymmetry is introduced by
the process of breaking the quadrilaterals into triangles to define the contour surface.
That is, the triangularization of the surface, which is done automatically, is not quite sym-
metric since the resulting right triangles face the same direction in all quadrants of the
plate. This effect would have diminished with a more refined model. Default scaling of
the data was used to assign a value of 100 to the point of maximum displacement at the
center of the plate. A listing of the contour-program output for this example is presented
in appendix F. Titles and headings are provided to indicate the meaning of the printed
output.

26
CONCLUDING REMARKS

User and programer documentation is presented for two programs for automatic plotting of digital data. One of the programs generates oblique orthographic projections of three-dimensional numerical models and the other program generates contour plots of data distributed in an arbitrary planar region. The user documentation gives a general description of the computational algorithms, user instructions, and complete listings of the programs. Several plots are included to illustrate various program options and a single example is described in detail to facilitate learning the use of the programs.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., November 18, 1974.
APPENDIX A

INCENTER OF A GENERAL TRIANGLE

The incenter of a triangle is the center of an inscribed circle as shown in figure 11. If the incenters of two similar triangles are coincident, the perpendicular distance between corresponding sides of the triangles will be equal. This geometric relationship is desired when locating the reference point for triangular elements in an exploded plot.

The coordinates of the incenter of a general triangle can be derived by using area coordinates. Areas of triangles $A_i$, $A_j$, and $A_k$ shown in figure 11 are

$$
\begin{align*}
A_i &= \frac{1}{2} ha \\
A_j &= \frac{1}{2} hb \\
A_k &= \frac{1}{2} hc
\end{align*}
$$

By use of equations (8) and (A1), the area coordinates are determined as

$$
\begin{align*}
L_i &= \frac{a}{s} \\
L_j &= \frac{b}{s} \\
L_k &= \frac{c}{s}
\end{align*}
$$

where $s = a + b + c$.

The area coordinates can be transformed to Cartesian coordinates by using equations (10) to (12) and equation (A2)

$$
\begin{align*}
x_p &= \frac{ax_i + bx_j + cx_k}{s} \\
y_p &= \frac{ay_i + by_j + cy_k}{s} \\
z_p &= \frac{az_i + bz_j + cz_k}{s}
\end{align*}
$$
where

\[
\begin{align*}
  a &= \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2} \\
  b &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2} \\
  c &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}
\end{align*}
\] (A4)
APPENDIX B

LISTING OF COMPUTER PROGRAM FOR OBLIQUE ORTHOGRAPHIC PROJECTIONS

An overall flow chart for this program is given in figure 12. The MAIN program is used to allocate blank COMMON storage and to call other subroutines necessary to read input data and to generate the desired plots. The purpose of each subroutine is described in comment cards in the listing. A subroutine called DOCMNT consisting entirely of comment cards is included in the program. Subroutine DOCMNT contains (1) a directory of selected variables used in the program, (2) user-input instructions, and (3) a description of plotting subroutines which are required from the Langley Graphic Output System library.
APPENDIX B – Continued

PROGRAM MAIN (INPUT=201, OUTPUT=201, TAPE5=INPUT, TAPE6=OUTPUT, 
1 TAPE9, TAPE10, TAPE20=201)

C
C *** THIS IS THE MAIN PROGRAM WHICH CALLS OTHER ROUTINES
C
INTEGER NUMPT, XPT, YPT, ZPT, UPT, VPT, WPT
COMMON/CTRL/ KEGM, KDATA, KPLUT, KSYMXY, KSYMZX, KSYMZY, NGAT, XLHT,
1 KVERT, PHII, THETA, PSI, NEWFR, ISCALE, PLOTSZ, XORG, YORG,
2 SCALE, KDISP, DMAG, KODE
COMMON/LIMITS/ XXMAX, YYMAX, ZZMAX, XXMIN, YYMIN, ZZMIN, NDMAX, NDMIN,
1 NELMAX, NELMIN
COMMON/CORGN/ XOABS, YOABS, XPMAX, XSPACE, PSIZE
COMMON/GLOB/ A(13,3)
COMMON/SAVEV/ DMAGS, DMAG
COMMON/KOUNT/ NNODE, NNDEST, NUDISP, NVDISP, NWDISP
COMMON/VALUES/ NVALLS
COMMON/CASEID/ ICCASE
COMMON/ABLK/ A(3,3)
COMMON/NLIST/ PICT/ KSYMXY, KSYMZX, KSYMZY, XXMAX, YYMAX, ZZMAX, XXMIN,
1 YYMIN, ZZMIN, NELMAX, NELMIN, XVMAX, YVMAX, ZZVMAX, XVMIN, YVMIN, ZZVMIN,
3 YVMAX, ZZVMAX, XVMIN, YVMIN, ZZVMIN, NDVMAX, NDVMIN, NVVALLS, NVVALLS, XVLHT

C *** TO ZERO NODE AND ELEMENT SUMMATION COUNTERS
C
ILOOP = 0
NNODE = 0
XOABS = 0.0
YOABS = 0.0
XPMAX = 0.0
500 CONTINUE
XSTRT = 0.0
YSTRT = 0.0
WRITE(6,1) 1 CONTINUE
8 FORMAT(1H1)

C *** TO READ TITLE CARD FOR RUN
C
READ(5,10) ABCO
10 FORMAT(8A10)
IF(EOF,3) Z2222,3333
2222 CALL PSTOP
3333 CONTINUE
WRITE(6,11) ABCO
11 FORMAT(///20X,8A10,///)
CALL INITAL
IF(KPLUT, NE.4, OR.KDATA, NE.5) GO TO 111
REWORD 20
CALL MESSAGE(1, 24HENTERING ORTHOGC PROGRAM, 24)
CALL MESSAGE(1, 19HSELECT DESIRED MODE, 19)
CALL MESSAGE(1, 18HBY PRESSING FN KEY, 18)
CALL NEXT(NKEY)
IMODE = NKEY-1
IF(IMODE, EQ.0) GO TO 111
DO 57 J=1, IMODE
KJUNT = 0
APPENDIX B – Continued

IF(NUDISP.NE.0) KOUNT = KOUNT+1
IF(NVDISP.NE.0) KOUNT = KOUNT+1
IF(NWISP.NE.0) KGUNT = KOUNT+1
DO 58 I=1,NVALUS
   READ(20) IDUM,(DSAV(K),K=1,KOUNT)
58 CONTINUE
57 CONTINUE
111 CONTINUE
HEIGHT = 0.15
XSTRT = XSTRT+2.0*HEIGHT
YSTRT = 1.0
CALL NOTATE(XSTRT,YSTRT,HEIGHT,ABCD,90.0,80)
CALL CALPLT(12.0,0.0,-3)

C *** TO SET POINTERS FOR BLANK COMMON STORAGE ZZZ
C *** WITH INTEGER NAMES OF ARRAYS USED IN CALLED SUBROUTINES
C
NUMPT = 1
XPT -= NUMPT *-NNOEST
YPT = XPT+NNDEST
ZPT = YPT+-NNDEST
UPT = ZPT+NNDEST
IF(NUDISP.EQ.0) VPT = UPT*-1
IF(NUDISP.NE.0) VPT = UPT+-NNDEST
IF(NVDISP.EQ.0) WPT = VPT+-1
IF(NVDISP.NE.0) WPT = VPT+-NNDEST
IF(NWISP.EQ.0) NENO = WPT+-1-1
IF(NWISP.NE.0) NEND = WPT+NNDEST-1
WRITE(6,15) NEND
15 FORMAT(///,20X,*8LANK COMMON STORAGE ZZZ REQUIRES AT LEAST *,I6,
1* LOCATIONS FOR THIS CASE*///)
IF(KGEOM.EQ.1) CALL GEOM1
1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
IF(KGEOM.EQ.2) CALL GEOM2
1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
IF(KGEOM.EQ.3) CALL GEOM9
1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
CALL PNTOUT1,
1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
600 CONTINUE
IF(IDCASE.EQ.0) GO TO 650
READ(5,10) ABCD
WRITE(6,11) ABCD
650 CONTINUE
CALL ZEROD
1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
IF(KDATA.EQ.1) CALL DATA1
1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
IF(KDATA.EQ.2) CALL DATA5
1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
IF(KDATA.EQ.3) CALL DATA9
1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
CALL PNTOUT12,
1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
700 CONTINUE
IF(KPLOT.EQ.4.AND.LOOP.NE.0) GO TO 6000
READ(5,PICT)
WRITE(6,PICT)

32
APPENDIX B – Continued

6000 CONTINUE
  IF(KPLOT.EQ.4) CALL CRRT1
  CALL USCALE
  1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
  CALL BOUND
  CALL ROTAT
  CALL PLUT
  1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
  IF(NOTAT.EQ.1) CALL NDLET
  1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
  IF(KPLOT.EQ.4) CALL CDRPLT(12.0,0.0,-3)
  IF(KPLOT.EQ.4) CALL CCRK2
  ILOOP = ILOOP+1
  IF(KODE.EQ.0) GO TO 800
  GO TO (700,600,500) KODE
800 CONTINUE
  CALL PSFOP
END OF MAIN
SUBROUTINE DOCMNT

*** THIS SUBROUTINE CONTAINS PROGRAM DOCUMENTATION

DESCRIPTION OF INPUT DATA CARDS

TITLE CARD - CONTAINS ANY DESIRED ALPHANUMERIC INFORMATION IN COLS.1-80.

NAMELIST OPTION - CONTAINS VALUES TO ALLOCATE STORAGE IN BLANK COMMON ZZZ,
AND CONTROL VALUES NEEDED BY THE PROGRAM.

THE FOLLOWING VALUES ARE INCLUDED---

NDEST = ESTIMATE NUMBER OF GRID POINTS TO BE USED. VALUE MUST BE
GREATER THAN OR EQUAL TO THE ACTUAL NUMBER OF GRID POINTS.
** DEFAULT = 200 **

NUDISP = 0 FOR NO DISPLACEMENT DATA IN X-DIRECTION.
= 1 FOR DATA INCLUDING DISPLACEMENTS IN X-DIRECTION.
** DEFAULT = 0 **

NVDISP = 0 FOR NO DISPLACEMENT DATA IN Y-DIRECTION.
= 1 FOR DATA INCLUDING DISPLACEMENTS IN Y-DIRECTION.
** DEFAULT = 0 **

NWDISP = 0 FOR NO DISPLACEMENT DATA IN Z-DIRECTION.
= 1 FOR DATA INCLUDING DISPLACEMENTS IN Z-DIRECTION.
** DEFAULT = 0 **

KGEOM SPECIFIES SUBROUTINE AND CORRESPONDING METHOD OF INPUT FOR
MODEL GEOMETRY.
KGEOM = 1 FOR GRID POINTS AND ELEMENTS READ FROM CARDS WITH USER
SPECIFIED FORMAT.
= 2 FOR NASTRAN DECK WITH CARD IDENTIFIERS LEFT ADJUSTED
AND DATA IN COLUMN WIDTHS OF 8.
= 9 FOR USER SUPPLIED SUBROUTINE - GEOM9.
** DEFAULT = 1 **

KDATA SPECIFIES SUBROUTINE AND CORRESPONDING METHOD OF INPUT FOR
DISPLACEMENT DATA.
KDATA = 1 FOR SUBROUTINE DATAG TO READ IN DISPLACEMENT DATA
FROM CARDS WITH USER SPECIFIED FORMAT.
= 5 FOR SUBROUTINE DATAG TO READ IN DISPLACEMENT DATA
FROM TAPE20.
= 9 FOR SUBROUTINE DATAG, A USER SUPPLIED SUBROUTINE.
** DEFAULT = 1 **

NVALUS - USED IF KDATA = 5 TO SPECIFY THE NUMBER OF SETS OF
DISPLACEMENTS TO BE READ FROM TAPE20.
** DEFAULT = 0 **

IRESEQ = 0 FOR NO RESEQUENCING OF GRID POINT NUMBERS.
= 1 TO RESEQUENCE GRID POINT NUMBERS IN SAME ORDER
AS THEY ARE INPUT.
** DEFAULT = 1 **

KPLUT SPECIFIES THE TYPE OF OUTPUT DEVICE TO BE USED.
KPLUT = 1 FOR CALCOMP.
= 2 FOR CALCOMP WITH PLOTTING SPEED REDUCED TO USE LEROY PENS.
APPENDIX B – Continued

= 3 FOR VARIAN.
= 4 FOR CRT (USE CDC250 SCIOPES AT LRC)
** DEFAULT = 1 **

XSPACE = SPACE BETWEEN PLOTS IN X-DIRECTION, IN INCHES.
** DEFAULT = 10.0 **

PSIZE = PAPER SIZE IN Y-DIRECTION IN INCHES, USED IN SCALING OF
    PLOTS TO INSURE THIS DIMENSION IS NOT EXCEEDED.
** DEFAULT = 25.0 **

ILOCASE = 0 FOR NO TITLE CARD PRECEDING
    DECKS OF DISPLACEMENT VALUES.
= 1 FOR TITLE CARD PRECEDING
    DECKS OF DISPLACEMENT VALUES.
** DEFAULT = 0 **

MODEL GEOMETRY IS NOW INPUT IN ONE OF THE FOLLOWING FORMS,
DEPENDING ON THE VALUE OF KGEOM SPECIFIED IN NAMELIST OPTION.

USE IF KGEOM = 1

(A) A SINGLE CARD CONTAINING THE WORD FORMAT IN COLUMNS 1-6 AND
    A VARIABLE FORMAT CORRESPONDING TO THE FORMAT OF THE GRID
    POINT CARDS WITH LEFT PARENTHESIS STARTING IN COLUMN 11
    AND UP TO 80 COLUMNS MAY BE USED.
(B) DECK OF GRID POINT CARDS. EACH CARD CONTAINS 4 VALUES, GRID
    POINT NUMBER (INTEGER), X-COORDINATE (REAL), Y-COORDINATE
    (REAL) AND Z-COORDINATE (REAL). THE FORMAT IS SPECIFIED
    IN (A) ABOVE.
(C) A SINGLE CARD CONTAINING THE WORD ENDDGRID IN COLUMNS 1-7.
(D) A SINGLE CARD CONTAINING THE WORD FORMAT IN COLUMNS 1-6 AND
    A VARIABLE FORMAT CORRESPONDING TO THE FORMAT OF THE
    ELEMENT CARDS WITH LEFT PARENTHESIS STARTING IN COLUMN 11
    AND UP TO 80 COLUMNS MAY BE USED.
(E) DECK OF ELEMENT CARDS. EACH CARD CONTAINS 5 INTEGER FIELDS
    WHICH ARE THE ELEMENT NUMBER, AND GRID POINT NUMBERS AT THE
    VERTICES OF THE ELEMENTS. FOR TRIANGULAR ELEMENTS THE
    LAST INTEGER FIELD MUST BE BLANK OR ZERO. FOR ROD OR BEAM
    ELEMENTS THE LAST TWO INTEGER FIELDS MUST BE BLANK OR ZERO.
    THE FORMAT IS SPECIFIED IN (D) ABOVE.
(F) A SINGLE CARD CONTAINING THE WORD ENDEOM IN COLUMNS 1-7.

USE IF KGEOM = 2

(A) A SINGLE CARD CONTAINING THE WORD LINEEL IN COLUMNS 1-6 AND
    UP TO NINE NASTRAN LINEAL ELEMENT CONNECTION NAMES,
    WHICH ARE LEFT-ADJUSTED IN FIELD WIDTHS OF 8, STARTING IN
    COLUMN 9 (COLS. 9-16, 17-24, ..., 73-80). THIS CARD CAN
    BE OMITTED IF LINEAL ELEMENTS ARE NOT USED FOR THE PLOT.
(B) A SINGLE CARD CONTAINING THE WORD TRIAEL IN COLUMNS 1-6 AND
    UP TO NINE NASTRAN TRIANGULAR ELEMENT CONNECTION NAMES,
    WHICH ARE LEFT-ADJUSTED IN FIELD WIDTHS OF 8, STARTING IN
    COLUMN 9 (COLS. 9-16, 17-24, ..., 73-80). THIS CARD CAN
    BE OMITTED IF TRIANGULAR ELEMENTS ARE NOT USED FOR THE PLOT.
APPENDIX B – Continued

(C) A single card containing the word QUADEL in columns 1-6 and
up to nine NASTRAN QUADRILATERAL connection names, which
are left-adjusted in field widths of 8, starting in col. 9
(cols. 9-16, 17-24, ..., 73-80). This card can be omitted
if quadrilateral elements are not used for the plot.

(D) A NASTRAN BULK DATA DECK. Only the grid cards and the
element connection cards with names matching those given on
the LINEEL, TRIAEL, and QUADEL cards will be used for the
plot. All other cards in the NASTRAN BULK DATA DECK will
be ignored.

(E) A single card containing the word ENDEGEOM in columns 1-7.

USE IF KGEOM = 9

CALL SUBROUTINE GEOM9 which is prepared by the user to read
GEOMETRY DATA.

CASE IDENTIFICATION CARD.

This card is omitted if IDCASE=0 is specified in $OPTION.
If present, this card contains any desired alphanumeric
information in cols. 1-80. Will appear before each data plot.

DATA TO BE PLotted IS NOW INPUT IN ONE OF THE FOLLOWING FORMS,
DEPENDING ON THE VALUE OF KDATA SPECIFIED IN NAMELIST OPTION.

USE IF KDATA = 1

(A) A single card containing the word FORMAT in columns 1-6 and
a variable format for the data cards with left parenthesis
starting in column 11 and up to 80 columns may be used. If
displacements are included for more than one grid point per
card, the number of grid points per card must be entered
as an integer in column 8.

(B) Deck of displacement sets. There can be multiple displace-
ment sets per card or the set can extend to more than one
card (often the case with NASTRAN punched output) which can
be handled with a format for reading multiple cards.
A displacement set for each grid point is defined to
contain from 2 to 4 values, a grid point number and
displacements corresponding to NUDISP, NVDISP, or NWDISP
equal to 1.

(C) Blank card or cards to end data deck. The number of blank
cards must correspond to the number of cards read at one
time by the specified variable format.

USE IF KDATA = 5

READS NVALUES (FROM NAMELIST OPTION) DISPLACEMENT SETS FROM
TAPE20. Each displacement set must have been written on TAPE20
as an unformatted record.
APPENDIX B - Continued

USE IF KDATA = 9

CALL SUBROUTINE DATA9 WHICH IS PREPARED BY THE USER TO READ
DISPLACEMENT DATA.

NAMELIST PICT - CONTAINS VALUES NEEDED TO GENERATE PLOTS.

THE FOLLOWING VALUES ARE INCLUDED---

KXURZ = INTEGER DESIGNATING HORIZONTAL AXIS OF VIEWING PLANE,
WHERE 1=X, 2=Y, 3=Z.
** DEFAULT = 1 **

KVERT = INTEGER DESIGNATING VERTICAL AXIS OF VIEWING PLANE,
WHERE 1=X, 2=Y, 3=Z.
** DEFAULT = 2 **

PHI = ANGULAR ROTATION OF MODEL ABOUT ITS X-AXIS, IN DEGREES
(MUST BE TAKEN THIRD).
** DEFAULT = 0.0 **

THETA = ANGULAR ROTATION OF MODEL ABOUT ITS Y-AXIS, IN DEGREES
(MUST BE TAKEN SECOND).
** DEFAULT = 0.0 **

PSI = ANGULAR ROTATION OF MODEL ABOUT ITS Z-AXIS, IN DEGREES
(MUST BE TAKEN FIRST).
** DEFAULT = 0.0 **

NEWFR = 1 FOR FRAME CHANGE BEFORE PLOT IS MADE.
(A FRAME CHANGE RESETS THE X-ORIGIN PAST PREVIOUS PLOT
BY XSPACE AND THE Y-ORIGIN AT 0.0).
NEWFR = 1 FOR NO FRAME CHANGE BEFORE PLOTTING.
** DEFAULT = 1 **

ISCALE = 1 FOR INTERNAL ORIGIN LOCATION AND SCALING.
= 2 FOR USER SPECIFIED ORIGIN AND SCALING.
** DEFAULT = 1 **

PLTSZ = MAXIMUM DIMENSION DESIRED ON COMPLETED PLOT.
(USED FOR SCALING IF ISCALE = 1)
** DEFAULT = 10.0 **

XORGN = X-LOCATION OF PLOT ORIGIN (USED IF ISCALE = 2).
** DEFAULT = 0.0 **

YOURGN = Y-LOCATION OF PLOT ORIGIN (USED IF ISCALE = 2).
** DEFAULT = 0.0 **

PScale = MODEL SIZE REDUCTION FACTOR, PScale = ACTUAL MODEL
SIZE/DESIRED PLOT SIZE (USED IF ISCALE = 2).
** DEFAULT = 1.0 **

NJAT = 0 FOR NO NUMBERING ON PLOTS.
= 1 FOR NUMBERING OF GRID POINTS.
= 2 FOR NUMBERING OF ELEMENTS.
** DEFAULT = 0 **

XLHT = HEIGHT OF INTEGERS SPECIFIED BY NOTAT, IN INCHES.
** DEFAULT = 0.15 **

KDISP = 0 FOR UNDEFORMED PLOT.
= 1 FOR DEFORMED PLOT.
= 2 FOR EXPLODED PLOT.
= 3 FOR DISPLACEMENTS REPRESENTED BY VECTORS.
** DEFAULT = 0 **

IDMAG = 1 FOR DIRECT SCALING OF DATA BY DMAGS.
= 2 FOR SCALING OF DATA TO A MAX. VALUE OF DMAGS.
** DEFAULT = 2 **
APPENDIX B – Continued

C DMAGS = MAGNIFICATION OF DISPLACEMENTS (IF KDISP=1).
C = REDUCTION FACTOR OF ELEMENTS (IF KDISP=2).
C ** DEFAULT = 1.0 **
C KSYMXY = 1 FOR SYMMETRY ABOUT X-Y PLANE.
C ** DEFAULT = 0 **
C KSYMxz = 1 FOR SYMMETRY ABOUT X-Z PLANE.
C ** DEFAULT = 0 **
C KSYMyz = 1 FOR SYMMETRY ABOUT Y-Z PLANE.
C ** DEFAULT = 0 **
C XXMAX, YYMAX, ZZMAX, XXMIN, YYMIN, ZZMIN LOCATE CUTTING PLANES
C PARALLEL TO PRINCIPAL (X-Y, X-Z, Y-Z) PLANES
C TO LIMIT PLOT.
C ** DEFAULT XXMAX=YYMAX=ZZMAX=1.0E+20 **
C ** DEFAULT XXMIN=YYMIN=ZZMIN=-1.0E+20 **
C NUMAX = MAXIMUM GRID PT. TO BE INCLUDED IN PLOT.
C ** DEFAULT = 9999999999 **
C NDMIN = MINIMUM GRID PT. TO BE INCLUDED IN PLOT.
C ** DEFAULT = 0 **
C NELMAX = MAXIMUM ELEMENT NUMBER TO BE INCLUDED IN PLOT.
C ** DEFAULT = 9999999999 **
C NELMIN = MINIMUM ELEMENT NUMBER TO BE INCLUDED IN PLOT.
C ** DEFAULT = 0 **
C KODE = 0, 1, 2, 3, LAST READ CASE IDENTIFICATION CARD IF PRESENT.
C = 1, READ ANOTHER NAMELIST PICT.
C = 2, READ A NEW SET OF DISPLACEMENT DATA, INCLUDING A
C CASE IDENTIFICATION CARD IF PRESENT.
C = 3, READ A COMPLETE NEW SET OF INPUT DATA,
C INCLUDING A TITLE CARD.
C ** DEFAULT = 0 **

C THE ABOVE COMPRISRE A COMPLETE BASIC SET OF INPUT DATA IF
C KODE = 0 IN $PICT. FOR KODE = 1, 2, OR 3, ADDITIONAL SECTIONS OF
C THE BASIC DECK MUST BE REPEATED. THE DECK MUST END WITH
C NAMELIST $PICT HAVING KODE = 0.

C**********************************************************************************

C DESCRIPTION OF GRAPHICS SUBROUTINES

C GRAPHICS SUBROUTINE USED BY FOLLOWING PLOTTING DEVICE CALLED BY FOLLOWING
C PROGRAM SUBROUTINES
C
C CALCUMP CALCUMP INITIAL
C LEKROY CALCUMP INITIAL
C PSEUDO VARIAN INITIAL
C CDC250 CRT INITIAL

38
APPENDIX B – Continued

SUBROUTINE CALCOMP

PURPOSE
THIS IS THE NORMAL MODE PROCESSOR. THE NECESSARY
PARAMETERS AND LINKAGE ARE SET UP TO OUTPUT A TAPE
FOR THE CALCOMP 780/763 0.010/0.005-INCH STEP PLOTTER.

USE
CALL CALCCMP

COMMENTS
THIS CALL MUST BE GIVEN BEFORE THE FIRST CALL TO A
PLOTTING ROUTINE.

SUBROUTINE LEROY

PURPOSE
THE PARAMETERS NECESSARY TO ACCOMODATE PLOTTING WITH THE
LIQUID INK PEN ARE SET UP BY CALL LEROY.

USE
CALL LEROY

COMMENTS
THIS CALL SHOULD ONLY BE USED WITH THE CALCOMP PROCESSOR.
IN ADDITION TO REDUCING THE SPEED OF THE PLOTTER FOR
ALL PLOTTING MOVEMENTS, THE NUMBER OF PLOT VECTORS IN ANY
ANNOTATION IS CONSIDERABLY INCREASED.

SUBROUTINE PSEUDO

PURPOSE
INITIALIZES PLOT VECTOR FILE FOR VARIAN PLOTTER

USE
CALL PSEUDO

SUBROUTINE CDC250

PURPOSE
INITIALIZES CATHODE RAY TUBE CONSOLE.

USE
CALL CDC250
SUBROUTINE CALPLT

PURPOSE
TO MOVE THE PLOTTER PEN TO A NEW LOCATION WITH PEN UP
OF DOWN AND TO SIGNAL THE END OF A JOB SEGMENT BY
INCREMENTING THE BLOCK ADDRESS NUMBER.

USE
CALL CALPLT(X,Y,IPEN)

WHERE
X,Y ARE THE FLOATING POINT VALUES FOR PEN MOVEMENT.

IPEN = 2 PEN DOWN
    = 3 PEN UP

NEGATIVE IPEN WILL ASSIGN X=0, Y=0
AS THE LOCATION OF THE PEN AFTER MOVING THE
X,Y (CREATE A NEW REFERENCE POINT) AND
INCREMENT THE BLOCK NUMBER BY ONE.

COMMENTS
ALL X AND Y COORDINATES MUST BE EXPRESSED AS FLOATING
POINT INCHES (ACTUAL PAGE DIMENSIONS) IN DEFLECTION FROM
THE ORIGIN.

SUBROUTINE NOTATE

PURPOSE
TO DRAW ALPHANUMERIC INFORMATION FOR ANNOTATION AND LABELING
AND PROVIDE SPECIAL CENTERED SYMBOLS FOR ANNOTATION OF
DATA POINTS.

USE
CALL NOTATE(X,Y,HEIGHT,BCD,THETA,N)

WHERE
X,Y ARE THE FLOATING POINT PAGE COORDINATES OF THE
FIRST CHARACTER. FOR ALPHANUMERIC CHARACTERS
THE COORDINATES OF THE LOWER LEFT-HAND CORNER
OF THE CHARACTERS ARE SPECIFIED.

HEIGHT SPECIFIES CHARACTER SIZE AND SPACING IN FLOATING
POINT INCHES FOR A FULL-SIZE CHARACTER. THE
WIDTH OF A CHARACTER WILL BE (4/7)*HEIGHT AND THE
SPACE BETWEEN CHARACTERS IS (2/7)*HEIGHT.

BCD IS THE STRING OF ALPHANUMERIC CHARACTERS TO BE
DRAWN.

THETA IS THE ANGLE IN FLOATING POINT DEGREES AT WHICH
THE INFORMATION IS TO BE DRAWN.

N IS THE NUMBER OF CHARACTERS, INCLUDING BLANKS, IN
THE LABEL.
APPENDIX B – Continued

SUBROUTINE NUMBER

PURPOSE
TU CONVERT A FLOATING NUMBER TO BCD (EXPRESSED IN F FORMAT), AND DRAW THE RESULTING APLHANUMERIC CHARACTERS.

USE
CALL NUMBER(X,Y,SIZE,FPN,THETA,N)

WHERE
X,Y ARE THE COORDINATES IN FLOATING POINT INCHES OF THE LEFT LOWER CORNER OF THE FIRST DIGIT OF OUTPUT.
SIZE IS THE HEIGHT OF THE PLOTTED NUMBER IN FLOATING POINT INCHES.
FPN IS THE FLOATING POINT NUMBER TO BE DRAWN.
THETA IS THE ANGLE IN FLOATING POINT DEGREES AT WHICH THE NUMBER IS TO BE DRAWN.
N IS THE NUMBER OF DECIMAL DIGITS TO THE RIGHT OF THE DECIMAL POINT FOR OUTPUT.
N = -1 AND N = 0 BOTH SPECIFY NO DECIMAL PLACES, HOWEVER, -1 SUPPRESSES THE DECIMAL POINT.

COMMENTS
THE NUMBER IS RESTRICTED TO A MAXIMUM OF 12 DIGITS.
The routine truncates the floating point number at the required decimal place.

SUBROUTINE NFRAME

PURPOSE
USED BY VARIAN PLOTTER TO ADVANCE PLOTTING FRAME.

USE
CALL NFRAME

SUBROUTINE NEXT

PURPOSE
PROVIDES A BREAK POINT OR HALT DURING APPLICATION PROGRAM EXECUTION. OPERATOR MUST PRESS FUNCTION KEY TO RESUME, AND NUMBER OF KEY IS RETURNED IN CALLING PARAMETER.

USE
CALL NEXT(NKEY)

WHERE
NKEY IS NUMBER OF FUNCTION KEY PRESSED.
APPENDIX B – Continued

SUBROUTINE MESSAGE

PURPOSE PROVIDES THE CAPABILITY TO DISPLAY A MESSAGE ON THE CRT250.

USE CALL MESSAGE(I,BCD,N)

WHERE

I INDICATES INTENSITY OF CHARACTER DISPLAY.

BCD IS ADDRESS OF ARRAY CONTAINING THE MESSAGE IN HOLLERITH FORM.

N IS THE NUMBER OF CHARACTERS IN THE MESSAGE (LESS THAN 50)

SUBROUTINE PARAMS

PURPOSE USED TO GENERATE A TABLE OF SYMBOLIC NAMES THAT CAN BE

ACCESSSED USING THE ALPHANUMERIC KEYBOARD ON THE CRT250.

USE CALL PARAMS(BCD,VAR)

WHERE

BCD IS THE HOLLERITH REPRESENTATION OF SYMBOLIC NAME.

VAR IS PROGRAM VARIABLE REFERRED TO BE SYMBOLIC NAME.

COMMENTS UP TO 3 PAIRS OF VARIABLES MAY BE SPECIFIED IN A SINGLE

CALL TO PARAMS. TABLE HAS CAPACITY FOR 42 PAIRS.

SUBROUTINE KFORMAT

PURPOSE ALLOWS PROGRAMMER TO CHANGE FORMAT FOR KEYBOARD INPUT.

USE CALL KFORMAT(NHBCD)

WHERE

N IS THE NUMBER OF CHARACTERS IN BCD.

H IS REQUIRED.

BCD IS THE REQUIRED FORMAT (I4,F4.2,A10,ETC.)

RETURN

END OF DOCMNT
SUBROUTINE CCRTL

C *** FOR CHANGING VALUES INPUT BY $PICT USING CRT.

C

C

C

COMMON/CONTRL/ KGEOM, KDATA, KPLT, KSYMXY, KSYMZX, KSYMZY, NOTAT, XLHT,
1KHORZ, KVERT, PHI, THETA, PSI, NEWFR, ISCALE, PLOTSZ, XORGN, YORGN,
2PSCALE, KDISP, IDMAG, IDORD

COMMON/LIMITS/ XXMAX, YYMAX, ZZMAX, XXMIN, YYMIN, ZZMIN, NDMAX, NDMIN,
1NELMAX, NELMIN

COMMON/SAVEV/ DMAGS, IDMAG

C *** FOR INTEGER CONTROL VALUES

C

CALL KFORMAT(3H110)
CALL PARAMS
CALL PARAMS(5LKHORZ, KHORZ, 5LKVERT, KVERT)
CALL PARAMS(6LNEWFR, NEWFR, 6LISCALE, ISCALE)
CALL PARAMS(5LNOTAT, NOTAT, 5LKDISP, KDISP, 5LIDMAG, IDMAG)
CALL PARAMS(6LKSYMX, KSYMXY, 6LKSYMZX, KSYMZX, 6LKSYMYZ, KSYMYZ)
CALL PARAMS(5LNDMAX, NDMAX, 5LNDMIN, NDMIN)
CALL PARAMS(6LNELMAX, NELMAX, 6LNELMIN, NELMIN)
CALL MESSAGE(1, 32HTO CHANGE INTEGER CONTROL VALUES, 32)
CALL MESSAGE(1, 13VARIABLES ARE, 13)
CALL MESSAGE(1, 27HKHORZ, KVERT, NEWFR, ISCALE, 27)
CALL MESSAGE(1, 19HNOTAT, KDISP, IDMAG, 19)
CALL MESSAGE(1, 22HKSYMXY, KSYMXY, 22)
CALL MESSAGE(1, 28HNDMAX, NDMAX, NDMIN, NELMIN, 28)
CALL MESSAGE(1, 17HANY KEY CONTINUES, 17)
CALL NEXT(NKEY)

C *** FOR FLOATING POINT CONTROL VALUES

C

CALL KFORMAT(5HF10.3)
CALL PARAMS
CALL PARAMS(3LYAW, PSI, 4LROLL, PHI, 5LPITCH, THETA)
CALL PARAMS(6LPLOTSZ, PLOTSZ, 6LPSCALE, PSCALE)
CALL PARAMS(5LXORGN, XORGN, 5LYORGN, YORGN)
CALL PARAMS(5LDMAGS, DMAGS)
CALL PARAMS(4LXLHT, XLHT)
CALL PARAMS(5LXXMAX, XXMAX, 5LXXMIN, XXMIN)
CALL PARAMS(5LYYMAX, YYMAX, 5LYYMIN, YYMIN)
CALL PARAMS(5LZZMAX, ZZMAX, 5LZZMIN, ZZMIN)
CALL MESSAGE(1, 39HTO CHANGE FLOATING POINT CONTROL VALUES, 39)
CALL MESSAGE(1, 13VARIABLES ARE, 13)
CALL MESSAGE(1, 32HYAW, ROLL, PITCH, PLOTSZ, PSCALE, 32)
CALL MESSAGE(1, 25HXORGN, YORGN, DMAGS, XLHT, 25)
CALL MESSAGE(1, 19HXXMAX, YYAX, ZZMAX, 19)
CALL MESSAGE(1, 19HXXMIN, YYMIN, ZZMIN, 19)
CALL MESSAGE(1, 17HANY KEY CONTINUES, 17)
CALL NEXT(NKEY)
RETURN

END OF CCRTL
APPENDIX B – Continued

SUBROUTINE CCRT2

*** FOR SELECTING CONTROL OPTION, KODE, AT END OF JOB USING CRT.

COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMZX,KSYMZY,NOTAT,XLHT,
IKHORZ,KVERT,PHI,THETA,PSI,NEWFR,LSCALE,PLOTSZ,XORGN,YORGN,
2PSCALE,KDISP,DMAG,KODE

10 CONTINUE
CALL MESSAGEC(1,1H,1)
CALL MESSAGEC(1,32HFN KEY 12 TO ALTER EXISTING PLOT,32)
CALL MESSAGEC(1,42HFN KEY 24 TO READ NEW SET OF DISPLACEMENTS,42)
CALL MESSAGEC(1,37HFN KEY 36 TO READ A COMPLETE NEW CASE,37)
CALL MESSAGEC(1,18HFN KEY 48 ENDS JOB,18)
CALL NEXT(NKEY)
KODE = 10
IF(NKEY.EQ.48) KODE = 0
IF(NKEY.EQ.12) KODE = 1
IF(NKEY.EQ.24) KODE = 2
IF(NKEY.EQ.36) KODE = 3
IF(KODE.EQ.10) GO TO 10
RETURN
END OF CCRT2

SUBROUTINE CCRT3

*** REMINDER TO PUT EOF ON PLOT FILE WHEN USING CRT.

CALL MESSAGEC(1,1H,1)
CALL MESSAGEC(1,45HLAST REMINDER TO PUT EOF ON PLOT FILE, IF ANY,45)
CALL MESSAGEC(1,1H,1)
CALL MESSAGEC(1,40HDO IT AT NEXT PLOT FILE COMPLETE MESSAGE,40)
CALL MESSAGEC(1,1H,1)
CALL MESSAGEC(1,17HANY KEY CONTINUES,17)
CALL NEXT(NKEY)
CALL CALPLT(12.0,0.0,-3)
RETURN
END OF CCRT3

SUBROUTINE PSTOP

*** TO TERMINATE JOB.

COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMZX,KSYMZY,NOTAT,XLHT,
IKHORZ,KVERT,PHI,THETA,PSI,NEWFR,LSCALE,PLOTSZ,XORGN,YORGN,
2PSCALE,KDISP,DMAG,KODE
CALL CALPLT(0.0,0.0,999)
IF(KPLOT.EQ.4) CALL CCRT3
STOP
END OF PSTOP
SUBROUTINE INITAL

*** TO SET UP VALUES FOR CONTROL PARAMETERS

COMMON/CONTRL/KGECM,KDATA,KPLOT,KSYMXY,KSYMZX,KSYMZY,NOTAT, XLHT,
IKHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
2PScale,KDISP,UMAG,KIODE
COMMON/LIMITS/XXMAX,YYMAX,ZZMAX,XXMIN,YYMIN,ZZMIN,NDMAX,NDMIN,
INELMAX,NELMIN
COMMON/CORGN/XCABS,YOABS,XPMAX,XSPACE,PSIZE
COMMON/SAVEV/DMAGS,UMAG
COMMON/SEQUENC/NNODE,NNODEST,NUDISP,NVDISP,NWDISP
COMMON/SEQUENC/IREDON
COMMON/VALUES/NVALUES
COMMON/CASEII/IOCASE
COMMON/VALUES/IVALUES
COMMON/VALUES/IVALUES
COMMON/VALUES/IVALUES

*** DESCRIPTION OF VALUES IN iOPTION GIVEN IN SUBROUTINE DOCMNT

*** TO SET DEFAULT VALUES FOR $OPTION

NNODEST = 200
NUDISP = 0
NVDISP = 0
NWDISP = 0
KGECM = 1
KDATA = 1
NVALUES = 0
IRESEQ = 1
KPLOT = 1
XSPACE = 10.0
PSIZE = 25.0
IOCASE = 0

*** TO SET DEFAULT VALUES FOR $PICT

KUHRZ = 1
KVERT = 2
PHI = 0.0
THETA = 0.0
PSI = 0.0
NEWFR = 1
ISCALE = 1
PLOTSZ = 10.0
XORGN = 0.0
YORGN = 0.0
PSIZE = 1.0
NOTAT = 0
XLHT = 0.15
KDISP = 0
IDMAIG = 2
DMAGS = 1.0
KSYMXY = 0
KSYMZX = 0
KSYMZY = 0
XXMAX = 1.0E20
APPENDIX B – Continued

YYMAX = 1.0E20
ZZMAX = 1.0E20
XXMIN = -1.0E20
YYMIN = -1.0E20
ZZMIN = -1.0E20
NOMAX = 9999999999
NDMIN = 0
NELMAX = 9999999999
NELMIN = 0
KODE = 0
READ(5,OPTION)
IF(.LOF,5) 100,200
100 CONTINUE
CALL PSTOP
200 CONTINUE
IF(KPLOT.LE.2) CALL CALCOMP
IF(KPLOT.EQ.2) CALL LEROY
IF(KPLOT.EQ.3) CALL PSEUDO
IF(KPLOT.EQ.4) CALL CDC250
WRITE(6,OPTION)
RETURN
END OF INITIAL
SUBROUTINE GEQMi(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)

*** TO READ GRID POINT INFORMATION AND ELEMENTS

COMMON/CTRL/ KGEOM,KDATA,KPLOT,KSXY,XSYMZ,SYMZ,SYMXY,NOTAT,XLHT,
1KHORZ,KERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
2PSCALE,XISP,YISP,XXZ,XYM,XYL
COMMON/KOONT/ NNOOE,NNDEST,NODISP,NVDISP,NWDISP
COMMON/SEQ/ ISEQ
DIMENSION NUMPT(l),XPT(l),YPT(l)
ZPT(1),UPT(1),VPT(1),WPT(1)
DIMENSION ABCD(8)
DIMENSION FORMT(7)
READ(5,10) ABCD
10 FORMAT(dAlO)
IF(ABCD(l).EQ.TESTl) GO TO 1000
IF(ABCD(l).EQ.TEST4) 50,60
50 CONTINUE
DECODE(lO,55,ABCD) FORMT
GO TO 100
60 CONTINUE
NNOOE = NNOOE-1
DECODE(lO,FORMT,ABCD) NUMPT(NNOOE),XPT(NNOOE),YPT(NNOOE)
ZPT(NNOOE)
GO TO 100
1000 CONTINUE

*** TO READ GRID INFORMATION

CONTINUE
READ(5,10) ABCD
10 FORMAT(3A10)
IF(ABCD(l).EQ.TESTl) GO TO 1000
IF(ABCD(l).EQ.TEST4) 50,60
50 CONTINUE
DECODE(lO,55,ABCD) FORMT
GO TO 100
60 CONTINUE
END FILE
GO TO 2000
285 CONTINUE
IF(ABCD(l).EQ.TEST4) 250,260
250 CONTINUE
DECODE(lO,55,ABCD) FORMT
GO TO 200
260 CONTINUE
IOM = 1
DECODE(lO,FORMT,ABCD) NUME1,NODE2,NODE3,NODE4
IF(ISEQ.NE.1) GO TO 700

CONTINUE
READ(5,10) ABCD
10 FORMAT(3A10)
IF(ABCD(l).EQ.TESTl) GO TO 1000
IF(ABCD(l).EQ.TEST4) 50,60
50 CONTINUE
DECODE(lO,55,ABCD) FORMT
GO TO 100
60 CONTINUE
END FILE
GO TO 2000
C *** TO RENUMBER ELEMENT NODES
C

NODE(1) = NODE1
NODE(2) = NODE2
NODE(3) = NODE3
NODE(4) = NODE4
DO 500 I=1,4
   IF(NODE(I).EQ.0) GO TO 550
   DO 510 J=1,NODE
      IF(NODE(I).EQ.NUMPT(J)) 511,510
   510 CONTINUE
  511 NODE(I) = J
   GO TO 500
  500 CONTINUE
  550 CONTINUE

NODE1 = NODE(1)
NODE2 = NODE(2)
NODE3 = NODE(3)
NODE4 = NODE(4)

CALL RECOUT(10,1,0,NUMEL,NODE1,NODE2,NODE3,NODE4)
GO TO 200

2000 CONTINUE
RETURN
END OF GEOM1
APPENDIX B — Continued

SUBROUTINE GEUM2(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)

C *** FOR INPUT OF NASTRAN DECK TO DESCRIBE GEOMETRY.

COMMON/CONTRL/ KGECM,KDATA,KPLOT,KSYMXY,KSYMXY,KSYMZY,KSYMZY,NOTAT,XLHT,
1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
2PSCALE,KDISP,DMAG,ISCALE
COMMON/SEQUENCE/ IRESEQ
COMMON/KOUNT/ NNOOE,NNOEST,NUDI5P,NVDISP,NWDISP
DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
DIMENSION ABCD(8)
DIMENSION NODE(4)
DIMENSION LINEEL(9),TRIAEL(9),QUADEL(9)
10 FORMAT(8A10)
REWIND 9
TEST0 = 10H
TEST1 = 10HGRID
TEST2 = 10HLINEEL
TEST3 = 10HTRIAEL
TEST4 = 10HQUADEL
TEST5 = 10HENDGECM
DO 15 I=1,9
LINEEL(I) = 10H
TRIAEL(I) = 10H
QUADEL(I) = 10H
15 CONTINUE
800 CONTINUE
READ(5,10) ABCD
DECODE (80,50,ABCD) WORD1
50 FORMAT(8A)
IF(WORD1.EQ.TEST0) GO TO 800
IF(WORD1.EQ.TEST1) GO TO 100
IF(WORD1.EQ.TEST2) 60,61
60 DECODE(80,111,ABCD) NUMPT(NNODE),XPT(NNODE),YPT(NNODE)
111 FORMAT(6X,9A8)
GO TO 800
61 CONTINUE
IF(WORD1.EQ.TEST3) 62,63
62 DECODE(80,111,ABCD) TRIAEL(I),I=1,9
GO TO 800
63 CONTINUE
IF(WORD1.EQ.TEST4) 64,65
64 DECODE(80,111,ABCD) QUADEL(I),I=1,9
GO TO 800
65 CONTINUE
DO 70 I=1,9
IF(WORD1.EQ.LINEEL(I)) GO TO 200
IF(WORD1.EQ.TRIAEL(I)) GO TO 300
IF(WORD1.EQ.QUADEL(I)) GO TO 400
70 CONTINUE
IF(WORD1.EQ.TEST5) GO TO 2000
GO TO 800

C *** TO READ GRID CARDS

100 CONTINUE
NNOOE = NNODE+1
DECODE(80,101,ABCD) NUMPT(NNOOE),XPT(NNODE),YPT(NNODE),ZPT(NNODE)
GO TO 100
APPENDIX B – Continued

101 FORMAT (8X, A8, 8X, 3F8.0)
   CALL IRITE(NUMPT(NNODE))
   GO TO 800

C *** TO READ CARDS CONTAINING ELEMENTS WITH 2 GRID POINTS
C
200 CONTINUE
   TEST100 = 10*HPLUTEL
   IF (WUKD1.EQ.TEST100) GO TO 250
   DECODE(80,201,ABCD) NEL,ND1,ND2
201 FORMAT (8X,A8,8X,2A8)
   CALL IRITE(NEL)
   CALL IRITE(ND1)
   CALL IRITE(ND2)
   ND3 = 0
   ND4 = 0
   CALL RECOUT(9,1,0,NEL,ND1,ND2,ND3,ND4)
   GO TO 800

C *** TO READ PLUTEL CARDS
C
250 CONTINUE
   DECODE(80,251,ABCD) NEL,ND1,ND2
251 FORMAT (8X,3A8)
   CALL IRITE(NEL)
   CALL IRITE(ND1)
   CALL IRITE(ND2)
   ND3 = 0
   ND4 = 0
   CALL RECOUT(9,1,0,NEL,ND1,ND2,ND3,ND4)
   GO TO 800

C *** TO READ CARDS CONTAINING ELEMENTS WITH 3 GRID POINTS
C
300 CONTINUE
   DECODE(80,301,ABCD) NEL,ND1,ND2,ND3
301 FORMAT (8X,A6,5X,3A8)
   CALL IRITE(NEL)
   CALL IRITE(ND1)
   CALL IRITE(ND2)
   CALL IRITE(ND3)
   ND4 = 0
   CALL RECOUT(9,1,0,NEL,ND1,ND2,ND3,ND4)
   GO TO 800

C *** TO READ CARDS CONTAINING ELEMENTS WITH 4 GRID POINTS
C
400 CONTINUE
   DECODE(80,401,ABCD) NEL,ND1,ND2,ND3,ND4
401 FORMAT (8X,A6,8X,4A8)
   CALL IRITE(NEL)
   CALL IRITE(ND1)
   CALL IRITE(ND2)
   CALL IRITE(ND3)
   CALL IRITE(ND4)
   CALL RECOUT(9,1,0,NEL,ND1,ND2,ND3,ND4)
   GO TO 800

2000 CONTINUE
   END FILE 9
C *** TO RENUMBER ELEMENT NODES AND WRITE ON TAPE 10
C
REWIND 9
REWIND 10
600 CONTINUE
   CALL RECIN(9,1,5,NUMEL,NODE1,NODE2,NODE3,NODE4)
   IF(EOF,9) 450,451
451 CONTINUE
   IF(IRESEQ.NE.1) GO TO 700
   NODE(1) = NODE1
   NODE(2) = NODE2
   NODE(3) = NODE3
   NODE(4) = NODE4
   DO 500 I=1,4
   IF(NODE(I).EQ.0) GO TO 550
   DO 510 J=1,NNODE
   IF(NODE(I).EQ.NUMPT(J)) 511,510
511 CONTINUE
   NODE(1) = J
   GO TO 500
510 CONTINUE
500 CONTINUE
550 CONTINUE
   NODE1 = NODE(1)
   NODE2 = NODE(2)
   NODE3 = NODE(3)
   NODE4 = NODE(4)
700 CONTINUE
   CALL RECOUT(10,1,0,NUMEL,NODE1,NODE2,NODE3,NODE4)
   GO TO 600
450 CONTINUE
END FILE 10
RETURN
END OF GEOM2
SUBROUTINE IRITE(NUM)

C
C *** TO RIGHT ADJUST INTEGERS IN A FIELD WIDTH OF EIGHT
C
DIMENSION N(8)
LANK = 1H
DISABLE(8,1,NUM) NSAVE
1 FORMAT(18)
DISABLE(8,2,NUM) (N(I), I = 1, 8)
2 FORMAT(8A1)
DO LO I = 1, 8
II = 9-I
IF (N(II), NE, LANK) GO TO 20
10 CONTINUE
20 NUM = NSAVE/(10**(8-II))
RETURN
END OF IRITE

SUBROUTINE GEOM9(NUMPT, XPT, YPT, ZPT, UPT, VPT, WPT)

C
C *** USER SUPPLIED GEOMETRY INPUT SUBROUTINE.
C
COMMON/CONTRL/KGEOM,KDATA,KPLOT,KSYMXY,KSYMZX,KSYMYZ,NOTAT,XLHT,
1KHORZ,KVERT PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
2PSCALE,KDISP,DMAG,KODE
COMMON/KOUNT/NNOUE,NNDEST,NUDISP,NVDISP,NWDISP
DIMENSION NUMPT(J), XPT(I), YPT(I), ZPT(I), UPT(I), VPT(I), WPT(I)

C
C *** INSERT ROUTINE HERE
C
RETURN
END OF GEOM9
APPENDIX B – Continued

SUBROUTINE BOUND(NUMPT, XPT, YPT, ZPT, UPT, VPT, WPT)

*** TO DETERMINE MAXIMUM DIMENSIONAL LIMITS OF BODY FOR USE IN SCALING PLOTS

COMMON/CCTRL/, KGEOM, KDATA, KPLOT, KSYMXY, KSYMXYZ, NOTAT, XLHT,
KHOKZ, KVERT, PHI, THETA, PSI, NEWFR, ISCALE, PLOTSZ, XORGN, YORGN,
2PSCALE, KDISP, DMAG, KODE
COMMON/LIMITS/, XXMAX, YYMAX, ZZMAX, XXMIN, YYMIN, ZZMIN, NDMAX, NDMIN,
INELMAX, INELMIN
COMMON/XYZUM/, XYZMAX(3), XYZMIN(3)
COMMON/KOUNT/, NNCDE, NNDEST, NUDISP, NVDISP, NWDISP
DIMENSION NUMPT(1), XPT(1), YPT(1), ZPT(1), UPT(1), VPT(1), WPT(1)
DIMENSION NODE(4)
DO 1 1=1,3
XYZMIN(I) = +1.0E20
XYZMAX(I) = -1.0E20
5 CONTINUE
RCWIND 10
100 CONTINUE
CALL RECIN(10, 1, 5, NUMEL, NODE1, NODE2, NODE3, NODE4)
IF(EOF, 10) 1000, 200
200 CONTINUE
IF(NUMEL .LT. NELMIN .OR. NUMEL .GT. NELMAX) GO TO 100
NODE(1) = NODE1
NODE(2) = NODE2
NODE(3) = NODE3
NODE(4) = NODE4
DO 10 1=1,4
ND = NODE(1)
IF(NODE(1).EQ.0) GO TO 15
IF(NUMPT(ND).LT.NDMIN .OR. NUMPT(ND) .GT. NDMAX) GO TO 100
10 CONTINUE
15 CONTINUE
DO 20 1=1,4
IF(NODE(1).EQ.0) GO TO 25
ND = NODE(1)
IF(XPT(ND) .LT. XXMAX) GO TO 20
IF(XPT(ND) .LT. XXMIN) GO TO 20
IF(YPT(ND) .LT. YYMAX) GO TO 20
IF(YPT(ND) .LT. YYMIN) GO TO 20
IF(ZPT(ND) .LT. ZZMAX) GO TO 20
IF(ZPT(ND) .LT. ZZMIN) GO TO 20
IF(XPT(ND) .LT. XYZMAX(1)) XYZMAX(1) = XPT(ND)
IF(XPT(ND) .LT. XYZMIN(1)) XYZMIN(1) = XPT(ND)
IF(YPT(ND) .LT. XYZMAX(2)) XYZMAX(2) = YPT(ND)
IF(YPT(ND) .LT. XYZMIN(2)) XYZMIN(2) = YPT(ND)
IF(ZPT(ND) .LT. XYZMAX(3)) XYZMAX(3) = ZPT(ND)
IF(ZPT(ND) .LT. XYZMIN(3)) XYZMIN(3) = ZPT(ND)
20 CONTINUE
25 CONTINUE
GO TO 100
1000 CONTINUE
DO 300 1=1,3
IF(1.EQ.1 .AND. KSYMZ .NE. 1) GO TO 300
IF(1.EQ.2 .AND. KSYMXY .NE. 1) GO TO 300
IF(1.EQ.3 .AND. KSYMXYZ .NE. 1) GO TO 300
300 CONTINUE
APPENDIX B – Continued

\[ XYZBIG = \text{ABS}(XYZMAX(I)) \]
\[ \text{IF}(\text{ABS}(XYZMIN(I)) > XYZBIG) \text{ XYZBIG} = \text{ABS}(XYZMIN(I)) \]
\[ XYZMAX(I) = XYZBIG \]
\[ XYZMIN(I) = -XYZBIG \]

300 CONTINUE
RETURN
END OF BOUND

SUBROUTINE ZEROD(NUMPT, XPT, YPT, ZPT, UPT, VPT, WPT)
C
C *** Initializes all displacements to zero.
C
COMMON/KQUNT/ NNODE, NNDEST, NUDISP, NVDISP, NWDISP
DIMENSION NUMPT(I), XPT(I), YPT(I), ZPT(I), UPT(I), VPT(I), WPT(I)
IF(NUDISP .EQ. 0) GO TO 200
DO 150 I = 1, NUDISP
    UPT(I) = 0.0
150 CONTINUE
IF(NVDISP .EQ. 0) GO TO 300
DO 250 I = 1, NVDISP
    VPT(I) = 0.0
250 CONTINUE
IF(NWDISP .EQ. 0) GO TO 400
DO 350 I = 1, NWDISP
    WPT(I) = 0.0
350 CONTINUE
RETURN
END OF ZEROD
SUBROUTINE OATAHNUMPT, XP!, YP!, ZP!, UPT, VPT, WPT)

C *** TO READ DISPLACEMENT DATA FROM CARDS
C *** WITH USER SPECIFIED FORMAT.
C
COMM/ON/CONTRL/KGEOM, KDATA, XPLSOT, KSYMXY, KSYMZX, KSYMZY, NOTAT, XLHT,
1KHORZ, KVERLF, PHI, THETA, PSI, NEWFR, ISCALE, PLTTSZ, XORGN, YORGN,
2PS, SCALE, XDISP, DMAG, KODE
COMMON/KOUNT/NGCD, NNDEST, NUDISP, NVDISP, NWDISP
COMMON/SElNC/IRESEQ
DIMENSION NUMPT(I), XPT(I), YPT(I), ZPT(I), UPT(I), VPT(I), WPT(I)
DIMENSION ABCD(8)
DIMENSION ISAV(10), OSAV(10, 3)
10 FORMAT(8A10)
C *** TO READ DISPLACEMENT INFORMATION
C
TEST = aHFORMAT
READ(5, 10) ABCD
DECODE(80, 45, ABCD) WORD, KVALU
45 FORMAT(A6, 1X, I 1)
IF (KVALU .EQ. 0) KVALU = 1
IF (WORD, EQ, TEST) 300, 200
200 WRITE(6,20)
20 FORMAT(IHI, ///, 20X, * SORRY, FORMAT FOR DATA NOT GIVEN*)
STOP
300 CONTINUE
DECODE(80, 50, ABCD) FORMT
50 FORMAT(10X, 7A10)
100 CONTINUE
KOUNT = 0
IF (NUDISP .NE. 0) KOUNT = KOUNT+1
IF (NVDISP .NE. 0) KOUNT = KOUNT+1
IF (NWDISP .NE. 0) KOUNT = KOUNT+1
READ(5, FORMT) (ISAV(K), (OSAV(K, I), I=1, KOUNT), K=1, KVALU)
DO 400 K=1, KVALU
400 IDM = ISAV(K)
IF (IDM .EQ. 0) GO TO 1000
IF (IRESEQ .NE. 1) GO TO 700
C *** FOR RESEQUENCED GRID POINTS
C
DO 500 J=1, NNODE
IF (NUMPT(J) .EQ. IDM) 501, 500
501 CONTINUE
KOUNT = 1
IF (NUDISP .NE. 0) 610, 620
510 UPT(J) = OSAV(K, KOUNT)
KOUNT = KOUNT+1
520 CONTINUE
IF (NVDISP .NE. 0) 630, 640
530 VPT(J) = OSAV(K, KOUNT)
KOUNT = KOUNT+1
540 CONTINUE
IF (NWDISP .NE. 0) 650, 660
550 WPT(J) = OSAV(K, KOUNT)
560 CONTINUE
APPENDIX B – Continued

GO TO 550
500 CONTINUE
550 CONTINUE
   GO TO 400
700 CONTINUE

C *** FOR NO RESEQUENCE OF GRID POINTS
C
KOUNT = 1
   IF(NUD.ISP.NE.0) 1610,1620
1610 UPT(IDUM) = DSAV(K,KOUNT)
   KOUNT = KOUNT+1
1620 CONTINUE
   IF(NVD.ISP.NE.0) 1630,1640
1630 VPT(IDUM) = OSAV(K,KOUNT)
   KOUNT = KOUNT+1
1640 CONTINUE
   IF(NW.DISP.NE.0) 1650,1660
1650 WPT(IDUM) = DSAV(K,KOUNT)
1660 CONTINUE
400 CONTINUE
   GO TO 100
1000 CONTINUE
   RETURN
END OF DATA I
SUBROUTINE DATA5(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)

*** TO READ DISPLACEMENT DATA FROM TAPE20.

COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,RLHT,
    LKHRZ,KVEKT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
    2PScale,DMAG,KODE
COMMON/KOUNT/ NNODE,NDEST,NUDISP,NVDisp,NWDisp
COMMON/KVALUES/ NVALUES
DIMENSION NUMPK(I),XPT(I),YPT(I),ZPT(I),UPT(I),VPT(I),WPT(I)
DIMENSION OSAV(3)
DO 10 I=1,NVALUES
KOUNT = 0
IF(NUDISP.NE.0) KOUNT = KOUNT+1
IF(NVDisp.NE.0) KOUNT = KOUNT+1
IF(NWDisp.NE.0) KOUNT = KOUNT+1
READ(20) IDUM,(OSAV(K),K=1,KOUNT)
KOUNT = 1
IF(NUDISP.NE.0) 610,620
610 UDUM = OSAV(KOUNT)
KOUNT = KOUNT+1
620 CONTINUE
IF(NVDisp.NE.0) 630,640
630 VDUM = OSAV(KOUNT)
KOUNT = KOUNT+1
640 CONTINUE
IF(NWDisp.NE.0) 650,660
650 WDUM = OSAV(KOUNT)
660 CONTINUE
IF(KESEQ.NE.1) GO TO 550
DO 500 J=1,NNODE
   IF(NUMPK(J).EQ.IDUM) 501,500
   501 CONTINUE
   IDUM = J
GO TO 550
500 CONTINUE
550 CONTINUE
IF(NUDISP.NE.0) UPT(IDUM) = UDUM
IF(NVDisp.NE.0) VPT(IDUM) = VDUM
IF(NWDisp.NE.0) WPT(IDUM) = WDUM
10 CONTINUE
RETURN
END OF DATA5

SUBROUTINE DATA9(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)

*** USER SUPPLIED DISPLACEMENT INPUT SUBROUTINE.

COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,RLHT,
    LKHRZ,KVEKT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
    2PScale,DMAG,KODE
COMMON/KCOUNT/ NNODE,NDEST,NUDISP,NVDisp,NWDisp
DIMENSION NUMPK(I),XPT(I),YPT(I),ZPT(I),UPT(I),VPT(I),WPT(I)

*** INSERT ROUTINE HERE

RETURN
END OF DATA9
APPENDIX B – Continued

SUBROUTINE PNTOUT(OUT, NUMPT, XPT, YPT, ZPT, UPT, VPT, WPT)
C
C *** FOR PRINTED OUTPUT OF INFORMATION IN BLANK COMMUN - 2 ZZ
C
COMMON/KOUNT/ NNODE, NNDEST, NVUISP, NVUISP, NWDISP
DIMENSION NUMPT(1), XPT(1), YPT(1), ZPT(1), UPT(1), VPT(1), WPT(1)
GO TO (1000, 2000) IOUT
1000 CONTINUE
C
C *** FOR OUTPUT OF GEOMETRY INFORMATION
C
WRITE(6, 16)
16 FORMAT(///, 5X, *GRID POINT INFORMATION*, ///)
WRITE(6, 17)
17 FORMAT(///, 5X, *RESEQUENCED*, 4X, *USER INPUT*/
15X, *GRID POINT*, 5X, *GRID POINT*/
DO 30 I = 1, NNODE
WRITE(6, 18) I, NUMPT(I), XPT(I), YPT(I), ZPT(I)
30 CONTINUE
C
WRITE(6, 19)
19 FORMAT(///, 5X, *ELEMENT INFORMATION - WITH RESEQUENCED GRID POINTS*
1**///)
WRITE(6, 21)
21 FORMAT(///, 5X, *RESEQUENCED*, 4X, *USER INPUT*/
15X, *ELEMENT*, 8X, *ELEMENT*/
REtn 10
I = 0
35 CONTINUE
I = I + 1
CALL RECIN(10, 1, NUMEL, NODE1, NODE2, NODE3, NODE4)
IF (EOF, 10) 36, 37
36 RETURN
37 CONTINUE
WRITE(6, 22) I, NUMEL, NODE1, NODE2, NODE3, NODE4
22 FORMAT(2X, I10, 5X, I10, 4X, I10, 4X, I10)
GO TO 35
2000 CONTINUE
C
C *** FOR OUTPUT OF DISPLACEMENT DATA
C
WRITE(6, 210)
210 FORMAT(///, 5X, *DISEPLACEMENTS TO BE PLOTTED*, ///)
WRITE(6, 17)
DU 230 I = 1, NNODE
U = 0.0
IF (NVUISP.NE.0) U = UPT(I)
V = 0.0
IF (NVUISP.NE.0) V = VPT(I)
W = 0.0
IF (NWDISP.NE.0) W = WPT(I)
WRITE(6, 18) I, NUMPT(I), U, V, W
230 CONTINUE
RETURN
END OF PNTOUT
APPENDIX B – Continued

SUBROUTINE DSCALE(NUMPT, XPT, YPT, ZPT, UPT, VPT, WPT)
C
C *** THIS SUBROUTINE DETERMINES THE SCALE FACTOR FOR DISPLACEMENTS
C
COMMON/CONTRL/ KGEOM, KDATA, KPLOT, KSYMXY, KSYMXZ, KSYMYZ, NOTAT, XLHT,
1 KHRZ, KVERT, PHI, THETA, PSI, NEWFR, ISCALE, PLOTSZ, XORGN, YORGN,
2 PSSCALE, KDISP, OMAG, KODE
COMMON/SAVEV/ OMAGStlDMAG
COMMON/KOUNT/ NNCDE, NNOEST, NUOISP, NVDISP, NWDISP
DIMENSION NUMPT(1), XPT(1), YPT(1), ZPT(1), UPT(1), VPT(1), WPT(1)
IF(KDISP.EQ.0.OR.KDISP.EQ.2) GO TO 10
GO TO (10, 20) IUMAG
10 CONTINUE
OMAG = OMAGS
GO TO 30
20 CONTINUE
DMAX = 0.0
GO 100 1=1, NNODE
IF(NUDISP.EQ.0) GO TO 500
IF(ABS(UPT(I)).GT.DMAX) DMAX = ABS(UPT(I))
500 CONTINUE
IF(NUDISP.EQ.0) GO TO 501
IF(ABS(VPT(I)).GT.DMAX) DMAX = ABS(VPT(I))
501 CONTINUE
IF(NWDISP.EQ.0) GO TO 502
IF(ABS(WPT(I)).GT.DMAX) DMAX = ABS(WPT(I))
502 CONTINUE
OMAG = DMAGS/DMAX
GO TO 30
30 CONTINUE
RETURN
END OF DSCALE
SUBROUTINE PLOT(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)

*** FOR GENERATING PLOTS.

COMMUN/CTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMZ,NOTAT,XLHT, 
1KVERT,PHI,THETA,PSI,NEWFR,LSCALE,PLTSZ,XORGN,YORGN, 
2KDISP,DMAG,KODE 
COMMUN/LIMITS/ XXMAX,YYMAX,ZZMAX,XXMIN,YYMIN,ZZMIN,NDMAX,NDMIN, 
INELMAX,INELMIN 
COMMUN/XYZLIM/ XYZMAX(3),XYZMIN(3) 
COMMUN/CORGN/ XGABS,YOABS,XMAX,XSPACE,PSIZE 
COMMUN/GLUOP/ ILUOP 
COMMON/ABLK/ A(3,3) 
COMMON/KCOUNT/ NNODE,NNODE,NUOISP,NVDISP,NWDISP 
COMMON/POELS/ DELX,DELY 
DIMENSION NUMPT,I,JNUMVE,NNODE,NUOISP,NVDISP,NWDISP 
DIMENSION NODE(4),X(4),Y(4),Z(4),XDISP(4),YDISP(4),ZDISP(4), 
1XRUT(4),YRUT(4) 
DIMENSION TEST(3) 
DIMENSION ABCD(8) 
1 FORMAT(8A10) 
2 FORMAT(1X,8A10) 

*** TO MAKE ALL GRID POINT NUMBERS NEGATIVE

DG 50 I=1,NNODE 
NUMPT(I) = -NUMPT(I) 
50 CONTINUE 
PI = 3.1415926 
XMOVE = 0.0 
IF(NEWFR.EQ.1) XMOVE = XMAX+XSPACE 
YMOVE = -YUABS 
CALL CALPLT(XMOVE,YMOVE,-3) 
XUABS = XABS+XMOVE 
YUABS = YUABS+YMOVE 
GO TO 701,701,703,701) KPLOT 
701 CONTINUE 
GO TO 710 
703 CONTINUE 
IF(NEWFR.EQ.1) CALL NFRAME 
710 CONTINUE 
DELX = 0.0 
DELY = 0.0 
IF(ISCALE.EQ.1) CALL XYSCL 
CALL CALPLT(XORGN,YORGN,-3) 
XUABS = XABS+XORGN 
YUABS = YUABS+YORGN 
XSHIFT = 0.0 
YSHIFT = 0.0 
ZSHIFT = 0.0 
XMAX = -1.0E20 

*** LOOPS TO ACCOUNT FOR SYMMETRY

ZSIGN = +1.0 
DG 500 II=1,2 
IF(II.EQ.2.AND.KSYMXY.NE.1) GO TO 500 
IF(II.EQ.2.AND.KSYMXY.EQ.1) ZSIGN = -1.0 
YSIGN = +1.0
APPENDIX B – Continued

DO 510 JJ=1,2
   IF(JJ.EQ.2.AND.KSYMZ.NE.1) GO TO 510
   IF(JJ.EQ.2.AND.KSYMZ.EQ.1) XSIGN = -1.0
   XSIGN = +1.0
DO 520 KK=1,2
   IF(KK.EQ.2.AND.KSYMZ.NE.1) GO TO 520
   IF(KK.EQ.2.AND.KSYMZ.EQ.1) XSIGN = -1.0
C
C *** TO DETERMINE PROJECTED COORDINATES OF ELEMENTS
C
REWIND 10
100 CONTINUE
   CALL RECIN(10, 1, 5, NUMEL, NODE1, NODE2, NODE3, NODE4)
   IF(EOF,10) 1000,200
200 CONTINUE
   IF(NUMEL.LT.NELMIN.OR.NUMEL.GT.NELMAX) GO TO 100
   NODE(1) = NODE1
   NODE(2) = NODE2
   NODE(3) = NODE3
   NODE(4) = NODE4
   DO 10 I=1,4
   ND = NODE(I)
   IF(NODE(I).EQ.0) GO TO 11
   C
   C *** TO MAKE GRID POINT NUMBERS CONNECTED BY ELEMENTS POSITIVE
   NUMPT(ND) = IABS(NUMPT(ND))
   IF(NUMPT(ND).LT.NCMIN.OR.NUMPT(ND).GT.NDMAX) GO TO 100
   NEND = I
10 CONTINUE
11 CONTINUE
   I = KHORZ
   J = KVERT
   DO 20 N=1,NEND
   ND = NODE(N)
   IF(XPT(ND).GT.XXMAX) GO TO 100
   IF(XPT(ND).LT.XXMIN) GO TO 100
   IF(YPT(ND).GT.YYMAX) GO TO 100
   IF(YPT(ND).LT.YYMIN) GO TO 100
   IF(ZPT(ND).GT.ZZMAX) GO TO 100
   IF(ZPT(ND).LT.ZZMIN) GO TO 100
   XDISP(N) = 0.0
   YDISP(N) = 0.0
   ZDISP(N) = 0.0
   IF(KDISP.EQ.1.AND.NUDISP.NE.0) XDISP(N) = UPT(ND)
   IF(KDISP.EQ.1.AND.NVDISP.NE.0) YDISP(N) = VPT(ND)
   IF(KDISP.EQ.1.AND.NWDISP.NE.0) ZDISP(N) = WPT(ND)
   X(N) = XSIGN*(XPT(ND)+XDISP(N)*DMAG+XSHIFT)/PSCALE
   Y(N) = YSIGN*(YPT(ND)+YDISP(N)*DMAG+YSHIFT)/PSCALE
   Z(N) = ZSIGN*(ZPT(ND)+ZDISP(N)*DMAG+ZSHIFT)/PSCALE
20 CONTINUE
   IF(KDISP.EQ.2) CALL XPLOD(NEND,X,Y,Z)
   XCENT = 0.0
   YCENT = 0.0
   DO 25 N=1,NEND
   XROT(N) = A(I,1)*X(N)+A(I,2)*Y(N)+A(I,3)*Z(N)
   YROT(N) = A(J,1)*X(N)+A(J,2)*Y(N)+A(J,3)*Z(N)
   XCENT = XCENT+XROT(N)
   YCENT = YCENT+YROT(N)
25 CONTINUE
   XCENT = XCENT/NEND
   YCENT = YCENT/NEND
61
APPENDIX B – Continued

XROT(N) = XROT(N) + DELX
YROT(N) = YROT(N) + DELY
IF(XROT(N) GT XPMAX) XPMAX = XROT(N)
29 CONTINUE
XCENT = XCENT / FLOAT(NEND) - (6.0/7.0) * XLHT
YCENT = YCENT / FLOAT(NEND) - XLHT/2.0
XCENT = XCENT + DELX
YCENT = YCENT + DELY
AL = NUMEL
IF(NOTAT.EQ.2) CALL NUMBER(XCENT, YCENT, XLHT, AL, 0.0, -1)

C *** TO PLOT ELEMENTS
C
CALL CALPLT(XROT(1), YROT(1), 3)
30 N = 2, NEND
CALL CALPLT(XROT(N), YROT(N), 2)
30 CONTINUE
CALL CALPLT(XROT(NEND), YROT(NEND), 3)
IF(NEND.GT.2) 35, 36
35 CALL CALPLT(XROT(1), YROT(1), 2)
CALL CALPLT(XROT(1), YROT(1), 3)
36 CONTINUE
GO TO 100
1000 CONTINUE
IF(KDISP.EQ.3) 600, 650
600 CONTINUE

C *** TO PLOT VECTORS AT GRID POINTS
C
601 ND = 1, NNODE
IF(NUMPT(NU) LE 0) GO TO 601
IF(NUMPT(ND), LT, NDMIN, OR, NUMPT(ND), GT, NDMIN) GO TO 601
IF(XPT(ND), GT, XYZMAX(1)) GO TO 601
IF(XPT(ND), LT, XYZMIN(1)) GO TO 601
IF(YPT(ND), GT, XYZMAX(2)) GO TO 601
IF(YPT(ND), LT, XYZMIN(2)) GO TO 601
IF(ZPT(ND), GT, XYZMAX(3)) GO TO 601
IF(ZPT(ND), LT, XYZMIN(3)) GO TO 601
X(1) = XSIGN*(XPT(NO) - Xispers*OMAG - XSHIFT)/PSCALE
Y(1) = YSIGN*(YPT(NO) + Yispers*DMAG + YSHIFT)/PSCALE
Z(1) = ZSIGN*(ZPT(NO) + ZSHIFT)/PSCALE
XDISP(1) = 0.0
YDISP(1) = 0.0
ZDISP(1) = 0.0
IF(NUDISP.NE.0) XDISP(1) = UPT(ND)
IF(NVDISP.NE.0) YDISP(1) = VPT(ND)
IF(NWDISP.NE.0) ZDISP(1) = WPT(ND)
X(2) = XSIGN*(XPT(NO) + XDISP(1) * DMAG + XSHIFT)/PSCALE
Y(2) = YSIGN*(YPT(NO) + YDISP(1) * DMAG + YSHIFT)/PSCALE
Z(2) = ZSIGN*(ZPT(NO) + ZDISP(1) * DMAG + ZSHIFT)/PSCALE
I = KH19K4
J = KVERT
DG 005 N = 1, 2
XROT(N) = A(1, 1)*X(N) + A(1, 2)*Y(N) + A(1, 3)*Z(N)
APPENDIX B – Continued

YROT(N) = A(J,1)*X(N)+A(J,2)*Y(N)+A(J,3)*Z(N)
XROT(N) = XROT(N)+DELX
YROT(N) = YROT(N)+DELY

605 CONTINUE
XARW = 0.06
YARW = XARW/3.0
CALL GARROW(XROT(1),YROT(1),XROT(2),YROT(2),1,XARW,YARW
601 CONTINUE
650 CONTINUE
920 CONTINUE
910 CONTINUE
500 CONTINUE
RETURN
END OF PLOT

SUBROUTINE ROTAT

C *** SETS UP COEFFICIENTS OF ROTATION MATRIX
C
COMMON/CURRL/KGECM, KOATA, KDATA, KPLT, KSYMXY, KSYMZX, KSYMZY, NOTAT, XLHT,
1KXORZ, KVQT, PHI, THETA, PSI, NEWFR, ISCALE, PLOTsz, XORGN, YORGN,
2PSCALE, KDISP, DMAG, KODE
COMMON/ABLx/ A(3,3)
PI = 3.1415926536
SINPHI = SIN(PHI*PI/180.0)
COSPH1 = COS(PHI*PI/180.0)
SINTHc = SIN(THETA*PI/180.0)
COSTHE = COS(THETA*PI/180.0)
SINPSI = SIN(PSI*PI/180.0)
COSPSi = COS(PSI*PI/160.0)
A(1,1) = COSTHE*COSPSI
A(1,2) = COSPSI*SINPHI*SINPHI-SINPSI*COSPH1
A(1,3) = SINPHI*COSPH1*COSPH1+SINPSI*SINPH1
A(2,1) = SINPSI*COSTHE
A(2,2) = SINPHI*SINPH1*SINPH1+COSPH1*COSPSI
A(2,3) = SINPH1*COSPH1*SINPSI-SINPH1*COSPSI
A(3,1) = -SINTHE
A(3,2) = CSTHE*SINPH1
A(3,3) = COSTHE*COSPH1
RETURN
END OF ROTAT
SUBROUTINE XYSCAL

*** TO DETERMINE SCALE FACTOR FOR MODEL GEOMETRY.

COMMON/CTRL/ KGEOM, KDATA, KPLUT, KSYMXY, KSYMXY, KSYMXY, NOTAT, XLHT, 1KHORZ, KVERT, PHI, THETA, PSI, NEWFR, SCALE, PLOTSZ, XORGN, YORGN,
2 SCALE, KDISP, DMAG, KODE
COMMON/XYZLIM/ XYZMAX(3), XYZMIN(3)
COMMON/CORGN/ XOABS, YOABS, XPMAX, XSPACE, PSIZE
COMMON/ABLK/ A(3, 3)
COMMON/DELX, DELY
I = KHORZ
J = KVERT
D MAX = 0.0
DO 5 N = 1, 3
VDUM = ABS(XYZMAX(N) - XYZMIN(N))
IF(VDUM .GT. DMAX) DMAX = VDUM
5 CONTINUE
P SCALE = DMAX / PLOTSZ
DO 10 L = 1, 2
DO 10 M = 1, 2
DO 10 N = 1, 2
X = XYZMIN(I)
IF(L .EQ. 2) X = XYZMAX(1)
Y = XYZMIN(2)
IF(M .EQ. 2) Y = XYZMAX(2)
Z = XYZMIN(3)
IF(N .EQ. 2) Z = XYZMAX(3)
XROT = A(I, 1) * X + A(I, 2) * Y + A(I, 3) * Z
YROT = A(J, 1) * X + A(J, 2) * Y + A(J, 3) * Z
IF(L * M * N .EQ. 1) 20, 30
20 CONTINUE
XRMIN = XROT
XRMAX = XROT
YRMIN = YROT
YRMAX = YROT
30 CONTINUE
IF(XROT .GT. XRMAX) XRMAX = XROT
IF(XROT .LT. XRMIN) XRMIN = XROT
IF(YROT .GT. YRMAX) YRMAX = YROT
IF(YROT .LT. YRMIN) YRMIN = YROT
10 CONTINUE
YR = ABS(YRMAX - YRMIN)
IF(YR / SCALE .GT. PSIZE) SCALE = YR / PSIZE
XRMAX = XRMAX / SCALE
YRMAX = YRMAX / SCALE
XRMIN = XRMIN / SCALE
YRMIN = YRMIN / SCALE
DELX = -XRMIN
DELY = -YRMIN
XORGN = 0.0
YORGN = 0.0
RETURN
END OF XYSCAL
SUBROUTINE XPLOD(NEND,X,Y,Z)

*** FOR GENERATING EXPLODED PLOTS.

COMMUN/CONTRO/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXX,KSYMYY,NOTAT,XLHT,
     LKHORZ, XVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLTsz,XORGX,YORGY,
     TPSCAL, KDISP, DMAG, KODE

DIMENSION X(4),Y(4),Z(4)

*** TO CALCULATE THE INCENTER OF TRIANGLES

IF(NEND.EQ.3) 10,20
10 CONTINUE

A = SQRT((X(2)-X(3))**2+(Y(2)-Y(3))**2+(Z(2)-Z(3))**2)
B = SQRT((X(1)-X(3))**2+(Y(1)-Y(3))**2+(Z(1)-Z(3))**2)
C = SQRT((X(1)-X(2))**2+(Y(1)-Y(2))**2+(Z(1)-Z(2))**2)

AC1 = A/(A+B+C)
AC2 = B/(A+B+C)
AC3 = C/(A+B+C)

XOC = AC1*X(1)+AC2*X(2)+AC3*X(3)
YOC = AC1*Y(1)+AC2*Y(2)+AC3*Y(3)
ZOC = AC1*Z(1)+AC2*Z(2)+AC3*Z(3)

GO TO 190

20 CONTINUE

*** TO CALCULATE THE CENTROID OF RODS, BARS, AND QUADS

XOC = 0.0
YOC = 0.0
ZOC = 0.0

DO 100 I=1,NEND

XOC = XOC+X(I)
YOC = YOC+Y(I)
ZOC = ZOC+Z(I)

100 CONTINUE

XOC = XOC/FLOAT(NEND)
YOC = YOC/FLOAT(NEND)
ZOC = ZOC/FLOAT(NEND)

190 CONTINUE

*** TO REDUCE THE SIZE OF THE ELEMENT

DO 200 I=1,NEND

XI(I) = X(I)*DMAG
Y(I) = Y(I)*DMAG
Z(I) = Z(I)*DMAG

200 CONTINUE

*** TO CALCULATE THE CENTROID OF THE REDUCED ELEMENT

XRC = XOC*DMAG
YRC = YOC*DMAG
ZRC = ZOC*DMAG

*** SHIFT CORNERS OF ORIGINAL AND REDUCED TO MAKE CENTROIDS MATCH

DO 400 I=1,NEND

X(I) = X(I)+(XOC-XRC)
400 CONTINUE
APPENDIX B – Continued

Y(I) = Y(I) + (YCC - YRC)
Z(I) = Z(I) + (ZCC - ZRC)
400 CONTINUE
RETURN
END OF XPL0D

SUBROUTINE GARROW(X1,Y1,X2,Y2,NC,XHEAD,YHEAD)
C
C *** TO DRAW ARROWS FROM X1,Y1 TO X2,Y2.
C
DEN = SQRT((X2-X1)**2 + (Y2-Y1)**2)
IF(DEN.EQ.0.0) GO TO 5000
C = (X1-X2)/DEN
S = (Y1-Y2)/DEN
CALL CALPLT(X1,Y1,3)
CALL CALPLT(X2,Y2,2)
IF(NC.LT.1) GO TO 1000
XA = X2 + (C*XHEAD - S*YHEAD)
YA = Y2 + (S*XHEAD + C*YHEAD)
CALL CALPLT(XA,YA,2)
IF(NC.LT.2) GO TO 1000
XB = X2 + (C*XHEAD - S*(-YHEAD))
YB = Y2 + (S*XHEAD + C*(-YHEAD))
CALL CALPLT(XB,YB,2)
IF(NC.LT.3) GO TO 1000
CALL CALPLT(X2,Y2,2)
IF(NC.LT.4) GO TO 1000
XC = X2 + (-S*YHEAD)
YC = Y2 + (+C*YHEAD)
CALL CALPLT(XC,YC,2)
IF(NC.LT.5) GO TO 1000
XD = X2 + (-S*(-YHEAD))
YD = Y2 + (+C*(-YHEAD))
CALL CALPLT(XD,YD,2)
1000 CONTINUE
CALL CALPLT(X2,Y2,3)
5000 CONTINUE
RETURN
END OF GARROW
SUBROUTINE NIDLET(NUMPT, XPT, YPT, ZPT, UPT, VPT, WPT)

*** FOR ANNOTATING GRID POINT NUMBERS ON PLOTS.

COMMON/CONTRL/ KGEOM, KDATA, KPLOT, KSYMMX, KSYMMZ, KSYMXY, KSYMZX, KSYMYZ, NOTAT, XLHT,
KHORZ, KVERT, PHI, THETA, PSI, NEWFR, ISCALE, PLOTSZ, XORGN, YORGN,
2PSCALE, KDISP, DMAG, KODE
COMMON/LIMITS/ XXMAX, YYMAX, ZZMAX, XXMIN, YYMIN, ZZMIN, NDMAX, NDMIN,
1NELMAX, NELMIN
COMMON/XYZLIN/ XYZMAX(3), XYZMIN(3)
COMMON/ABLK/ A(3,3)
COMMON/KOUNT/ NNODE, NNDEST, NUOISP, NVOISP, NWDISP
COMMON/PVELS/ DELX, DELY
DIMENSION NUMPT(I), XPT(I), YPT(I), ZPT(I), UPT(I), VPT(I), WPT(I)

II = KHORZ
JJ = KVERT
XSHIFT = 0.0
YSHIFT = 0.0
ZSHIFT = 0.0

DO 500 I = 1, NNODE
  IF(NUMPT(I).LE.0) GO TO 500
  IF(NUMPT(I).GT.NDMIN.OR.NUMPT(I).GT.NDMAX) GO TO 500
  IF(XPT(I).GT.XYZMAX(1)) GO TO 500
  IF(XPT(I).LT.XYZMIN(1)) GO TO 500
  IF(YPT(I).GT.XYZMAX(2)) GO TO 500
  IF(YPT(I).LT.XYZMIN(2)) GO TO 500
  IF(ZPT(I).GT.XYZMAX(3)) GO TO 500
  IF(ZPT(I).LT.XYZMIN(3)) GO TO 500
  X = (XPT(I)*XSHIFT)/PSCALE
  Y = (YPT(I)*YSHIFT)/PSCALE
  Z = (ZPT(I)*ZSHIFT)/PSCALE
  XRUT = A(II,1)*X+A(II,2)*Y+A(II,3)*Z
  YRUT = A(JJ,1)*X+A(JJ,2)*Y+A(JJ,3)*Z
  XL = XRUT+XLHT/2.0
  YL = YRUT+YLHT/2.0
  X = XL+DELX
  Y = YL+DELY
  AL = NUMPT(I)
  CALL NUMBER(XL, YL, XLHT, AL, 0.0, -1)

500 CONTINUE
RETURN
END OF NIDLET
APPENDIX C

LISTING OF COMPUTER PROGRAM FOR CONTOUR PLOTS

An overall flow chart for this program is given in figure 13. The MAIN program is used to allocate blank COMMON storage and to call other subroutines necessary to read input data and to generate the desired plots. The purpose of each subroutine is described in comment cards in the listing. A subroutine called DOCMNT consisting entirely of comment cards is included in the program. Subroutine DOCMNT contains (1) a directory of selected variables used in the program, (2) user-input instructions, and (3) a description of plotting subroutines which are required from the Langley Graphic Output System library.
APPENDIX C – Continued

PROGRAM MAIN(INFLT=201, OUTPUT=201, TAPE5= INPUT, TAPE6 = OUTPUT, 
1 TAPE20 = 201)

C
C *** THIS IS MAIN PROGRAM WHICH CALLS OTHER SUBROUTINES
C
C
C *** USER DOCUMENTATION IS GIVEN IN SUBROUTINE DOCMNT
C
INTEGER NUMPT, XPT, YPT, WPT, RAAPT, NUMEL, NODE1, NODE2, NODE3, NODE4, 
ICENTR
COMMON/KOUNT/ NNGD, NELPT, NNDEST, NELES
COMMON/ CONTROL/ INFOR, KGECM, KDATA, KSIGN, KPLUT, XORGN, YORGN,
IPSCALE, ISPERSK, WMSG, ICNTR, XLHT, XLAB, YLAB(10), NYLAB, YLAB(10)
COMMON/GKUDE/ KCLN, NPLUT
COMMON/ VALUES/ NVALUES
COMMON/ CASEID/ ICCASE
COMMON/ SAVE V/ SMMAGS, ICNTRS
COMMON/ SIZEXY/ XMNIM, XMAXIM, YMNIM, YMAXIM, XSPACE, YSPACE
COMMON ZZZ(11)
DIMENSION ABCO(81), BCD(2)
NAMELIST/PICL/ NVALUES
COMMON/ SCALE/ WMSG, ICNTRS, XLHT, XLAB, YLAB, KCLN
500 CONTINUE
ILOOP = 0
C
C *** TC ZERO NODE AND ELEMENT COUNTERS AND SET TEST VALUES
C
ANODE = 0
NELM = 0
WRITE(6,0)
8 FORMAT(IH1)
C
C *** TO READ TITLE CARD FOR RUN
C
READ(5,10) ABCD
10 FORMAT(IH10)
IF(EHOF,5) 2222,3333
2222 CALL PSTUP
3333 CONTINUE
WRITE(6,11) ABCD
11 FORMAT(///,20X,AH10,///)
CALL INITIAL
IF(KPLUT .NE. 4 OR KDATA .NE. 5) GO TO 111
REWRIND 20
CALL MESSAGE(1, 24 HENTERING CONTOUR PROGRAM, 24)
CALL MESSAGE(1, 19FSELECT DESIRED MODE, 19)
CALL MESSAGE(1, 18HBY PRESSING FN KEY, 18)
CALL NEXT(NKEY)
IMODE = NKEY-1
IF(IMODE .EQ. 0) GO TO 111
DO 57 J=1,IMODE
DO 58 I=1,NVALUES
READ(20) IDUM, WDUM
58 CONTINUE
57 CONTINUE
111 CONTINUE
HEIGHT = 0.15
XSTRT = 2.0*HEIGHT+2.0
YSTRT = 1.0
CALL NUTATE (XSTRT, YSTRT, HEIGHT, ABCD, 90.0, 80)
CALL NFRAME

*** TO SET POINTERS FOR BLANK COMMON STORAGE ZZZ

*** (WITH INTEGER NAMES OF ARRAYS USED IN CALLED SUBROUTINES)

NUMPT = 1
XPT = XNUMPT+NNDEST
YPT = XPT+NNDEST
WPT = YPT+NNDEST
RADPT = WPT+NNDEST
NUMEL = RADPT+NNDEST
NODE1 = NUMEL+NELEST
NODE2 = NODE1+NELEST
NODE3 = NODE2+NELEST
NODE4 = NODE3+NELEST
CENTR = NODE4+NELEST
NEND = CENTR+NELEST-1

WRITE (*, 15) NEND
15 FORMAT (///, 20X, 'BLANK COMMON STORAGE ZZZ REQUIRES AT LEAST **,I6, 
* LOCATIONS FOR THIS CASE***///)

*** CALL SUBROUTINES FOR INPUT OF CONTROL VALUES AND GEOMETRY

IF (KDEUm .EQ. 1) CALL GEMr1
LLL NUMPT), ZZZ XPT), ZZZ YPT), ZZZ WPT), ZZZ RADPT),
ZZZ NUMEL), ZZZ NODE1), ZZZ NODE2), ZZZ NODE3), ZZZ NODE4), ZZZ CENTR)
IF (KDE Um .EQ. 2) CALL GEMr2
LLL NUMPT), ZZZ XPT), ZZZ YPT), ZZZ WPT), ZZZ RADPT),
ZZZ NUMEL), ZZZ NODE1), ZZZ NODE2), ZZZ NODE3), ZZZ NODE4), ZZZ CENTR)
IF (KDEUm .EQ. 9) CALL GEMr9
LLL NUMPT), ZZZ XPT), ZZZ YPT), ZZZ WPT), ZZZ RADPT),
ZZZ NUMEL), ZZZ NODE1), ZZZ NODE2), ZZZ NODE3), ZZZ NODE4), ZZZ CENTr)
CALL XYSCAl
LLL NUMPT), ZZZ XPT), ZZZ YPT), ZZZ WPT), ZZZ RADPT),
ZZZ NUMEL), ZZZ NODE1), ZZZ NODE2), ZZZ NODE3), ZZZ NODE4), ZZZ CENTr)
CALL ENCLOS
LLL NUMPT), ZZZ XPT), ZZZ YPT), ZZZ WPT), ZZZ RADPT),
ZZZ NUMEl), ZZZ NODE1), ZZZ NODE2), ZZZ NODE3), ZZZ NODE4), ZZZ CENTr)
CALL PUTOUT
LLL NUMPT), ZZZ XPT), ZZZ YPT), ZZZ WPT), ZZZ RADPT),
ZZZ NUMEl), ZZZ NODE1), ZZZ NODE2), ZZZ NODE3), ZZZ NODE4), ZZZ CENTr)
600 CONTINUE
IF (10CASE .EQ. 0) GO TO 650
READ (5, 10) ABCD
WRITE (6, 11) ABCD
HEIGHT = 0.15
XSTRT = 2.0*HEIGHT+5.0
YSTRT = 1.0
CALL NUTATE (XSTRT, YSTRT, HEIGHT, ABCD, 90.0, 80)
CALL NFRAME
650 CONTINUE
CALL ZEROW
1 (LLL NUMPT), ZZZ WPT), ZZZ NUMEL), ZZZ CENTr)
IF (KDATA .EQ. 1) CALL DATAl
1 (LLL NUMPT), ZZZ WPT), ZZZ NUMEl), ZZZ CENTr)
IF (KDATA .EQ. 5) CALL DATAs
APPENDIX C – Continued

1(LLL1(NUML),LLL1(WPT),LLL1(NODE1),LLL1(NODE2),LLL1(NODE3),LLL1(NODE4),LLL1(CENTR))
1(IF(KDATA.EQ.9) CALL DATA9
1(LLL1(NUMPT),LLL1(WPT),LLL1(NODE1),LLL1(NODE2),LLL1(NODE3),LLL1(NODE4),LLL1(CENTR))
CALL PNTOUT12,
1(LLL1(NUMPT),LLL1(XPT),LLL1(YPT),LLL1(WPT),LLL1(RADPT),
LLL1(NODE1),LLL1(NODE2),LLL1(NODE3),LLL1(NODE4),LLL1(CENTR))
C
C *** CALL SUBROUTINES FOR PLOTTING GEOMETRIC LAYOUTS
C
700 CONTINUE
IF( KPU1T.EQ.4.AND.ILCOP.NE.0) GO TO 6000
READ(5,PICT)
WRITE(5,PICT)
6000 CONTINUE
IF(KPU1T.EQ.4) CALL CCRT1
IF(NPLU1T.EQ.4.OR.NPLUT.EQ.5) 211,212
211 CONTINUE
CALL SCALEW(LLL1(WPT),LLL1(CENTR))
LABEL = dH WMAG =
ENCODE (0,333,BED) LABEL, WMAG
333 FORMAT(A8,E12.4)
HEIGHT = 0.15
XSTRT = XSPACE
YSTRT = 2.0*HEIGHT
CALL NOTATE(XSTRT,YSTRT,HEIGHT,BED,0.0,20)
212 CONTINUE
YSPACE = 0.0
IF(YSPACE.LE.5.0*HEIGHT) YSPACE = 5.0*HEIGHT
XURGN = -XMINIM/PSCALE*XSPACE
YURGN = FLOAT(KSIGN)*(-YMINIM)/PSCALE*YSPACE
CALL CALPLT(XURGN,YURGN,-3)
IF(NPLU1T.LE.3) 201,202
201 CONTINUE
CALL LAYOUT(NPLUT,
1(LLL1(NUMPT),LLL1(XPT),LLL1(YPT),LLL1(WPT),LLL1(RADPT),
LLL1(NODE1),LLL1(NODE2),LLL1(NODE3),LLL1(NODE4),LLL1(CENTR))
GO TO 900
202 CONTINUE
C
C *** FOR CONTOUR PLOTS
C
KPOINT = 1
KIND = -2
ISIZE = 1
IF(NPLUT.EQ.5) CALL POINTS(KIND,ISIZE,LLL1(XPT),LLL1(YPT))
1(IF(NPLUT.EQ.5) CALL WNODE1,
1(LLL1(NUMPT),LLL1(XPT),LLL1(YPT),LLL1(WPT),LLL1(RADPT),
LLL1(NODE1),LLL1(NODE2),LLL1(NODE3),LLL1(NODE4),LLL1(CENTR))
1(IF(NPLUT.EQ.2) CALL WELMT,
1(LLL1(NUMPT),LLL1(XPT),LLL1(YPT),LLL1(WPT),LLL1(RADPT),
LLL1(NODE1),LLL1(NODE2),LLL1(NODE3),LLL1(NODE4),LLL1(CENTR))
1(CALL BOUNC,
1(LLL1(NUMPT),LLL1(XPT),LLL1(YPT),LLL1(WPT),LLL1(RADPT),
LLL1(NODE1),LLL1(NODE2),LLL1(NODE3),LLL1(NODE4),LLL1(CENTR))
YOU CONTINUE
CALL NFRAME
IF(KPLOT.EQ.4) CALL CCRT2
ILOOP = ILOOP + 1
IF(KODE.EQ.0) GO TO 150
GO TO (700,600,500) KODE
150 CONTINUE
CALL PSTOP
STOP
END OF MAIN
APPENDIX C - Continued

SUBROUTINE DOCMNT
C *** THIS SUBROUTINE CONTAINS PROGRAM DOCUMENTATION
C
C
C
DESCRIPTION OF INPUT DATA CARDS
C
C
TITLE CARD - CONTAINS ANY DESIRED ALPHANUMERIC INFORMATION IN COLS.1-80.
C
NAMELIST OPTION - CONTAINS VALUES TO ALLOCATE STORAGE IN BLANK COMMON ZZZ, AND CONTROL VALUES NEEDED BY THE PROGRAM.
C
THE FOLLOWING VALUES ARE INCLUDED——-
C
NNODEST = ESTIMATE NUMBER OF GRID POINTS TO BE USED. VALUE MUST BE GREATER THAN OR EQUAL TO THE ACTUAL NUMBER OF GRID POINTS.
** DEFAULT = 200 **
C
NELEST = ESTIMATED NUMBER OF ELEMENTS TO BE USED. VALUE MUST BE GREATER THAN OR EQUAL TO THE ACTUAL NUMBER OF ELEMENTS.
** DEFAULT = 200 **
C
KGEOM SPECIFIES SUBROUTINE AND CORRESPONDING METHOD OF INPUT FOR MODEL GEOMETRY.
KGEOM = 1 FOR GRID POINTS AND ELEMENTS READ FROM CARDS WITH USER SPECIFIED FORMAT.
= 2 FOR NASTRAN DECK WITH CARD IDENTIFIERS LEFT ADJUSTED AND DATA IN COLUMN WIDTHS OF 8.
= 9 FOR USER SUPPLIED SUBROUTINE - GEOM9.
** DEFAULT = 1 **
C
KDATA SPECIFIES SUBROUTINE AND CORRESPONDING METHOD OF INPUT FOR DATA TO BE REPRESENTED BY CONTOUR LINES.
KDATA = 1 FOR SUBROUTINE DATA1 TO READ IN DATA TO BE PLOTTED FROM CARDS WITH USER SPECIFIED FORMAT.
= 5 FOR SUBROUTINE DATA5 TO READ IN DATA TO BE PLOTTED FROM TAPE20.
= 9 FOR SUBROUTINE DATA9, A USER SUPPLIED SUBROUTINE.
** DEFAULT = 1 **
C
NVALUES - USE IF KDATA = 5 TO SPECIFY THE NUMBER OF SETS OF DATA TO BE PLOTTED WHICH ARE READ FROM TAPE20.
** DEFAULT = 0 **
C
IRSEQ = 0 FOR NO RESEQUENCING OF GRID POINT NUMBERS.
= 1 TO RESEQUENCE GRID POINT NUMBERS IN SAME ORDER AS THEY ARE INPUT.
** DEFAULT = 1 **
C
KPLUT SPECIFIES THE TYPE OF OUTPUT DEVICE TO BE USED.
KPLUT = 1 FOR CALCOMP.
= 2 FOR CALCOMP WITH PLOTTING SPEED REDUCED TO USE LEROY PENS.
= 3 FOR VARIAN.
= 4 FOR CRT (USE CGC250 SCOPES AT LRC)
** DEFAULT = 1 **
C
INFOR = 1 IF DATA TO BE PLOTTED IS SPECIFIED AT THE GRID POINTS.
= 2 IF DATA TO BE PLOTTED IS SPECIFIED AT THE ELEMENT CENTROIDS.
** DEFAULT = 1 **
C
XSPACE = SPACE BETWEEN PLOTS IN X-DIRECTION, IN INCHES.
** DEFAULT = 10.0 **
APPENDIX C - Continued

C ** KSIGN = -1 TO CHANGE SIGN OF Y ORIGINATES. **
C ** 1 DO NOT CHANGE SIGN OF Y COORDINATES. **
C ** DEFAULT = 1 **
C ** UCASE = 0 FOR NO TITLE CARD PRECEDING **
C ** DECK OF DATA TO BE PLOTTED. **
C ** DEFAULT = 0 **

C MODEL GEOMETRY IS NOW INPUT IN ONE OF THE FOLLOWING FORMS,
C DEPENDING ON THE VALUE OF KGEO SPECIFIED IN NAMELIST OPTION.

C USE IF KGEO = 1

C (A) A SINGLE CARD CONTAINING THE WORD FORMAT IN COLUMNS 1-6 AND
C A VARIABLE FORMAT CORRESPONDING TO THE FORMAT OF THE GRID
C POINT CARDS WITH LEFT PARENTHESIS STARTING IN COLUMN 11
C AND UP TO 80 COLUMNS MAY BE USED.
C (B) DECK OF GRID POINT CARDS. EACH CARD CONTAINS 3 VALUES, GRID
C POINT NUMBER (INTEGER), X-COORDINATE (REAL), AND
C Y-COORDINATE (REAL). THE FORMAT IS SPECIFIED BY (A) ABOVE.
C (C) A SINGLE CARD CONTAINING THE WORD ENGRID IN COLUMNS 1-7.
C (D) A SINGLE CARD CONTAINING THE WORD FORMAT IN COLUMNS 1-6 AND
C A VARIABLE FORMAT CORRESPONDING TO THE FORMAT OF THE
C ELEMENT CARDS WITH LEFT PARENTHESIS STARTING IN COLUMN 11
C AND UP TO 80 COLUMNS MAY BE USED.
C (E) DECK OF ELEMENT CARDS. EACH CARD CONTAINS 5 INTEGER FIELDS
C WHICH ARE THE ELEMENT NUMBER, AND GRID POINT NUMBERS AT THE
C VERTICES OF THE ELEMENTS. FOR TRIANGULAR ELEMENTS THE
C LAST INTEGER FIELD MUST BE BLANK OR ZERO.
C THE FORMAT IS SPECIFIED IN (D) ABOVE.
C (F) A SINGLE CARD CONTAINING THE WORD ENDEOM IN COLUMNS 1-7.

C USE IF KGEO = 2

C (A) A SINGLE CARD CONTAINING THE WORD TRIAEL IN COLUMNS 1-6 AND
C UP TO NINE NASTRAN TRIANGULAR ELEMENT CONNECTION NAMES,
C WHICH ARE LEFT-ADJUSTED IN FIELD WIDTHS OF 8, STARTING IN
C COLUMN 9 (COLS. 9-16, 17-24, ..., 73-80). THIS CARD CAN
C BE OMITTED IF TRIANGULAR ELEMENTS ARE NOT USED FOR THE PLOT
C (B) A SINGLE CARD CONTAINING THE WORD QUADEL IN COLUMNS 1-6 AND
C UP TO NINE NASTRAN QUADRILATERAL CONNECTION NAMES, WHICH
C ARE LEFT-ADJUSTED IN FIELD WIDTHS OF 8, STARTING IN COL. 9
C (COLS. 9-16, 17-24, ..., 73-80). THIS CARD CAN BE OMITTED
C IF QUADRILATERAL ELEMENTS ARE NOT USED FOR THE PLOT.
C (C) A NASTRAN BULK DATA DECK. ONLY THE GRID CARDS AND THE
C ELEMENT CONNECTION CARDS WITH NAMES MATCHING THOSE GIVEN ON
C THE TRIAEL AND QUADEL CARDS WILL BE USED FOR THE PLOT. ALL
C OTHER CARDS IN THE NASTRAN BULK DATA DECK WILL BE IGNORED.
C (D) A SINGLE CARD CONTAINING THE WORD ENDEOM IN COLUMNS 1-7.
APPENDIX C – Continued

USE IF KGEOM = 9
CALL SUBROUTINE GEOM9 WHICH IS PREPARED BY THE USER TO READ
GEOMETRY DATA.

CASE IDENTIFICATION CARD.
THIS CARD IS OMITTED IF IDCASE=0 IS SPECIFIED IN $OPTION.
IF PRESENT, THIS CARD CONTAINS ANY DESIRED ALPHANUMERIC
INFORMATION IN COLS.1-80. WILL APPEAR BEFORE EACH DATA PLOT.

DATA TO BE PLOTTED IS NOW INPUT IN ONE OF THE FOLLOWING FORMS,
DEPENDING ON THE VALUE OF KDATA SPECIFIED IN NAMELIST OPTION.

USE IF KDATA = 1

(A) A SINGLE CARD CONTAINING THE WORD FORMAT IN COLUMNS 1-6 AND
A VARIABLE FORMAT FOR THE DATA CARDS WITH LEFT PARENTHESIS
STARTING IN COLUMN 11 AND UP TO 80 COLUMNS MAY BE USED. IF
CONTROL VARIABLES ARE INCLUDED FOR MORE THAN ONE GRID
POINT OR ELEMENT PER CARD, THE NUMBER OF GRID POINTS OR
ELEMENTS PER CARD MUST BE ENTERED AS AN INTEGER IN COL. 8.
(B) DECK OF DATA TO BE PLOTTED. THERE CAN BE MULTIPLE DATA
VALUE SETS PER CARD OR THE SET CAN EXTEND TO MORE THAN ONE
CARD (OFTEN THE CASE WITH NASTRAN PUNCHED OUTPUT) WHICH CAN
BE HANDLED WITH A FORMAT FOR READING MULTIPLE CARDS.
(C) BLANK CARD OR CARDS TO END DATA DECK. THE NUMBER OF BLANK
CARDS MUST CORRESPOND TO THE NUMBER OF CARDS READ AT ONE
TIME BY THE SPECIFIED VARIABLE FORMAT.

USE IF KDATA = 5
READS NVALUES (FROM NAMELIST OPTION) SETS OF CONTROL VARIABLE
VALUES FROM TAPE20. EACH SET OF CONTROL VARIABLES MUST HAVE
BEEN WRITTEN ON TAPE20 AS AN UNFORMATTED RECORD.

USE IF KDATA = 9
CALL SUBROUTINE DATA9 WHICH IS PREPARED BY THE USER TO READ
DATA TO BE PLOTTED.

NAMELIST PICT - CONTAINS VALUES NEEDED TO GENERATE PLOTS.
THE FOLLOWING VALUES ARE INCLUDED---
APPENDIX C - Continued

NPLOT SPECIFIES TYPE OF PLOT TO BE GENERATED.
NPLOT = 1 IF ELEMENT LAYOUT WITHOUT LABELS IS DESIRED.
NPLOT = 2 IF ELEMENT LAYOUT WITH GRID PT. LABELS IS DESIRED.
NPLOT = 3 IF ELEMENT LAYOUT WITH ELEMENT LABELS IS DESIRED.
NPLOT = 4 IF CONTOUR PLOTS ARE DESIRED WITHOUT SYMBOLS AT GRID POINTS.
NPLOT = 5 IF CONTOUR PLOTS ARE DESIRED WITH SYMBOLS AT GRID POINTS.
** DEFAULT = 4 **

XORGN = X-LOCATION OF ORIGIN OF FIRST PLOT, IN INCHES.
** DEFAULT = 0.0 **

YORGN = Y-LOCATION OF ORIGIN OF FIRST PLOT, IN INCHES.
** DEFAULT = 0.0 **

PScale = MODEL SIZE REDUCTION FACTOR, PScale = ACTUAL MODEL SIZE/DESIRED PLOT SIZE.
** DEFAULT = 1.0 **

IScale = METHOD OF SCALING CONTROL VARIABLE DATA TO BE PLOTTED.

IScale = 1, FOR USER SPECIFICATION OF SCALE FACTORS.
= 2, FOR PROGRAM CALCULATION OF SCALE FACTORS TO GIVE THE USER SPECIFIED NUMBER OF SIGNIFICANT DIGITS IN ANNOTATION OF THE MAXIMUM ABSOLUTE CONTOUR LINE.
= 3, FOR PROGRAM CALCULATION OF SCALE FACTORS TO GIVE WMAGS AS THE MAXIMUM VALUE OF DATA.
** DEFAULT = 3 **

WMAGS = (FOR IScale = 1), MAGNIFICATION OF CONTROL VARIABLES FOR ANNOTATION OF CONTOUR LINES ON PLOT.
(FOR IScale = 2), NUMBER OF SIGNIFICANT DIGITS IN ANNOTATION OF MAXIMUM ABSOLUTE CONTOUR LINE.
(WMAGS = 1.0, 2.0, 3.0, ETC.)
(FOR IScale = 3), MAXIMUM VALUE OF SCALED DATA, WMAGS MUST BE LESS THAN OR EQUAL TO 2 (MAXIMUM CONTOUR LINE IS INTEGER TRUNCATION OF WMAGS).
** DEFAULT = 100.0 **

ICNTRS = (FOR IScale = 1), USER SPECIFIED CONTOUR INTERVAL (DIFFERENCE IN INTEGER VALUES OF ADJACENT CONTOUR LINES).
(FOR IScale = 2 OR 3), APPROXIMATE NUMBER OF DIFFERENT CONTOUR LINE VALUES, THE CONTOUR INTERVAL IS CALCULATED BY THE PROGRAM.
** DEFAULT = 1.0 **

XLHT = HEIGHT OF INTEGERS TO BE ANNOTATED ON PLOTS, IN INCHES.
** DEFAULT = 0.10 **

NXLAb = TOTAL NUMBER OF LINES PARALLEL TO Y-AXIS ALONG WHICH CONTOUR LINES ARE TO BE LABELED. MUST BE .LE. 10, (ALL CONTOUR LINES ARE LABELED WHERE THEY INTERSECT WITH THESE SELECTED LINES). THESE LABELS ARE IN ADDITION TO THOSE AUTOMATICALLY PROVIDED AT BOUNDARIES OF THE CONTOUR SURFACE.
** DEFAULT = 0 **

XLAb = ARRAY OF DISTANCES IN X-DIRECTION FROM THE ORIGIN TO LINES PARALLEL TO THE Y-AXIS ALONG WHICH CONTOUR LINES ARE LABELED. THERE MUST BE NXLAb OF THESE VALUES AND THEY MUST BE IN UNITS OF THE ORIGINAL (UNSCALED) MODEL.
** DEFAULT = ALL ZEROS **

NYLAb = SAME AS NXLAb FOR LABEL LOCATIONS PARALLEL TO X-AXIS.
** DEFAULT = 0 **
YLAB = SAME AS XLAB FOR LABEL LOCATIONS PARALLEL TO X-AXIS.
** DEFAULT = ALL ZEROS **
KODE - SPECIFIES CONTROL OPTION AFTER PLOT IS COMPLETE.
= 0, FOR LAST PLOT, EXIT FROM PROGRAM.
= 1, READ ANOTHER NAMELIST PICT.
= 2, READ A NEW SET OF CONTROL VARIABLE VALUES TO BE
   PLOTTED, INCLUDING A CASE IDENTIFICATION CARD IF PRESENT
= 3, READ A COMPLETE NEW SET OF INPUT DATA, INCLUDING A
   TITLE CARD.
** DEFAULT = 0 **

THE ABOVE CUMPRIS A COMPLETE BASIC SET OF INPUT DATA IF
KODE = 0 IN $PICT. FOR KODE = 1, 2, OR 3, ADDITIONAL SECTIONS OF
THE BASIC DECK MUST BE REPEATED. THE DECK MUST END WITH
NAMELIST $PICT HAVING KODE = 0.

***************************************************************

DESCRIPTION OF GRAPHICS SUBROUTINES

GRAPHICS SUBROUTINE  USED BY FOLLOWING PLOTTING DEVICE  CALLED BY FOLLOWING PROGRAM SUBROUTINES
CALCUMP            CALCCMP           INITAL       CALCCMP
LEGOY             CALCCMP           INITAL       INITAL
PSEUDO            VARIAN           INITAL       INITAL
DOC250            CRT              INITAL       INITAL
CALPLT            CALCCMP,VARIAN,CRT  MAIN,CCRT3,PSTOP,TRIN,BOUND,LAYOUT
PNTPLT            CALCCMP,VARIAN,CRT  POINTS
NGDATE            CALCCMP,VARIAN,CRT  MAIN       MAIN
NUMBER            CALCCMP,VARIAN,CRT  TRIN,BOUND,LAYOUT
NFRAME            VARIAN           MAIN,CCRT1,CCRT2,CCRT3
NEXT              CRT              MAIN,CCRT1,CCRT2,CCRT3
MESSAGE           CRT              CCRT1
PARAMS             CRT              CCRT1
KFORMAT           CRT              CCRT1

77
SUBROUTINE CALCUMP

PURPOSE: THIS IS THE NORMAL MODE PROCESSOR. THE NECESSARY
PARAMETERS AND LINKAGE ARE SET UP TO OUTPUT A TAPE
FOR THE CALCUMP 780/763 0.010/0.005-INCH STEP PLOTTER.

USE CALL CALCUMP

COMMENTS THIS CALL MUST BE GIVEN BEFORE THE FIRST CALL TO A
PLOTTING ROUTINE.

SUBROUTINE LEROY

PURPOSE THE PARAMETERS NECESSARY TO ACCOMODATE PLOTTING WITH THE
LIQUID INK PEN ARE SET UP BY CALL LEROY.

USE CALL LEROY

COMMENTS THIS CALL SHOULD ONLY BE USED WITH THE CALCUMP PROCESSOR.
IN ADDITION TO REDUCING THE SPEED OF THE PLOTTER FOR
ALL PLOTTING MOVEMENTS, THE NUMBER OF PLOT VECTORS IN ANY
ANNOTATION IS CONSIDERABLY INCREASED.

SUBROUTINE PSEUDO

PURPOSE INITIALIZES PLOT VECTOR FILE FOR VARIAN PLOTTER

USE CALL PSEUDO

SUBROUTINE CDC250

PURPOSE INITIALIZES CATHODE RAY TUBE CONSOLE.

USE CALL CDC250
APPENDIX C – Continued

SUBROUTINE CALPLT

PURPOSE
TO MOVE THE PLOTTER PEN TO A NEW LOCATION WITH PEN UP
OF DOWN AND TO SIGNAL THE END OF A JOB SEGMENT BY
INCREMENTING THE BLOCK ADDRESS NUMBER.

USE
CALL CALPLT(X,Y,IPEN)
WHERE
X,Y ARE THE FLOATING POINT VALUES FOR PEN MOVEMENT.

IPEN = 2 PEN DOWN
= 3 PEN UP

NEGATIVE IPEN WILL ASSIGN X=0, Y=0
AS THE LOCATION OF THE PEN AFTER MOVING THE
X,Y (CREATE A NEW REFERENCE POINT) AND
INCREASE THE BLOCK NUMBER BY ONE.

COMMARTS
ALL X AND Y COORDINATES MUST BE EXPRESSED AS FLOATING
POINT INCHES (ACTUAL PAGE DIMENSIONS) IN DEFLECTION FROM
THE ORIGIN.

SUBROUTINE PNTPLT

PURPOSE
TO DRAW NASA STANDARD PLOT SYMBOLS CENTERED ON A GIVEN
COORDINATE VALUE.

USE
CALL PNTPLT(A,B,NO,IS)
WHERE

A IS THE X COORDINATE FOR THE CENTERED SYMBOL
IN FLOATING POINT INCHES.

B IS THE Y COORDINATE FOR THE CENTERED SYMBOL
IN FLOATING POINT INCHES.

NO IS AN INTEGER SPECIFYING THE SYMBOL TO BE USED.

IS IS AN INTEGER VALUE SPECIFYING THE SIZE SYMBOL
TO BE USED.
SUBROUTINE NOTATE

PURPOSE TO DRAW ALPHANUMERIC INFORMATION FOR ANNOTATION AND LABELING AND PROVIDE SPECIAL CENTERED SYMBOLS FOR ANNOTATION OF DATA POINTS.

USE CALL NOTATE(X,Y,HEIGHT,BCD,THETA,N)

WHERE

X, Y ARE THE FLOATING POINT PAGE COORDINATES OF THE FIRST CHARACTER. FOR ALPHANUMERIC CHARACTERS THE COORDINATES OF THE LOWER LEFT-HAND CORNER OF THE CHARACTERS ARE SPECIFIED.

HEIGHT SPECIFIES CHARACTER SIZE AND SPACING IN FLOATING POINT INCHES FOR A FULL-SIZE CHARACTER. THE WIDTH OF A CHARACTER WILL BE (4/7) * HEIGHT AND THE SPACE BETWEEN CHARACTERS IS (2/7) * HEIGHT.

BCD IS THE STRING OF ALPHANUMERIC CHARACTERS TO BE DRAWN.

THETA IS THE ANGLE IN FLOATING POINT DEGREES AT WHICH THE INFORMATION IS TO BE DRAWN.

N IS THE NUMBER OF CHARACTERS, INCLUDING BLANKS, IN THE LABEL.
APPENDIX C – Continued

SUBROUTINE NUMBER

PURPOSE TO CONVERT A FLOATING NUMBER TO BCD (EXPRESSED IN F FORMAT), AND DRAW THE RESULTING APLHANUMERIC CHARACTERS.

USE CALL NUMBER(X,Y,SIZE,FPN,THETA,N)

WHERE

X,Y ARE THE COORDINATES IN FLOATING POINT INCHES OF THE LEFT LOWER CORNER OF THE FIRST DIGIT OF OUTPUT.

SIZE IS THE HEIGHT OF THE PLOTTED NUMBER IN FLOATING POINT INCHES.

FPN IS THE FLOATING POINT NUMBER TO BE DRAWN.

THETA IS THE ANGLE IN FLOATING POINT DEGREES AT WHICH THE NUMBER IS TO BE DRAWN.

N IS THE NUMBER OF DECIMAL DIGITS TO THE RIGHT OF THE DECIMAL POINT FOR OUTPUT. N = -1 AND N = 0 BOTH SPECIFY NO DECIMAL PLACES, HOWEVER, -1 SUPPRESSES THE DECIMAL POINT.

COMMENTS THE NUMBER IS RESTRICTED TO A MAXIMUM OF 12 DIGITS. THE ROUTINE TRUNCATES THE FLOATING POINT NUMBER AT THE REQUIRED DECIMAL PLACE.

SUBROUTINE NFRAME

PURPOSE USED BY VARIAH PLOTTER TO ADVANCE PLOTTING FRAME.

USE CALL NFRAME

SUBROUTINE NEXT

PURPOSE PROVIDES A BREAK POINT OR HALT DURING APPLICATION PROGRAM EXECUTION. OPERATOR MUST PRESS FUNCTION KEY TO RESUME, AND NUMBER OF KEY IS RETURNED IN CALLING PARAMETER.

USE CALL NEXT(NKEY)

WHERE

NKEY IS NUMBER OF FUNCTION KEY PRESSED.
APPENDIX C – Continued

SUBROUTINE MESAGE

PURPOSE PROVIDES THE CAPABILITY TO DISPLAY A MESSAGE ON THE CRT250.

USE CALL MESAGE(I,BCD,N)

WHERE

I INDICATES INTENSITY OF CHARACTER DISPLAY.

BCD IS ADDRESS OF ARRAY CONTAINING THE MESSAGE IN HOLLERITH FORM.

N IS THE NUMBER OF CHARACTERS IN THE MESSAGE (LESS THAN 50).

SUBROUTINE PARAMS

PURPOSE USED TO GENERATE A TABLE OF SYMBOLIC NAMES THAT CAN BE ACCESSED USING THE ALPHANUMERIC KEYBOARD ON THE CRT250.

USE CALL PARAMS(BCD,VAR)

WHERE

BCD IS THE HOLLERITH REPRESENTATION OF SYMBOLIC NAME.

VAR IS PROGRAM VARIABLE REFERRED TO BE SYMBOLIC NAME.

COMMENTS UP TO 3 PAIRS OF VARIABLES MAY BE SPECIFIED IN A SINGLE CALL TO PARAMS. TABLE HAS CAPACITY FOR 42 PAIRS.

SUBROUTINE KFORMAT

PURPOSE ALLOWS PROGRAMMER TO CHANGE FORMAT FOR KEYBOARD INPUT.

USE CALL KFORMAT(NHBCD)

WHERE

N IS THE NUMBER OF CHARACTERS IN BCD.

H IS REQUIRED.

BCD IS THE REQUIRED FORMAT (I4,F4.2,A10,ETC.)

RETURN
END OF DOCMNT
SUBROUTINE CCRT1

*** FOR CHANGING VALUES INPUT BY SPECT USING CRT.

COMMON/CTRL/ INFOR,KGECM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
LSCALE,ISCALE,WMA,ICNTRS,XLHT,NXLAB,XYLAB(10),NYLAB,YLAB(10)
COMMON/SAVEV/ WMAGS,ICNTRS
COMMON/GKOE/ KGDE,NPLOT
COMMON/SLCPXY/ XMINIM,XMAXIM,YMINIM,YMAXIM,XSPACE,YSPACE

*** FOR INTEGER CONTROL VALUES

CALL KF CMMAT(3H110)
CALL PARAMS
CALL PARAMS(5LNPLCT,NPLOT,6LISCALE,ISCALE,6LICNTRS,ICNTRS)
CALL PARAMS(5LNXLAB,NXLAB,5LYLAB,YLAB)
CALL MESSAGE(1,32HTO CHANGE INTEGER CONTROL VALUES,32)
CALL MESSAGE(1,13HVARIALES ARE,13)
CALL MESSAGE(1,21HNPLOT, ISCALE, 4CNTRS,21)
CALL MESSAGE(1,12HXYLAB, NYLAB,12)
CALL MESSAGE(1,17HANY KEY CONTINUES,17)
CALL NEXTINKEY

*** FOR FLOATING POINT CONTROL VALUES

CALL KFORMAT(5HF10.3)
CALL PARAMS
CALL PARAMS(5LXORGN,XORGN,5LYORGN,YORGN,6LPSCALE,PSCALE)
CALL PARAMS(5LWMAGS,WMAGS,4LXLHT,XLHT)
CALL PARAMS(5LXYLAB,YLAB(1),5LYLAB,YLAB(1))
CALL MESSAGE(1,39HTO CHANGE FLOATING POINT CONTROL VALUES,39)
CALL MESSAGE(1,13HVARIALES ARE,13)
CALL MESSAGE(1,20HXORGN, YORGN, PSSCALE,20)
CALL MESSAGE(1,11HMAGS, XLHT,11)
CALL MESSAGE(1,12HXYLAB, YLAB,12)
CALL MESSAGE(1,17HANY KEY CONTINUES,17)
CALL NEXTINKEY
RETURN
END OF CCRT1

SUBROUTINE CCRT2

*** FOR SELECTING CONTROL OPTION, KODE, AT END OF JOB USING CRT.

COMMON/GKOE/ KGDE,NPLOT
CALL MESSAGE(1,18HFN KEY 34 ENDS JOB,18)
CALL MESSAGE(1,32HFN KEY 35 TO ALTER EXISTING PLOT,32)
CALL MESSAGE(1,42HFN KEY 36 TO READ NEW SET OF DISPLACEMENTS,42)
CALL MESSAGE(1,37HFN KEY 37 TO READ A COMPLETE NEW CASE,37)
CONTINUE
CALL NEXTINKEY
IF(INKEY.LT.34.OR.INKEY.GT.37) GO TO 10
IF(INKEY.EQ.34) KGDE = 0
IF(INKEY.EQ.35) KGDE = 1
IF(INKEY.EQ.36) KGDE = 2
IF(INKEY.EQ.37) KGDE = 3
RETURN
END OF CCRT2
APPENDIX C - Continued

SUBROUTINE CCRT3

*** REMINDER TO PUT EOF ON PLOT FILE WHEN USING CRT.

CALL MESAGE(1,1H ,1)
CALL MESAGE(1,45H:LAST REMINDER TO PUT EOF ON PLOT FILE, IF ANY,45)
CALL MESAGE(1,1H ,1)
CALL MESAGE(1,40H:DO IT AT NEXT PLOT FILE COMPLETE MESSAGE,40)
CALL MESAGE(1,1H ,1)
CALL MESAGE(1,17H:ANY KEY CONTINUES,17)
CALL NEXTINKEY
CALL CALPLT(0.0,0.0,-3)
RETURN
END OF CCRT3

SUBROUTINE PSTOP

*** TC TERMINATE JOB

COMMON/CONTRL/ INFGK,KGECM,KUATA,KSIGN,KPLOT,XORGN,YORGN,
IPSCALE, ISCALE, WMAG, ICNTR, XLHT, NXLAB, XLAB(10), NYLAB, YLAB(10)
CALL CALPLT(0.0,0.0,999)
IF(KPLOT.Eq.4) CALL CCRT3
STOP
END OF PSTOP
APPENDIX C – Continued

SUBROUTINE INITIAL

*** TO SET VALUES FOR CONTROL PARAMETERS

COMMON/KCNT/ NNCUE, NELMT, NNDEST, NELEST
COMMON/CTRL/ INFOR, KGEOM, KDATA, KSIGN, KPLUT, XORGN, YORGN,
ISCALE, ISAVE, WMASS, IGNTRS
COMMON/SegNE/ NVALUS
COMMON/VALUES/ NVALUS
COMMON/SAVE/ WMAGS, IGNTRS
COMMON/SEtNC/ IRESEQ, KGEOM, KDATA, NVALUS, IRESEQ, KPLUT,
INFOR, XSPACE, YSPACE
NAMELIST/OPTION/ NNDEST, NELEST, KGEOM, KDATA, NVALUS, IRESEQ, KPLUT,
INFOR, XSPACE, YSPACE

*** TO SET DEFAULT VALUES FOR OPTION

NNDEST = 200
NELEST = 200
KGEOM = 1
KDATA = 1
NVALUS = 0
IRESEQ = 1
KPLUT = 1
INFOR = 1
XSPACE = 10.0
KSIGN = 1
ILCASE = 0

*** TO SET DEFAULT VALUES FOR $PICT

NPLUT = 4
XORGN = 0.0
YORGN = 0.0
PSCALE = 1.0
ISCALE = 3
WMASS = 100.0
IGNTRS = 10
XLAB = 0.15
NXLAB = 0
NYLAB = 0
DO 10 I = 1, 10
   XLAB(I) = 0.0
10 CONTINUE
KODE = 0
READ(5, OPTION)
IF(KPLUT.LE.2) CALL CACOMP
IF(KPLUT.EQ.2) CALL LERCY
IF(KPLUT.EQ.3) CALL PSEUDO
IF(KPLUT.EQ.4) CALL CDC250
WRITE(6, OPTION)
RETURN
END OF INITIAL
APPENDIX C – Continued

SUBROUTINE GEOM1(NUMPT,XPT,YPT,WPT,RAOPT,NMEL,NODE1,NODE2, INODE3,NODE4,CNTR)

*** TO READ GEOMETRY DATA FROM CARDS HAVING A GENERAL FORMAT

COMMON/KCOUNT/ NCODE,NELMT,NNDEST,NELEST
COMMON/CONTROL/ INFOR,KEOM,KDATA,KSIGN,KPLOT,PORTG,YORTG,
IPTSAL,ISAL,ICNTR,XMLT,NXLAB,XLAB(10),NYLAB,YLAB(10)
DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RAOPT(1),NMEL(1), INODE1(1),NODE2(1),NODE3(1),NODE4(1),CNTR(1)
DIMENSION ABCD(8)
DIMENSION FORMAT(7)
TEST1 = 10HENDGRID
TEST2 = 10HENDGECM
TEST3 = 10HFORMAT

*** TO READ GRID INFORMATION

100 CONTINUE
READ(5,10) ABCD
10 FORMAT(8A10)
IF(ABCD(1).EQ.TEST1) GO TO 1000
IF(ABCD(1).EQ.TEST2) 50,60
50 CONTINUE
DECODE(80,55,ABCD) FORMT
55 FORMAT(10X,7A10)
GO TO 100
60 CONTINUE
NNODE = NNODE+1
IF(NNODE.GT.NNDEST) 550,551
550 CONTINUE
WRITE(6,555)
555 FORMAT(1X,///,10X,*XXXXX SOORY, THE ESTIMATE OF NUMBER OF NODE POINTS WAS EXCEEDED XXXXX*)
STOP
551 CONTINUE
DECODE(80,FORMAT,ABCD) NUMPT(NNODE),XPT(NNODE),YPT(NNODE)
GO TO 100
1000 CONTINUE

*** TO READ ELEMENT CONNECTION INFORMATION

200 CONTINUE
READ(5,10) ABCD
IF(ABCD(1).EQ.TEST2) GO TO 2000
IF(ABCD(1).EQ.TEST3) 250,260
250 CONTINUE
DECODE(80,55,ABCD) FORMT
GO TO 200
260 CONTINUE
NELMT = NELMT+1
IF(NELMT.GT.NELEST) 650,651
650 CONTINUE
WRITE(6,655)
655 FORMAT(1X,///,10X,*XXXXX SOORY, THE ESTIMATE OF NUMBER OF ELEM LENTS WAS EXCEEDED XXXXX*)
STOP
APPENDIX C – Continued

651 CONTINUE
  DECODE(80,FMT,ABCD) NUMEL(NELMT),NODE1(NELMT),NODE2(NELMT),
  INODE3(NELMT),NODE4(NELMT)
  GO TO 200
2000 CONTINUE
  RETURN
  END OF GEUM1
SUBROUTINE GEOM2 (NUMPT, XPT, YPT, WPT, RADPT, NUMEL, NODE1, NODE2, INODE3, INODE4, CENTR)

C *** TO READ NASTRAN GEOMETRY DATA

COMMON/KOUNT/NNODE, NELMT, NNDEST, NELEST
COMMON/CONTRL/INFOR, KGEOM, KDATA, KSIGN, KPLT, XORGN, YORGN,
  LPScale, IScale, MAG, ICNTR, XLHT, NXLAB, XLAB(10), NYLAB, YLAB(10)
DIMENSION NUMPT(1), XPT(1), YPT(1), WPT(1), RADPT(1), NUMEL(1),
  INODE1(1), NODE2(1), NODE3(1), NODE4(1), CENTR(1)
DIMENSION ABCO(8)
DIMENSION NDPT(4)
DIMENSION TRIAEL(9), QUADEL(9)

10 FORMAT (8A10)
  TEST0 = 'LOH'
  TEST1 = 'LOHGRID'
  TEST3 = 'LOHTRIAEL'
  TEST4 = 'LOHQUADEL'
  TEST5 = 'LOHENDGECM'
  DD 15 I = 1, 9
  TRIAEL(I) = 'LOH'
  QUADEL(I) = 'LOH'
15 CONTINUE

800 CONTINUE
  READ(5,10) ABCD
  IF(EOF(5), 1111, 750)
750 CONTINUE
  DECODE(80, 50, ABCD, WORD1)
50 FORMAT (8A8)
  IF(WORD1.EQ.TEST0) GO TO 800
  IF(WORD1.EQ.TEST1) GO TO 200
  IF(WORD1.EQ.TEST3) 62, 63
62 DECODE(80, 101, ABCD) (TRIAEL(I), I = 1, 9)
101 FORMAT (8X, 9A8)
  GO TO 600
63 CONTINUE
  IF(WORD1.EQ.TEST4) 64, 65
64 DECODE(80, 101, ABCD) (QUADEL(I), I = 1, 9)
  GO TO 800
65 CONTINUE
  DD 70 I = 1, 9
  IF(WORD1.EQ.TRIAEL(I)) GO TO 300
  IF(WORD1.EQ.QUADEL(I)) GO TO 400
70 CONTINUE
  IF(WORD1.EQ.TEST5) GO TO 2000
  GO TO 800

C *** TO READ GRID CARDS

C

200 CONTINUE
  DECODE(60, 201, ABCD) IDUM, XDUM, YDUM
201 FORMAT (8X, A8, 8X, 2F8.0)
  CALL IKITE(IDUM)
  NNODE = NNODE+1
  IF(NNODE.GT.NNDEST) 550, 551
APPENDIX C – Continued

550 CONTINUE
WRITE(6,555)
555 FORMAT(1X,///,1X,*XXXXXXX SORRY, THE ESTIMATE OF NUMBER OF NODE
1 POINTS WAS EXCEEDED XXXXXX*)
STOP
551 CONTINUE
NUMPT(INODE) = ICUM
XPT(INODE) = XDUM
YPT(INODE) = YDUM
GO TO 300
C
C *** TO READ CARDS CONTAINING ELEMENTS WITH 3 GRID POINTS
C
300 CONTINUE
DECOD(180,301,ABCC) IDUM,NDPT(1),NDPT(2),NDPT(3)
301 FORMAT(8X,A8,8X,3A8)
CALL IRITE(IDUM)
CALL IRITE(NDPT(1))
CALL IRITE(NDPT(2))
CALL IRITE(NDPT(3))
NELMT = NELMT+1
IF(NELMT.GT.NELEST) 750,651
650 CONTINUE
WRITE(6,556)
656 FORMAT(1X,///,1X,*XXXXXXX SORRY, THE ESTIMATE OF NUMBER OF ELEM
ENTS WAS EXCEEDED XXXXXX*)
STOP
651 CONTINUE
NUMEL(NELMT) = IDUM
NODE1(NELMT) = NDPT(1)
NODE2(NELMT) = NDPT(2)
NODE3(NELMT) = NDPT(3)
NODE4(NELMT) = 0
GO TO 800
C
C *** TO READ CARDS CONTAINING ELEMENTS WITH 4 GRID POINTS
C
400 CONTINUE
DECOD(180,401,ABCC) IDUM,NDPT(1),NDPT(2),NDPT(3),NDPT(4)
401 FORMAT(8X,A8,8X,4A8)
CALL IRITE(IDUM)
CALL IRITE(NDPT(1))
CALL IRITE(NDPT(2))
CALL IRITE(NDPT(3))
CALL IRITE(NDPT(4))
NELMT = NELMT+1
IF(NELMT.GT.NELEST) 751,751
751 CONTINUE
WRITE(6,556)
STOP
751 CONTINUE
NUMEL(NELMT) = IDUM
NODE1(NELMT) = NDPT(1)
NODE2(NELMT) = NDPT(2)
NODE3(NELMT) = NDPT(3)
NODE4(NELMT) = NDPT(4)
GO TO 800
C
APPENDIX C - Continued

SUBROUTINE IRITE(NUM)
C
C *** TO RIGHT ADJUST INTEGERS IN A FIELD WIDTH OF EIGHT
C
DIMENSION N(8)
LANK = 1H
DECOD(8,1,NUM) NSAVE
1 FORMAT(18)
DECOD(8,2,NUM) (N(I),I=1,8)
2 FORMAT(8A1)
DU 10 I=1,8
II = 9-1
IF(N(II).NE.LANK) GO TO 20
10 CONTINUE
20 NUM = NSAVE/(10**(8-II))
RETURN
END OF IRITE

SUBROUTINE GEOM9(NUMPT,XPT,YPT,WPT,RADPT,NUMEL,NODE1,NODE2,NODE3,NODE4,CENTR)
C
C *** USER SUPPLIED GEOMETRY INPUT ROUTINE
C
COMMON/KOUNT/NNODE,NELMT,NNEST,NELEST,
COMMON/CONTROL/INFOR,KGECM,KCATA,KSIGN,XPLOT,XORGN,YORGN,
IPSCALE,IPSCALE,WMAG,ICNTR,XLHT,NXLAB,XYLAB(10),NYLAB,YLAB(10)
DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
INODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
C
C *** INSERT ROUTINE HERE
C
RETURN
END OF GEOM9
SUBROUTINE ZEROW(NUMPT, UPT, NUMEL, CENTR)

*** TO ZERO OUT VALUES OF DATA TO BE PLOTTED.

COMMON/KCOUNT/ NNCHE, NELMT, NNDEST, NELEST
 COMMON/CONTROL/ INFOR, KGEOM, KDATA, KSIGN, KBLOT, XORG, YORG,
                  IPSCALE, ISCALE, wMAG, ICNTR, XLHT, NXLAB, YLAB, YLAB(10)
 DIMENSION NUMPT(1), wPT(1), NUMEL(1), CENTR(1)

GO TO (100, 200) INFOR
100 CONTINUE
   DO 110 I=1, NNDEST
       UPT(I) = 0.0
 110 CONTINUE
   GO TO 500
200 CONTINUE
   DO 210 I=1, NELEST
       CENTR(I) = 0.0
 210 CONTINUE
500 CONTINUE
RETURN
END OF ZEROW
SUBROUTINE DATAU(NUMPT, HPT, NUMEL, CENTR)

*** TO READ THE DATA TO BE PLOTTED ***

COMMON/KUUNT/ NNGOE, NLEM, NNNEEST, NELEST
COMMON/CONTRL/ INFOR, KGEOM, KDATA, KSIGN, KPLTR, XORGN, YORGN,
IPSCALE, ISCALE, WMAG, ICNTR, XLHT, NXLAB, XLAB(10), NYLAB, YLAB(10)
COMMON/SEQUENCE/ IRESEQ
DIMENSION NUMPT(1), HPT(1), NUMEL(1), CENTR(1)
DIMENSION ABCD(8)
DIMENSION FORMT(7)
DIMENSION ISAV(10), OSAV(10)

10 FORMAT (dAlu)
TEST = 'HFFORMAT
READ(5, 10) ABCD
DECODE(80, 45, ABCD) WORD, KVALU

5 FORMAT (AI, IX, 1 l)
IF (KVALU.EQ.O) KVALU = 1
IF (WORD.EQ. TEST) 300, 200
200 WRITE(6, 20)
20 FORMAT (' H1, ///, 2Gx, * SORRY, FORMAT FOR DATA NOT GIVEN*)
STOP
300 CONTINUE
DECODE(80, 50, ABCD) FORMT
50 FORMAT (10X, 7AI)
100 CONTINUE
READ(5, FORMT) (ISAV(K), OSAV(K), K=1, KVALU)
GO TO 1310, 610) INFOR

*** FOR DATA AT GRID POINTS ***

510 CONTINUE
DO 550 K=1, KVALU
IDUM = ISAV(K)
IF (IDUM.EQ.O) GO TO 1000
WDUM = OSAV(K)
IF (IRSEQ.EQ.1) GO TO 520
HPT(IDUM) = WDUM
GO TO 560
520 CONTINUE
DO 550 J=1, NNGOE
IF (NUMPT(J).EQ.IDUM) 501, 500
501 HPT(J) = WDUM
GO TO 550
500 CONTINUE
550 CONTINUE
GO TO 100

*** FOR DATA AT ELEMENT CENTROIDS ***

610 CONTINUE
DO 660 K=1, KVALU
IDUM = ISAV(K)
IF (IDUM.EQ.O) GO TO 1000
WDUM = OSAV(K)
IF (IRSEQ.EQ.1) GO TO 620
CENTR(IDUM) = WDUM
GO TO 660

92
APPENDIX C - Continued

520 CONTINUE
DO 600 J=1,NELMT
IF(NUMEL(J).EQ.IDUM) 601,600
501 CENTR(J) = WDOM
GO TO 500
600 CONTINUE
650 CONTINUE
660 CONTINUE
GO TO 100
1000 CONTINUE
RETURN
END OF DATA1

SUBROUTINE DATA5(NUMPT,WPT,NUMEL,CENTR)
C *** TO READ DISPLACEMENT DATA FROM TAPE20
C
C COMMUN/ KOUNT/ KCODE,NELMT,KADEST,NELEST
C COMMUN/ CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORG,
C IPSCALE,ISCALE,CMAG,ICNTR,XLFT,NXLAB,XLAB(10),NYLAB,YLAB(10)
C COMMUN/ VALUES/ NVALUES
C COMMUN/ SEQUENCE/ IRESEQ
C DIMENSION NUMPT(1),WPT(1),NUMEL(1),CENTR(1)
C DO 10 I=1,NVALUES
READ(20) (IDUM,WDOM)
GO TO (510,510) INFOR
C
C *** FOR DATA AT GRID POINTS
C
510 CONTINUE
IF(IRESEQ.EQ.1) GO TO 520
WPT(IDUM) = WDOM
GO TO 10
520 CONTINUE
DO 500 J=1,NNODE
IF(NUMPT(J).EQ.IDUM) 501,500
501 WPT(J) = WDOM
GO TO 10
500 CONTINUE
GO TO 10
C
C *** FOR DATA AT ELEMENT CENTROIDS
C
610 CONTINUE
IF(IRESEQ.EQ.1) GO TO 620
WPT(IDUM) = WDOM
GO TO 10
620 CONTINUE
DO 600 J=1,NELMT
IF(NUMEL(J).EQ.IDUM) 601,600
601 CENTR(J) = WDOM
GO TO 10
600 CONTINUE
10 CONTINUE
RETURN
END OF DATA5

93
APPENDIX C – Continued

SUBROUTINE DATA9(NUMPT,WPT,NUMEL,CENTR)

*** USER SUPPLIED DATA INPUT ROUTINE

COMMON/KOUNT/, NNODE, NELMT, NADEST, NELEST
COMMON/CTRL/, INFOR, KGEOM, KDATA, KSIGN, KPLOT, XORGN, YORGN,
IPSCALE, ISCALE, WMAG, ICNTR, XLHT, NXLAB, XLAB(10), NYLAB, YLAB(10)
DIMENSION NUMPT(1), WPT(1), NUMEL(1), CENTR(1)

*** INSERT ROUTINE HERE

RETURN
END OF DATA9
APPENDIX C – Continued

SUBROUTINE PNTOUT(IOUT,NUMPT,XPT,YPT,WPT,RADPT,NUMEL,
INODE1,NODE2,NODE3,NODE4,CENTR)

C *** FOR PRINTED OUTPUT OF INFORMATION IN BLANK COMMON – ZZZ

COMMON/KMNUN/KOUNT/ NNUCE,NELMT,NNCEST,NELEST
COMMON/CONTRL/ INFOR,KGECM,KDATA,KSIGN,KPLOT,XXORGN,YORGN,
IPSCL,ISCL,WMAG,ICNTR,XLAB,YLAB10,1,NYLAB,YLAB10
DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1)
DIMENSION NUMEU1,NODE1,NODE2,NODE3,NODE4,CENTR(1)

GO TO (1000,2000) ICUT

1000 CONTINUE

C *** FOR OUTPUT OF GEOMETRY INFORMATION

WRITE(6,16)
16 FORMAT(///,5X,*GRID POINT INFORMATION*,///)
WRITE(6,17)
17 FORMAT(5X,*RESEQUENCED*,4X,*USER INPUT*,35X,*BOUNDARY*/
16X,*GRID POINT*,5X,*GRID POINT*,7X,*INPUT*,10X,*INPUT*,
26X,*POINT*/
36X,*NUMBER*,9X,*NUMBER*,13X,*X*,14X,*Y*,10X,*INDICATOR*/)
DO 30 I = 1,NNODE
WRITE(6,18) I,NUMPT(I),XPT(I),YPT(I)
30 CONTINUE
WRITE(6,19)
19 FORMAT(///,5X,*ELEMENT INFORMATION – WITH RESEQUENCED GRID POINTS
1/*/)
WRITE(6,21)
21 FORMAT(5X,*RESEQUENCED*,4X,*USER INPUT*,19X,*GRID POINTS*/
15X,*ELEMENT*,8X,*ELEMENT*/
26X,*NUMBER*,9X,*NUMBER*,13X,*X*,14X,*Y*,10X,*INDICATOR*/)
DO 35 I = 1,NELMT
WRITE(6,22) I,NUMEU(I),NODE1(I),NODE2(I),NODE3(I)
35 CONTINUE
RETURN
2000 CONTINUE

C *** FOR OUTPUT OF DATA TO BE PLOTTED

WRITE(6,210)
210 FORMAT(///,5X,*DATA TO BE PLOTTED*,///)
IF(INFOR.EQ.1) WRITE(6,220)
220 FORMAT(5X,*RESEQUENCED GRID POINT NUMBERS AND DATA VALUES*/)
IF(INFOR.EQ.2) WRITE(6,230)
230 FORMAT(5X,*RESEQUENCED ELEMENT NUMBERS AND DATA VALUES*/)
IF(INFOR.EQ.1) WRITE(6,240) ((I,WPT(I)),I=1,NNODE)
IF(INFOR.EQ.2) WRITE(6,240) ((I,CENTR(I)),I=1,NELMT)
240 FORMAT(2X,18,E15.5)
RETURN
END OF PNTOUT
APPENDIX C - Continued

SUBROUTINE XYSCAL(NUMPT, XPT, YPT, WPT, RADPT, NUMEL, NODE1, NODE2, INODE3, NODE4, CENTR)

*** TO DETERMINE MAXIMUM DIMENSIONS OF THE DATUM SURFACE

*** AND RENUMBER ELEMENT GRID POINTS

*** If DATA IS SCALED IN SUBROUTINE SCALEW

COMMON/KOUNT/ NNODE, NELMT, NNODE2, NNODE3, NNODE4
COMMON/CTRL/ INFOR, KECM, KDATA, KSIGN, KPOINT, XORG, YORG,
COMMON/SCALE/ ISCALE, WMAG, ICNTR, XMAX, YMAX, XLAB(10), YLAB(10)
COMMON/SEGNCE/ IRESEQ
COMMON/SCALE/ XMINIM, XMAXIM, YMINIM, YMAXIM, XSACE, YSPACE
DIMENSION NUMPT(1), XPT(1), YPT(1), WPT(1), RADPT(1), NUMEL(1), INODE1(1), NODE2(1), NODE3(1), NODE4(1), CENTR(1)
DIMENSION NNODE(4)

*** TO SCALE DATA AND DETERMINE X MINIMUM AND MAXIMUM

XMINIM = XPT(1)
XMAXIM = XPT(1)
YMINIM = YPT(1)
YMAXIM = YPT(1)
DO 110 I=1, NNODE
   IF(XPT(I).LT.XMINIM) XMINIM = XPT(I)
   IF(XPT(I).GT.XMAXIM) XMAXIM = XPT(I)
   IF(YPT(I).LT.YMINIM) YMINIM = YPT(I)
   IF(YPT(I).GT.YMAXIM) YMAXIM = YPT(I)
110 CONTINUE
   IF(IRESEQ.NE.1) GO TO 700

*** TO RENUMBER ELEMENT NODES

DO 600 I=1, NELMT
   NOPT(1) = NODE1(I)
   NOPT(2) = NODE2(I)
   NOPT(3) = NODE3(I)
   NOPT(4) = NODE4(I)
   DO 605 J=1, NNODE
      IF(NOPT(J).EQ.0) GO TO 605
      DO 610 K=1, NNODE
         IF(NUMPT(K).EQ.NOPT(J)) GO TO 615, 610
      610 CONTINUE
   605 CONTINUE
   WRITE(6,555) NOPT(J), NUMEL(I)
555 FORMAT(5X, *, NODE *, I10, *, IN ELEMENT *, I10, *, NOT IN USER DATA*)
STOP
  600 CONTINUE
  605 CONTINUE
  600 CONTINUE
RETURN
END OF XYSCAL
APPENDIX C – Continued

SUBROUTINE SCALEW(wPT, CENTR)

*** THIS SUBROUTINE SCALES W DATA

COMMON/XK_TIME/ NNODE, NELMT, NNODEST, NELEST
COMMON/CURTCL/ INFOR, KGEGM, KDATA, KSIGN, KPLTS, XORGN, YORGN,
  ISCALE, ISCALE, WMAG, ICNTR, XLMT, NLAB, XLAB(I0), NLAB, YLAB(I0)
COMMON/SAVEV/ WMAGS, ICNTRS
DIMENSION wPT, CENTR

GO TO (100, 200) INFOR

*** FOR NODAL DATA

100 CONTINUE
WMIN = WPT(I)
WMAX = WPT(I)
DO 120 I = 1, NNODE
IF(WPT(I) .LT. WMIN) WMIN = WPT(I)
IF(WPT(I) .GT. WMAX) WMAX = WPT(I)
120 CONTINUE
WBIG = ABS(WMIN)
IF(ABS(WMAX) .GT. WBIG) WBIG = ABS(WMAX)
WRANG = ABS(WMAX - WMIN)
WRITE(I2, 121) WMAX, WMIN,
121 FORMAT('MAXIMUM VALUE OF DATA =*', E20.8, 'MINIMUM VALUE OF DATA =', E20.8)
GO TO (140, 160, 180) ISCALE
140 CONTINUE
WMAG = WMAGS
ICNTR = ICNTRS
GO TO 190
160 CONTINUE
WMAG = WMAGS
LEXP = ALOG10(WBIG)
IF(LEXP .LT. 0) LEXP = LEXP - 1
IF(LEXP .EQ. 0) GO TO 161
WMAG = WMAGS/(10.0**LEXP)
161 CONTINUE
WRANG = WRANG*WMAG
CINT = WRANG/FLOAT(ICNTRS)
GO TO 800
180 CONTINUE
IF(WMAGS .LT. 2.0) 333, 334
333 WRITE(6, 335)
335 FORMAT('WMAG MUST BE 2.0 OR GREATER')
STOP
334 CONTINUE
WMAG = WMAGS/WBIG
WRANG = WRANG*WMAG
CINT = WRANG/FLOAT(ICNTRS)
800 CALL INTRVL(CINT, ICNTR)
190 CONTINUE
WRITE(I2, 122) WMAG, ICNTR
122 FORMAT('WMAG =*', E20.8, 'ICNTR =*', I6)
GO TO 300

*** FOR ELEMENT DATA

200 CONTINUE
CMIN = CENTR(i)  
CMAX = CENTR(i)  
DO 220 I = 1,NELMT  
IF(CENTR(I).LT.CMIN) CMIN = CENTR(I)  
IF(CENTR(I).GT.CMAX) CMAX = CENTR(I)  
CBIG = ABS(CMIN)  
IF(ABS(CMAX).GT.CBIG) CBIG = ABS(CMAX)  
CHANG = ABSCMAX-CMIN)  
220 CONTINUE  
WRITE(6,121) CMAX,CMIN  
GO TO (240,260,280) ISCALE  
240 CONTINUE  
WMAG = WMAGS  
ICNTR = ICNTRS  
GO TO 290  
260 CONTINUE  
WMAG = WMAGS  
LEXP = ALOG10(CBIG)  
IF(LEXP.LT.0) LEXP = LEXP-1  
IF(LEXP.EQ.0) GO TO 261  
WMAG = WMAGS/(10.0**LEXP)  
261 CONTINUE  
CRANG = CRANG*WMAG  
CINT = CRANG/FLOAT(ICNTRS)  
GO TO 900  
280 CONTINUE  
WMAG = WMAGS/CBIG  
CRANG = CRANG*WMAG  
CINT = CRANG/FLOAT(ICNTRS)  
900 CALL INTRVL(CINT,ICNTR)  
290 CONTINUE  
WRITE(6,122) WMAG,ICNTR  
300 CONTINUE  
RETURN  
END OF SCALEW
SUBROUTINE INTRVL(CNT, ICNT)

C *** TO CALCULATE COUNTER INTERVAL, ICNT
C
DO 165 I = 1, 10
CUP = 1.5*(10.0**I)/10.0
IF(CNT<.LT.CUP) 166, 167
160 ICNT = I*(10**I)/10
GO TO 140
167 CONTINUE
CUP = 3.5*(10.0**I)/10.0
IF(CNT<.LT.CUP) 166, 169
168 ICNT = 2*(10**I)/10
GO TO 140
169 CONTINUE
CUP = 7.5*(10.0**I)/10.0
IF(CNT<.LT.CUP) 170, 171
170 ICNT = 3*(10**I)/10
GO TO 140
171 CONTINUE
165 CONTINUE
WRITE(6,175)
175 FORMAT(1X,///,5X,*SURRY, COUNTER INTERVAL NOT FOUND, CHECK INPUT*)
STOP
140 CONTINUE
RETURN
END OF INTRVAL
SUBROUTINE ENCLOSED(XOPT, XPT, YPT, WPT, RADPT, NUMEL, NODE1, NODE2,
                  NODE3, NODE4, CENTR)

C *** TO CALCULATE ENCLOSED ANGLES AROUND NODE POINTS
C
COMMON/KOONT/ NNODE, NELMT, NNDEST, NELEST
DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
                  NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
DIMENSION XX(3),YY(3),RRK(3),RRAD(3)
DO 25 I=1,NNDEST
  25 RADPT(I) = 0.0
  DO 500 I=1,NELMT
     ND1 = NODE1(I)
     ND2 = NODE2(I)
     ND3 = NODE3(I)
     ND4 = NODE4(I)
     XX(1) = XPT(ND1)
     YY(1) = YPT(ND1)
     XX(2) = XPT(ND2)
     YY(2) = YPT(ND2)
     XX(3) = XPT(ND3)
     YY(3) = YPT(ND3)
     CALL ARCSINO(XX,YY,RRAD)
     RADPT(ND1) = RADPT(ND1)+RRAD(1)
     RADPT(ND2) = RADPT(ND2)+RRAD(2)
     RADPT(ND3) = RADPT(ND3)+RRAD(3)
     IF(NODE(I).EQ.0) GO TO 505
     XX(2) = XPT(NODE(I))
     YY(2) = YPT(NODE(I))
     CALL ARCSINO(XX,YY,RRAD)
     RADPT(ND1) = RADPT(ND1)+RRAD(1)
     RADPT(ND2) = RADPT(ND2)+RRAD(2)
     RADPT(ND3) = RADPT(ND3)+RRAD(3)
C 505 CONTINUE
  500 CONTINUE
  PI = 3.14159265
  EPSIL = 5.0*PI/180.0
  TEST = 2.0*PI-EPSIL
  DO 600 I=1,NNODE
     IF(RADPT(I).LT.TEST) GO TO 601
  601 CONTINUE
     RADPT(I) = 1.0
     GO TO 600
  602 CONTINUE
     RADPT(I) = 0.0
  600 CONTINUE
RETURN
END OF ENCLOSED

100
SUBROUTINE AROUND(X,Y,RAD)
C
C *** DETERMINES A SPECIFIED CORNER ANGLE OF AN ELEMENT
C
DIMENSION X(3),Y(3),RAD(3)
UGNE = SQRT((X(3)-X(2))**2+(Y(3)-Y(2))**2)
UTWO = SQRT((X(3)-X(1))**2+(Y(3)-Y(1))**2)
OTRI = SQRT((X(2)-X(1))**2+(Y(2)-Y(1))**2)
COS1 = (UTWO**2+CTRI**2-UGNE**2)/(2.0*UTWO*OTRI)
COS2 = (UGNE**2+CTRI**2-UTWO**2)/(2.0*UGNE*OTRI)
COS3 = (UGNE**2+CTWO**2-OTRI**2)/(2.0*UGNE*OTWO)
RAD(1) = ACOS(COS1)
RAD(2) = ACOS(COS2)
RAD(3) = ACOS(COS3)
RETURN
END OF AROUND

SUBROUTINE POINTS(KIND,ISIZE,XPT,YPT)
C
C *** TO PLOT SYMBOLS AT NODE POINTS
C
COMMON/KCOUNT/ NCDE,NELMT,NNDEST,NELEST
COMMON/CTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
LSCALE,LSCALE,WMAJ,INCNI,NLHT,NXLAB,YLAB(10)
DIMENSION XPT(ISIZE),YPT(ISIZE)
DO 100 I=1,NCDE
X = XPT(I)/SCALE
Y = FLAT(KSIGN)*YPT(I)/SCALE
CALL PNTPLT(X,Y,KIND,ISIZE)
100 CONTINUE
RETURN
END OF POINTS
SUBROUTINE WNOOE(NUMPT, XPT, YPT, WPT, RADPT, NUMEL, NODE1, NODE2, NODE3, NODE4, CENTR)
C
C *** FOR PLOTTING NODAL DATA
C
COMMON/NODE, NELMT, NNDST, NELEST
COMMON/CONTROL/INCR, KGEOM, KDATA, KSIGN, KPLOT, XORGN, YORGN,
IPSCALE, ISCALE, WMAG, ICNTR, XLHT, NXLAB, XLAB(10), YLAB, YLAB(10)
DIMENSION NUMPT(1), XPT(1), YPT(1), WPT(1), RADPT(1), NUMEL(1),
NODE1(1), NODE2(1), NODE3(1), NODE4(1), CENTR(1)
DIMENSION XX(3), YY(3), WW(3), RRAD(3)
DO 500 I=1, NELMT
   NO1 = NODE1(I)
   NO2 = NODE2(I)
   NO3 = NODE3(I)
   NO4 = NODE4(I)
   XX(1) = XPT(NO1)
   YY(1) = YPT(NO1)
   WW(1) = WPT(NO1)
   XX(2) = XPT(NO2)
   YY(2) = YPT(NO2)
   WW(2) = WPT(NO2)
   XX(3) = XPT(NO3)
   YY(3) = YPT(NO3)
   WW(3) = WPT(NO3)
   CALL TRIN(XX, YY, WW)
   IF(NO4.EQ.0) GO TO 505
   XX(2) = XPT(NO4)
   YY(2) = YPT(NO4)
   WW(2) = WPT(NO4)
   CALL TRIN(XX, YY, WW)
505 CONTINUE
500 CONTINUE
RETURN
END OF WNODE
APPENDIX C – Continued

SUBROUTINE WELMT( NUMPT, XPT, YPT, WPT, RAOPT, NUMEL, NODE1, NODE2, INODE3, NODE4, CENTR)

C *** FOR PLOTTING ELEMENT DATA

    COMMON/KUNT/ NNCUE, NELMT, NNOEST, NELEST
    COMMON/CUNTRL/ INFOR, KGEOM, KCATA, KSIGN, KPLOT, XORGN, YORGN,
    ISCALE, SCALE, WMAG, ICNTR, XLHT, NXLAB, XLAB(10), YLAB(10)
    DIMENSION NUMPT(1), XPT(1), WPT(1), RAOPT(1), NUMEL(1),
    NODE1(1), NODE2(1), NODE3(1), NODE4(1), CENTR(1)
    DIMENSION NDSAV(4), NEL(20), X(20), Y(20), R(20), ANG(20)
    DIMENSION XX(3), YY(3), WM(3)
    DIMENSION ISAV(4), JSAV(4)
    PI = 3.1415926536
    DO 100 I = 1, NNUUE

C *** DETERMINE ELEMENTS CONNECTED TO NODE 1

    KKEEP = 0
    DO 500 J = 1, NELMT
    NDSAV(1) = NODE1(J)
    NDSAV(2) = NODE2(J)
    NDSAV(3) = NODE3(J)
    NDSAV(4) = NODE4(J)
    DO 505 JJ = 1, 4
    IF(NDSAV(JJ) .EQ. 0) 510, 505
    505 NKEEP = NKEEP + 1
    NEL(NKEEP) = J
    CONTINUE
    500 CONTINUE
    IF(NKEEP .EQ. 0) GC TO 100

C *** CALCULATE INFORMATION FOR CENTROID OF EACH CONNECTED ELEMENT

    DO 550 K = 1, NKEEP
    IDUM = NEL(K)
    ND1 = NODE1(IDUM)
    ND2 = NODE2(IDUM)
    ND3 = NODE3(IDUM)
    ND4 = NODE4(IDUM)
    IF(ND4 .EQ. 0) 560, 561
    550 CONTINUE
    X(K) = (XPT(ND1) + XPT(ND2) + XPT(ND3)) / 3.0
    Y(K) = (YPT(ND1) + YPT(ND2) + YPT(ND3)) / 3.0
    GO TO 565

560 CONTINUE
    X(K) = (XPT(ND1) + XPT(ND2) + XPT(ND3) + XPT(ND4)) / 4.0
    Y(K) = (YPT(ND1) + YPT(ND2) + YPT(ND3) + YPT(ND4)) / 4.0

555 CONTINUE
    XDIST = X(K) - XPT(I)
    YDIST = Y(K) - YPT(I)
    K(K) = SQRT(XDIST**2 + YDIST**2)
    IF(XDIST .EQ. 0.0 .AND. YDIST .GE. 0.0) ANG(K) = PI / 2.0
    IF(XDIST .EQ. 0.0 .AND. YDIST .LT. 0.0) ANG(K) = 3.0 * PI / 2.0
    IF(XDIST .NE. 0.0) ANG(K) = ATAN2(YDIST, XDIST)
    IF(ANG(K) .LT. 0.0) ANG(K) = ANG(K) + 2.0 * PI
    550 CONTINUE

103
APPENDIX C - Continued

C *** TO REORDER INFORMATION IN ORDER OF INCREASING ANGLE
C
IF(NKEEP.EQ.1) GO TO 600
DO 570 K=1,NKEEP
KK = NKEEP-K
DO 570 J=1, KK
IF(ANG(J)-ANG(J+1)) 575, 575, 570, 570, 575
575 TEMP1 = ANG(J)
ITEMP2 = NEL(J)
TEMP3 = X(J)
TEMP4 = Y(J)
TEMP5 = R(J)
ANG(J) = ANG(J+1)
NEL(J) = NEL(J+1)
X(J) = X(J+1)
Y(J) = Y(J+1)
R(J) = R(J+1)
ANG(J+1) = TEMP1
NEL(J+1) = TEMP2
X(J+1) = TEMP3
Y(J+1) = TEMP4
R(J+1) = TEMP5
570 CONTINUE
600 CONTINUE
C
C *** TO CALCULATE WEIGHTED AVERAGE AT NODE POINT
C
DENOM = 0.0
DO 610 K=1,NKEEP
DENOM = DENOM+1.0/R(K)
610 CONTINUE
WEIGHT = 1.0/R(K)/DENOM
WPT(I) = WPT(I)+WEIGHT*CENTR(IDUM)
615 CONTINUE
C
C *** TO PLOT CONTOURS AROUND POINT
C
IF(NKEEP.EQ.1) GO TO 625
XX(I) = XPT(I)
YY(I) = YPT(I)
WM(I) = WPT(I)
DO 620 K=1,NKEEP
11 = K
I2 = K+1
IF(K.EQ.NKEEP) I2 = 1
IF(RAOP(1).EQ.C.0) 660, 650, 660, 650
C *** TO CHECK CONNECTIVITY OF CENTROIDS ABOUT A BOUNDARY POINT
650 CONTINUE
IDUM = NEL(I1)
J Dum = NEL(I2)
ISAV(1) = NODE1(Icum)
ISAV(2) = NODE2(Icum)
ISAV(3) = NODE3(Icum)
ISAV(4) = NODE4(Icum)
JSAV(I) = NODE1(Jcum)

104
APPENDIX C – Continued

JSAV(2) = NODE2(JDUM)
JSAV(3) = NODE3(JDUM)
JSAV(4) = NODE4(JDUM)
DU 651 III=1,4
IF(ISA1(1II).EQ.0.OR.JSA1(II1).EQ.1) GO TO 651
DU 652 JJJ=1,4
IF(JSAV(JJJ).EQ.0.OR.JSAV(JJJJ).EQ.1) GO TO 652
IF(ISAV(1II).EQ.JSAV(JJJJ)) 660,652
652 CONTINUE
651 CONTINUE
GO TO 620
660 CONTINUE
XX(2) = X(I1)
YY(2) = Y(I1)
IDUM = NEL(I1)
WW(2) = CENTK(IDUM)
XX(3) = X(I2)
YY(3) = Y(I2)
IDUM = NEL(I2)
WW(3) = CENTK(IDUM)
CALL TRLN(XX,YY,WW)
620 CONTINUE
625 CONTINUE
100 CONTINUE
RETURN
END OF \WELMT
SUBROUTINE TRIN(XG,YG,WG)

C *** PLOTS CONTOUR LINES WITHIN A GIVEN TRIANGLE

COMMON/CTRL/ INFOR,KGECM,KCATA,KSIGN,KPLOT,XORGN,YORGN,
PSCALE,QSCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
DIMENSION XG(3),YG(3),WG(3)
DIMENSION X(3),Y(3),W(3)
DIMENSION XPNT(3),YPNT(3)

C *** TO SCALE w DATA BEFORE PLOTTING

DO 10 I = 1,3
   XI(I) = XG(I)/PSCALE
   YI(I) = FLOAT(KSIGN)*YG(I)/QSCALE
   WI(I) = WG(I)*WMAG
10 CONTINUE

C *** TO FIND MIN AND MAX VALUES OF W

   WMIN = WI(1)
   WMAX = WI(1)
   DO 20 I = 2,3
      IF(WI(I).LT.WMIN) WMIN = WI(I)
      IF(WI(I).GT.WMAX) WMAX = WI(I)
20 CONTINUE

C *** TO DETERMINE CONSTANT VALUES OF w WITHIN TRIANGLE

   IDUM = WMIN/ICNTR
   JDOM = IDUM*ICNTR
   IF(JDOM.GT.0) ISTRT = JDOM+ICNTR
   IF(JDOM.LE.0) ISTRT = JDOM
   IF(FLOAT(JDOM).EQ.WMIN) ISTRT = JDOM
   IDOM = WMAX/ICNTR
   JDUM = IDOM*ICNTR
   IF(JDOM.GE.0) ISTOP = JDOM
   IF(JDOM.LT.0) ISTOP = JDOM+ICNTR
   IF(FLOAT(JDOM).EQ.WMAX) ISTOP = JDOM
   IF(ISTOP.LT.0) GO TO 100
   I = ISTRT,ISTOP,ICNTR
   WCONST = FLOAT(I)
   ICNT = 0
   DEN = WI(I)-W(3)
   IF(DEN.EQ.0.0) GO TO 120
   XLI = (WCONST-W(3))/DEN
   IF(XLI.GE.ELOW.AND.XLI.LE.EUP) 110,120
110 ICNT = ICNT+1
   APNT(ICNT) = XLI*XI(I)+1.0-XLI)*X(3)
   YPNT(ICNT) = XLI*Y(I)+1.0-XLI)*Y(3)
120 CONTINUE
APPENDIX C – Continued

\[ U_n = r_{13} - W_{21} \]

IF \( U_n \geq 0.0 \) GO TO 140

\[ XLM = \frac{w_k(n) - W_{21}}{U_n} \]

IF \( XLM \leq 0.0 \) AND \( XLM \geq U_p \) GO TO 130

ICNT = ICNT + 1

\[ XPN_{1+1}(ICNT) = (1.0 - XLM) \times X(2) + XLM \times X(3) \]

\[ YPN_{1+1}(ICNT) = (1.0 - XLM) \times Y(2) + XLM \times Y(3) \]

CONTINUE

\[ DEN = r(3) - W(2) \]

IF \( DEN \equiv 0.0 \) GO TO 140

\[ XLM = \frac{W_{12} - W(2)}{DEN} \]

IF \( XLM \equiv 0.0 \) AND \( XLM \leq U_p \) GO TO 150

ICNT = ICNT + 1

\[ XPN_{1+1}(ICNT) = (1.0 - XLM) \times X(1) + XLM \times X(2) \]

\[ YPN_{1+1}(ICNT) = (1.0 - XLM) \times Y(1) + XLM \times Y(2) \]

CONTINUE

IF \( ICNT \leq 1 \) GO TO 100

IF \( ICNT \geq 2 \) GO TO 160

CONTINUE

IF \( ABS(XPN(1) - XPN(2)) \lt \text{ALLOW} \) GO TO 170

CONTINUE

\[ XPN(2) = XPN(3) \]

\[ YPN(2) = YPN(3) \]

CONTINUE

CALL CALPLT \( XPN(1), YPN(1), 3 \)

CALL CALPLT \( XPN(2), YPN(2), 2 \)

CALL CALPLT \( XPN(2), YPN(2), 3 \)

IF \( \text{XLAB} \equiv 0.0 \) GO TO 300

IF \( \text{LAB} \equiv 0.0 \) AND \( \text{LAB} \equiv 0.0 \) GO TO 350

GO TO 310

\[ XSTRT = XLAB \]

\[ YSTRT = YPN(1) \times (YPN(2) - YPN(1)) \times (XLAB - XPN(1)) \]

\[ X(2) - XPN(2) - XPN(1) \]

\[ YSTRT = YSTRT - XLHT/2.0 \]

IF \( \text{LAB} \lt \text{ALLOW} \) GO TO 310

VSAVE = YSTRT

CALL NUMBER \( XSTRT, YSTRT, XLHT, WCONST, 0.0, -1 \)

CONTINUE

\[ XSTRT = XLAB \]

\[ YSTRT = YPN(1) \times (YPN(2) - YPN(1)) \times (XLAB - YPN(1)) \]

\[ X(2) - XPN(2) - YPN(1) \]

\[ YSTRT = YSTRT + XLHT/2.0 \]

IF \( \text{LAB} \lt \text{ALLOW} \) GO TO 400

\[ XSTRT = XSTRT + XLHT/2.0 \]

IF \( \text{LAB} \lt \text{ALLOW} \) GO TO 410

VSAVE = XSTRT

CALL NUMBER \( XSTRT, YSTRT, XLHT, WCONST, 0.0, -1 \)

CONTINUE
APPENDIX C – Continued

400 CONTINUE
100 CONTINUE
1000 CONTINUE
   DO 500 I=1,3
      W(I) = H(I)/MAG
500 CONTINUE
   RETURN
   END OF TRIN
APPENDIX C – Continued

SUBROUTINE BOUND(NUMPT, XPT, YPT, WPT, RADPT, NUMEL, NODE1, NODE2, 
INODE3, NODE4, CENTR)

C *** PLOTS BOUNDARY OF STRUCTURE AND LABELS CONTOUR LINES

COMMON/CONT/ INFOR, KGECM, KDATA, KSIGN, KPLOT, XORGN, YORGN,
LSCALE, ISCALE, WMAG, ICNTR, XLFT, XLAB(10), NYLAB, YLAB(10)
DIMENSION NUMPT(1), XPT(1), YPT(1), WPT(1), RADPT(1), NUMEL(1),
INODE1(1), NODE2(1), NODE3(1), NODE4(1), CENTR(1)
DIMENSION ND(5), NOCON(50, 2)
DIMENSION XX(3), YY(3), MM(3)

C *** EXAMINE ALL ELEMENT CONNECTIONS AND ADD 1.0 TO RAOPTS WHEN TWO
C *** NODES ARE CONNECTED WHICH HAVE RAOPT .NE. 0.0. ALSO FLAG
C *** ELEMENTS CONTAINING SUCH CONNECTIONS BY MAKING NUMEL NEGATIVE.

DO 400 J=1, NELMT
   K = ND(J), L = ND(J+1)
   IF(K.EQ.J) GO TO 400
   IF(L.EQ.J) L = ND(J+2)
   IF(RADPT(K).NE.0.0 .AND. RADPT(L).NE.0.0) 411, 400
411 CONTINUE

RADPT(K) = RADPT(K) + 1.0
RADPT(L) = RADPT(L) + 1.0
IF(NUMEL(J).GT.0) NUMEL(J) = -NUMEL(J)
410 CONTINUE
400 CONTINUE

C *** EXAMINE ALL ELEMENTS AND SAVE ALL CONNECTIONS WITH BOTH
C *** ENDS HAVING RAOPT.GE.4.0 IN NOCON.

NPOSS = 0
DO 420 J=1, NELMT
   IF(NUMEL(J).GT.0) GO TO 420
   K = ND(J), L = ND(J+1)
   IF(K.EQ.J) GO TO 420
   IF(L.EQ.J) L = ND(J+2)
   IF(RADPT(K).GE.4.0 .AND. RADPT(L).GE.4.0) 431, 430
431 CONTINUE

IF(NPOSS.EQ.50) GC TO 430
   NPOSS = NPOSS+1
   NOCON(NPOSS,1) = K
   NOCON(NPOSS,2) = L
430 CONTINUE
APPENDIX C - Continued

C *** CHECK POSSIBLE NG CONNECTS AND SAVE ANY MATCHED PAIRS

NBAU = 0
IF(NPOSS.LE.1) GO TO 470
IEND = NPOSS-1
DO 450 I=1,IEND
JGO = I+1
JEND = NPOSS
DO 460 J=JGO,JEND
IF(NOCOUNT(I,1).EQ.NUCON(J,1).AND.NOCOUNT(I,2).EQ.NUCON(J,2))
GO TO 461
IF(NOCOUNT(I,1).EQ.NUCON(J,2).AND.NOCOUNT(I,2).EQ.NUCON(J,1))
GO TO 461
GO TO 460
461 CONTINUE
NBAU = NBAU+1
NUCON(NBAU,1) = NOCON(I,1)
NUCON(NBAU,2) = NOCON(I,2)
460 CONTINUE
450 CONTINUE
470 CONTINUE
DO 100 I=1,NELMT
IF(NUMEL(I).GT.0) GO TO 100
ND(1) = NUDE1(I)
ND(2) = NUDE2(I)
ND(3) = NUDE3(I)
ND(4) = NUDE4(I)
ND(5) = NUDE5(I)
DO 50 J=1,4
K = ND(J)
IF(K.EQ.0) GO TO 100
L = ND(J+1)
IF(L.EQ.0) L = ND(J+2)
IF(RADPT(K).GE.3.GT.RADPT(L).GE.3.0) 111,50
111 CONTINUE
NU1 = K
NU2 = L
IF(J.EQ.1) IDUM = J-1
IF(J.EQ.1) IDUM = 3
NAUX = ND(IDUM)
C *** TO PLOT BOUNDARY LINES

IF(NBAU.EQ.0) GO TO 40
DU 30 JJ = 1,NBAC
IF(ND1.EQ.NUCON(JJ,1).AND.ND2.EQ.NUCON(JJ,2)) GO TO 50
IF(ND1.EQ.NUCON(JJ,2).AND.ND2.EQ.NUCON(JJ,1)) GO TO 50
30 CONTINUE
40 CONTINUE
A1 = XPT(ND1)/PSCALE
Y1 = FLOAT(KSIGN)*YPT(ND1)/PSCALE
X2 = XPT(ND2)/PSCALE
Y2 = FLOAT(KSIGN)*YPT(ND2)/PSCALE
CALL CALPLT(X1,Y1,3)
CALL CALPLT(X2,Y2,2)
CALL CALPLT(X2,Y2,3)
IF(INFOR.EQ.2) 888,889
888 CONTINUE

110
APPENDIX C - Continued

*** TO PLOT CONTOUR LINES TO EDGE OF ELEMENT DATA PLOTS

\[
\begin{align*}
AX(1) &= XPT(ND1) \\
AY(1) &= YPT(ND1) \\
AW(1) &= WPT(ND1) \\
AX(2) &= XPT(ND2) \\
AY(2) &= YPT(ND2) \\
AW(2) &= WPT(ND2) \\
N1 &= NU(1) \\
N2 &= NU(2) \\
N3 &= NU(3) \\
N4 &= NU(4)
\end{align*}
\]

IF(N4.EQ.0) 910,920

910 CONTINUE
\[
\begin{align*}
AXCEN = (AX(1) + AX(2) + AX(3)) / 3.0 \\
AYCEN = (AY(1) + AY(2) + AY(3)) / 3.0
\end{align*}
\]
GO TO 950

920 CONTINUE
\[
\begin{align*}
AXCEN &= (AX(1) + AX(2) + AX(3) + AX(4)) / 4.0 \\
AYCEN &= (AY(1) + AY(2) + AY(3) + AY(4)) / 4.0
\end{align*}
\]
GO TO 950

950 CONTINUE
\[
\begin{align*}
AX(3) &= AXCEN \\
AY(3) &= AYCEN \\
AW(3) &= CALL T1NXX,YY,WW
\end{align*}
\]
889 CONTINUE

*** TO LABEL CONTOUR LINES

\[
\begin{align*}
WPT(ND1) &= WPT(ND1) * WMAG \\
WPT(ND2) &= WPT(ND2) * WMAG \\
IF(WPT(ND1).EQ.WPT(ND2)) GO TO 666 \\
WMIN &= WPT(ND1) \\
WMAX &= WPT(ND2) \\
IF(WPT(ND1).GT.WPT(ND2)) WMAX &= WPT(ND1) \\
IF(WPT(ND2).LT.WPT(ND1)) WMIN &= WPT(ND2) \\
JDOUM &= WMIN/ICNTR \\
JDOUM &= JDOUM*ICNTR \\
IF(JDOUM.GT.0) LSTART &= JDOUM+ICNTR \\
IF(JDOUM.GT.0) LSTART &= JDOUM \\
IF(REAL(JDOUM).EQ.WMIN) LSTART &= JDOUM \\
JDOUM &= WMAX/ICNTR \\
JDOUM &= JDOUM*ICNTR \\
IF(JDOUM.GE.0) ISTOP &= JDOUM \\
IF(JDOUM.LE.0) ISTOP &= JDOUM-ICNTR \\
IF(REAL(JDOUM).EQ.WMAX) ISTOP &= JDOUM \\
IF(ISTOP.LT.LSTART) GO TO 666 \\
YLINE &= 1.0E+20 \\
XLINE &= 1.0E+20 \\
IF(AYPT(ND2)-AYPT(ND1).NE.0.0) YLINE &= YPT(ND1)+(YPT(ND2)-YPT(ND1)) \\
* (XPT(NAUX)-XPT(ND1))/(XPT(ND2)-XPT(ND1)) \\
IF(AYPT(ND2)-AYPT(ND1).NE.0.0) XLINE &= XPT(ND1)+(XPT(ND2)-XPT(ND1)) \\
* (YPT(NAUX)-YPT(ND1))/(YPT(ND2)-YPT(ND1)) \\
AXDIST &= XPT(NAUX)-XLINE \\
AYDIST &= YPT(NAUX)-YLINE \\
THETA &= 90.0 \\
IF(ABS(YDIST/AXDIST).LT.1.0) THETA &= 90.0 \\
ASAVE &= 1.0E+20
\end{align*}
\]

111
APPENDIX C - Continued

YSAVE = 1.0E+20
DO 660 JJ = ISTRT,ISTOP,ICNTR
  WCONST = FLOAT(JJ)
  IF(WCONST.LT.WMIN.OR.WCONST.GT.WMAX) GO TO 660
  X = XPT(ND1)+1(XPT(ND2)-XPT(ND1))*(WCONST-WPT(ND1))
  X = X/PSCALE
  Y = YPT(NDI)+1(YPT(ND2)-YPT(ND1))*(WCONST-WPT(ND1))
  Y = FLAT(KSIGN)*Y/PSCALE
  DU 650 IG = 1,10
  100 CONTINUE
  IF(ABS(JJ).LT.IDUM) GO 655,650
  NDIG = IG
  IF(JJ.LT.0) NDIG = NDIG+1
  GO TO 660
650 CONTINUE
660 CONTINUE
  IF(THETA.EQ.0.0) GO TO 670
  IF(THETA.EQ.90.0) GO TO 690
  IF(ABS(YSAVE-Y).LE.XLHT) GO TO 660
  YNUM = Y-XLHT/2.0
  IF(XDIST.LE.0.0) XNUM = X+(6.0/7.0)*XLHT
  IF(XDIST.GT.0.0) XNUM = X-(6.0/7.0)*XLHT*FLOAT(NDIG+1)
  YSAVE = Y
  GO TO 690
680 CONTINUE
  IF(ABS(ASAVE-X).LE.XLHT) GO TO 660
  XNUM = X+XLHT/2.0
  IF(YDIST.LE.0.0) YNUM = Y+(6.0/7.0)*XLHT
  IF(YDIST.GT.0.0) YNUM = Y-(6.0/7.0)*XLHT*FLOAT(NDIG+1)
  ASAVE = Y
  GO TO 690
690 CONTINUE
  CALL NUMBER(XNUM,YNUM,XLHT,KCNST,THETA,-1)
660 CONTINUE
665 CONTINUE
  WPT(ND1) = WPT(N)1)/WMAG
  WPT(ND2) = WPT(N)2)/WMAG
50 CONTINUE
100 CONTINUE
C
C *** TO RESTORE RADPT OF BOUNDARY POINTS TO 1.0
C *** AND NUMEL TO POSITIVE VALUE.
C
DO 310 I=1,NNODE
  IF(RADPT(I).GT.0.0) RADPT(I) = 1.0
310 CONTINUE
DO 315 I=1,NELMT
  NUMEL(I) = IABS(NUMEL(I))
315 CONTINUE
RETURN
END OF BOUND.
SUBROUTINE LAYOUT(NPLOT, NUMPT, XPT, YPT, WPT, RAUPT, NUMEL, NODE1, NODE2, INGUD3, NODE4, CENTR)

C *** TO PLOT LAYOUT OF ELEMENTS

C COMMON/KOUNT/ NNODE, NELMT, NNODE1, NNODE2, NNODE3, NNODE4, CENTR
C COMMON/CONTROLL/ INFOR, AGEN, KBDA, KSIGN, KPLUT, XORGN, YORGN,
ICOLN, ISCALE, IMAG, INCTR, XLAB, YLAB, XLAB(10), YLAB(10)
DIMENSION NUMPT(1), XPT(1), YPT(1), WPT(1), RAUPT(1), NUMEL(1),
NODE1(1), NODE2(1), NODE3(1), NODE4(1), CENTR(1)

DO 400 I = 1, NNODE
NUMPT (1) = -NUMPT (1)
400 CONTINUE

DO 100 I = 1, NELMT
ND1 = NODE1(I)
ND2 = NODE2(I)
ND3 = NODE3(I)
ND4 = NODE4(I)

C *** TO MAKE ALL GRID POINT NUMBERS NEGATIVE

C *** TO MAKE ALL GRID POINT NUMBERS CONNECTED BY ELEMENTS POSITIVE

IF (NPLOT .NE. 1) GO TO 200

X = XCEN T - (6.0/7.0) * XLHT
Y = YCENT - XLHT/2.0
A = NUMEL(I)
CALL NUMDER(X, Y, XLHT, A, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
200 CONTINUE

C *** TO NUMBER TRIANGLES

X = XCENT - (6.0/7.0) * XLHT
Y = YCENT - XLHT/2.0
A = NUMEL(I)
CALL NUMDER(X, Y, XLHT, A, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
300 CONTINUE

C *** TO NUMBER QUADRILATERALS

X = XCENT - (6.0/7.0) * XLHT
Y = YCENT - XLHT/2.0
A = NUMEL(I)
CALL NUMDER(X, Y, XLHT, A, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

APPENDIX C – Continued
Y = YCENT-XLHT/2.0
A = NUMEL(I)
CALL NUMBERIX,Y,XLHT,A,0.0,-1)
350 CONTINUE
100 CONTINUE
IF(NPLOT.NE.2) GC TO 450
C *** TO NUMBER GRID POINTS
DO 400 I=1,NNODE
IF(NUMPT(I).LE.0) GO TO 400
X = XPT(I)/PSCALE+0.5*XLHT
Y = FLOAT(KSIGN)*YPT(I)/PSCALE+0.5*XLHT
A = NUMPT(I)
CALL NUMBER(X,Y,XLHT,A,0.0,-1)
400 CONTINUE
430 CONTINUE
RETURN
END OF LAYOUT
APPENDIX D

INPUT DATA FOR OBLIQUE ORTHOGRAPHIC PROJECTION OF EXAMPLE PROBLEM

FLAT PLATE UNDER UNIFORM PRESSURE - OBLIQUE ORTHOGRAPHIC PROJECTIONS

$OPTION
NMODEST = 36, NUDISP = 1, NVISP = 1, NWDISP = 1,
KGEOM = 2, KDATA = 1, KPLAN = 3, IDCASE = 1
$

QUADEL CQUAD1
I0 GILES,TEST
APP DISPLACEMENT
SOL 1,0
TIME 60
END

TITLE = FLAT PLATE UNDER UNIFORM PRESSURE
QLOAD = ALL
SPCFORCES = ALL
DISPLACEMENT(PRINT,PUNCH) = ALL
STRESS(PRINT,PUNCH) = ALL
LOAD = 100
SPC = 100
BEGIN BULK

PQUAD1 1 1 0.01 1 8.333E-9 0.01 0.0 +GARY1
+GARY1 0.005 -0.005
MAT1 1
SPC1 100 123 10 20 30 40 50 60 +S1
+S1 100 130 190 250 310 320 330 340 +S2
+S2 350 360 120 180 240 300
PLOAD2 100 100.0 6 THRU 10
PLLOAD2 100 100.0 16 THRU 20
PLLOAD2 100 100.0 26 THRU 30
PLLOAD2 100 100.0 36 THRU 40
PLLOAD2 100 100.0 46 THRU 50
GROSET 0
GRID 10 0.0 0.0 0.0
GRID 20 2.0 0.0 0.0
GRID 30 4.0 0.0 0.0
GRID 40 6.0 0.0 0.0
GRID 50 0.0 0.0 0.0
GRID 60 10.0 0.0 0.0
GRID 70 0.0 2.0 0.0
GRID 80 2.0 2.0 0.0
GRID 90 4.0 2.0 0.0
GRID 100 6.0 2.0 0.0
GRID 110 8.0 2.0 0.0
GRID 120 10.0 2.0 0.0
GRID 130 0.0 4.0 0.0
GRID 140 2.0 4.0 0.0
GRID 150 4.0 4.0 0.0
GRID 160 6.0 4.0 0.0
GRID 170 8.0 4.0 0.0
GRID 180 10.0 4.0 0.0
GRID 190 0.0 6.0 0.0
GRID 200 2.0 6.0 0.0
GRID 210 4.0 6.0 0.0
GRID 220 6.0 6.0 0.0
GRID 230 8.0 6.0 0.0

115
<table>
<thead>
<tr>
<th>GRID</th>
<th>240</th>
<th>10.0</th>
<th>6.0</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRID</td>
<td>250</td>
<td>0.0</td>
<td>8.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>260</td>
<td>2.0</td>
<td>8.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>270</td>
<td>4.0</td>
<td>8.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>280</td>
<td>6.0</td>
<td>8.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>290</td>
<td>8.0</td>
<td>8.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>300</td>
<td>10.0</td>
<td>6.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>310</td>
<td>0.0</td>
<td>10.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>320</td>
<td>2.0</td>
<td>10.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>330</td>
<td>4.0</td>
<td>10.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>340</td>
<td>6.0</td>
<td>10.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>350</td>
<td>8.0</td>
<td>10.0</td>
<td>0.0</td>
</tr>
<tr>
<td>GRID</td>
<td>360</td>
<td>10.0</td>
<td>10.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

CQUADI 6 1 10 20 80 70
CQUADI 7 1 20 30 90 80
CQUADI 8 1 30 40 100 90
CQUADI 9 1 40 50 110 100
CQUADI 10 1 50 60 120 110
CQUADI 11 1 70 80 140 130
CQUADI 12 1 80 90 150 140
CQUADI 13 1 90 100 160 150
CQUADI 14 1 100 110 170 160
CQUADI 15 1 110 120 180 170
CQUADI 16 1 130 140 200 190
CQUADI 17 1 140 150 210 200
CQUADI 18 1 150 160 220 210
CQUADI 19 1 160 170 230 220
CQUADI 20 1 170 180 240 230
CQUADI 21 1 190 200 260 250
CQUADI 22 1 200 210 270 260
CQUADI 23 1 210 220 280 270
CQUADI 24 1 220 230 290 280
CQUADI 25 1 230 240 300 290
CQUADI 26 1 250 260 320 310
CQUADI 27 1 260 270 330 320
CQUADI 28 1 270 280 340 330
CQUADI 29 1 280 290 350 340
CQUADI 30 1 290 300 360 350

ENDDATA
ENDFILE

Z - DISPLACEMENTS

<table>
<thead>
<tr>
<th>FORMAT 1 (110,8X,3E18.6/)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 G 0. -5.741610E+01 5.741610E+01 0. 0. 0. 0.</td>
</tr>
<tr>
<td>20 G 0. 2.831d30E+03 4.045395E+01 0. 0. 0. 0.</td>
</tr>
<tr>
<td>30 G 0. 4.462056E+03 2.751137E+01 0. 0. 0. 0.</td>
</tr>
<tr>
<td>40 G 0. 4.462056E+03 -2.751137E+01 0. 0. 0. 0.</td>
</tr>
<tr>
<td>50 G 0. 2.831d30E+03 -4.045395E+01 0. 0. 0. 0.</td>
</tr>
<tr>
<td>60 G 0. -5.741610E+01 -5.741610E+01 0. 0. 0. 0.</td>
</tr>
<tr>
<td>70 G 0. -4.045395E+01 -2.831d30E+03 0. 0. 0. 0.</td>
</tr>
<tr>
<td>80 G 0. 2.192913E+03 -2.192913E+03 0. 0. 0. 0.</td>
</tr>
</tbody>
</table>

116
<table>
<thead>
<tr>
<th>90</th>
<th>G</th>
<th>0.</th>
<th>3.472762E+03</th>
<th>0.</th>
<th>8.351675E+03</th>
<th>26</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>G</td>
<td>0.</td>
<td>7.800294E+02</td>
<td>0.</td>
<td>8.351675E+03</td>
<td>27</td>
</tr>
<tr>
<td>110</td>
<td>G</td>
<td>0.</td>
<td>7.800294E+02</td>
<td>0.</td>
<td>8.351675E+03</td>
<td>28</td>
</tr>
<tr>
<td>120</td>
<td>G</td>
<td>0.</td>
<td>2.192913E+03</td>
<td>0.</td>
<td>5.284250E+03</td>
<td>29</td>
</tr>
<tr>
<td>130</td>
<td>G</td>
<td>0.</td>
<td>-4.045355E+01</td>
<td>0.</td>
<td>5.284250E+03</td>
<td>30</td>
</tr>
<tr>
<td>140</td>
<td>G</td>
<td>0.</td>
<td>-4.62056E+03</td>
<td>0.</td>
<td>5.284250E+03</td>
<td>31</td>
</tr>
<tr>
<td>150</td>
<td>G</td>
<td>0.</td>
<td>-3.472762E+03</td>
<td>0.</td>
<td>5.284250E+03</td>
<td>32</td>
</tr>
<tr>
<td>160</td>
<td>G</td>
<td>0.</td>
<td>-1.250892E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>33</td>
</tr>
<tr>
<td>170</td>
<td>G</td>
<td>0.</td>
<td>-1.250892E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>34</td>
</tr>
<tr>
<td>180</td>
<td>G</td>
<td>0.</td>
<td>-3.472762E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>35</td>
</tr>
<tr>
<td>190</td>
<td>G</td>
<td>0.</td>
<td>4.462056E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>36</td>
</tr>
<tr>
<td>200</td>
<td>G</td>
<td>0.</td>
<td>-4.462056E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>37</td>
</tr>
<tr>
<td>210</td>
<td>G</td>
<td>0.</td>
<td>-3.472762E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>38</td>
</tr>
<tr>
<td>220</td>
<td>G</td>
<td>0.</td>
<td>-1.250892E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>39</td>
</tr>
<tr>
<td>230</td>
<td>G</td>
<td>0.</td>
<td>-1.250892E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>40</td>
</tr>
<tr>
<td>240</td>
<td>G</td>
<td>0.</td>
<td>-3.472762E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>41</td>
</tr>
<tr>
<td>250</td>
<td>G</td>
<td>0.</td>
<td>4.462056E+03</td>
<td>0.</td>
<td>1.324540E+04</td>
<td>42</td>
</tr>
<tr>
<td>260</td>
<td>G</td>
<td>0.</td>
<td>2.751137E+01</td>
<td>2.831830E+03</td>
<td>0.</td>
<td>43</td>
</tr>
<tr>
<td>270</td>
<td>G</td>
<td>0.</td>
<td>4.045395E+01</td>
<td>2.831830E+03</td>
<td>0.</td>
<td>44</td>
</tr>
<tr>
<td>280</td>
<td>G</td>
<td>0.</td>
<td>-2.831830E+03</td>
<td>0.</td>
<td>2.831830E+03</td>
<td>45</td>
</tr>
<tr>
<td>290</td>
<td>G</td>
<td>0.</td>
<td>-2.192913E+03</td>
<td>0.</td>
<td>2.831830E+03</td>
<td>46</td>
</tr>
<tr>
<td>300</td>
<td>G</td>
<td>0.</td>
<td>2.751137E+01</td>
<td>2.831830E+03</td>
<td>0.</td>
<td>47</td>
</tr>
<tr>
<td>310</td>
<td>G</td>
<td>0.</td>
<td>2.831830E+03</td>
<td>2.831830E+03</td>
<td>0.</td>
<td>48</td>
</tr>
<tr>
<td>320</td>
<td>G</td>
<td>0.</td>
<td>8.351675E+03</td>
<td>5.741610E+01</td>
<td>0.</td>
<td>49</td>
</tr>
<tr>
<td>330</td>
<td>G</td>
<td>0.</td>
<td>5.741610E+01</td>
<td>5.741610E+01</td>
<td>0.</td>
<td>50</td>
</tr>
<tr>
<td>340</td>
<td>G</td>
<td>0.</td>
<td>5.741610E+01</td>
<td>5.741610E+01</td>
<td>0.</td>
<td>51</td>
</tr>
<tr>
<td>350</td>
<td>G</td>
<td>0.</td>
<td>5.741610E+01</td>
<td>5.741610E+01</td>
<td>0.</td>
<td>52</td>
</tr>
<tr>
<td>360</td>
<td>G</td>
<td>0.</td>
<td>5.741610E+01</td>
<td>5.741610E+01</td>
<td>0.</td>
<td>53</td>
</tr>
</tbody>
</table>

(BLANK CARD)

(BLANK CARD)
APPENDIX D – Concluded

$PICT
KHORZ = 1, KVERT = 3, PHI = 20.0, THETA = 0.0, PSI = 10.0,
ISCALE = 2, XORGN = 1.0, YORGN = 1.0, PSSCALE = 2.0,
KD1SP = 1, IDMAG = 2, DMAGS = 2.0, KDE = 1
$

$PICT
KD1SP = 3, KDE = 0
$
APPENDIX E

INPUT DATA FOR CONTOUR PLOT OF EXAMPLE PROBLEM

FLAT PLATE UNDER UNIFORM PRESSURE - CONTOUR PLOT

$OPTION
NNDEST = 36, NELEST = 25,
KPLT = 3, IUGCASE = 1,
$FOMAT (18X,13.5X,2F8.1)

<table>
<thead>
<tr>
<th>GRID</th>
<th>10</th>
<th>0.0</th>
<th>0.0</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>30</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>40</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>50</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>60</td>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>70</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>80</td>
<td>2.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>90</td>
<td>4.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>100</td>
<td>6.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>110</td>
<td>8.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>120</td>
<td>10.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>130</td>
<td>0.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>140</td>
<td>2.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>150</td>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>160</td>
<td>6.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>170</td>
<td>8.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>180</td>
<td>10.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>190</td>
<td>0.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>200</td>
<td>2.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>210</td>
<td>4.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>220</td>
<td>6.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>230</td>
<td>8.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>240</td>
<td>10.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>250</td>
<td>0.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>260</td>
<td>2.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>270</td>
<td>4.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>280</td>
<td>6.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>290</td>
<td>8.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>300</td>
<td>10.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>310</td>
<td>0.0</td>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>320</td>
<td>2.0</td>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>330</td>
<td>4.0</td>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>340</td>
<td>6.0</td>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>350</td>
<td>8.0</td>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>360</td>
<td>10.0</td>
<td>10.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
ENDGRID

FORMAT (18X,12,14X,4(13,5X))

<table>
<thead>
<tr>
<th>CQUAD1</th>
<th>6</th>
<th>1</th>
<th>10</th>
<th>20</th>
<th>80</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQUAD1</td>
<td>7</td>
<td>1</td>
<td>20</td>
<td>30</td>
<td>90</td>
<td>80</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>8</td>
<td>1</td>
<td>30</td>
<td>40</td>
<td>100</td>
<td>90</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>9</td>
<td>1</td>
<td>40</td>
<td>50</td>
<td>110</td>
<td>100</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>10</td>
<td>1</td>
<td>50</td>
<td>60</td>
<td>120</td>
<td>110</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>11</td>
<td>1</td>
<td>60</td>
<td>70</td>
<td>140</td>
<td>130</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>12</td>
<td>1</td>
<td>70</td>
<td>80</td>
<td>150</td>
<td>140</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>13</td>
<td>1</td>
<td>80</td>
<td>90</td>
<td>160</td>
<td>150</td>
</tr>
</tbody>
</table>
### APPENDIX E – Concluded

<table>
<thead>
<tr>
<th>CQUAD1</th>
<th>19</th>
<th>1</th>
<th>100</th>
<th>110</th>
<th>170</th>
<th>160</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQUAD1</td>
<td>20</td>
<td>1</td>
<td>110</td>
<td>120</td>
<td>180</td>
<td>170</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>26</td>
<td>1</td>
<td>130</td>
<td>140</td>
<td>200</td>
<td>190</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>27</td>
<td>1</td>
<td>140</td>
<td>150</td>
<td>210</td>
<td>200</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>28</td>
<td>1</td>
<td>150</td>
<td>160</td>
<td>220</td>
<td>210</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>29</td>
<td>1</td>
<td>160</td>
<td>170</td>
<td>230</td>
<td>220</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>30</td>
<td>1</td>
<td>170</td>
<td>180</td>
<td>240</td>
<td>230</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>36</td>
<td>1</td>
<td>190</td>
<td>200</td>
<td>260</td>
<td>250</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>37</td>
<td>1</td>
<td>200</td>
<td>210</td>
<td>270</td>
<td>260</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>38</td>
<td>1</td>
<td>210</td>
<td>220</td>
<td>280</td>
<td>270</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>39</td>
<td>1</td>
<td>220</td>
<td>230</td>
<td>290</td>
<td>280</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>40</td>
<td>1</td>
<td>230</td>
<td>240</td>
<td>300</td>
<td>290</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>46</td>
<td>1</td>
<td>250</td>
<td>260</td>
<td>320</td>
<td>310</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>47</td>
<td>1</td>
<td>260</td>
<td>270</td>
<td>330</td>
<td>320</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>48</td>
<td>1</td>
<td>270</td>
<td>280</td>
<td>340</td>
<td>330</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>49</td>
<td>1</td>
<td>280</td>
<td>290</td>
<td>350</td>
<td>340</td>
</tr>
<tr>
<td>CQUAD1</td>
<td>50</td>
<td>1</td>
<td>290</td>
<td>300</td>
<td>360</td>
<td>350</td>
</tr>
</tbody>
</table>

### 2 - DISPLACEMENTS

**FORMAT 4 (14(15,E15.6))**

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.</td>
<td>20</td>
<td>0.</td>
<td>30</td>
<td>0.</td>
<td>40</td>
</tr>
<tr>
<td>50</td>
<td>0.</td>
<td>60</td>
<td>0.</td>
<td>70</td>
<td>0.</td>
<td>80</td>
</tr>
<tr>
<td>90</td>
<td>8.351675E+03</td>
<td>100</td>
<td>8.351675E+03</td>
<td>110</td>
<td>5.284250E+03</td>
<td>120</td>
</tr>
<tr>
<td>130</td>
<td>0.</td>
<td>140</td>
<td>8.351675E+03</td>
<td>150</td>
<td>1.324540E+04</td>
<td>160</td>
</tr>
<tr>
<td>170</td>
<td>8.351675E+03</td>
<td>180</td>
<td>0.</td>
<td>190</td>
<td>0.</td>
<td>200</td>
</tr>
<tr>
<td>210</td>
<td>1.324540E+04</td>
<td>220</td>
<td>1.324540E+04</td>
<td>230</td>
<td>8.351675E+03</td>
<td>240</td>
</tr>
<tr>
<td>250</td>
<td>0.</td>
<td>260</td>
<td>0.</td>
<td>270</td>
<td>0.</td>
<td>280</td>
</tr>
<tr>
<td>290</td>
<td>5.284250E+03</td>
<td>300</td>
<td>0.</td>
<td>310</td>
<td>0.</td>
<td>320</td>
</tr>
<tr>
<td>330</td>
<td>0.</td>
<td>340</td>
<td>0.</td>
<td>350</td>
<td>0.</td>
<td>360</td>
</tr>
</tbody>
</table>

*(BLANK CARD)*

**SPTC**

- PS SCALE = 2.0;
- NXLAB = 1, XLAB(1) = 5.0, NYLAB = 1, YLAB(1) = 5.0

$
APPENDIX F

LISTING OF CONTOUR PROGRAM OUTPUT FOR EXAMPLE PROBLEM

FLAT PLATE UNDER UNIFORM PRESSURE—CONTOUR PLOT

$OPTION
NNODES = 36,
NELEST = 25,
KGEOM = 1,
KDQTA = 1,
NVALUES = 0,
IRESEQ = 1,
KPLUT = 3,
INFOR = 1,
XSPACE = 0.1E+02,
KSIGN = 1,
ICASE = 1,
$END

BLANK COMMON STORAGE ZZZ REQUIRES AT LEAST 330 LOCATIONS FOR THIS CASE

GRID POINT INFORMATION

<table>
<thead>
<tr>
<th>RESEQUENCED</th>
<th>USER INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRID POINT NUMBER</td>
<td>GRID POINT NUMBER</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>INPUT X</td>
<td>INPUT Y</td>
</tr>
<tr>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>2.0000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>4.0000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>6.0000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>8.0000E+00</td>
<td>0.</td>
</tr>
<tr>
<td>1.0000E+01</td>
<td>0.</td>
</tr>
<tr>
<td>BOUNDARY POINT INDICATOR</td>
<td></td>
</tr>
<tr>
<td>1.0000E+00</td>
<td></td>
</tr>
<tr>
<td>1.0000E+00</td>
<td></td>
</tr>
<tr>
<td>1.0000E+00</td>
<td></td>
</tr>
<tr>
<td>1.0000E+00</td>
<td></td>
</tr>
<tr>
<td>1.0000E+00</td>
<td></td>
</tr>
<tr>
<td>1.0000E+00</td>
<td></td>
</tr>
</tbody>
</table>

121
APPENDIX F - Continued

<table>
<thead>
<tr>
<th>ELEMENT NUMBER</th>
<th>USER INPUT</th>
<th>GRID POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>21</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>22</td>
<td>14</td>
</tr>
<tr>
<td>13</td>
<td>23</td>
<td>16</td>
</tr>
<tr>
<td>14</td>
<td>24</td>
<td>17</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>19</td>
</tr>
<tr>
<td>16</td>
<td>26</td>
<td>20</td>
</tr>
</tbody>
</table>

ELEMENT INFORMATION - WITH RESEQUENCED GRID POINTS
APPENDIX F – Continued

| 18 | 38 | 21 | 22 | 28 | 27 |
| 19 | 39 | 22 | 23 | 29 | 28 |
| 20 | 40 | 23 | 24 | 30 | 29 |
| 21 | 46 | 25 | 26 | 32 | 31 |
| 22 | 47 | 26 | 27 | 33 | 32 |
| 23 | 48 | 27 | 28 | 34 | 33 |
| 24 | 49 | 28 | 29 | 35 | 34 |
| 25 | 50 | 29 | 30 | 36 | 35 |

Z – DISPLACEMENTS

DATA TO BE PLOTTED

RESEQUENCED GRID POINT NUMBERS AND DATA VALUES

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>5.28425E+03</td>
<td>1.32454E+04</td>
<td>5.26425E+03</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>11</td>
<td>0.</td>
<td>8.35167E+03</td>
<td>1.32454E+04</td>
<td>8.35167E+03</td>
<td>0.</td>
</tr>
<tr>
<td>16</td>
<td>1.32454E+04</td>
<td>8.35167E+03</td>
<td>5.28425E+03</td>
<td>8.35167E+03</td>
<td>0.</td>
</tr>
<tr>
<td>21</td>
<td>5.28425E+03</td>
<td>8.35167E+03</td>
<td>0.</td>
<td>8.35167E+03</td>
<td>0.</td>
</tr>
<tr>
<td>26</td>
<td>5.28425E+03</td>
<td>8.35167E+03</td>
<td>0.</td>
<td>8.35167E+03</td>
<td>0.</td>
</tr>
</tbody>
</table>

These items appear to the right of the above items in the actual printout

\[
\begin{align*}
4 & : 0.0 \\
9 & : 8.35167E+03 \\
14 & : 8.35167E+03 \\
19 & : 0.0 \\
24 & : 0.0 \\
29 & : 5.28425E+03 \\
34 & : 0.0 \\
4 & : 0.0 \\
9 & : 8.35167E+03 \\
14 & : 8.35167E+03 \\
19 & : 0.0 \\
24 & : 0.0 \\
29 & : 5.28425E+03 \\
34 & : 0.0 \\
5 & : 0.0 \\
10 & : 8.35167E+03 \\
15 & : 1.32454E+04 \\
20 & : 8.35167E+03 \\
25 & : 0.0 \\
30 & : 0.0 \\
35 & : 0.0
\end{align*}
\]

SPICT
NPLGT = 4,
XORGN = 0.0,
YORGN = 0.0,
PScale = 0.2E+01,
IScale = 3,
WMAGS = 0.1E+03,
ICNTRS = 10,
APPENDIX F – Concluded

XLHT = 0.15E+00,
NXLAB = 1,
XLAB = 0.5E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
NYLAB = 1,
YLAB = 0.5E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
KODE = 0,
SEND

MAXIMUM VALUE OF DATA = 1.32454000CE+04  MINIMUM VALUE OF DATA = 0.

WMAG = 7.54979087E-03  ICCATR = 10
REFERENCES


Figure 2. - Coordinate systems and Euler angles (rotations) for an oblique orthographic projection shown in $X_0-Z_0$ viewing plane.
Figure 3.- Exploded plot of outboard portion of aircraft wing shown in figure 1.
Figure 4. Exploded plots of elements at a selected fuselage frame-wing spar location of figure 1.
(a) Contour plot.

(b) View from side.

(c) View from rear.

Figure 5 - Contour plot and oblique orthographic projections of vibration mode shapes of aircraft in figure 1.
Figure 6. - Triangular meshes on datum surface used in generating contour plots.
Figure 7.- Diagram showing subareas of a triangle required to locate a generic point p by using area coordinates.

- $A_i$ = area of $\triangle jkp$
- $A_j$ = area of $\triangle kip$
- $A_k$ = area of $\triangle ijp$
- $\overline{A}$ = area of $\triangle ijk$
Figure 8.- A proper sequence for input cards.
Figure 9. - Oblique orthographic projections of a simply supported, square, flat plate under uniform pressure.
Figure 10.- Contour plot of a simply supported, square, flat plate under uniform pressure.
Figure 11.- Diagram showing subareas of a triangle required to locate its incenter $p$ by using area coordinates.
Figure 12.- Flow chart for program which generates oblique-orthographic-projection plots.
Figure 13. Flow chart for program which generates contour plots.
The aeronautical and space activities of the United States shall be conducted to as to contribute to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results therefrom.

—National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENIFIC AND TECHNICAL INFORMATION OFFICE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546