75N17124

NASA TECHNICAL NOTE

NASA TN D-7797

NASA TN D-7797

DIGITAL COMPUTER PROGRAMS FOR GENERATING OBLIQUE ORTHOGRAPHIC PROJECTIONS AND CONTOUR PLOTS

by Gary L. Giles

Langley Research Center
Hampton, Va. 23665

RECEIVED
FEB 1 1975
AERONAUTICAL LIBRARY

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JANUARY 1975

1. Report No. NASA TN D-7797	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle DIGITAL COMPUTER PROGRAMS FOR GENERATING OBLIQUE ORTHOGRAPHIC PROJECTIONS AND CONTOUR PLOTS		5. Report Date January 1975
		6. Performing Organization Code
7. Author(s) Gary L. Giles		8. Performing Organization Report No. L-9766
		10. Work Unit No. 501-22-11
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23665		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

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.

17. Key Words (Suggested by Author(s)) Computer graphics; Contour plots; Orthographic projections	18. Distribution Statement Unclassified – Unlimited; STAR Category 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	140	$5.75

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

Page Intentionally Left Blank

CONTENTS

Page

DIGITAL COMPUTER PROGRAMS FOR GENERATING OBLIQUE ORTHOGRAPHIC PROJECTIONS AND CONTOUR PLOTS

By Gary L. Giles
Langley Research Center

SUMMARY

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.

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 programer 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_θ, A_ϕ, A_ψ — 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_i,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_o,Y_o,Z_o	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_o,y_o,z_o	coordinates of a point in X_o,Y_o,Z_o coordinate system
x_p,y_p,z_p	coordinates of generic point p
β_i	weighting function to interpolate data values from ith element centroid to a particular grid point
θ	Euler angle rotation about Y-axis, performed second
ϕ	Euler angle rotation about X-axis, performed third
ψ	Euler angle rotation about Z-axis, performed first

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 lineal elements with two grid points and triangular or quadrilateral plate elements with three or four grid points.

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 (ϕ,θ,ψ) 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 ψ, θ, and ϕ in the program and transform coordinates of a point on the model (x,y,z) to viewing plane coordinates (x_o,y_o,z_o) as

$$\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = \begin{bmatrix} A_\psi \end{bmatrix}\begin{bmatrix} A_\theta \end{bmatrix}\begin{bmatrix} A_\phi \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1}$$

where

$$\begin{bmatrix} A_\psi \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} A_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} A_\phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{4}$$

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_o-Y_o, X_o-Z_o, or Y_o-Z_o planes and is shown projected on the X_o-Z_o 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

$$\bar{x} = \frac{\sum_{i=1}^{M} x_i}{M}, \quad \bar{y} = \frac{\sum_{i=1}^{M} y_i}{M}, \quad \text{and} \quad \bar{z} = \frac{\sum_{i=1}^{M} z_i}{M} \tag{5}$$

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:

$$w_G = \sum_{i=1}^{N} w_i \beta_i \tag{6}$$

where w_i is the given data value at the centroid of the ith element, β_i is a weighting function for that value, and N is the total number of elements connected to grid point G (N = 5 in fig. 6(b)). The weighting function selected for use herein is

$$\beta_i = \frac{\frac{1}{l_i}}{\sum_{j=1}^{N} \frac{1}{l_j}} \tag{7}$$

where l_i 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

$$\left.\begin{aligned} L_i &= \frac{A_i}{A} \\ L_j &= \frac{A_j}{A} \\ L_k &= \frac{A_k}{A} \end{aligned}\right\} \tag{8}$$

and

$$L_i + L_j + L_k = 1 \tag{9}$$

The coordinates of a generic point p on the triangular plane (one facet of the contour surface) with vertices $\left(x_i,y_i,w_i\right)$, $\left(x_j,y_j,w_j\right)$, and $\left(x_k,y_k,w_k\right)$ can be expressed as

$$x_p = L_i x_i + L_j x_j + L_k x_k \tag{10}$$

$$y_p = L_i y_i + L_j y_j + L_k y_k \tag{11}$$

$$w_p = L_i w_i + L_j w_j + L_k w_k \tag{12}$$

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

$$L_i = \frac{w_c - w_j}{w_i - w_j} \tag{13}$$

Similarly along the edge j-k,

$$L_i = 0 \qquad L_k = 1 - L_j$$

Thus,

$$L_j = \frac{w_c - w_k}{w_j - w_k} \tag{14}$$

Along the edge k-i,

$$L_j = 0 \qquad L_i = 1 - L_k$$

which yields

$$L_k = \frac{w_c - w_i}{w_k - w_i} \tag{15}$$

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

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

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

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° (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

$$FL_{10} = 18\ 000 + (4 + \mathrm{NUDISP} + \mathrm{NVDISP} + \mathrm{NWDISP})\mathrm{NNDEST}$$

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

$$FL_{10} = 17\ 000 + 5(\mathrm{NNDEST}) + 6(\mathrm{NELEST})$$

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.

COMPUTER 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.

FORTRAN name	Default value	Description
NNDEST	200	Estimated number of grid points, must be equal to or greater than the actual number of grid points
NUDISP	0	0 no displacement data in x-direction 1 data including displacements in x-direction
NVDISP	0	0 no displacement data in y-direction 1 data including displacements in y-direction
NWDISP	0	0 no displacement data in z-direction 1 data including displacements in z-direction
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 displacement data 1 subroutine DATA1 to read in displacement data from cards with user-specified format 5 subroutine DATA5 to read in displacement data from TAPE20 9 subroutine DATA9, a user-supplied subroutine
NVALUS	0	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)
IRESEQ	1	Grid point numbers are stored in the program from 1 to the total number of grid points

FORTRAN name	Default 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
KPLOT	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
PSIZE	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.

FORTRAN name	Default value	Description
KHORZ	1	Integer designating the horizontal axis of the viewing plane where $1 = X_O$, $2 = Y_O$, and $3 = Z_O$
KVERT	2	Integer designating the vertical axis of the viewing plane where $1 = X_O$, $2 = Y_O$, and $3 = Z_O$
PHI	0.0	Angular rotation of model about its X-axis in degrees (must be performed third)
THETA	0.0	Angular rotation of model about its Y-axis in degrees (must be performed second)
PSI	0.0	Angular rotation of model about its Z-axis in degrees (must be performed first)
NEWFR	1	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)

FORTRAN name	Default value	Description
NEWFR	1	0 no frame change before plotting
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 $\geqq 0.07$)
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
KSYMXZ	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 KSYMXZ and KSYMYZ equal to one would yield a complete plate).

FORTRAN name	Default value	Description
XXMAX, YYMAX, ZZMAX XXMIN, YYMIN, ZZMIN	1.0 E+20 -1.0 E+20	Locate cutting planes parallel to principal planes (X-Y, X-Z, Y-Z) to limit plot
NDMAX	9999999999	Maximum grid point identification number to be included in plot
NDMIN	0	Minimum grid point identification number to be included in plot
NELMAX	9999999999	Maximum element identification number to be included in plot
NELMIN	0	Minimum element identification number to be included in plot
KODE	0	Specifies control option after plot is complete 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.

FORTRAN name	Default value	Description
NNDEST	200	Estimated number of grid points, must be equal to or greater than the actual number of grid points

FORTRAN name	Default value	Description
NELEST	200	Estimated number of elements, must be equal to or greater than the actual number of elements
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 name	Default value	Description
KPLOT	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)
INFOR	1	1 data to be plotted is specified at grid points 2 data to be plotted is specified at element centroids
XSPACE	10.0	Space between plots in x-direction, in inches
KSIGN	1	-1 change signs of y-coordinates 1 do not change signs of y-coordinates
IDCASE	0	0 no identification card preceding decks of data to be plotted 1 identification card preceding decks of data to be plotted

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 correspond 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.

FORTRAN name	Default value	Description
NPLOT	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
PSCALE	1.0	Model size reduction factor (i.e., PSCALE 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 values 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

FORTRAN name	Default value	Description
ISCALE	3	2 program calculation of scale factors to give the user-specified number of significant digits in annotation of the maximum absolute contour line 3 program calculation of scale factors to give WMAGS as the maximum value of data
WMAGS	100.0	If ISCALE = 1, magnification of control variables for annotation of contour lines on plot If ISCALE = 2, number of significant digits in annotation of maximum absolute contour line (WMAGS = 1.0, 2.0, 3.0, etc.) If ISCALE = 3, maximum value of scaled data, WMAGS $\geqq$ 2.0 (maximum contour line is integer truncation of WMAGS)
ICNTRS	10	If ISCALE = 1, user-specified contour interval (difference in integer values of adjacent contour lines) If ISCALE = 2 or 3, approximate number of different contour line values; the contour interval is calculated by the program
XLHT	0.15	Height of integers to be annotated on plots, in inches (must be $\geqq$0.07)
NXLAB	0	Total number of lines parallel to Y-axis along which contour lines are labeled, must be $\leqq$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
XLAB	all zeros	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

FORTRAN name	Default value	Description
NYLAB	0	Same as NXLAB for label locations parallel to X-axis
YLAB	all zeros	Same as XLAB for label locations parallel to X-axis
KODE	0	Specifies control option after plot is complete 0 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

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 illustrated. 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 symmetric 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.

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

$$\left.\begin{aligned} A_i &= \frac{1}{2} ha \\ A_j &= \frac{1}{2} hb \\ A_k &= \frac{1}{2} hc \end{aligned}\right\} \tag{A1}$$

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

$$\left.\begin{aligned} L_i &= \frac{a}{s} \\ L_j &= \frac{b}{s} \\ L_k &= \frac{c}{s} \end{aligned}\right\} \tag{A2}$$

where $s = a + b + c$.

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

$$\left.\begin{aligned} x_p &= \frac{ax_i + bx_j + cx_k}{s} \\ y_p &= \frac{ay_i + by_j + cy_k}{s} \\ z_p &= \frac{az_i + bz_j + cz_k}{s} \end{aligned}\right\} \tag{A3}$$

where

$$\left.\begin{aligned}
a &= \sqrt{\left(x_j - x_k\right)^2 + \left(y_j - y_k\right)^2 + \left(z_j - z_k\right)^2} \\
b &= \sqrt{\left(x_k - x_i\right)^2 + \left(y_k - y_i\right)^2 + \left(z_k - z_i\right)^2} \\
c &= \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2}
\end{aligned}\right\} \tag{A4}$$

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.

```
      PROGRAM MAIN(INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE9,TAPE10,TAPE20=201)
C
C *** THIS IS THE MAIN PROGRAM WHICH CALLS OTHER SUBROUTINES
C
      INTEGER NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,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/CORGN/ XOABS,YOABS,XPMAX,XSPACE,PSIZE
      COMMON/GLOOP/ ILOOP
      COMMON/ABLK/ A(3,3)
      COMMON/SAVEV/ DMAGS,IDMAG
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      COMMON/VALUES/ NVALLS
      COMMON/CASEID/ ICCASE
      COMMON ZZZ(1)
      DIMENSION DSAV(3)
      DIMENSION ABCD(8)
      NAMELIST/PICT/ KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,
     1PLOTSZ,XORGN,YORGN,PSCALE,NOTAT,KDISP,IDMAG,DMAGS,KODE,
     2KSYMXY,KSYMXZ,KSYMYZ,XXMAX,YYMAX,ZZMAX,XXMIN,
     3YYMIN,ZZMIN,NDMAX,NDMIN,NELMAX,NELMIN,XLHT
C
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(6,8)
    8 FORMAT(1H1)
C
C *** TO READ TITLE CARD FOR RUN
C
      READ(5,10) ABCD
   10 FORMAT(8A10)
      IF(EOF,5) 2222,3333
 2222 CALL PSTOP
 3333 CONTINUE
      WRITE(6,11) ABCD
   11 FORMAT(///,20X,8A10,///)
      CALL INITAL
      IF(KPLOT.NE.4.OR.KDATA.NE.5) GO TO 111
      REWIND 20
      CALL MESAGE(1,24HENTERING ORTHOGC PROGRAM,24)
      CALL MESAGE(1,19HSELECT DESIRED MODE,19)
      CALL MESAGE(1,18HBY PRESSING FN KEY,18)
      CALL NEXT(NKEY)
      IMODE = NKEY-1
      IF(IMODE.EQ.0) GO TO 111
      DO 57 J=1,IMODE
      KOUNT = 0
```

```
      IF(NUDISP.NE.0) KOUNT = KOUNT+1
      IF(NVDISP.NE.0) KOUNT = KOUNT+1
      IF(NWDISP.NE.0) KOUNT = 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
C *** TO SET POINTERS FOR BLANK COMMON STORAGE ZZZ
C *** (WITH INTEGER NAMES OF ARRAYS USED IN CALLED SUBROUTINES)
C
      NUMPT = 1
      XPT = NUMPT+NNDEST
      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(NWDISP.EQ.0) NEND = WPT+1-1
      IF(NWDISP.NE.0) NEND = WPT+NNDEST-1
      WRITE(6,15) NEND
   15 FORMAT(///,20X,*BLANK 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.9) CALL GEOM9
     1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
      CALL PNTOUT(1,
     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.5) CALL DATA5
     1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
      IF(KDATA.EQ.9) CALL DATA9
     1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
      CALL PNTOUT(2,
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
  700 CONTINUE
      IF(KPLOT.EQ.4.AND.ILOOP.NE.0) GO TO 6000
      READ(5,PICT)
      WRITE(6,PICT)
```

```
6000 CONTINUE
      IF(KPLOT.EQ.4) CALL CCRT1
      CALL DSCALE
     1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
      CALL BOUND
     1(ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(ZPT),ZZZ(UPT),ZZZ(VPT),ZZZ(WPT))
      CALL ROTAT
      CALL PLOT
     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 CALPLT(12.0,0.0,-3)
     IF(KPLOT.EQ.4) CALL CCRT2
     ILOOP = ILOOP+1
     IF(KODE.EQ.0) GO TO 800
     GO TO (700,600,500) KODE
 800 CONTINUE
     CALL PSTOP
     END OF MAIN
```

```
      SUBROUTINE DOCMNT
C
C *** THIS SUBROUTINE CONTAINS PROGRAM DOCUMENTATION
C
C
C
C                           DESCRIPTION OF INPUT DATA CARDS
C
C
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,
C                       AND CONTROL VALUES NEEDED BY THE PROGRAM.

C          THE FOLLOWING VALUES ARE INCLUDED---

C          NNDEST = ESTIMATE NUMBER OF GRID POINTS TO BE USED.  VALUE MUST BE
C                   GREATER THAN OR EQUAL TO THE ACTUAL NUMBER OF GRID POINTS.
C               ** DEFAULT = 200 **
C          NUDISP = 0 FOR NO DISPLACEMENT DATA IN X-DIRECTION.
C                 = 1 FOR DATA INCLUDING DISPLACEMENTS IN X-DIRECTION.
C               ** DEFAULT = 0 **
C          NVDISP = 0 FOR NO DISPLACEMENT DATA IN Y-DIRECTION.
C                 = 1 FOR DATA INCLUDING DISPLACEMENTS IN Y-DIRECTION.
C               ** DEFAULT = 0 **
C          NWDISP = 0 FOR NO DISPLACEMENT DATA IN Z-DIRECTION.
C                 = 1 FOR DATA INCLUDING DISPLACEMENTS IN Z-DIRECTION.
C               ** DEFAULT = 0 **
C
C          KGEOM SPECIFIES SUBROUTINE AND CORRESPONDING METHOD OF INPUT FOR
C          MODEL GEOMETRY.
C          KGEOM = 1 FOR GRID POINTS AND ELEMENTS READ FROM CARDS WITH USER
C                    SPECIFIED FORMAT.
C                = 2 FOR NASTRAN DECK WITH CARD IDENTIFIERS LEFT ADJUSTED
C                    AND DATA IN COLUMN WIDTHS OF 8.
C                = 9 FOR USER SUPPLIED SUBROUTINE - GEOM9.
C               ** DEFAULT = 1 **
C          KDATA SPECIFIES SUBROUTINE AND CORRESPONDING METHOD OF INPUT FOR
C          DISPLACEMENT DATA.
C          KDATA = 1 FOR SUBROUTINE DATA1 TO READ IN DISPLACEMENT DATA
C                    FROM CARDS WITH USER SPECIFIED FORMAT.
C                = 5 FOR SUBROUTINE DATA5 TO READ IN DISPLACEMENT DATA
C                    FROM TAPE20.
C                = 9 FOR SUBROUTINE DATA9, A USER SUPPLIED SUBROUTINE.
C               ** DEFAULT = 1 **
C          NVALUS - USED IF KDATA = 5 TO SPECIFY THE NUMBER OF SETS OF
C          DISPLACEMENTS TO BE READ FROM TAPE20.
C               ** DEFAULT = 0 **
C          IRESEQ = 0 FOR NO RESEQUENCING OF GRID POINT NUMBERS.
C                 = 1 TO RESEQUENCE GRID POINT NUMBERS IN SAME ORDER
C                     AS THEY ARE INPUT.
C               ** DEFAULT = 1 **
C          KPLOT SPECIFIES THE TYPE OF OUTPUT DEVICE TO BE USED.
C          KPLOT = 1 FOR CALCOMP.
C                = 2 FOR CALCOMP WITH PLOTTING SPEED REDUCED TO USE LEROY PENS.
```

```
C                = 3 FOR VARIAN.
C                = 4 FOR CRT (USE CDC250 SCOPES AT LRC)
C              ** DEFAULT = 1 **
C          XSPACE = SPACE BETWEEN PLOTS IN X-DIRECTION, IN INCHES.
C              ** DEFAULT = 10.0 **
C          PSIZE = PAPER SIZE IN Y-DIRECTION IN INCHES, USED IN SCALING OF
C                  PLOTS TO INSURE THIS DIMENSION IS NOT EXCEEDED.
C              ** DEFAULT = 25.0 **
C          IDCASE = 0 FOR NO TITLE CARD PRECEDING
C                     DECKS OF DISPLACEMENT VALUES.
C                 = 1 FOR TITLE CARD PRECEDING
C                     DECKS OF DISPLACEMENT VALUES.
C              ** DEFAULT = 0 **


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


C      USE IF KGEOM = 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 4 VALUES, GRID
C              POINT NUMBER (INTEGER), X-COORDINATE (REAL), Y-COORDINATE
C              (REAL) AND Z-COORDINATE (REAL).  THE FORMAT IS SPECIFIED
C              IN (A) ABOVE.
C          (C) A SINGLE CARD CONTAINING THE WORD ENDGRID 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.  FOR ROD OR BEAM
C              ELEMENTS THE LAST TWO INTEGER FIELDS MUST BE BLANK OR ZERO.
C              THE FORMAT IS SPECIFIED IN (D) ABOVE.
C          (F) A SINGLE CARD CONTAINING THE WORD ENDGEOM IN COLUMNS 1-7.


C      USE IF KGEOM = 2

C          (A) A SINGLE CARD CONTAINING THE WORD LINEEL IN COLUMNS 1-6 AND
C              UP TO NINE NASTRAN LINEAL 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 LINEAL ELEMENTS ARE NOT USED FOR THE PLOT.
C          (B) 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         (C) 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         (D) A NASTRAN BULK DATA DECK.  ONLY THE GRID CARDS AND THE
C             ELEMENT CONNECTION CARDS WITH NAMES MATCHING THOSE GIVEN ON
C             THE LINEEL, TRIAEL, AND QUADEL CARDS WILL BE USED FOR THE
C             PLOT.  ALL OTHER CARDS IN THE NASTRAN BULK DATA DECK WILL
C             BE IGNORED.
C         (E) A SINGLE CARD CONTAINING THE WORD ENDGEOM IN COLUMNS 1-7.


C      USE IF KGEOM = 9

C         CALL SUBROUTINE GEOM9 WHICH IS PREPARED BY THE USER TO READ
C         GEOMETRY DATA.


C     CASE IDENTIFICATION CARD.

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


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


C      USE IF KDATA = 1

C         (A) A SINGLE CARD CONTAINING THE WORD FORMAT IN COLUMNS 1-6 AND
C             A VARIABLE FORMAT FOR THE DATA CARDS WITH LEFT PARENTHESIS
C             STARTING IN COLUMN 11 AND UP TO 80 COLUMNS MAY BE USED.  IF
C             DISPLACEMENTS ARE INCLUDED FOR MORE THAN ONE GRID POINT PER
C             CARD, THE NUMBER OF GRID POINTS PER CARD MUST BE ENTERED
C             AS AN INTEGER IN COLUMN 8.
C         (B) DECK OF DISPLACEMENT SETS.  THERE CAN BE MULTIPLE DISPLACE-
C             MENT SETS PER CARD OR THE SET CAN EXTEND TO MORE THAN ONE
C             CARD (OFTEN THE CASE WITH NASTRAN PUNCHED OUTPUT) WHICH CAN
C             BE HANDLED WITH A FORMAT FOR READING MULTIPLE CARDS.
C             A DISPLACEMENT SET FOR EACH GRID POINT IS DEFINED TO
C             CONTAIN FROM 2 TO 4 VALUES, A GRID POINT NUMBER AND
C             DISPLACEMENTS CORRESPONDING TO NUDISP, NVDISP, OR NWDISP
C             EQUAL TO 1.
C         (C) BLANK CARD OR CARDS TO END DATA DECK.  THE NUMBER OF BLANK
C             CARDS MUST CORRESPOND TO THE NUMBER OF CARDS READ AT ONE
C             TIME BY THE SPECIFIED VARIABLE FORMAT.


C      USE IF KDATA = 5

C         READS NVALUS (FROM NAMELIST OPTION) DISPLACEMENT SETS FROM
C         TAPE20.  EACH DISPLACEMENT SET MUST HAVE BEEN WRITTEN ON TAPE20
C             AS AN UNFORMATTED RECORD.
```

```
C     USE IF KDATA = 9

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



C     NAMELIST PICT - CONTAINS VALUES NEEDED TO GENERATE PLOTS.

C         THE FOLLOWING VALUES ARE INCLUDED---

C         KHORZ = INTEGER DESIGNATING HORIZONTAL AXIS OF VIEWING PLANE,
C                 WHERE 1=X, 2=Y, 3=Z.
C              ** DEFAULT = 1 **
C         KVERT = INTEGER DESIGNATING VERTICAL AXIS OF VIEWING PLANE,
C                 WHERE 1=X, 2=Y, 3=Z.
C              ** DEFAULT = 2 **
C         PHI = ANGULAR ROTATION OF MODEL ABOUT ITS X-AXIS, IN DEGREES
C               (MUST BE TAKEN THIRD).
C              ** DEFAULT = 0.0 **
C         THETA = ANGULAR ROTATION OF MODEL ABOUT ITS Y-AXIS, IN DEGREES
C                 (MUST BE TAKEN SECOND).
C              ** DEFAULT = 0.0 **
C         PSI = ANGULAR ROTATION OF MODEL ABOUT ITS Z-AXIS, IN DEGREES
C               (MUST BE TAKEN FIRST).
C              ** DEFAULT = 0.0 **
C         NEWFR = 1 FOR FRAME CHANGE BEFORE PLOT IS MADE.
C                  (A FRAME CHANGE RESETS THE X-ORGIN PAST PREVIOUS PLOT
C                   BY XSPACE AND THE Y-ORGIN AT 0.0).
C         NEWFR .NE. 1 FOR NO FRAME CHANGE BEFORE PLOTTING.
C              ** DEFAULT = 1 **
C         ISCALE = 1 FOR INTERNAL ORGIN LOCATION AND SCALING.
C                = 2 FOR USER SPECIFIED ORGIN AND SCALING.
C              ** DEFAULT = 1 **
C         PLOTSZ = MAXIMUM DIMENSION DESIRED ON COMPLETED PLOT.
C                 (USED FOR SCALING IF ISCALE = 1)
C              ** DEFAULT = 10.0 **
C         XORGN = X-LOCATION OF PLOT ORIGIN (USED IF ISCALE = 2).
C              ** DEFAULT = 0.0 **
C         YORGN = Y-LOCATION OF PLOT ORIGIN (USED IF ISCALE = 2).
C              ** DEFAULT = 0.0 **
C         PSCALE = MODEL SIZE REDUCTION FACTOR, PSCALE = ACTUAL MODEL
C                  SIZE/DESIRED PLOT SIZE (USED IF ISCALE = 2).
C              ** DEFAULT = 1.0 **
C         NOTAT = 0 FOR NO NUMBERING ON PLOTS.
C               = 1 FOR NUMBERING OF GRID POINTS.
C               = 2 FOR NUMBERING OF ELEMENTS.
C              ** DEFAULT = 0 **
C         XLHT = HEIGHT OF INTEGERS SPECIFIED BY NOTAT, IN INCHES.
C              ** DEFAULT = 0.15 **
C         KDISP = 0 FOR UNDEFORMED PLOT.
C               = 1 FOR DEFORMED PLOT.
C               = 2 FOR EXPLODED PLOT.
C               = 3 FOR DISPLACEMENTS REPRESENTED BY VECTORS.
C              ** DEFAULT = 0 **
C         IDMAG = 1 FOR DIRECT SCALING OF DATA BY DMAGS.
C               = 2 FOR SCALING OF DATA TO A MAX. VALUE OF DMAGS.
C              ** DEFAULT = 2 **
```

```
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          NDMAX = 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 SPECIFIES CONTROL OPTION AFTER PLOT IS COMPLETE.
C          KODE = 0, LAST PLOT, EXIT FROM PROGRAM.
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 COMPRISES 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
C
C*******************************************************************************
C
C
C                          DESCRIPTION OF GRAPHICS SUBROUTINES
C
C
C
C    GRAPHICS         USED BY FOLLOWING        CALLED BY FOLLOWING
C    SUBROUTINE       PLOTTING DEVICE          PROGRAM SUBROUTINES
C
C
C    CALCOMP          CALCOMP                  INITAL
C
C    LEROY            CALCOMP                  INITAL
C
C    PSEUDO           VARIAN                   INITAL
C
C    CDC250           CRT                      INITAL
C
```

```
C     CALPLT          CALCOMP,VARIAN,CRT     MAIN,CCRT3,PSTOP,PLOT,GARROW
C
C     NOTATE          CALCOMP,VARIAN,CRT     MAIN
C
C     NUMBER          CALCOMP,VARIAN,CRT     PLOT,NDLET
C
C     NFRAME          VARIAN                 PLOT
C
C     NEXT            CRT                    MAIN,CCRT1,CCRT2,CCRT3
C
C     MESAGE          CRT                    MAIN,CCRT1,CCRT2,CCRT3
C
C     PARAMS          CRT                    CCRT1
C
C     KFORMAT         CRT                    CCRT1
C
C
C                              SUBROUTINE CALCOMP
C
C
C       PURPOSE         THIS IS THE NORMAL MODE PROCESSOR.  THE NECESSARY
C                       PARAMETERS AND LINKAGE ARE SET UP TO OUTPUT A TAPE
C                       FOR THE CALCOMP 780/763 0.010/0.005-INCH STEP PLOTTER.
C
C       USE             CALL CALCOMP
C
C       COMMENTS        THIS CALL MUST BE GIVEN BEFORE THE FIRST CALL TO A
C                       PLOTTING ROUTINE.
C
C
C                              SUBROUTINE LEROY
C
C
C       PURPOSE         THE PARAMETERS NECESSARY TO ACCOMODATE PLOTTING WITH THE
C                       LIQUID INK PEN ARE SET UP BY CALL LEROY.
C
C       USE             CALL LEROY
C
C       COMMENTS        THIS CALL SHOULD ONLY BE USED WITH THE CALCOMP PROCESSOR.
C                       IN ADDITION TO REDUCING THE SPEED OF THE PLOTTER FOR
C                       ALL PLOTTING MOVEMENTS, THE NUMBER OF PLOT VECTORS IN ANY
C                       ANNOTATION IS CONSIDERABLY INCREASED.
C
C
C                              SUBROUTINE PSEUDO
C
C
C       PURPOSE         INITIALIZES PLOT VECTOR FILE FOR VARIAN PLOTTER
C
C       USE             CALL PSEUDO
C
C
C                              SUBROUTINE CDC250
C
C
C       PURPOSE         INITIALIZES CATHODE RAY TUBE CONSOLE.
C
C       USE             CALL CDC250
```

```
C
C
C                                 SUBROUTINE CALPLT
C
C
C      PURPOSE          TO MOVE THE PLOTTER PEN TO A NEW LOCATION WITH PEN UP
C                       OF DOWN AND TO SIGNAL THE END OF A JOB SEGMENT BY
C                       INCREMENTING THE BLOCK ADDRESS NUMBER.
C
C      USE              CALL CALPLT(X,Y,IPEN)
C
C                       WHERE
C
C                       X,Y      ARE THE FLOATING POINT VALUES FOR PEN MOVEMENT.
C
C                       IPEN = 2 PEN DOWN
C
C                            = 3 PEN UP
C
C                                NEGATIVE IPEN WILL ASSIGN X=0, Y=0
C                                AS THE LOCATION OF THE PEN AFTER MOVING THE
C                                X,Y (CREATE A NEW REFERENCE POINT) AND
C                                INCREASE THE BLOCK NUMBER BY ONE.
C
C      COMMENTS         ALL X AND Y COORDINATES MUST BE EXPRESSED AS FLOATING
C                       POINT INCHES (ACTUAL PAGE DIMENSIONS) IN DEFLECTION FROM
C                       THE ORIGIN.
C
C
C                                SUBROUTINE NOTATE
C
C
C      PURPOSE          TO DRAW ALPHANUMERIC INFORMATION FOR ANNOTATION AND LABELIN
C                       AND PROVIDE SPECIAL CENTERED SYMBOLS FOR ANNOTATION OF
C                       DATA POINTS.
C
C      USE              CALL NOTATE(X,Y,HEIGHT,BCD,THETA,N)
C
C                       WHERE
C
C                       X,Y      ARE THE FLOATING POINT PAGE COORDINATES OF THE
C                                FIRST CHARACTER.  FOR ALPHANUMERIC CHARACTERS
C                                THE COORDINATES OF THE LOWER LEFT-HAND CORNER
C                                OF THE CHARACTERS ARE SPECIFIED.
C
C                       HEIGHT   SPECIFIES CHARACTER SIZE AND SPACING IN FLOATING
C                                POINT INCHES FOR A FULL-SIZE CHARACTER.  THE
C                                WIDTH OF A CHARACTER WILL BE (4/7)*HEIGHT AND THE
C                                SPACE BETWEEN CHARACTERS IS (2/7)*HEIGHT.
C
C                       BCD      IS THE STRING OF ALPHANUMERIC CHARACTERS TO BE
C                                DRAWN.
C
C                       THETA    IS THE ANGLE IN FLOATING POINT DEGREES AT WHICH
C                                THE INFORMATION IS TO BE DRAWN.
C
C                       N        IS THE NUMBER OF CHARACTERS, INCLUDING BLANKS, IN
C                                THE LABEL.
C
C
```

```
C                                   SUBROUTINE NUMBER
C
C
C      PURPOSE             TO CONVERT A FLOATING NUMBER TO BCD (EXPRESSED IN
C                          F FORMAT), AND DRAW THE RESULTING APLHANUMERIC CHARACTERS.
C
C      USE                 CALL NUMBER(X,Y,SIZE,FPN,THETA,N)
C
C                          WHERE
C
C                          X,Y        ARE THE COORDINATES IN FLOATING POINT INCHES
C                                     OF THE LEFT LOWER CORNER OF THE FIRST DIGIT
C                                     OF OUTPUT.
C
C                          SIZE       IS THE HEIGHT OF THE PLOTTED NUMBER IN FLOATING
C                                     POINT INCHES.
C                          FPN        IS THE FLOATING POINT NUMBER TO BE DRAWN.
C
C                          THETA      IS THE ANGLE IN FLOATING POINT DEGREES AT WHICH
C                                     THE NUMBER IS TO BE DRAWN.
C
C                          N          IS THE NUMBER OF DECIMAL DIGITS TO THE RIGHT
C                                     OF THE DECIMAL POINT FOR OUTPUT.
C                                     N = -1 AND N = 0 BOTH SPECIFY NO DECIMAL PLACES,
C                                     HOWEVER, -1 SUPPRESSES THE DECIMAL POINT.
C
C      COMMENTS            THE NUMBER IS RESTRICTED TO A MAXIMUM OF 12 DIGITS.
C                          THE ROUTINE TRUNCATES THE FLOATING POINT NUMBER AT THE
C                          REQUIRED DECIMAL PLACE.
C
C
C                                   SUBROUTINE NFRAME
C
C
C      PURPOSE             USED BY VARIAN PLOTTER TO ADVANCE PLOTTING FRAME.
C
C      USE                 CALL NFRAME
C
C
C                                   SUBROUTINE NEXT
C
C
C      PURPOSE             PROVIDES A BREAK POINT OR HALT DURING APPLICATION PROGRAM
C                          EXECUTION.  OPERATOR MUST PRESS FUNCTION KEY TO RESUME,
C                          AND NUMBER OF KEY IS RETURNED IN CALLING PARAMETER.
C
C      USE                 CALL NEXT(NKEY)
C
C                          WHERE
C
C                          NKEY IS NUMBER OF FUNCTION KEY PRESSED.
C
C
```

```
C                               SUBROUTINE MESAGE
C
C
C      PURPOSE         PROVIDES THE CAPABILITY TO DISPLAY A MESSAGE ON THE CRT250.
C
C      USE             CALL MESAGE(I,BCD,N)
C
C                      WHERE
C
C                      I INDICATES INTENSITY OF CHARACTER DISPLAY.
C
C                      BCD IS ADDRESS OF ARRAY CONTAINING THE MESSAGE
C                      IN HOLLERITH FORM.
C
C                      N IS THE NUMBER OF CHARACTERS IN THE MESSAGE (LESS THAN 50)
C
C
C                               SUBROUTINE PARAMS
C
C
C      PURPOSE         USED TO GENERATE A TABLE OF SYMBOLIC NAMES THAT CAN BE
C                      ACCESSED USING THE ALPHANUMERIC KEYBOARD ON THE CRT250.
C
C      USE             CALL PARAMS(BCD,VAR)
C
C                      WHERE
C
C                      BCD IS THE HOLLERITH REPRESENTATION OF SYMBOLIC NAME.
C
C                      VAR IS PROGRAM VARIABLE REFERRED TO BE SYMBOLIC NAME.
C
C      COMMENTS        UP TO 3 PAIRS OF VARIABLES MAY BE SPECIFIED IN A SINGLE
C                      CALL TO PARAMS.  TABLE HAS CAPACITY FOR 42 PAIRS.
C
C
C                               SUBROUTINE KFORMAT
C
C
C      PURPOSE         ALLOWS PROGRAMMER TO CHANGE FORMAT FOR KEYBOARD INPUT.
C
C      USE             CALL KFORMAT(NHBCD)
C
C                      WHERE
C
C                      N IS THE NUMBER OF CHARACTERS IN BCD.
C
C                      H IS REQUIRED.
C
C                      BCD IS THE REQUIRED FORMAT (I4,F4.2,A10,ETC.)
C
C
       RETURN
       END OF DOCMNT
```

```
      SUBROUTINE CCRT1
C
C *** FOR CHANGING VALUES INPUT BY $PICT USING CRT.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,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/SAVEV/ DMAGS,IDMAG
C
C *** FOR INTEGER CONTROL VALUES
C
      CALL KFORMAT(3HI10)
      CALL PARAMS
      CALL PARAMS(5LKHORZ,KHORZ,5LKVERT,KVERT)
      CALL PARAMS(5LNEWFR,NEWFR,6LISCALE,ISCALE)
      CALL PARAMS(5LNOTAT,NOTAT,5LKDISP,KDISP,5LIDMAG,IDMAG)
      CALL PARAMS(6LKSYMXY,KSYMXY,6LKSYMXZ,KSYMXZ,6LKSYMYZ,KSYMYZ)
      CALL PARAMS(5LNDMAX,NDMAX,5LNDMIN,NDMIN)
      CALL PARAMS(6LNELMAX,NELMAX,6LNELMIN,NELMIN)
      CALL MESAGE(1,32HTO CHANGE INTEGER CONTROL VALUES,32)
      CALL MESAGE(1,13HVARIABLES ARE,13)
      CALL MESAGE(1,27HKHORZ, KVERT, NEWFR, ISCALE,27)
      CALL MESAGE(1,19HNOTAT, KDISP, IDMAG,19)
      CALL MESAGE(1,22HKSYMXY, KSYMXZ, KSYMYZ,22)
      CALL MESAGE(1,28HNDMAX, NDMIN, NELMAX, NELMIN,28)
      CALL MESAGE(1,17HANY KEY CONTINUES,17)
      CALL NEXT(NKEY)
C
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 MESAGE(1,39HTO CHANGE FLOATING POINT CONTROL VALUES,39)
      CALL MESAGE(1,13HVARIABLES ARE,13)
      CALL MESAGE(1,32HYAW, ROLL, PITCH, PLOTSZ, PSCALE,32)
      CALL MESAGE(1,25HXORGN, YORGN, DMAGS, XLHT,25)
      CALL MESAGE(1,19HXXMAX, YYMAX, ZZMAX,19)
      CALL MESAGE(1,19HXXMIN, YYMIN, ZZMIN,19)
      CALL MESAGE(1,17HANY KEY CONTINUES,17)
      CALL NEXT(NKEY)
      RETURN
      END OF CCRT1
```

```
      SUBROUTINE CCRT2
C
C *** FOR SELECTING CONTROL OPTION, KODE, AT END OF JOB USING CRT.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,LSCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
   10 CONTINUE
      CALL MESAGE(1,1H ,1)
      CALL MESAGE(1,32HFN KEY 12 TO ALTER EXISTING PLOT,32)
      CALL MESAGE(1,42HFN KEY 24 TO READ NEW SET OF DISPLACEMENTS,42)
      CALL MESAGE(1,37HFN KEY 36 TO READ A COMPLETE NEW CASE,37)
      CALL MESAGE(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
C
C *** REMINDER TO PUT EOF ON PLOT FILE WHEN USING CRT.
C
      CALL MESAGE(1,1H ,1)
      CALL MESAGE(1,45HLAST REMINDER TO PUT EOF ON PLOT FILE, IF ANY,45)
      CALL MESAGE(1,1H ,1)
      CALL MESAGE(1,40HDO IT AT NEXT PLOT FILE COMPLETE MESSAGE,40)
      CALL MESAGE(1,1H ,1)
      CALL MESAGE(1,17HANY KEY CONTINUES,17)
      CALL NEXT(NKEY)
      CALL CALPLT(12.0,0.0,-3)
      RETURN
      END OF CCRT3

      SUBROUTINE PSTOP
C
C *** TO TERMINATE JOB.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,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
C
C *** TO SET UP VALUES FOR CONTROL PARAMETERS
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,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/CORGN/ XOABS,YOABS,XPMAX,XSPACE,PSIZE
      COMMON/SAVEV/ DMAGS,IDMAG
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      COMMON/SEQNCE/ IRESEQ
      COMMON/VALUES/ NVALUS
      COMMON/CASEID/ IDCASE
      NAMELIST/OPTION/ NNDEST,NUDISP,NVDISP,NWDISP,
     1KGEOM,KDATA,NVALUS,IRESEQ,KPLOT,XSPACE,PSIZE,IDCASE
C
C *** DESCRIPTION OF VALUES IN $OPTION GIVEN IN SUBROUTINE DOCMNT
C
C
C *** TO SET DEFAULT VALUES FOR $OPTION
C
      NNDEST = 200
      NUDISP = 0
      NVDISP = 0
      NWDISP = 0
      KGEOM = 1
      KDATA = 1
      NVALUS = 0
      IRESEQ = 1
      KPLOT = 1
      XSPACE = 10.0
      PSIZE = 25.0
      IDCASE = 0
C
C *** TO SET DEFAULT VALUES FOR $PICT
C
      KHORZ = 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
      PSCALE = 1.0
      NOTAT = 0
      XLHT = 0.15
      KDISP = 0
      IDMAG = 2
      DMAGS = 1.0
      KSYMXY = 0
      KSYMXZ = 0
      KSYMYZ = 0
      XXMAX = 1.0E20
```

```
      YYMAX = 1.0E20
      ZZMAX = 1.0E20
      XXMIN = -1.0E20
      YYMIN = -1.0E20
      ZZMIN = -1.0E20
      NDMAX = 9999999999
      NDMIN = 0
      NELMAX = 9999999999
      NELMIN = 0
      KODE = 0
      READ(5,OPTION)
      IF(EOF,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 INITAL
```

```
      SUBROUTINE GEOM1(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)
C
C *** TO READ GRID POINT INFORMATION AND ELEMENTS
C *** FROM CARDS WITH USER SPECIFIED FORMAT.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      COMMON/SEQNCE/ IRESEQ
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
      DIMENSION ABCD(8)
      DIMENSION FORMT(7)
      DIMENSION NODE(4)
      REWIND 10
      TEST1 = 10HENDGRID
      TEST2 = 10HENDGEOM
      TEST4 = 10HFORMAT
C
C *** TO READ GRID INFORMATION
C
  100 CONTINUE
      READ(5,10) ABCD
   10 FORMAT(8A10)
      IF(ABCD(1).EQ.TEST1) GO TO 1000
      IF(ABCD(1).EQ.TEST4) 50,60
   50 CONTINUE
      DECODE(80,55,ABCD) FORMT
   55 FORMAT(10X,7A10)
      GO TO 100
   60 CONTINUE
      NNODE = NNODE+1
      DECODE(80,FORMT,ABCD) NUMPT(NNODE),XPT(NNODE),YPT(NNODE)
     1,ZPT(NNODE)
      GO TO 100
 1000 CONTINUE
C
C *** TO READ ELEMENT CONNECTION INFORMATION
C
  200 CONTINUE
      READ(5,10) ABCD
      IF(ABCD(1).EQ.TEST2) 275,285
  275 CONTINUE
      END FILE 10
      GO TO 2000
  285 CONTINUE
      IF(ABCD(1).EQ.TEST4) 250,260
  250 CONTINUE
      DECODE(80,55,ABCD) FORMT
      GO TO 200
  260 CONTINUE
      IDUM = 1
      DECODE(80,FORMT,ABCD) NUMEL,NODE1,NODE2,NODE3,NODE4
      IF(IRESEQ.NE.1) GO TO 700
```

```
C
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,NNODE
      IF(NODE(I).EQ.NUMPT(J)) 511,510
  511 CONTINUE
      NODE(I) = 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 200
 2000 CONTINUE
      RETURN
      END OF GEOM1
```

```
      SUBROUTINE GEOM2(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)
C
C *** FOR INPUT OF NASTRAN DECK TO DESCRIBE GEOMETRY.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/SEQNCE/ IRESEQ
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,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 = 10HENDGEOM
      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(A8)
      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) (LINEEL(I),I=1,9)
  111 FORMAT(8X,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
C *** TO READ GRID CARDS
C
  100 CONTINUE
      NNODE = NNODE+1
      DECODE(80,101,ABCD) NUMPT(NNODE),XPT(NNODE),YPT(NNODE)
     1,ZPT(NNODE)
```

```
  101 FORMAT(8X,A8,8X,3F8.0)
      CALL IRITE(NUMPT(NNODE))
      GO TO 800
C
C *** TO READ CARDS CONTAINING ELEMENTS WITH 2 GRID POINTS
C
  200 CONTINUE
      TEST100 = 10HPLOTEL
      IF(WORD1.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
C *** TO READ PLOTEL 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
C *** TO READ CARDS CONTAINING ELEMENTS WITH 3 GRID POINTS
C
  300 CONTINUE
      DECODE(80,301,ABCD) NEL,ND1,ND2,ND3
  301 FORMAT(8X,A8,8X,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
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,A8,8X,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
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(I) = 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
      DECODE(8,1,NUM) NSAVE
    1 FORMAT(I8)
      DECODE(8,2,NUM) (N(I),I=1,8)
    2 FORMAT(8A1)
      DO 10 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,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
C
C *** INSERT ROUTINE HERE
C
      RETURN
      END OF GEOM9
```

```
      SUBROUTINE BOUND(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)
C
C *** TO DETERMINE MAXIMUM DIMENSIONAL LIMITS OF BODY FOR USE
C     IN SCALING PLOTS
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,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/XYZLIM/ XYZMAX(3),XYZMIN(3)
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
      DIMENSION NODE(4)
      DO 5 I=1,3
      XYZMIN(I) = +1.0E20
      XYZMAX(I) = -1.0E20
    5 CONTINUE
      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 15
      IF(NUMPT(ND).LT.NDMIN.OR.NUMPT(ND).GT.NDMAX) GO TO 100
   10 CONTINUE
   15 CONTINUE
      DO 20 I=1,4
      IF(NODE(I).EQ.0) GO TO 25
      ND = NODE(I)
      IF(XPT(ND).GT.XXMAX) GO TO 20
      IF(XPT(ND).LT.XXMIN) GO TO 20
      IF(YPT(ND).GT.YYMAX) GO TO 20
      IF(YPT(ND).LT.YYMIN) GO TO 20
      IF(ZPT(ND).GT.ZZMAX) GO TO 20
      IF(ZPT(ND).LT.ZZMIN) GO TO 20
      IF(XPT(ND).GT.XYZMAX(1)) XYZMAX(1) = XPT(ND)
      IF(XPT(ND).LT.XYZMIN(1)) XYZMIN(1) = XPT(ND)
      IF(YPT(ND).GT.XYZMAX(2)) XYZMAX(2) = YPT(ND)
      IF(YPT(ND).LT.XYZMIN(2)) XYZMIN(2) = YPT(ND)
      IF(ZPT(ND).GT.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 I=1,3
      IF(I.EQ.1.AND.KSYMYZ.NE.1) GO TO 300
      IF(I.EQ.2.AND.KSYMXZ.NE.1) GO TO 300
      IF(I.EQ.3.AND.KSYMXY.NE.1) GO TO 300
```

```
      XYZBIG = ABS(XYZMAX(I))
      IF(ABS(XYZMIN(I)).GT.XYZBIG) XYZBIG = 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/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
      IF(NUDISP.EQ.0) GO TO 200
      DO 150 I=1,NUDISP
      UPT(I) = 0.0
  150 CONTINUE
  200 CONTINUE
      IF(NVDISP.EQ.0) GO TO 300
      DO 250 I=1,NVDISP
      VPT(I) = 0.0
  250 CONTINUE
  300 CONTINUE
      IF(NWDISP.EQ.0) GO TO 400
      DO 350 I=1,NWDISP
      WPT(I) = 0.0
  350 CONTINUE
  400 CONTINUE
      RETURN
      END OF ZEROD
```

```
      SUBROUTINE DATA1(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)
C
C *** TO READ DISPLACEMENT DATA FROM CARDS
C *** WITH USER SPECIFIED FORMAT.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      COMMON/SEQNCE/ IRESEQ
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
      DIMENSION ABCD(8)
      DIMENSION FORMT(7)
      DIMENSION ISAV(10),DSAV(10,3)
   10 FORMAT(8A10)
C
C *** TO READ DISPLACEMENT INFORMATION
C
      TEST = 6HFORMAT
      READ(5,10) ABCD
      DECODE(80,45,ABCD) WORD,KVALU
   45 FORMAT(A6,1X,I1)
      IF(KVALU.EQ.0) KVALU = 1
      IF(WORD.EQ.TEST) 300,200
  200 WRITE(6,20)
   20 FORMAT(1H1,///,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),(DSAV(K,I),I=1,KOUNT),K=1,KVALU)
      DO 400 K=1,KVALU
      IDUM = ISAV(K)
      IF(IDUM.EQ.0) GO TO 1000
      IF(IRESEQ.NE.1) GO TO 700
C
C *** FOR RESEQUENCED GRID POINTS
C
      DO 500 J=1,NNODE
      IF(NUMPT(J).EQ.IDUM) 501,500
  501 CONTINUE
      KOUNT = 1
      IF(NUDISP.NE.0) 610,620
  610 UPT(J) = DSAV(K,KOUNT)
      KOUNT = KOUNT+1
  620 CONTINUE
      IF(NVDISP.NE.0) 630,640
  630 VPT(J) = DSAV(K,KOUNT)
      KOUNT = KOUNT+1
  640 CONTINUE
      IF(NWDISP.NE.0) 650,660
  650 WPT(J) = DSAV(K,KOUNT)
  660 CONTINUE
```

```
      GO TO 550
  500 CONTINUE
  550 CONTINUE
      GO TO 400
  700 CONTINUE
C
C *** FOR NO RESEQUENCE OF GRID POINTS
C
      KOUNT = 1
      IF(NUDISP.NE.0) 1610,1620
 1610 UPT(IDUM) = DSAV(K,KOUNT)
      KOUNT = KOUNT+1
 1620 CONTINUE
      IF(NVDISP.NE.0) 1630,1640
 1630 VPT(IDUM) = DSAV(K,KOUNT)
      KOUNT = KOUNT+1
 1640 CONTINUE
      IF(NWDISP.NE.0) 1650,1660
 1650 WPT(IDUM) = DSAV(K,KOUNT)
 1660 CONTINUE
  400 CONTINUE
      GO TO 100
 1000 CONTINUE
      RETURN
      END OF DATA1
```

```
      SUBROUTINE DATA5(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)
C
C *** TO READ DISPLACEMENT DATA FROM TAPE20.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      COMMON/SEQNCE/ IRESEQ
      COMMON/VALUES/ NVALUS
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
      DIMENSION DSAV(3)
      DO 10 I=1,NVALUS
      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,(DSAV(K),K=1,KOUNT)
      KOUNT = 1
      IF(NUDISP.NE.0) 610,620
  610 UDUM = DSAV(KOUNT)
      KOUNT = KOUNT+1
  620 CONTINUE
      IF(NVDISP.NE.0) 630,640
  630 VDUM = DSAV(KOUNT)
      KOUNT = KOUNT+1
  640 CONTINUE
      IF(NWDISP.NE.0) 650,660
  650 WDUM = DSAV(KOUNT)
  660 CONTINUE
      IF(IRESEQ.NE.1) GO TO 550
      DO 500 J=1,NNODE
      IF(NUMPT(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)
C
C *** USER SUPPLIED DISPLACEMENT INPUT SUBROUTINE.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
C
C *** INSERT ROUTINE HERE
C
      RETURN
      END OF DATA9
```

```
      SUBROUTINE PNTOUT(IOUT,NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)
C
C *** FOR PRINTED OUTPUT OF INFORMATION IN BLANK COMMON - ZZZ
C
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,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 GECMETRY 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*/
     25X,*NUMBER*,9X,*NUMBER*,13X,*X*,14X,*Y*,14X,*Z*//)
      DO 30 I=1,NNODE
      WRITE(6,18) I,NUMPT(I),XPT(I),YPT(I),ZPT(I)
   18 FORMAT(2X,I10,5X,I10,3X,3E15.4)
   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*/
     25X,*NUMBER*,9X,*NUMBER*,13X,*1*,9X,*2*,9X,*3*,9X,*4*//)
      REWIND 10
      I = 0
   35 CONTINUE
      I = I+1
      CALL RECIN(10,1,5,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,4I10)
      GO TO 35
 2000 CONTINUE
C
C *** FOR OUTPUT OF DISPLACEMENT DATA
C
      WRITE(6,210)
  210 FORMAT(///,5X,*DISPLACEMENTS TO BE PLOTTED*,///)
      WRITE(6,17)
      DO 230 I=1,NNODE
      U = 0.0
      IF(NUDISP.NE.0) U = UPT(I)
      V = 0.0
      IF(NVDISP.NE.0) V = VPT(I)
      W = 0
      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
```

```
      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,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/SAVEV/ DMAGS,IDMAG
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,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) IDMAG
   10 CONTINUE
      DMAG = DMAGS
      GO TO 30
   20 CONTINUE
      DMAX = 0.0
      DO 100 I=1,NNODE
      IF(NUDISP.EQ.0) GO TO 500
      IF(ABS(UPT(I)).GT.DMAX) DMAX = ABS(UPT(I))
  500 CONTINUE
      IF(NVDISP.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
  100 CONTINUE
      DMAG = DMAGS/DMAX
   30 CONTINUE
      RETURN
      END OF DSCALE
```

```
      SUBROUTINE PLOT(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)
C
C *** FOR GENERATING PLOTS.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,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/XYZLIM/ XYZMAX(3),XYZMIN(3)
      COMMON/CORGN/ XOABS,YOABS,XPMAX,XSPACE,PSIZE
      COMMON/GLOOP/ ILOOP
      COMMON/ABLK/ A(3,3)
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      COMMON/PDELS/ DELX,DELY
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
      DIMENSION NODE(4),X(4),Y(4),Z(4),XDISP(4),YDISP(4),ZDISP(4),
     1XROT(4),YROT(4)
      DIMENSION TEST(3)
      DIMENSION ABCD(8)
    1 FORMAT(8A10)
    2 FORMAT(1X,8A10)
C
C *** TO MAKE ALL GRID POINT NUMBERS NEGATIVE
C
      DO 50 I=1,NNODE
      NUMPT(I) = -NUMPT(I)
   50 CONTINUE
      PI = 3.1415926
      XMOVE = 0.0
      IF(NEWFR.EQ.1) XMOVE = XPMAX+XSPACE
      YMOVE = -YOABS
      CALL CALPLT(XMOVE,YMOVE,-3)
      XOABS = XOABS+XMOVE
      YOABS = YOABS+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 XYSCAL
      CALL CALPLT(XORGN,YORGN,-3)
      XOABS = XOABS+XORGN
      YOABS = YOABS+YORGN
      XSHIFT = 0.0
      YSHIFT = 0.0
      ZSHIFT = 0.0
      XPMAX = -1.0E20
C
C *** LOOPS TO ACCOUNT FOR SYMMETRY
C
      ZSIGN = +1.0
      DO 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
```

```
      DO 510 JJ=1,2
      IF(JJ.EQ.2.AND.KSYMXZ.NE.1) GO TO 510
      IF(JJ.EQ.2.AND.KSYMXZ.EQ.1) YSIGN = -1.0
      XSIGN = +1.0
      DO 520 KK=1,2
      IF(KK.EQ.2.AND.KSYMYZ.NE.1) GO TO 520
      IF(KK.EQ.2.AND.KSYMYZ.EQ.1) XSIGN = -1.0
C
C *** TO DETERMINE PRCJECTED 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+YRCT(N)
```

```
      XROT(N) = XROT(N)+DELX
      YROT(N) = YROT(N)+DELY
      IF(XROT(N).GT.XPMAX) XPMAX = XROT(N)
   25 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
C *** TO PLOT ELEMENTS
C
      CALL CALPLT(XROT(1),YROT(1),3)
      DO 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
C *** TO PLOT VECTORS AT GRID POINTS
C
      DO 601 ND=1,NNODE
      IF(NUMPT(ND).LE.0) GO TO 601
      IF(NUMPT(ND).LT.NDMIN.OR.NUMPT(ND).GT.NDMAX) 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(ND)+XSHIFT)/PSCALE
      Y(1) = YSIGN*(YPT(ND)+YSHIFT)/PSCALE
      Z(1) = ZSIGN*(ZPT(ND)+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(ND)+XDISP(1)*DMAG+XSHIFT)/PSCALE
      Y(2) = YSIGN*(YPT(ND)+YDISP(1)*DMAG+XSHIFT)/PSCALE
      Z(2) = ZSIGN*(ZPT(ND)+ZDISP(1)*DMAG+ZSHIFT)/PSCALE
      I = KHORZ
      J = KVERT
      DO 605 N=1,2
      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)
      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
  520 CONTINUE
  510 CONTINUE
  500 CONTINUE
      RETURN
      END OF PLOT


      SUBROUTINE ROTAT
C
C *** SETS UP COEFFICIENTS OF ROTATION MATRIX
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,LSCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/ABLK/ A(3,3)
      PI = 3.1415926536
      SINPHI  =  SIN(PHI*PI/180.0)
      COSPHI  =  COS(PHI*PI/180.0)
      SINTHE  =  SIN(THETA*PI/180.0)
      COSTHE  =  COS(THETA*PI/180.0)
      SINPSI  =  SIN(PSI*PI/180.0)
      COSPSI  =  COS(PSI*PI/180.0)
      A(1,1)  =  COSTHE*COSPSI
      A(1,2)  =  COSPSI*SINTHE*SINPHI-SINPSI*COSPHI
      A(1,3)  =  SINTHE*COSPHI*COSPSI+SINPHI*SINPSI
      A(2,1)  =  SINPSI*COSTHE
      A(2,2)  =  SINTHE*SINPHI*SINPSI+COSPHI*COSPSI
      A(2,3)  =  SINTHE*COSPHI*SINPSI-SINPHI*COSPSI
      A(3,1)  =  -SINTHE
      A(3,2)  =  COSTHE*SINPHI
      A(3,3)  =  COSTHE*COSPHI
      RETURN
      END OF ROTAT
```

```
      SUBROUTINE XYSCAL
C
C *** TO DETERMINE SCALE FACTOR FOR MODEL GEOMETRY.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      COMMON/XYZLIM/ XYZMAX(3),XYZMIN(3)
      COMMON/CORGN/ XOABS,YOABS,XPMAX,XSPACE,PSIZE
      COMMON/ABLK/ A(3,3)
      COMMON/PDELS/ DELX,DELY
      I = KHORZ
      J = KVERT
      DMAX = 0.0
      DO 5 N=1,3
      VDUM = ABS(XYZMAX(N)-XYZMIN(N))
      IF(VDUM.GT.DMAX) DMAX = VDUM
    5 CONTINUE
      PSCALE = DMAX/PLOTSZ
      DO 10 L=1,2
      DO 10 M=1,2
      DO 10 N=1,2
      X = XYZMIN(1)
      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/PSCALE.GT.PSIZE) PSCALE = YR/PSIZE
      XRMAX = XRMAX/PSCALE
      YRMAX = YRMAX/PSCALE
      XRMIN = XRMIN/PSCALE
      YRMIN = YRMIN/PSCALE
      DELX = -XRMIN
      DELY = -YRMIN
      XORGN = 0.0
      YORGN = 0.0
      RETURN
      END OF XYSCAL
```

```
      SUBROUTINE XPLOD(NEND,X,Y,Z)
C
C *** FOR GENERATING EXPLODED PLOTS.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,KVERT,PHI,THETA,PSI,NEWFR,ISCALE,PLOTSZ,XORGN,YORGN,
     2PSCALE,KDISP,DMAG,KODE
      DIMENSION X(4),Y(4),Z(4)
C
C *** TO CALCULATE THE INCENTER OF TRIANGLES
C
      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
C
C *** TO CALCULATE THE CENTROID OF RODS,BARS,AND QUADS
C
      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
C
C *** TO REDUCE THE SIZE OF THE ELEMENT
C
      DO 200 I=1,NEND
      X(I) = X(I)*DMAG
      Y(I) = Y(I)*DMAG
      Z(I) = Z(I)*DMAG
  200 CONTINUE
C
C *** TO CALCULATE THE CENTROID OF THE REDUCED ELEMENT
C
      XRC = XOC*DMAG
      YRC = YOC*DMAG
      ZRC = ZOC*DMAG
C
C *** SHIFT CORNERS OF ORIGINAL AND REDUCED TO MAKE CENTROIDS MATCH
C
      DO 400 I=1,NEND
      X(I) = X(I)+(XOC-XRC)
```

```
      Y(I) = Y(I)+(YOC-YRC)
      Z(I) = Z(I)+(ZOC-ZRC)
  400 CONTINUE
      RETURN
      END OF XPLOD


      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 NDLET(NUMPT,XPT,YPT,ZPT,UPT,VPT,WPT)
C
C *** FOR ANNOTATING GRID POINT NUMBERS ON PLOTS.
C
      COMMON/CONTRL/ KGEOM,KDATA,KPLOT,KSYMXY,KSYMXZ,KSYMYZ,NOTAT,XLHT,
     1KHORZ,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/XYZLIM/ XYZMAX(3),XYZMIN(3)
      COMMON/ABLK/ A(3,3)
      COMMON/KOUNT/ NNODE,NNDEST,NUDISP,NVDISP,NWDISP
      COMMON/PDELS/ DELX,DELY
      DIMENSION NUMPT(1),XPT(1),YPT(1),ZPT(1),UPT(1),VPT(1),WPT(1)
      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).LT.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
      XROT = A(II,1)*X+A(II,2)*Y+A(II,3)*Z
      YROT = A(JJ,1)*X+A(JJ,2)*Y+A(JJ,3)*Z
      XL = XROT+XLHT/2.0
      YL = YROT+XLHT/2.0
      XL = XL+DELX
      YL = YL+DELY
      AL = NUMPT(I)
      CALL NUMBER(XL,YL,XLHT,AL,0.0,-1)
  500 CONTINUE
      RETURN
      END OF NDLET
```

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.

```
      PROGRAM MAIN(INPUT=201,OUTPUT=201,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE20=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,RADPT,NUMEL,NODE1,NODE2,NODE3,NODE4,
     1CENTR
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      COMMON/GKODE/ KODE,NPLOT
      COMMON/VALUES/ NVALUS
      COMMON/CASEID/ ICCASE
      COMMON/SAVEV/ WMAGS,ICNTRS
      COMMON/SIZEXY/ XMINIM,XMAXIM,YMINIM,YMAXIM,XSPACE,YSPACE
      COMMON ZZZ(1)
      DIMENSION ABCD(8),BCD(2)
      NAMELIST/PICT/ NPLOT,XORGN,YORGN,PSCALE,ISCALE,WMAGS,ICNTRS,XLHT,
     1NXLAB,XLAB,NYLAB,YLAB,KODE
  500 CONTINUE
      ILOOP = 0
C
C *** TO ZERO NODE AND ELEMENT COUNTERS AND SET TEST VALUES
C
      NNODE = 0
      NELMT = 0
      WRITE(6,8)
    8 FORMAT(1H1)
C
C *** TO READ TITLE CARD FOR RUN
C
      READ(5,10) ABCD
   10 FORMAT(8A10)
      IF(EOF,5) 2222,3333
 2222 CALL PSTOP
 3333 CONTINUE
      WRITE(6,11) ABCD
   11 FORMAT(///,20X,8A10,///)
      CALL INITAL
      IF(KPLOT.NE.4.OR.KDATA.NE.5) GO TO 111
      REWIND 20
      CALL MESAGE(1,24HENTERING CONTOUR PROGRAM,24)
      CALL MESAGE(1,19HSELECT DESIRED MODE,19)
      CALL MESAGE(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,NVALUS
      READ(20) IDUM,WDUM
   58 CONTINUE
   57 CONTINUE
  111 CONTINUE
      HEIGHT = 0.15
      XSTRT = 2.0*HEIGHT+2.0
```

```
      YSTRT = 1.0
      CALL NOTATE(XSTRT,YSTRT,HEIGHT,ABCD,90.0,80)
      CALL NFRAME
C
C *** TO SET POINTERS FOR BLANK COMMON STORAGE ZZZ
C ***   (WITH INTEGER NAMES OF ARRAYS USED IN CALLED SUBROUTINES)
C
      NUMPT = 1
      XPT = NUMPT+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(6,15) NEND
   15 FORMAT(///,20X,*BLANK COMMON STORAGE ZZZ REQUIRES AT LEAST *,I6,
     1* LOCATIONS FOR THIS CASE*///)
C
C *** CALL SUBROUTINES FOR INPUT OF CONTROL VALUES AND GEOMETRY
C
      IF(KGEOM.EQ.1) CALL GEOM1(
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
      IF(KGEOM.EQ.2) CALL GEOM2(
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
      IF(KGEOM.EQ.9) CALL GEOM9(
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
      CALL XYSCAL(
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
      CALL ENCLOS(
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
      CALL PNTOUT(1,
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
  600 CONTINUE
      IF(IDCASE.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 NOTATE(XSTRT,YSTRT,HEIGHT,ABCD,90.0,80)
      CALL NFRAME
  650 CONTINUE
      CALL ZEROW
     1(ZZZ(NUMPT),ZZZ(WPT),ZZZ(NUMEL),ZZZ(CENTR))
      IF(KDATA.EQ.1) CALL DATA1
     1(ZZZ(NUMPT),ZZZ(WPT),ZZZ(NUMEL),ZZZ(CENTR))
      IF(KDATA.EQ.5) CALL DATA5
```

```
     1(ZZZ(NUMPT),ZZZ(WPT),ZZZ(NUMEL),ZZZ(CENTR))
      IF(KDATA.EQ.9) CALL DATA9
     1(ZZZ(NUMPT),ZZZ(WPT),ZZZ(NUMEL),ZZZ(CENTR))
      CALL PNTOUT(2,
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
C
C *** CALL SUBROUTINES FOR PLOTTING GEOMETRIC LAYOUTS
C
  700 CONTINUE
      IF(KPLOT.EQ.4.AND.ILOOP.NE.0) GO TO 6000
      READ(5,PICT)
      WRITE(6,PICT)
 6000 CONTINUE
      IF(KPLOT.EQ.4) CALL CCRT1
      IF(NPLOT.EQ.4.OR.NPLOT.EQ.5) 211,212
  211 CONTINUE
      CALL SCALEW(ZZZ(WPT),ZZZ(CENTR))
      LABEL = 8H WMAG =
      ENCODE(20,333,BCD) LABEL,WMAG
  333 FORMAT(A8,E12.4)
      HEIGHT = 0.15
      XSTRT = XSPACE
      YSTRT = 2.0*HEIGHT
      CALL NOTATE(XSTRT,YSTRT,HEIGHT,BCD,0.0,20)
  212 CONTINUE
      YSPACE = 0.0
      IF(YSPACE.LT.5.0*HEIGHT) YSPACE = 5.0*HEIGHT
      XORGN = -XMINIM/PSCALE+XSPACE
      YORGN = FLOAT(KSIGN)*(-YMINIM)/PSCALE+YSPACE
      CALL CALPLT(XORGN,YORGN,-3)
      IF(NPLOT.LE.3) 201,202
  201 CONTINUE
      CALL LAYOUT(NPLOT,
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
      GO TO 900
  202 CONTINUE
C
C *** FOR CONTOUR PLOTS
C
      KPOINT = 1
      KIND = -2
      ISIZE = 1
      IF(NPLOT.EQ.5) CALL POINTS(KIND,ISIZE,ZZZ(XPT),ZZZ(YPT))
      IF(INFOR.EQ.1) CALL WNODE(
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
      IF(INFOR.EQ.2) CALL WELMT(
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
      CALL BOUND(
     1ZZZ(NUMPT),ZZZ(XPT),ZZZ(YPT),ZZZ(WPT),ZZZ(RADPT),
     2ZZZ(NUMEL),ZZZ(NODE1),ZZZ(NODE2),ZZZ(NODE3),ZZZ(NODE4),ZZZ(CENTR))
  900 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
```

```
      SUBROUTINE DOCMNT
C
C *** THIS SUBROUTINE CONTAINS PROGRAM DOCUMENTATION
C
C
C
C                         DESCRIPTION OF INPUT DATA CARDS
C
C
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,
C                      AND CONTROL VALUES NEEDED BY THE PROGRAM.

C        THE FOLLOWING VALUES ARE INCLUDED---

C        NNDEST = ESTIMATE NUMBER OF GRID POINTS TO BE USED.  VALUE MUST BE
C                 GREATER THAN OR EQUAL TO THE ACTUAL NUMBER OF GRID POINTS.
C             ** DEFAULT = 200 **
C        NELEST = ESTIMATED NUMBER OF ELEMENTS TO BE USED.  VALUE MUST BE
C                 GREATER THAN OR EQUAL TO THE ACTUAL NUMBER OF ELEMENTS.
C             ** DEFAULT = 200 **
C
C        KGEOM SPECIFIES SUBROUTINE AND CORRESPONDING METHOD OF INPUT FOR
C        MODEL GEOMETRY.
C        KGEOM = 1 FOR GRID POINTS AND ELEMENTS READ FROM CARDS WITH USER
C                  SPECIFIED FORMAT.
C              = 2 FOR NASTRAN DECK WITH CARD IDENTIFIERS LEFT ADJUSTED
C                  AND DATA IN COLUMN WIDTHS OF 8.
C              = 9 FOR USER SUPPLIED SUBROUTINE - GEOM9.
C             ** DEFAULT = 1 **
C        KDATA SPECIFIES SUBROUTINE AND CORRESPONDING METHOD OF INPUT FOR
C        DATA TO BE REPRESENTED BY CONTOUR LINES.
C        KDATA = 1 FOR SUBROUTINE DATA1 TO READ IN DATA TO BE PLOTTED
C                  FROM CARDS WITH USER SPECIFIED FORMAT.
C              = 5 FOR SUBROUTINE DATA5 TO READ IN DATA TO BE PLOTTED
C                  FROM TAPE20.
C              = 9 FOR SUBROUTINE DATA9, A USER SUPPLIED SUBROUTINE.
C             ** DEFAULT = 1 **
C        NVALUS - USED IF KDATA = 5 TO SPECIFY THE NUMBER OF SETS OF
C        DATA TO BE PLOTTED WHICH ARE READ FROM TAPE20.
C             ** DEFAULT = 0 **
C        IRESEQ = 0 FOR NO RESEQUENCING OF GRID POINT NUMBERS.
C               = 1 TO RESEQUENCE GRID POINT NUMBERS IN SAME ORDER
C                   AS THEY ARE INPUT.
C             ** DEFAULT = 1 **
C        KPLOT SPECIFIES THE TYPE OF OUTPUT DEVICE TO BE USED.
C        KPLOT = 1 FOR CALCOMP.
C              = 2 FOR CALCOMP WITH PLOTTING SPEED REDUCED TO USE LEROY PENS.
C              = 3 FOR VARIAN.
C              = 4 FOR CRT (USE CDC250 SCOPES AT LRC)
C             ** DEFAULT = 1 **
C        INFOR = 1 IF DATA TO BE PLOTTED IS SPECIFIED AT THE GRID POINTS.
C              = 2 IF DATA TO BE PLOTTED IS SPECIFIED AT THE ELEMENT CENTROIDS.
C             ** DEFAULT = 1 **
C        XSPACE = SPACE BETWEEN PLOTS IN X-DIRECTION, IN INCHES.
C             ** DEFAULT = 10.0 **
```

```
C        KSIGN = -1 TO CHANGE SIGN OF Y ORDINATES.
C              = 1 DO NOT CHANGE SIGN OF Y COORDINATES.
C             ** DEFAULT = 1 **
C        ICCASE = 0 FOR NO TITLE CARD PRECEDING
C                   DECK OF DATA TO BE PLOTTED.
C               = 1 FOR 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 KGEOM SPECIFIED IN NAMELIST OPTION.


C     USE IF KGEOM = 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 ENDGRID 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 ENDGEOM IN COLUMNS 1-7.


C     USE IF KGEOM = 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 ENDGEOM IN COLUMNS 1-7.
```

```
C      USE IF KGEOM = 9

C         CALL SUBROUTINE GEOM9 WHICH IS PREPARED BY THE USER TO READ
C         GEOMETRY DATA.


C    CASE IDENTIFICATION CARD.

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


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


C      USE IF KDATA = 1


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


C      USE IF KDATA = 5

C         READS NVALUS (FROM NAMELIST OPTION) SETS OF CONTROL VARIABLE
C         VALUES FROM TAPE20.  EACH SET OF CONTROL VARIABLES MUST HAVE
C         BEEN WRITTEN ON TAPE20 AS AN UNFORMATTED RECORD.


C      USE IF KDATA = 9

C         CALL SUBROUTINE DATA9 WHICH IS PREPARED BY THE USER TO READ
C         DATA TO BE PLOTTED.


C     NAMELIST PICT - CONTAINS VALUES NEEDED TO GENERATE PLOTS.

C         THE FOLLOWING VALUES ARE INCLUDED---
```

```
C          NPLOT SPECIFIES TYPE OF PLOT TO BE GENERATED.
C          NPLOT = 1 IF ELEMENT LAYOUT WITHOUT LABELS IS DESIRED.
C                = 2 IF ELEMENT LAYOUT WITH GRID PT. LABELS IS DESIRED.
C                = 3 IF ELEMENT LAYOUT WITH ELEMENT LABELS IS DESIRED.
C                = 4 IF CONTOUR PLOTS ARE DESIRED WITHOUT SYMBOLS AT GRID POINTS.
C                = 5 IF CONTOUR PLOTS ARE DESIRED WITH SYMBOLS AT GRID POINTS.
C               ** DEFAULT = 4 **
C          XORGN = X-LOCATION OF ORIGIN OF FIRST PLOT, IN INCHES.
C               ** DEFAULT = 0.0 **
C          YORGN = Y-LOCATION OF ORIGIN OF FIRST PLOT, IN INCHES.
C               ** DEFAULT = 0.0 **
C          PSCALE = MODEL SIZE REDUCTION FACTOR, PSCALE = ACTUAL MODEL
C                   SIZE/DESIRED PLOT SIZE.
C               ** DEFAULT = 1.0 **
C          ISCALE = METHOD OF SCALING CONTROL VARIABLE DATA TO BE PLOTTED.
C                   CONTOUR LINES HAVE ONLY INTEGER VALUES ANNOTATED ON
C                   THE PLOT AND, THUS, THE DATA MUST BE SCALED SUCH THAT
C                   THESE INTEGERS WILL CONTAIN THE DESIRED NUMBER OF
C                   SIGNIFICANT DIGITS.  THE DEFINITION OF WMAGS AND
C                   ICNTRS DEPENDS ON THE VALUE OF ISCALE.
C          ISCALE = 1, FOR USER SPECIFICATION OF SCALE FACTORS.
C                 = 2, FOR PROGRAM CALCULATION OF SCALE FACTORS TO GIVE
C                   THE USER SPECIFIED NUMBER OF SIGNIFICANT DIGITS IN
C                   ANNOTATION OF THE MAXIMUM ABSOLUTE CONTOUR LINE.
C                 = 3, FOR PROGRAM CALCULATION OF SCALE FACTORS TO GIVE
C                   WMAGS AS THE MAXIMUM VALUE OF DATA.
C               ** DEFAULT = 3 **
C          WMAGS - (FOR ISCALE = 1), MAGNIFICATION OF CONTROL VARIABLES
C                 FOR ANNOTATION OF CONTOUR LINES ON PLOT.
C                 (FOR ISCALE = 2), NUMBER OF SIGNIFICANT DIGITS IN
C                 ANNOTATION OF MAXIMUM ABSOLUTE CONTOUR LINE.
C                 (WMAGS = 1.0, 2.0, 3.0, ETC.)
C                 (FOR ISCALE = 3), MAXIMUM VALUE OF SCALED DATA,
C                 WMAGS MUST BE LESS THAN OR EQUAL TO 2 (MAXIMUM CONTOUR
C                 LINE IS INTEGER TRUNCATION OF WMAGS).
C               ** DEFAULT = 100.0 **
C          ICNTRS - (FOR ISCALE = 1), USER SPECIFIED CONTOUR INTERVAL
C                   (DIFFERENCE IN INTEGER VALUES OF ADJACENT CONTOUR
C                   LINES).
C                   (FOR ISCALE = 2 OR 3), APPROXIMATE NUMBER OF DIFFERENT
C                   CONTOUR LINE VALUES, THE CONTOUR INTERVAL IS
C                   CALCULATED BY THE PROGRAM.
C               ** DEFAULT = 1.0 **
C          XLHT = HEIGHT OF INTEGERS TO BE ANNOTATED ON PLOTS, IN INCHES.
C               ** DEFAULT = 0.15 **
C          NXLAB = TOTAL NUMBER OF LINES PARALLEL TO Y-AXIS ALONG WHICH
C                  CONTOUR LINES ARE TO BE LABELED, MUST BE .LE. 10, (ALL
C                  CONTOUR LINES ARE LABELED WHERE THEY INTERSECT WITH
C                  THESE SELECTED LINES).  THESE LABELS ARE IN ADDITION
C                  TO THOSE AUTOMATICALLY PROVIDED AT BOUNDARIES OF THE
C                  CONTOUR SURFACE.
C               ** DEFAULT = 0 **
C          XLAB = ARRAY OF DISTANCES IN X-DIRECTION FROM THE ORIGIN TO
C                 LINES PARALLEL TO THE Y-AXIS ALONG WHICH CONTOUR LINES
C                 ARE LABELED.  THERE MUST BE NXLAB OF THESE VALUES AND
C                 THEY MUST BE IN UNITS OF THE ORIGINAL (UNSCALED) MODEL.
C               ** DEFAULT = ALL ZEROS **
C          NYLAB = SAME AS NXLAB FOR LABEL LOCATIONS PARALLEL TO X-AXIS.
C               ** DEFAULT = 0 **
```

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


C    THE ABOVE COMPRISES 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
C
C*******************************************************************************
C
C
C                        DESCRIPTION OF GRAPHICS SUBROUTINES
C
C
C
C    GRAPHICS          USED BY FOLLOWING        CALLED BY FOLLOWING
C    SUBROUTINE        PLOTTING DEVICE          PROGRAM SUBROUTINES
C
C
C    CALCOMP           CALCOMP                  INITAL
C
C    LEROY             CALCOMP                  INITAL
C
C    PSEUDO            VARIAN                   INITAL
C
C    CDC250            CRT                      INITAL
C
C    CALPLT            CALCOMP,VARIAN,CRT       MAIN,CCRT3,PSTOP,TRIN,BOUND,LAYOUT
C
C    PNTPLT            CALCOMP,VARIAN,CRT       POINTS
C
C    NOTATE            CALCOMP,VARIAN,CRT       MAIN
C
C    NUMBER            CALCOMP,VARIAN,CRT       TRIN,BOUND,LAYOUT
C
C    NFRAME            VARIAN                   MAIN,LAYOUT
C
C    NEXT              CRT                      MAIN,CCRT1,CCRT2,CCRT3
C
C    MESAGE            CRT                      MAIN,CCRT1,CCRT2,CCRT3
C
C    PARAMS            CRT                      CCRT1
C
C    KFORMAT           CRT                      CCRT1
C
```

```
C
C
C                              SUBROUTINE CALCOMP
C
C
C      PURPOSE          THIS IS THE NORMAL MODE PROCESSOR.  THE NECESSARY
C                       PARAMETERS AND LINKAGE ARE SET UP TO OUTPUT A TAPE
C                       FOR THE CALCOMP 780/763 0.010/0.005-INCH STEP PLOTTER.
C
C      USE              CALL CALCOMP
C
C      COMMENTS         THIS CALL MUST BE GIVEN BEFORE THE FIRST CALL TO A
C                       PLOTTING ROUTINE.
C
C
C                              SUBROUTINE LEROY
C
C
C      PURPOSE          THE PARAMETERS NECESSARY TO ACCOMODATE PLOTTING WITH THE
C                       LIQUID INK PEN ARE SET UP BY CALL LEROY.
C
C      USE              CALL LEROY
C
C      COMMENTS         THIS CALL SHOULD ONLY BE USED WITH THE CALCOMP PROCESSOR.
C                       IN ADDITION TO REDUCING THE SPEED OF THE PLOTTER FOR
C                       ALL PLOTTING MOVEMENTS, THE NUMBER OF PLOT VECTORS IN ANY
C                       ANNOTATION IS CONSIDERABLY INCREASED.
C
C
C                              SUBROUTINE PSEUDO
C
C
C      PURPOSE          INITIALIZES PLOT VECTOR FILE FOR VARIAN PLOTTER
C
C      USE              CALL PSEUDO
C
C
C                              SUBROUTINE CDC250
C
C
C      PURPOSE          INITIALIZES CATHODE RAY TUBE CONSOLE.
C
C      USE              CALL CDC250
```

```
C                             SUBROUTINE CALPLT
C
C
C     PURPOSE          TO MOVE THE PLOTTER PEN TO A NEW LOCATION WITH PEN UP
C                      OF DOWN AND TO SIGNAL THE END OF A JOB SEGMENT BY
C                      INCREMENTING THE BLOCK ADDRESS NUMBER.
C
C     USE              CALL CALPLT(X,Y,IPEN)
C
C                      WHERE
C
C                      X,Y       ARE THE FLOATING POINT VALUES FOR PEN MOVEMENT.
C
C                      IPEN = 2 PEN DOWN
C
C                           = 3 PEN UP
C
C                                NEGATIVE IPEN WILL ASSIGN X=0, Y=0
C                                AS THE LOCATION OF THE PEN AFTER MOVING THE
C                                X,Y (CREATE A NEW REFERENCE POINT) AND
C                                INCREASE THE BLOCK NUMBER BY ONE.
C
C     COMMENTS         ALL X AND Y COORDINATES MUST BE EXPRESSED AS FLOATING
C                      POINT INCHES (ACTUAL PAGE DIMENSIONS) IN DEFLECTION FROM
C                      THE ORIGIN.
C
C
C                            SUBROUTINE PNTPLT
C
C
C     PURPOSE          TO DRAW NASA STANDARD PLOT SYMBOLS CENTERED ON A GIVEN
C                      COORDINATE VALUE.
C
C     USE              CALL PNTPLT(A,B,NO,IS)
C
C                      WHERE
C
C                      A         IS THE X COORDINATE FOR THE CENTERED SYMBOL
C                                IN FLOATING POINT INCHES.
C
C                      B         IS THE Y COORDINATE FOR THE CENTERED SYMBOL
C                                IN FLOATING POINT INCHES.
C
C                      NO        IS AN INTEGER SPECIFYING THE SYMBOL TO BE USED,
C
C                      IS        IS AN INTEGER VALUE SPECIFYING THE SIZE SYMBOL
C                                TO BE USED.
C
```

```
C
C                          SUBROUTINE NOTATE
C
C
C      PURPOSE        TO DRAW ALPHANUMERIC INFORMATION FOR ANNOTATION AND LABELIN
C                     AND PROVIDE SPECIAL CENTERED SYMBOLS FOR ANNOTATION OF
C                     DATA POINTS.
C
C      USE            CALL NOTATE(X,Y,HEIGHT,BCD,THETA,N)
C
C                     WHERE
C
C                     X,Y       ARE THE FLOATING POINT PAGE COORDINATES OF THE
C                               FIRST CHARACTER.  FOR ALPHANUMERIC CHARACTERS
C                               THE COORDINATES OF THE LOWER LEFT-HAND CORNER
C                               OF THE CHARACTERS ARE SPECIFIED.
C
C                     HEIGHT    SPECIFIES CHARACTER SIZE AND SPACING IN FLOATING
C                               POINT INCHES FOR A FULL-SIZE CHARACTER.  THE
C                               WIDTH OF A CHARACTER WILL BE (4/7)*HEIGHT AND THE
C                               SPACE BETWEEN CHARACTERS IS (2/7)*HEIGHT.
C
C                     BCD       IS THE STRING OF ALPHANUMERIC CHARACTERS TO BE
C                               DRAWN.
C
C                     THETA     IS THE ANGLE IN FLOATING POINT DEGREES AT WHICH
C                               THE INFORMATION IS TO BE DRAWN.
C
C                     N         IS THE NUMBER OF CHARACTERS, INCLUDING BLANKS, IN
C                               THE LABEL.
C
```

```
C
C                            SUBROUTINE NUMBER
C
C
C     PURPOSE          TO CONVERT A FLOATING NUMBER TO BCD (EXPRESSED IN
C                      F FORMAT), AND DRAW THE RESULTING APLHANUMERIC CHARACTERS.
C
C     USE              CALL NUMBER(X,Y,SIZE,FPN,THETA,N)
C
C                      WHERE
C
C                      X,Y        ARE THE COORDINATES IN FLOATING POINT INCHES
C                                 OF THE LEFT LOWER CORNER OF THE FIRST DIGIT
C                                 OF OUTPUT.
C
C                      SIZE       IS THE HEIGHT OF THE PLOTTED NUMBER IN FLOATING
C                                 POINT INCHES.
C                      FPN        IS THE FLOATING POINT NUMBER TO BE DRAWN.
C
C                      THETA      IS THE ANGLE IN FLOATING POINT DEGREES AT WHICH
C                                 THE NUMBER IS TO BE DRAWN.
C
C                      N          IS THE NUMBER OF DECIMAL DIGITS TO THE RIGHT
C                                 OF THE DECIMAL POINT FOR OUTPUT.
C                                 N = -1 AND N = 0 BOTH SPECIFY NO DECIMAL PLACES,
C                                 HOWEVER, -1 SUPPRESSES THE DECIMAL POINT.
C
C     COMMENTS         THE NUMBER IS RESTRICTED TO A MAXIMUM OF 12 DIGITS.
C                      THE ROUTINE TRUNCATES THE FLOATING POINT NUMBER AT THE
C                      REQUIRED DECIMAL PLACE.
C
C
C                            SUBROUTINE NFRAME
C
C
C     PURPOSE          USED BY VARIAN PLOTTER TO ADVANCE PLOTTING FRAME.
C
C     USE              CALL NFRAME
C
C
C                            SUBROUTINE NEXT
C
C
C     PURPOSE          PROVIDES A BREAK POINT OR HALT DURING APPLICATION PROGRAM
C                      EXECUTION.  OPERATOR MUST PRESS FUNCTION KEY TO RESUME,
C                      AND NUMBER OF KEY IS RETURNED IN CALLING PARAMETER.
C
C     USE              CALL NEXT(NKEY)
C
C                      WHERE
C
C                      NKEY IS NUMBER OF FUNCTION KEY PRESSED.
C
```

```
C
C                             SUBROUTINE MESAGE
C
C
C     PURPOSE         PROVIDES THE CAPABILITY TO DISPLAY A MESSAGE ON THE CRT250.
C
C     USE             CALL MESAGE(I,BCD,N)
C
C                     WHERE
C
C                     I INDICATES INTENSITY OF CHARACTER DISPLAY.
C
C                     BCD IS ADDRESS OF ARRAY CONTAINING THE MESSAGE
C                     IN HOLLERITH FORM.
C
C                     N IS THE NUMBER OF CHARACTERS IN THE MESSAGE (LESS THAN 50)
C
C
C                             SUBROUTINE PARAMS
C
C
C     PURPOSE         USED TO GENERATE A TABLE OF SYMBOLIC NAMES THAT CAN BE
C                     ACCESSED USING THE ALPHANUMERIC KEYBOARD ON THE CRT250.
C
C     USE             CALL PARAMS(BCD,VAR)
C
C                     WHERE
C
C                     BCD IS THE HOLLERITH REPRESENTATION OF SYMBOLIC NAME.
C
C                     VAR IS PROGRAM VARIABLE REFERRED TO BE SYMBOLIC NAME.
C
C     COMMENTS        UP TO 3 PAIRS OF VARIABLES MAY BE SPECIFIED IN A SINGLE
C                     CALL TO PARAMS.  TABLE HAS CAPACITY FOR 42 PAIRS.
C
C
C                             SUBROUTINE KFORMAT
C
C
C     PURPOSE         ALLOWS PROGRAMMER TO CHANGE FORMAT FOR KEYBOARD INPUT.
C
C     USE             CALL KFORMAT(NHBCD)
C
C                     WHERE
C
C                     N IS THE NUMBER OF CHARACTERS IN BCD.
C
C                     H IS REQUIRED.
C
C                     BCD IS THE REQUIRED FORMAT (I4,F4.2,A10,ETC.)
C
C
      RETURN
      END OF DOCMNT
```

```
      SUBROUTINE CCRT1
C
C *** FOR CHANGING VALUES INPUT BY $PICT USING CRT.
C
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      COMMON/SAVEV/ WMAGS,ICNTRS
      COMMON/GKODE/ KODE,NPLOT
      COMMON/SIZEXY/ XMINIM,XMAXIM,YMINIM,YMAXIM,XSPACE,YSPACE
C
C *** FOR INTEGER CONTROL VALUES
C
      CALL KFORMAT(3HI10)
      CALL PARAMS
      CALL PARAMS(5LNPLOT,NPLOT,6LISCALE,ISCALE,6LICNTRS,ICNTRS)
      CALL PARAMS(5LNXLAB,NXLAB,5LNYLAB,NYLAB)
      CALL MESAGE(1,32HTO CHANGE INTEGER CONTROL VALUES,32)
      CALL MESAGE(1,13HVARIABLES ARE,13)
      CALL MESAGE(1,21HNPLOT, ISCALE, ICNTRS,21)
      CALL MESAGE(1,12HNXLAB, NYLAB,12)
      CALL MESAGE(1,17HANY KEY CONTINUES,17)
      CALL NEXT(NKEY)
C
C *** FOR FLOATING POINT CONTROL VALUES
C
      CALL KFORMAT(5HF10.3)
      CALL PARAMS
      CALL PARAMS(5LXORGN,XORGN,5LYORGN,YORGN,6LPSCALE,PSCALE)
      CALL PARAMS(5LWMAGS,WMAGS,4LXLHT,XLHT)
      CALL PARAMS(5LXLAB1,XLAB(1),5LYLAB1,YLAB(1))
      CALL MESAGE(1,39HTO CHANGE FLOATING POINT CONTROL VALUES,39)
      CALL MESAGE(1,13HVARIABLES ARE,13)
      CALL MESAGE(1,20HXORGN, YORGN, PSCALE,20)
      CALL MESAGE(1,11HWMAGS, XLHT,11)
      CALL MESAGE(1,12HXLAB1, YLAB1,12)
      CALL MESAGE(1,17HANY KEY CONTINUES,17)
      CALL NEXT(NKEY)
      RETURN
      END OF CCRT1


      SUBROUTINE CCRT2
C
C *** FOR SELECTING CONTROL OPTION, KODE, AT END OF JOB USING CRT.
C
      COMMON/GKODE/ KODE,NPLOT
      CALL MESAGE(1,18HFN KEY 34 ENDS JOB,18)
      CALL MESAGE(1,32HFN KEY 35 TO ALTER EXISTING PLOT,32)
      CALL MESAGE(1,42HFN KEY 36 TO READ NEW SET OF DISPLACEMENTS,42)
      CALL MESAGE(1,37HFN KEY 37 TO READ A COMPLETE NEW CASE,37)
   10 CONTINUE
      CALL NEXT(NKEY)
      IF(NKEY.LT.34.OR.NKEY.GT.37) GO TO 10
      IF(NKEY.EQ.34) KODE = 0
      IF(NKEY.EQ.35) KODE = 1
      IF(NKEY.EQ.36) KODE = 2
      IF(NKEY.EQ.37) KODE = 3
      RETURN
      END OF CCRT2
```

```
      SUBROUTINE CCRT3
C
C *** REMINDER TO PUT EOF ON PLOT FILE WHEN USING CRT.
C
      CALL MESAGE(1,1H ,1)
      CALL MESAGE(1,45HLAST REMINDER TO PUT EOF ON PLOT FILE, IF ANY,45)
      CALL MESAGE(1,1H ,1)
      CALL MESAGE(1,40HDO IT AT NEXT PLOT FILE COMPLETE MESSAGE,40)
      CALL MESAGE(1,1H ,1)
      CALL MESAGE(1,17HANY KEY CONTINUES,17)
      CALL NEXT(NKEY)
      CALL CALPLT(12.0,0.0,-3)
      RETURN
      END OF CCRT3


      SUBROUTINE PSTOP
C
C *** TO TERMINATE JOB
C
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,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
```

```
      SUBROUTINE INITAL
C
C *** TO SET VALUES FOR CONTROL PARAMETERS
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      COMMON/GRODE/ KODE,NPLOT
      COMMON/CASEID/ IDCASE
      COMMON/SEQNCE/ IRESEQ
      COMMON/VALUES/ NVALUS
      COMMON/SAVEV/ WMAGS,ICNTRS
      COMMON/SIZEXY/ XMINIM,XMAXIM,YMINIM,YMAXIM,XSPACE,YSPACE
      NAMELIST/OPTION/ NNDEST,NELEST,KGEOM,KDATA,NVALUS,IRESEQ,KPLOT,
     1INFOR,XSPACE,KSIGN,IDCASE
C
C *** DESCRIPTION OF VALUES IN $OPTION GIVEN IN SUBROUTINE DOCMNT
C
C
C *** TO SET DEFAULT VALUES FOR $OPTION
C
      NNDEST = 200
      NELEST = 200
      KGEOM = 1
      KDATA = 1
      NVALUS = 0
      IRESEQ = 1
      KPLOT = 1
      INFOR = 1
      XSPACE = 10.0
      KSIGN = 1
      IDCASE = 0
C
C *** TO SET DEFAULT VALUES FOR $PICT
C
      NPLOT = 4
      XORGN = 0.0
      YORGN = 0.0
      PSCALE = 1.0
      ISCALE = 3
      WMAGS = 100.0
      ICNTRS = 10
      XLHT = 0.15
      NXLAB = 0
      NYLAB = 0
      DO 10 I=1,10
      XLAB(I) = 0.0
      YLAB(I) = 0.0
   10 CONTINUE
      KODE = 0
      READ(5,OPTION)
      IF(KPLOT.LE.2) CALL CALCOMP
      IF(KPLOT.EQ.2) CALL LERCY
      IF(KPLOT.EQ.3) CALL PSEUDO
      IF(KPLOT.EQ.4) CALL CDC250
      WRITE(6,OPTION)
      RETURN
      END OF INITAL
```

```
      SUBROUTINE GEOM1(NUMPT,XPT,YPT,WPT,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** TO READ GEOMETRY DATA FROM CARDS HAVING A GENERAL FORMAT
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
      DIMENSION ABCD(8)
      DIMENSION FORMT(7)
      TEST1 = 10HENDGRID
      TEST2 = 10HENDGELM
      TEST3 = 10HFORMAT
C
C *** TO READ GRID INFORMATION
C
  100 CONTINUE
      READ(5,10) ABCD
   10 FORMAT(8A10)
      IF(ABCD(1).EQ.TEST1) GO TO 1000
      IF(ABCD(1).EQ.TEST3) 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      SORRY, THE ESTIMATE OF NUMBER OF NODE
     1 POINTS WAS EXCEEDED      XXXXX*)
      STOP
  551 CONTINUE
      DECODE(80,FORMT,ABCD) NUMPT(NNODE),XPT(NNODE),YPT(NNODE)
      GO TO 100
 1000 CONTINUE
C
C *** TO READ ELEMENT CONNECTION INFORMATION
C
  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(6,556)
  556 FORMAT(1X,///,10X,*XXXXX      SORRY, THE ESTIMATE OF NUMBER OF ELEM
     1ENTS WAS EXCEEDED      XXXXX*)
      STOP
```

```
 651 CONTINUE
     DECODE(80,FORMT,ABCD) NUMEL(NELMT),NODE1(NELMT),NODE2(NELMT),
    1NODE3(NELMT),NODE4(NELMT)
     GO TO 200
2000 CONTINUE
     RETURN
     END OF GEOM1
```

```
      SUBROUTINE GEOM2(NUMPT,XPT,YPT,WPT,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** TO READ NASTRAN GEOMETRY DATA
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
      DIMENSION ABCD(8)
      DIMENSION NDPT(4)
      DIMENSION TRIAEL(9),QUADEL(9)
   10 FORMAT(8A10)
      TEST0 = 10H
      TEST1 = 10HGRID
      TEST3 = 10HTRIAEL
      TEST4 = 10HQUADEL
      TEST5 = 10HENDGEOM
      DO 15 I=1,9
      TRIAEL(I) = 10H
      QUADEL(I) = 10H
   15 CONTINUE
  800 CONTINUE
      READ(5,10) ABCD
      IF(EOF,5) 1111,750
  750 CONTINUE
      DECODE(80,50,ABCD) WORD1
   50 FORMAT(A8)
      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 800
   63 CONTINUE
      IF(WORD1.EQ.TEST4) 64,65
   64 DECODE(80,101,ABCD) (QUADEL(I),I=1,9)
      GO TO 800
   65 CONTINUE
      DO 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
C *** TO READ GRID CARDS
C
C
  200 CONTINUE
      DECODE(80,201,ABCD) IDUM,XDUM,YDUM
  201 FORMAT(8X,A8,8X,2F8.0)
      CALL IRITE(IDUM)
      NNODE = NNODE+1
      IF(NNODE.GT.NNDEST) 550,551
```

```
  550 CONTINUE
      WRITE(6,555)
  555 FORMAT(1X,///,10X,*XXXXX      SORRY, THE ESTIMATE OF NUMBER OF NODE
     1 POINTS WAS EXCEEDED       XXXXX*)
      STOP
  551 CONTINUE
      NUMPT(NNODE) = IDUM
      XPT(NNODE) = XDUM
      YPT(NNODE) = YDUM
      GO TO 800
C
C *** TO READ CARDS CONTAINING ELEMENTS WITH 3 GRID POINTS
C
  300 CONTINUE
      DECODE(80,301,ABCD) 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) 650,651
  650 CONTINUE
      WRITE(6,556)
  556 FORMAT(1X,///,10X,*XXXXX      SORRY, THE ESTIMATE OF NUMBER OF ELEM
     1ENTS WAS EXCEEDED       XXXXX*)
      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
      DECODE(80,401,ABCD) 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) 761,751
  761 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
```

```
 1111 CONTINUE
 2000 CONTINUE
      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
      DECODE(8,1,NUM) NSAVE
    1 FORMAT(I8)
      DECODE(8,2,NUM) (N(I),I=1,8)
    2 FORMAT(8A1)
      DO 10 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,WPT,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** USER SUPPLIED GEOMETRY INPUT ROUTINE
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
C
C *** INSERT ROUTINE HERE
C
      RETURN
      END OF GEOM9
```

```
      SUBROUTINE ZEROW(NUMPT,WPT,NUMEL,CENTR)
C
C *** TO ZERO OUT VALUES OF DATA TO BE PLOTTED.
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),WPT(1),NUMEL(1),CENTR(1)
      GO TO (100,200) INFOR
  100 CONTINUE
      DO 110 I=1,NNDEST
      WPT(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 DATA1(NUMPT,WPT,NUMEL,CENTR)
C
C *** TO READ THE DATA TO BE PLOTTED
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      COMMON/SEQNCE/ IRESEQ
      DIMENSION NUMPT(1),WPT(1),NUMEL(1),CENTR(1)
      DIMENSION ABCD(8)
      DIMENSION FORMT(7)
      DIMENSION ISAV(10),DSAV(10)
   10 FORMAT(8A10)
      TEST = 6HFORMAT
      READ(5,10) ABCD
      DECODE(80,45,ABCD) WORD,KVALU
   45 FORMAT(A6,1X,I1)
      IF(KVALU.EQ.0) KVALU = 1
      IF(WORD.EQ.TEST) 300,200
  200 WRITE(6,20)
   20 FORMAT(1H1,///,20X,* SORRY, FORMAT FOR DATA NOT GIVEN*)
      STOP
  300 CONTINUE
      DECODE(80,50,ABCD) FORMT
   50 FORMAT(10X,7A10)
  100 CONTINUE
      READ(5,FORMT) (ISAV(K),DSAV(K),K=1,KVALU)
      GO TO (510,610) INFOR
C
C *** FOR DATA AT GRID POINTS
C
  510 CONTINUE
      DO 560 K=1,KVALU
      IDUM = ISAV(K)
      IF(IDUM.EQ.0) GO TO 1000
      WDUM = DSAV(K)
      IF(IRESEQ.EQ.1) GO TO 520
      WPT(IDUM) = WDUM
      GO TO 560
  520 CONTINUE
      DO 500 J=1,NNODE
      IF(NUMPT(J).EQ.IDUM) 501,500
  501 WPT(J) = WDUM
      GO TO 550
  500 CONTINUE
  550 CONTINUE
  560 CONTINUE
      GO TO 100
C
C *** FOR DATA AT ELEMENT CENTROIDS
C
  610 CONTINUE
      DO 660 K=1,KVALU
      IDUM = ISAV(K)
      IF(IDUM.EQ.0) GO TO 1000
      WDUM = DSAV(K)
      IF(IRESEQ.EQ.1) GO TO 620
      CENTR(IDUM) = WDUM
      GO TO 660
```

```
  620 CONTINUE
      DO 600 J=1,NELMT
      IF(NUMEL(J).EQ.IDUM) 601,600
  601 CENTR(J) = WDUM
      GO TO 650
  600 CONTINUE
  650 CONTINUE
  660 CONTINUE
      GO TO 100
 1000 CONTINUE
      RETURN
      END OF DATA1


      SUBROUTINE DATA5(NUMPT,WPT,NUMEL,CENTR)
C
C *** TO READ DISPLACEMENT DATA FROM TAPE20
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      COMMON/VALUES/ NVALUS
      COMMON/SEQNCE/ IRESEQ
      DIMENSION NUMPT(1),WPT(1),NUMEL(1),CENTR(1)
      DO 10 I=1,NVALUS
      READ(20) IDUM,WDUM
      GO TO (510,610) INFOR
C
C *** FOR DATA AT GRID POINTS
C
  510 CONTINUE
      IF(IRESEQ.EQ.1) GO TO 520
      WPT(IDUM) = WDUM
      GO TO 10
  520 CONTINUE
      DO 500 J=1,NNODE
      IF(NUMPT(J).EQ.IDUM) 501,500
  501 WPT(J) = WDUM
      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) = WDUM
      GO TO 10
  620 CONTINUE
      DO 600 J=1,NELMT
      IF(NUMEL(J).EQ.IDUM) 601,600
  601 CENTR(J) = WDUM
      GO TO 10
  600 CONTINUE
   10 CONTINUE
      RETURN
      END OF DATA5
```

```
      SUBROUTINE DATA9(NUMPT,WPT,NUMEL,CENTR)
C
C *** USER SUPPLIED W DATA INPUT ROUTINE
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),WPT(1),NUMEL(1),CENTR(1)
C
C *** INSERT ROUTINE HERE
C
      RETURN
      END OF DATA9
```

```
      SUBROUTINE PNTOUT(IOUT,NUMPT,XPT,YPT,WPT,RADPT,NUMEL,
     1NODE1,NODE2,NODE3,NODE4,CENTR)
C
C *** FOR PRINTED OUTPUT OF INFORMATION IN BLANK COMMON - ZZZ
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(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*,35X,*BOUNDARY*/
     15X,*GRID POINT*,5X,*GRID POINT*,7X,*INPUT*,10X,*INPUT*,
     28X,*POINT*/
     35X,*NUMBER*,9X,*NUMBER*,13X,*X*,14X,*Y*,10X,*INDICATOR*//)
      DO 30 I = 1,NNODE
      WRITE(6,18) I,NUMPT(I),XPT(I),YPT(I),RADPT(I)
   18 FORMAT(2X,I10,5X,I10,3X,3E15.4)
   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*/
     25X,*NUMBER*,9X,*NUMBER*,13X,*1*,9X,*2*,9X,*3*,9X,*4*//)
      DO 35 I = 1,NELMT
      WRITE(6,22) I,NUMEL(I),NODE1(I),NODE2(I),NODE3(I),NODE4(I)
   22 FORMAT(2X,I10,5X,I10,4X,4I10)
   35 CONTINUE
      RETURN
 2000 CONTINUE
C
C *** FOR OUTPUT OF DATA TO BE PLOTTED
C
      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(5(2X,I8,E15.5))
      RETURN
      END OF PNTOUT
```

```
      SUBROUTINE XYSCAL(NUMPT,XPT,YPT,WPT,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** TO DETERMINE MAXIMUM DIMENSIONS OF THE DATUM SURFACE
C     AND RENUMBER ELEMENT GRID POINTS.
C *** W DATA IS SCALED IN SUBROUTINE SCALEW
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      COMMON/SEQNCE/ IRESEQ
      COMMON/SIZEXY/ XMINIM,XMAXIM,YMINIM,YMAXIM,XSPACE,YSPACE
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
      DIMENSION NDPT(4)
C
C *** TO SCALE DATA AND DETERMINE X MINIMUM AND MAXIMUM
C
      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
C
C *** TO RENUMBER ELEMENT NODES
C
      DO 600 I=1,NELMT
      NDPT(1) = NODE1(I)
      NDPT(2) = NODE2(I)
      NDPT(3) = NODE3(I)
      NDPT(4) = NODE4(I)
      DO 605 J=1,4
      IF(NDPT(J).EQ.0) GO TO 605
      DO 610 K=1,NNODE
      IF(NUMPT(K).EQ.NDPT(J)) 615,610
  615 IF(J.EQ.1) NODE1(I) = K
      IF(J.EQ.2) NODE2(I) = K
      IF(J.EQ.3) NODE3(I) = K
      IF(J.EQ.4) NODE4(I) = K
      GO TO 601
  610 CONTINUE
      WRITE(6,555) NDPT(J),NUMEL(I)
  555 FORMAT(5X,* NODE *,I10,* IN ELEMENT *,I10,* NOT IN USER DATA*)
      STOP
  601 CONTINUE
  605 CONTINUE
  600 CONTINUE
  700 CONTINUE
      RETURN
      END OF XYSCAL
```

```
      SUBROUTINE SCALEW(WPT,CENTR)
C
C *** THIS SUBROUTINE SCALES W DATA
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      COMMON/SAVEV/ WMAGS,ICNTRS
      DIMENSION WPT(1),CENTR(1)
      GO TO (100,200) INFOR
C
C *** FOR NODAL DATA
C
  100 CONTINUE
      WMIN = WPT(1)
      WMAX = WPT(1)
      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(6,121) WMAX,WMIN
  121 FORMAT(1X,///,5X,*MAXIMUM VALUE OF DATA =*,E20.8,10X,*MINIMUM VALU
     1E 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(1X,///,10X,*SORRY, VALUE OF 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(6,122) WMAG,ICNTR
  122 FORMAT(1X,///,5X,*WMAG =*,E20.8,10X,*ICNTR =*,I6)
      GO TO 300
C
C *** FOR ELEMENT DATA
C
  200 CONTINUE
```

```
      CMIN = CENTR(1)
      CMAX = CENTR(1)
      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)
      CRANG = ABS(CMAX-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(CINT,ICNTR)
C
C *** TO CALCULATE CONTOUR INTERVAL, ICNTR
C
      DO 165 I=1,10
      CUP = 1.5*(10.0**I)/10.0
      IF(CINT.LT.CUP) 166,167
  166 ICNTR = 1*(10**I)/10
      GO TO 140
  167 CONTINUE
      CUP = 3.5*(10.0**I)/10.0
      IF(CINT.LT.CUP) 168,169
  168 ICNTR = 2*(10**I)/10
      GO TO 140
  169 CONTINUE
      CUP = 7.5*(10.0**I)/10.0
      IF(CINT.LT.CUP) 170,171
  170 ICNTR = 5*(10**I)/10
      GO TO 140
  171 CONTINUE
  165 CONTINUE
      WRITE(6,175)
  175 FORMAT(1X,///,5X,*SORRY, CONTOUR INTERVAL NOT FOUND, CHECK INPUT*)
      STOP
  140 CONTINUE
      RETURN
      END OF INTRVL
```

```
      SUBROUTINE ENCLOS(NUMPT,XPT,YPT,WPT,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** TO CALCULATE ENCLOSED ANGLES AROUND NODE POINTS
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
      DIMENSION XX(3),YY(3),WW(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 AROUND(XX,YY,RRAD)
      RADPT(ND1) = RADPT(ND1)+RRAD(1)
      RADPT(ND2) = RADPT(ND2)+RRAD(2)
      RADPT(ND3) = RADPT(ND3)+RRAD(3)
      IF(ND4.EQ.0) GO TO 505
      XX(2) = XPT(ND4)
      YY(2) = YPT(ND4)
      CALL AROUND(XX,YY,RRAD)
      RADPT(ND1) = RADPT(ND1)+RRAD(1)
      RADPT(ND4) = RADPT(ND4)+RRAD(2)
      RADPT(ND3) = RADPT(ND3)+RRAD(3)
  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) 601,602
  601 CONTINUE
      RADPT(I) = 1.0
      GO TO 600
  602 CONTINUE
      RADPT(I) = 0.0
  600 CONTINUE
      RETURN
      END OF ENCLOS
```

```
      SUBROUTINE AROUND(X,Y,RAD)
C
C *** DETERMINES A SPECIFIED CORNER ANGLE OF AN ELEMENT
C
      DIMENSION X(3),Y(3),RAD(3)
      OONE = SQRT((X(3)-X(2))**2+(Y(3)-Y(2))**2)
      OTWO = SQRT((X(3)-X(1))**2+(Y(3)-Y(1))**2)
      OTRI = SQRT((X(2)-X(1))**2+(Y(2)-Y(1))**2)
      COS1 = (OTWO**2+OTRI**2-OONE**2)/(2.0*OTWO*OTRI)
      COS2 = (OONE**2+OTRI**2-OTWO**2)/(2.0*OONE*OTRI)
      COS3 = (OONE**2+OTWO**2-OTRI**2)/(2.0*OONE*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/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION XPT(1),YPT(1)
      DO 100 I=1,NNODE
      X = XPT(I)/PSCALE
      Y = FLOAT(KSIGN)*YPT(I)/PSCALE
      CALL PNTPLT(X,Y,KIND,ISIZE)
  100 CONTINUE
      RETURN
      END OF POINTS
```

```
      SUBROUTINE WNODE(NUMPT,XPT,YPT,WPT,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** FOR PLOTTING NODAL DATA
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
      DIMENSION XX(3),YY(3),WW(3),RRAD(3)
      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)
      WW(1) = WPT(ND1)
      XX(2) = XPT(ND2)
      YY(2) = YPT(ND2)
      WW(2) = WPT(ND2)
      XX(3) = XPT(ND3)
      YY(3) = YPT(ND3)
      WW(3) = WPT(ND3)
      CALL TRIN(XX,YY,WW)
      IF(ND4.EQ.0) GO TO 505
      XX(2) = XPT(ND4)
      YY(2) = YPT(ND4)
      WW(2) = WPT(ND4)
      CALL TRIN(XX,YY,WW)
  505 CONTINUE
  500 CONTINUE
      RETURN
      END OF WNODE
```

```
      SUBROUTINE WELMT( NUMPT,XPT,YPT,WPT,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** FOR PLOTTING ELEMENT DATA
C
      COMMON/KOUNT/ NNODE,NELMT,NNODEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(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),WW(3)
      DIMENSION ISAV(4),JSAV(4)
      PI = 3.1415926536
      DO 100 I=1,NNODE
C
C *** TO DETERMINE ELEMENTS CONNECTED TO NODE I
C
      NKEEP = 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.I) 510,505
  510 NKEEP = NKEEP+1
      NEL(NKEEP) = J
  505 CONTINUE
  500 CONTINUE
      IF(NKEEP.EQ.0) GO TO 100
C
C *** CALCULATE INFORMATION FOR CENTROID OF EACH CONNECTED ELEMENT
C
      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
  560 CONTINUE
      X(K) = (XPT(ND1)+XPT(ND2)+XPT(ND3))/3.0
      Y(K) = (YPT(ND1)+YPT(ND2)+YPT(ND3))/3.0
      GO TO 565
  561 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
  565 CONTINUE
      XDIST = X(K)-XPT(I)
      YDIST = Y(K)-YPT(I)
      R(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
```

```
C
C *** TO REORDER INFORMATICN 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)) 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) = ITEMP2
      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
      WPT(I) = 0.0
      DO 615 K=1,NKEEP
      WEIGHT = (1.0/R(K))/DENCM
      IDUM = NEL(K)
      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(1) = XPT(I)
      YY(1) = YPT(I)
      WW(1) = WPT(I)
      DO 620 K=1,NKEEP
      I1 = K
      I2 = K+1
      IF(K.EQ.NKEEP) I2 = 1
      IF(RADPT(I).EQ.0.0) 660,650
C *** TO CHECK CONNECTIVITY OF CENTROIDS ABOUT A BOUNDARY POINT
  650 CONTINUE
      IDUM = NEL(I1)
      JDUM = NEL(I2)
      ISAV(1) = NODE1(ICUM)
      ISAV(2) = NODE2(ICUM)
      ISAV(3) = NODE3(ICUM)
      ISAV(4) = NODE4(IDUM)
      JSAV(1) = NODE1(JDUM)
```

```
      JSAV(2) = NODE2(JDUM)
      JSAV(3) = NODE3(JDUM)
      JSAV(4) = NODE4(JDUM)
      DO 651 III=1,4
      IF(ISAV(III).EQ.0.OR.ISAV(III).EQ.I) GO TO 651
      DO 652 JJJ=1,4
      IF(JSAV(JJJ).EQ.0.OR.JSAV(JJJ).EQ.I) GO TO 652
      IF(ISAV(III).EQ.JSAV(JJJ)) 660,652
  652 CONTINUE
  651 CONTINUE
      GO TO 620
  660 CONTINUE
      XX(2) = X(I1)
      YY(2) = Y(I1)
      IDUM = NEL(I1)
      WW(2) = CENTR(IDUM)
      XX(3) = X(I2)
      YY(3) = Y(I2)
      IDUM = NEL(I2)
      WW(3) = CENTR(IDUM)
      CALL TRIN(XX,YY,WW)
  620 CONTINUE
  625 CONTINUE
  100 CONTINUE
      RETURN
      END OF WELMT
```

```
      SUBROUTINE TRIN(XG,YG,WG)
C
C *** PLOTS CONTOUR LINES WITHIN A GIVEN TRIANGLE
C
      COMMON/CONTRL/  INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,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
C *** TO SCALE W DATA BEFORE PLOTTING
C
      DO 10 I=1,3
      X(I) = XG(I)/PSCALE
      Y(I) = FLOAT(KSIGN)*YG(I)/PSCALE
      W(I) = WG(I)*WMAG
   10 CONTINUE
C
C *** TO FIND MIN AND MAX VALUES OF W
C
      WMIN = W(1)
      WMAX = W(1)
      DO 20 I=2,3
      IF(W(I).LT.WMIN) WMIN = W(I)
      IF(W(I).GT.WMAX) WMAX = W(I)
   20 CONTINUE
C
C *** TO DETERMINE CONSTANT VALUES OF W WITHIN TRIANGLE
C
      IDUM = WMIN/ICNTR
      JDUM = IDUM*ICNTR
      IF(JDUM.GT.0) ISTRT = JDUM+ICNTR
      IF(JDUM.LE.0) ISTRT = JDUM
      IF(FLOAT(JDUM).EQ.WMIN) ISTRT = JDUM
      IDUM = WMAX/ICNTR
      JDUM = IDUM*ICNTR
      IF(JDUM.GE.0) ISTOP = JDUM
      IF(JDUM.LT.0) ISTOP = JDUM-ICNTR
      IF(FLOAT(JDUM).EQ.WMAX) ISTOP = JDUM
      IF(ISTOP.LT.ISTRT) GO TO 1000
C
C *** DETERMINES INTERSECTIONS OF CONSTANT W LINES AND TRIANGLE SIDES
C
      ALLOW = 1.0E-05
      EPSILN = 1.0E-10
      ELOW = 0.0-EPSILN
      EUP = 1.0+EPSILN
      VSAVE = 1.0E+20
      DO 100 I=ISTRT,ISTOP,ICNTR
      WCONST = FLOAT(I)
      ICNT = 0
      DEN = W(1)-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
      XPNT(ICNT) = XLI*X(1)+(1.0-XLI)*X(3)
      YPNT(ICNT) = XLI*Y(1)+(1.0-XLI)*Y(3)
  120 CONTINUE
```

```
      DEN = W(3)-W(2)
      IF(DEN.EQ.0.0) GO TO 140
      XLM = (WCONST-W(2))/DEN
      IF(XLM.GE.ELOW.AND.XLM.LE.EUP) 130,140
  130 ICNT = ICNT+1
      XPNT(ICNT) = (1.0-XLM)*X(2)+XLM*X(3)
      YPNT(ICNT) = (1.0-XLM)*Y(2)+XLM*Y(3)
  140 CONTINUE
      DEN = W(2)-W(1)
      IF(DEN.EQ.0.0) GO TO 160
      XLJ = (WCONST-W(1))/DEN
      IF(XLJ.GE.ELOW.AND.XLJ.LE.EUP) 150,160
  150 ICNT = ICNT+1
      XPNT(ICNT) = (1.0-XLJ)*X(1)+XLJ*X(2)
      YPNT(ICNT) = (1.0-XLJ)*Y(1)+XLJ*Y(2)
  160 CONTINUE
      IF(ICNT.LE.1) GO TO 100
      IF(ICNT.EQ.2) GO TO 180
      IF(ABS(XPNT(1)-XPNT(2)).LT.ALLOW.AND.
     1ABS(YPNT(1)-YPNT(2)).LT.ALLOW) 170,180
  170 CONTINUE
      XPNT(2) = XPNT(3)
      YPNT(2) = YPNT(3)
  180 CONTINUE
      CALL CALPLT(XPNT(1),YPNT(1),3)
      CALL CALPLT(XPNT(2),YPNT(2),2)
      CALL CALPLT(XPNT(2),YPNT(2),3)
      IF(NXLAB.EQ.0) GO TO 300
      IF(ABS(XPNT(1)-XPNT(2)).LT.ALLOW) GO TO 300
      DO 310 K=1,NXLAB
      XLABEL = XLAB(K)/PSCALE
      IF(XPNT(1).LE.XLABEL.AND.XPNT(2).GE.XLABEL) GO TO 350
      IF(XPNT(2).LE.XLABEL.AND.XPNT(1).GE.XLABEL) GO TO 350
      GO TO 310
  350 CONTINUE
      XSTRT = XLABEL
      YSTRT = YPNT(1)+(YPNT(2)-YPNT(1))*(XLABEL-XPNT(1))/
     1(XPNT(2)-XPNT(1))
      YSTRT = YSTRT-XLHT/2.0
      IF(ABS(VSAVE-YSTRT).LT.XLHT) GO TO 310
      VSAVE = YSTRT
      CALL NUMBER(XSTRT,YSTRT,XLHT,WCONST,0.0,-1)
  310 CONTINUE
  300 CONTINUE
      IF(NYLAB.EQ.0) GO TO 400
      IF(ABS(YPNT(1)-YPNT(2)).LT.ALLOW) GO TO 400
      DO 410 K=1,NYLAB
      YLABEL = FLOAT(KSIGN)*YLAB(K)/PSCALE
      IF(YPNT(1).LE.YLABEL.AND.YPNT(2).GE.YLABEL) GO TO 450
      IF(YPNT(2).LE.YLABEL.AND.YPNT(1).GE.YLABEL) GO TO 450
      GO TO 410
  450 CONTINUE
      YSTRT = YLABEL
      XSTRT = XPNT(1)+(XPNT(2)-XPNT(1))*(YLABEL-YPNT(1))/
     1(YPNT(2)-YPNT(1))
      XSTRT = XSTRT+XLHT/2.0
      IF(ABS(VSAVE-XSTRT).LT.XLHT) GO TO 410
      VSAVE = XSTRT
      CALL NUMBER(XSTRT,YSTRT,XLHT,WCONST,90.0,-1)
  410 CONTINUE
```

```
  400 CONTINUE
  100 CONTINUE
 1000 CONTINUE
      DO 500 I=1,3
      W(I) = W(I)/WMAG
  500 CONTINUE
      RETURN
      END OF TRIN
```

```
      SUBROUTINE BOUND(NUMPT,XPT,YPT,WPT,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** PLOTS BOUNDARY OF STRUCTURE AND LABELS CONTOUR LINES
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
      DIMENSION ND(5),NOCON(50,2)
      DIMENSION XX(3),YY(3),WW(3)
C
C *** EXAMINE ALL ELEMENT CONNECTIONS AND ADD 1.0 TO RADPTS WHEN TWO
C *** NODES ARE CONNECTED WHICH HAVE RADPT.NE.0.0.  ALSO FLAG
C *** ELEMENTS CONTAINING SUCH CONNECTIONS BY MAKING NUMEL NEGATIVE.
C
      DO 400 I=1,NELMT
      ND(1) = NODE1(I)
      ND(2) = NODE2(I)
      ND(3) = NODE3(I)
      ND(4) = NODE4(I)
      ND(5) = NODE1(I)
      DO 410 J=1,4
      K = ND(J)
      IF(K.EQ.0) GO TO 400
      L = ND(J+1)
      IF(L.EQ.0) L = ND(J+2)
      IF(RADPT(K).NE.0.0.AND.RADPT(L).NE.0.0) 411,410
  411 CONTINUE
      RADPT(K) = RADPT(K)+1.0
      RADPT(L) = RADPT(L)+1.0
      IF(NUMEL(I).GT.0) NUMEL(I) = -NUMEL(I)
  410 CONTINUE
  400 CONTINUE
C
C *** EXAMINE ALL ELEMENTS AND SAVE ALL CONNECTIONS WITH BOTH
C *** ENDS HAVING RADPT.GE.4.0 IN NOCON.
C
      NPOSS = 0
      DO 420 I=1,NELMT
      IF(NUMEL(I).GT.0) GO TO 420
      ND(1) = NODE1(I)
      ND(2) = NODE2(I)
      ND(3) = NODE3(I)
      ND(4) = NODE4(I)
      ND(5) = NODE1(I)
      DO 430 J=1,4
      K = ND(J)
      IF(K.EQ.0) GO TO 420
      L = ND(J+1)
      IF(L.EQ.0) L = ND(J+2)
      IF(RADPT(K).GE.4.0.AND.RADPT(L).GE.4.0) 431,430
  431 CONTINUE
      IF(NPOSS.EQ.50) GO TO 430
      NPOSS = NPOSS+1
      NOCON(NPOSS,1) = K
      NOCON(NPOSS,2) = L
  430 CONTINUE
```

```
  420 CONTINUE
C
C *** CHECK POSSIBLE NC CONNECTS AND SAVE ANY MATCHED PAIRS
C
      NBAD = 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(NOCON(I,1).EQ.NOCON(J,1).AND.NOCON(I,2).EQ.NOCON(J,2))
     1GO TO 461
      IF(NOCON(I,1).EQ.NOCON(J,2).AND.NOCON(I,2).EQ.NOCON(J,1))
     1 GO TO 461
      GO TO 460
  461 CONTINUE
      NBAD = NBAD+1
      NOCON(NBAD,1) = NOCON(I,1)
      NOCON(NBAD,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) = NODE1(I)
      ND(2) = NODE2(I)
      ND(3) = NODE3(I)
      ND(4) = NODE4(I)
      ND(5) = NODE1(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.3.0.AND.RADPT(L).GE.3.0) 111,50
  111 CONTINUE
      ND1 = K
      ND2 = L
      IF(J.GT.1) IDUM = J-1
      IF(J.EQ.1) IDUM = 3
      NAUX = ND(IDUM)
C
C *** TO PLOT BOUNDARY LINES
C
      IF(NBAD.EQ.0) GO TO 40
      DO 30 JJ = 1,NBAD
      IF(ND1.EQ.NOCON(JJ,1).AND.ND2.EQ.NOCON(JJ,2)) GO TO 50
      IF(ND1.EQ.NOCON(JJ,2).AND.ND2.EQ.NOCON(JJ,1)) GO TO 50
   30 CONTINUE
   40 CONTINUE
      X1 = 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
```

```
C
C *** TO PLOT CONTOUR LINES TO EDGE OF ELEMENT DATA PLOTS
C
      XX(1) = XPT(ND1)
      YY(1) = YPT(ND1)
      WW(1) = WPT(ND1)
      XX(2) = XPT(ND2)
      YY(2) = YPT(ND2)
      WW(2) = WPT(ND2)
      N1 = ND(1)
      N2 = ND(2)
      N3 = ND(3)
      N4 = ND(4)
      IF(N4.EQ.0) 910,920
  910 CONTINUE
      XCENT = (XPT(N1)+XPT(N2)+XPT(N3))/3.0
      YCENT = (YPT(N1)+YPT(N2)+YPT(N3))/3.0
      GO TO 950
  920 CONTINUE
      XCENT = (XPT(N1)+XPT(N2)+XPT(N3)+XPT(N4))/4.0
      YCENT = (YPT(N1)+YPT(N2)+YPT(N3)+YPT(N4))/4.0
  950 CONTINUE
      XX(3) = XCENT
      YY(3) = YCENT
      WW(3) = CENTR(I)
      CALL TRIN(XX,YY,WW)
  889 CONTINUE
C
C *** TO LABEL CONTOUR LINES
C
      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)
      IDUM = WMIN/ICNTR
      JDUM = IDUM*ICNTR
      IF(JDUM.GT.0) ISTRT = JDUM+ICNTR
      IF(JDUM.LE.0) ISTRT = JDUM
      IF(FLOAT(JDUM).EQ.WMIN) ISTRT = JDUM
      IDUM = WMAX/ICNTR
      JDUM = IDUM*ICNTR
      IF(JDUM.GE.0) ISTOP = JDUM
      IF(JDUM.LT.0) ISTOP = JDUM-ICNTR
      IF(FLOAT(JDUM).EQ.WMAX) ISTOP = JDUM
      IF(ISTOP.LT.ISTRT) GO TO 666
      YLINE = 1.0E+20
      XLINE = 1.0E+20
      IF(XPT(ND2)-XPT(ND1).NE.0.0) YLINE = YPT(ND1)+(YPT(ND2)-YPT(ND1))
     1*(XPT(NAUX)-XPT(ND1))/(XPT(ND2)-XPT(ND1))
      IF(YPT(ND2)-YPT(ND1).NE.0.0) XLINE = XPT(ND1)+(XPT(ND2)-XPT(ND1))
     1*(YPT(NAUX)-YPT(ND1))/(YPT(ND2)-YPT(ND1))
      XDIST = XPT(NAUX)-XLINE
      YDIST = YPT(NAUX)-YLINE
      THETA = 0.0
      IF(ABS(YDIST/XDIST).LT.1.0) THETA = 90.0
      XSAVE = 1.0E+20
```

```
      YSAVE = 1.0E+20
      DO 600 JJ = ISTRT,ISTOP,ICNTR
      WCONST = FLOAT(JJ)
      IF(WCONST.LT.WMIN.OR.WCONST.GT.WMAX) GO TO 600
      X = XPT(ND1)+(XPT(ND2)-XPT(ND1))*(WCONST-WPT(ND1))
     1/(WPT(ND2)-WPT(ND1))
      X = X/PSCALE
      Y = YPT(ND1)+(YPT(ND2)-YPT(ND1))*(WCONST-WPT(ND1))
     1/(WPT(ND2)-WPT(ND1))
      Y = FLOAT(KSIGN)*Y/PSCALE
      DO 650 IG = 1,10
      IDUM = 10**IG
      IF(IABS(JJ).LT.IDUM) 655,650
  655 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 680
  670 CONTINUE
      IF(ABS(YSAVE-Y).LE.XLHT) GO TO 600
      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(XSAVE-X).LE.XLHT) GO TO 600
      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)
      XSAVE = X
  690 CONTINUE
      CALL NUMBER(XNUM,YNUM,XLHT,WCONST,THETA,-1)
  600 CONTINUE
  666 CONTINUE
      WPT(ND1) = WPT(ND1)/WMAG
      WPT(ND2) = WPT(ND2)/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,RADPT,NUMEL,NODE1,NODE2,
     1NODE3,NODE4,CENTR)
C
C *** TO PLOT LAYOUT OF ELEMENTS
C
      COMMON/KOUNT/ NNODE,NELMT,NNDEST,NELEST
      COMMON/CONTRL/ INFOR,KGEOM,KDATA,KSIGN,KPLOT,XORGN,YORGN,
     1PSCALE,ISCALE,WMAG,ICNTR,XLHT,NXLAB,XLAB(10),NYLAB,YLAB(10)
      DIMENSION NUMPT(1),XPT(1),YPT(1),WPT(1),RADPT(1),NUMEL(1),
     1NODE1(1),NODE2(1),NODE3(1),NODE4(1),CENTR(1)
C *** TO MAKE ALL GRID POINT NUMBERS NEGATIVE
      DO 490 I=1,NNODE
      NUMPT(I) = -NUMPT(I)
  490 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 CONNECTED BY ELEMENTS POSITIVE
      NUMPT(ND1) = IABS(NUMPT(ND1))
      NUMPT(ND2) = IABS(NUMPT(ND2))
      NUMPT(ND3) = IABS(NUMPT(ND3))
      NUMPT(ND4) = IABS(NUMPT(ND4))
      X1 = XPT(ND1)/PSCALE
      X2 = XPT(ND2)/PSCALE
      X3 = XPT(ND3)/PSCALE
      Y1 = FLOAT(KSIGN)*YPT(ND1)/PSCALE
      Y2 = FLOAT(KSIGN)*YPT(ND2)/PSCALE
      Y3 = FLOAT(KSIGN)*YPT(ND3)/PSCALE
      CALL CALPLT(X1,Y1,3)
      CALL CALPLT(X2,Y2,2)
      CALL CALPLT(X3,Y3,2)
      IF(ND4.EQ.0) 200,300
  200 CONTINUE
      CALL CALPLT(X1,Y1,2)
      CALL CALPLT(X1,Y1,3)
      IF(NPLOT.NE.3) GO TO 250
C *** TO NUMBER TRIANGLES
      XCENT = (X1+X2+X3)/3.0
      YCENT = (Y1+Y2+Y3)/3.0
      X = XCENT-(6.0/7.0)*XLHT
      Y = YCENT-XLHT/2.0
      A = NUMEL(I)
      CALL NUMBER(X,Y,XLHT,A,0.0,-1)
  250 CONTINUE
      GO TO 100
  300 CONTINUE
      X4 = XPT(ND4)/PSCALE
      Y4 = FLOAT(KSIGN)*YPT(ND4)/PSCALE
      CALL CALPLT(X4,Y4,2)
      CALL CALPLT(X1,Y1,2)
      CALL CALPLT(X1,Y1,3)
      IF(NPLOT.NE.3) GO TO 350
C *** TO NUMBER QUADRILATERALS
      XCENT = (X1+X2+X3+X4)/4.0
      YCENT = (Y1+Y2+Y3+Y4)/4.0
      X = XCENT-(6.0/7.0)*XLHT
```

```
      Y = YCENT-XLHT/2.0
      A = NUMEL(I)
      CALL NUMBER(X,Y,XLHT,A,0.0,-1)
  350 CONTINUE
  100 CONTINUE
      IF(NPLOT.NE.2) GO 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
  450 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
 NNDEST = 36, NUDISP = 1, NVDISP = 1, NWDISP = 1,
 KGEOM = 2, KDATA = 1, KPLOT = 3, IDCASE = 1
 $
QUADEL   CQUAD1
ID       GILES,TEST
APP      DISPLACEMENT
SOL      1,0
TIME     60
CEND
TITLE = FLAT PLATE UNDER UNIFORM PRESSURE
OLOAD = 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-91       0.01    0.0     +GARY1
+GARY1   0.005   -0.005
MAT1     1          30.+6             0.3     0.3
SPC1     100      123     10      20      30      40      50      60      +S1
+S1      70       130     190     250     310     320     330     340     +S2
+S2      350      360     120     180     240     300
PLOAD2   100      100.0   6       THRU    10
PLOAD2   100      100.0   16      THRU    20
PLOAD2   100      100.0   26      THRU    30
PLOAD2   100      100.0   36      THRU    40
PLOAD2   100      100.0   46      THRU    50
GRDSET            0                               0       6
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              8.0     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
```

```
GRID     240                  10.0      6.0      0.0
GRID     250                  0.0       8.0      0.0
GRID     260                  2.0       8.0      0.0
GRID     270                  4.0       8.0      0.0
GRID     280                  6.0       8.0      0.0
GRID     290                  8.0       8.0      0.0
GRID     300                  10.0      8.0      0.0
GRID     310                  0.0       10.0     0.0
GRID     320                  2.0       10.0     0.0
GRID     330                  4.0       10.0     0.0
GRID     340                  6.0       10.0     0.0
GRID     350                  8.0       10.0     0.0
GRID     360                  10.0      10.0     0.0
CQUAD1   6        1           10        20       80       70
CQUAD1   7        1           20        30       90       80
CQUAD1   8        1           30        40       100      90
CQUAD1   9        1           40        50       110      100
CQUAD1   10       1           50        60       120      110
CQUAD1   16       1           70        80       140      130
CQUAD1   17       1           80        90       150      140
CQUAD1   18       1           90        100      160      150
CQUAD1   19       1           100       110      170      160
CQUAD1   20       1           110       120      180      170
CQUAD1   26       1           130       140      200      190
CQUAD1   27       1           140       150      210      200
CQUAD1   28       1           150       160      220      210
CQUAD1   29       1           160       170      230      220
CQUAD1   30       1           170       180      240      230
CQUAD1   36       1           190       200      260      250
CQUAD1   37       1           200       210      270      260
CQUAD1   38       1           210       220      280      270
CQUAD1   39       1           220       230      290      280
CQUAD1   40       1           230       240      300      290
CQUAD1   46       1           250       260      320      310
CQUAD1   47       1           260       270      330      320
CQUAD1   48       1           270       280      340      330
CQUAD1   49       1           280       290      350      340
CQUAD1   50       1           290       300      360      350
ENDDATA
ENDGEOM
      Z - DISPLACEMENTS
FORMAT 1   (I10,8X,3E18.6/)
         10       G          0.                  0.                  0.
-CONT-                      -5.741610E+01        5.741610E+01        0.
         20       G          0.                  0.                  0.
-CONT-                       2.831830E+03        4.045395E+01        0.
         30       G          0.                  0.                  0.
-CONT-                       4.462056E+03        2.751137E+01        0.
         40       G          0.                  0.                  0.
-CONT-                       4.462056E+03       -2.751137E+01        0.
         50       G          0.                  0.                  0.
-CONT-                       2.831830E+03       -4.045395E+01        0.
         60       G          0.                  0.                  0.
-CONT-                      -5.741610E+01       -5.741610E+01        0.
         70       G          0.                  0.                  0.
-CONT-                      -4.045395E+01       -2.831830E+03        0.
         80       G          0.                  0.                  5.284250E+03
-CONT-                       2.192913E+03       -2.192913E+03        0.
```

	90	G	0.	0.	8.351675E+03	26
-CONT-			3.472762E+03	-7.800294E+02	0.	27
	100	G	0.	0.	8.351675E+03	28
-CONT-			3.472762E+03	7.800294E+02	0.	29
	110	G	0.	0.	5.284250E+03	30
-CONT-			2.192913E+03	2.192913E+03	0.	31
	120	G	0.	0.	0.	32
-CONT-			-4.045395E+01	2.831830E+03	0.	33
	130	G	0.	0.	0.	34
-CONT-			-2.751137E+01	-4.462056E+03	0.	35
	140	G	0.	0.	8.351675E+03	36
-CONT-			7.800294E+02	-3.472762E+03	0.	37
	150	G	0.	0.	1.324540E+04	38
-CONT-			1.250892E+03	-1.250892E+03	0.	39
	160	G	0.	0.	1.324540E+04	40
-CONT-			1.250892E+03	1.250892E+03	0.	41
	170	G	0.	0.	8.351675E+03	42
-CONT-			7.800294E+02	3.472762E+03	0.	43
	180	G	0.	0.	0.	44
-CONT-			-2.751137E+01	4.462056E+03	0.	45
	190	G	0.	0.	0.	46
-CONT-			2.751137E+01	-4.462056E+03	0.	47
	200	G	0.	0.	8.351675E+03	48
-CONT-			-7.800294E+02	-3.472762E+03	0.	49
	210	G	0.	0.	1.324540E+04	50
-CONT-			-1.250892E+03	-1.250892E+03	0.	51
	220	G	0.	0.	1.324540E+04	52
-CONT-			-1.250892E+03	1.250892E+03	0.	53
	230	G	0.	0.	8.351675E+03	54
-CONT-			-7.800294E+02	3.472762E+03	0.	55
	240	G	0.	0.	0.	56
-CONT-			2.751137E+01	4.462056E+03	0.	57
	250	G	0.	0.	0.	58
-CONT-			4.045395E+01	-2.831830E+03	0.	59
	260	G	0.	0.	5.284250E+03	60
-CONT-			-2.192913E+03	-2.192913E+03	0.	61
	270	G	0.	0.	8.351675E+03	62
-CONT-			-3.472762E+03	-7.800294E+02	0.	63
	280	G	0.	0.	8.351675E+03	64
-CONT-			-3.472762E+03	7.800294E+02	0.	65
	290	G	0.	0.	5.284250E+03	66
-CONT-			-2.192913E+03	2.192913E+03	0.	67
	300	G	0.	0.	0.	68
-CONT-			4.045395E+01	2.831830E+03	0.	69
	310	G	0.	0.	0.	70
-CONT-			5.741610E+01	5.741610E+01	0.	71
	320	G	0.	0.	0.	72
-CONT-			-2.831830E+03	4.045395E+01	0.	73
	330	G	0.	0.	0.	74
-CONT-			-4.462056E+03	2.751137E+01	0.	75
	340	G	0.	0.	0.	76
-CONT-			-4.462056E+03	-2.751137E+01	0.	77
	350	G	0.	0.	0.	78
-CONT-			-2.831830E+03	-4.045395E+01	0.	79
	360	G	0.	0.	0.	80
-CONT-			5.741610E+01	-5.741610E+01	0.	81

(BLANK CARD)
(BLANK CARD)

```
$PICT
KHORZ = 1, KVERT = 3, PHI = 20.0, THETA = 0.0, PSI = 10.0,
ISCALE = 2, XORGN = 1.0, YORGN = 1.0, PSCALE = 2.0,
KDISP = 1, IDMAG = 2, DMAGS = 2.0, KODE = 1,
$
$PICT
KDISP = 3, KODE = 0,
$
```

APPENDIX E

INPUT DATA FOR CONTOUR PLOT OF EXAMPLE PROBLEM

```
     FLAT PLATE UNDER UNIFORM PRESSURE - CONTOUR PLOT
 $OPTION
 NNDEST = 36, NELEST = 25,
 KPLOT = 3, IDCASE = 1,
 $
FORMAT     (8X,I3,13X,2F8.1)
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                 8.0       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
GRID      240                 10.0      6.0       0.0
GRID      250                 0.0       8.0       0.0
GRID      260                 2.0       8.0       0.0
GRID      270                 4.0       8.0       0.0
GRID      280                 6.0       8.0       0.0
GRID      290                 8.0       8.0       0.0
GRID      300                 10.0      8.0       0.0
GRID      310                 0.0       10.0      0.0
GRID      320                 2.0       10.0      0.0
GRID      330                 4.0       10.0      0.0
GRID      340                 6.0       10.0      0.0
GRID      350                 8.0       10.0      0.0
GRID      360                 10.0      10.0      0.0
ENDGRID
FORMAT     (8X,I2,14X,4(I3,5X))
CQUAD1     6        1          10        20        80        70
CQUAD1     7        1          20        30        90        80
CQUAD1     8        1          30        40       100        90
CQUAD1     9        1          40        50       110       100
CQUAD1    10        1          50        60       120       110
CQUAD1    16        1          70        80       140       130
CQUAD1    17        1          80        90       150       140
CQUAD1    18        1          90       100       160       150
```

```
CQUAD1  19    1    100   110   170   160
CQUAD1  20    1    110   120   180   170
CQUAD1  26    1    130   140   200   190
CQUAD1  27    1    140   150   210   200
CQUAD1  28    1    150   160   220   210
CQUAD1  29    1    160   170   230   220
CQUAD1  30    1    170   180   240   230
CQUAD1  36    1    190   200   260   250
CQUAD1  37    1    200   210   270   260
CQUAD1  38    1    210   220   280   270
CQUAD1  39    1    220   230   290   280
CQUAD1  40    1    230   240   300   290
CQUAD1  46    1    250   260   320   310
CQUAD1  47    1    260   270   330   320
CQUAD1  48    1    270   280   340   330
CQUAD1  49    1    280   290   350   340
CQUAD1  50    1    290   300   360   350
ENDGEOM
     Z - DISPLACEMENTS
FORMAT 4  (4(I5,E15.6))
   10  0.              20  0.              30  0.              40  0.
   50  0.              60  0.              70  0.              80  5.284250E+03
   90  8.351675E+03   100  8.351675E+03   110  5.284250E+03   120  0.
  130  0.             140  8.351675E+03   150  1.324540E+04   160  1.324540E+04
  170  8.351675E+03   180  0.             190  0.             200  8.351675E+03
  210  1.324540E+04   220  1.324540E+04   230  8.351675E+03   240  0.
  250  0.             260  5.284250E+03   270  8.351675E+03   280  8.351675E+03
  290  5.284250E+03   300  0.             310  0.             320  0.
  330  0.             340  0.             350  0.             360  0.
(BLANK CARD)
 $PICT
 PSCALE = 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
NNDEST  =  36,
NELEST  =  25,
KGEOM   =  1,
KDATA   =  1,
NVALUS  =  0,
IRESEQ  =  1,
KPLOT   =  3,
INFOR   =  1,
XSPACE  =  0.1E+02,
KSIGN   =  1,
IDCASE  =  1,
$END

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


GRID POINT INFORMATION
```

RESEQUENCED GRID POINT NUMBER	USER INPUT GRID POINT NUMBER	INPUT X	INPUT Y	BOUNDARY POINT INDICATOR
1	10	0.	0.	1.0000E+00
2	20	2.0000E+00	0.	1.0000E+00
3	30	4.0000E+00	0.	1.0000E+00
4	40	6.0000E+00	0.	1.0000E+00
5	50	8.0000E+00	0.	1.0000E+00
6	60	1.0000E+01	0.	1.0000E+00

7	70	0.	2.0000E+00	1.0000E+00
8	80	2.0000E+00	2.0000E+00	0.
9	90	4.0000E+00	2.0000E+00	0.
10	100	6.0000E+00	2.0000E+00	0.
11	110	8.0000E+00	2.0000E+00	0.
12	120	1.0000E+01	2.0000E+00	1.0000E+00
13	130	0.	4.0000E+00	1.0000E+00
14	140	2.0000E+00	4.0000E+00	0.
15	150	4.0000E+00	4.0000E+00	0.
16	160	6.0000E+00	4.0000E+00	0.
17	170	8.0000E+00	4.0000E+00	0.
18	180	1.0000E+01	4.0000E+00	1.0000E+00
19	190	0.	6.0000E+00	1.0000E+00
20	200	2.0000E+00	6.0000E+00	0.
21	210	4.0000E+00	6.0000E+00	0.
22	220	6.0000E+00	6.0000E+00	0.
23	230	8.0000E+00	6.0000E+00	0.
24	240	1.0000E+01	6.0000E+00	1.0000E+00
25	250	0.	8.0000E+00	1.0000E+00
26	260	2.0000E+00	8.0000E+00	0.
27	270	4.0000E+00	8.0000E+00	0.
28	280	6.0000E+00	8.0000E+00	0.
29	290	8.0000E+00	8.0000E+00	0.
30	300	1.0000E+01	8.0000E+00	1.0000E+00
31	310	0.	1.0000E+01	1.0000E+00
32	320	2.0000E+00	1.0000E+01	1.0000E+00
33	330	4.0000E+00	1.0000E+01	1.0000E+00
34	340	6.0000E+00	1.0000E+01	1.0000E+00
35	350	8.0000E+00	1.0000E+01	1.0000E+00
36	360	1.0000E+01	1.0000E+01	1.0000E+00

ELEMENT INFORMATION – WITH RESEQUENCED GRID POINTS

RESEQUENCED ELEMENT NUMBER	USER INPUT ELEMENT NUMBER	GRID POINTS 1	2	3	4
1	6	1	2	8	7
2	7	2	3	9	8
3	8	3	4	10	9
4	9	4	5	11	10
5	10	5	6	12	11
6	16	7	8	14	13
7	17	8	9	15	14
8	18	9	10	16	15
9	19	10	11	17	16
10	20	11	12	18	17
11	26	13	14	20	19
12	27	14	15	21	20
13	28	15	16	22	21
14	29	16	17	23	22
15	30	17	18	24	23
16	36	19	20	26	25
17	37	20	21	27	26

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

1	0.	2	0.	3	0.
6	0.	7	0.	8	5.28425E+03
11	5.28425E+03	12	0.	13	0.
16	1.32454E+04	17	8.35167E+03	18	0.
21	1.32454E+04	22	1.32454E+04	23	8.35167E+03
26	5.28425E+03	27	8.35167E+03	28	8.35167E+03
31	0.	32	0.	33	0.
36	0.				

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

4	0.	5	0.
9	8.35167E+03	10	8.35167E+03
14	8.35167E+03	15	1.32454E+04
19	0.	20	8.35167E+03
24	0.	25	0.
29	5.28425E+03	30	0.
34	0.	35	0.

```
$PICT
NPLOT   =  4,
XORGN   =  0.0,
YORGN   =  0.0,
PSCALE  =  0.2E+01,
ISCALE  =  3,
WMAGS   =  0.1E+03,
ICNTRS  =  10,
```

```
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,
$END

    MAXIMUM VALUE OF DATA = 1.3245400CE+04   MINIMUM VALUE OF DATA = 0.

    WMAG =        7.54979087E-03            ICNTR =     10
```

REFERENCES

1. Everstine, Gordon C.; and McKee, James M.: A Survey of Pre- and Postprocessors for NASTRAN. Structural Mechanics Computer Programs, W. Pilkey, K. Saczalski, and H. Schaeffer, eds., Univ. Press of Virginia, c.1974, pp. 825-847.
2. Tischler, V. A.; and Bernier, L. J. D.: Considerations for Developing a General Finite Element Pre- and Postprocessing System. Structural Mechanics Computer Programs, W. Pilkey, K. Saczalski, and H. Schaeffer, eds., Univ. Press of Virginia, c.1974, pp. 849-886.
3. Giles, Gary L.; and Blackburn, Charles L.: Procedure for Efficiently Generating, Checking, and Displaying NASTRAN Input and Output Data for Analysis of Aerospace Vehicle Structures. NASTRAN: Users' Experiences, NASA TM X-2378, 1971, pp. 679-696.
4. Giles, Gary L.; Blackburn, Charles L.; and Dixon, Sidney C.: Automated Procedures for Sizing Aerospace Vehicle Structures (SAVES). J. Aircraft, vol. 9, no. 12, Dec. 1972, pp. 812-819.
5. Giles, Gary L.: Procedure for Automating Aircraft Wing Structural Design. J. Struct. Div., Amer. Soc. Civil Eng., vol. 97, no. ST1, Jan. 1971, pp. 99-113.
6. Craidon, Charlotte B.: Description of a Digital Computer Program for Airplane Configuration Plots. NASA TM X-2074, 1970.
7. Sandford, Maynard C.; Ruhlin, Charles L.; and Abel, Irving: Transonic Flutter Study of a 50.5° Cropped-Delta Wing With Two Rearward-Mounted Nacelles. NASA TN D-7544, 1974.
8. Zienkiewicz, O. C.: The Finite Element Method in Engineering Science. McGraw-Hill Book Co., Inc., c.1971, pp. 116-117.
9. McCormick, Caleb W., ed.: The NASTRAN User's Manual (Level 15). NASA SP-222(01), 1972.

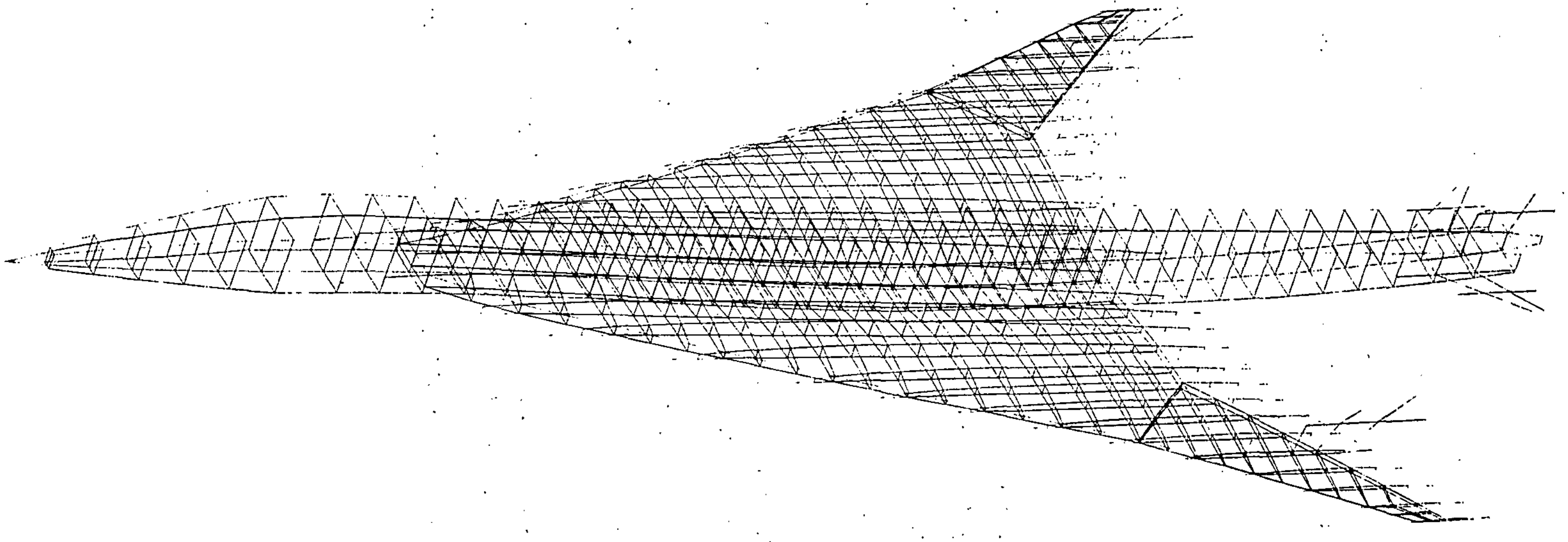


Figure 1.- Oblique orthographic projection of a finite-element model of an aircraft structure.

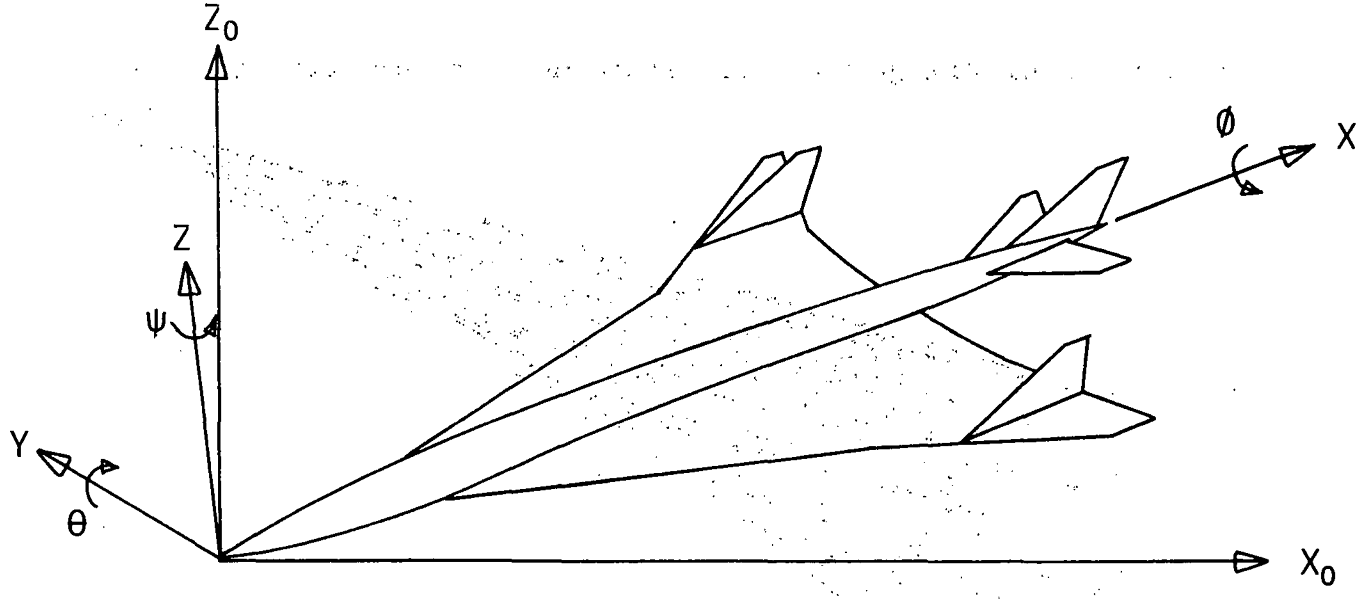


Figure 2.- Coordinate systems and Euler angles (rotations) for an oblique orthographic projection shown in X_0-Z_0 viewing plane.

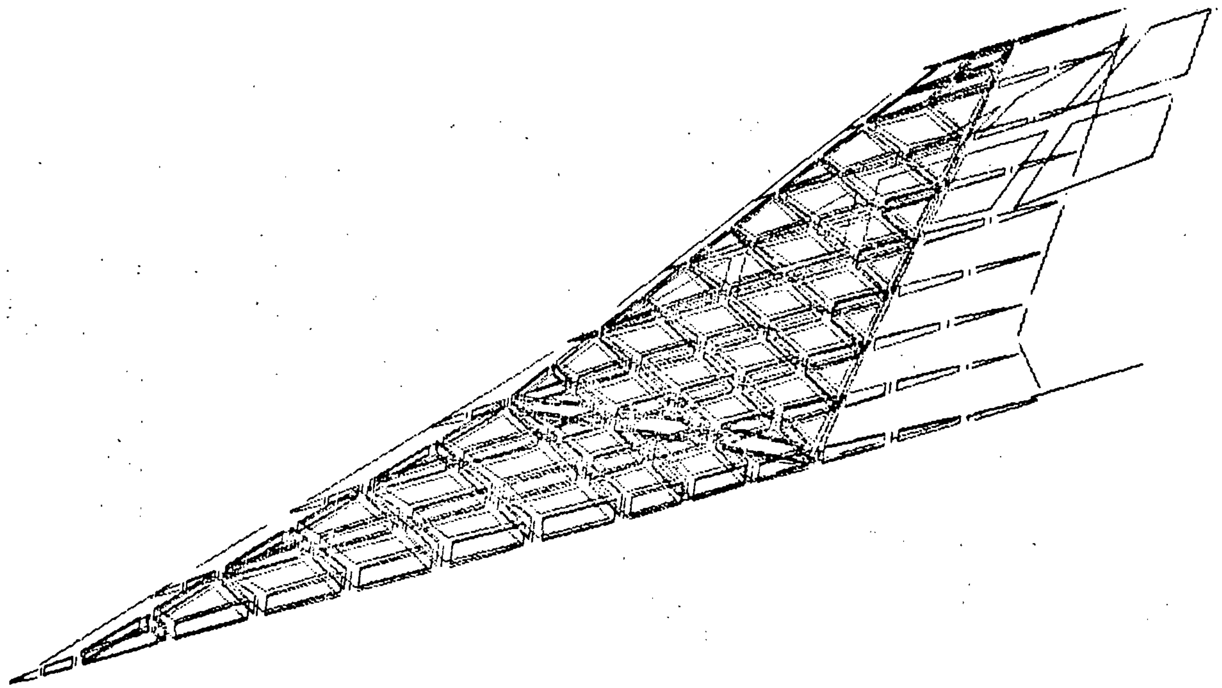
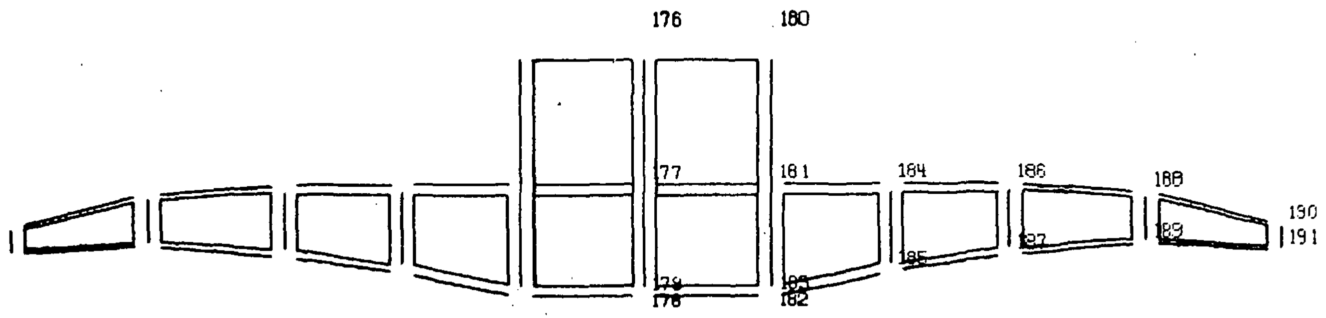
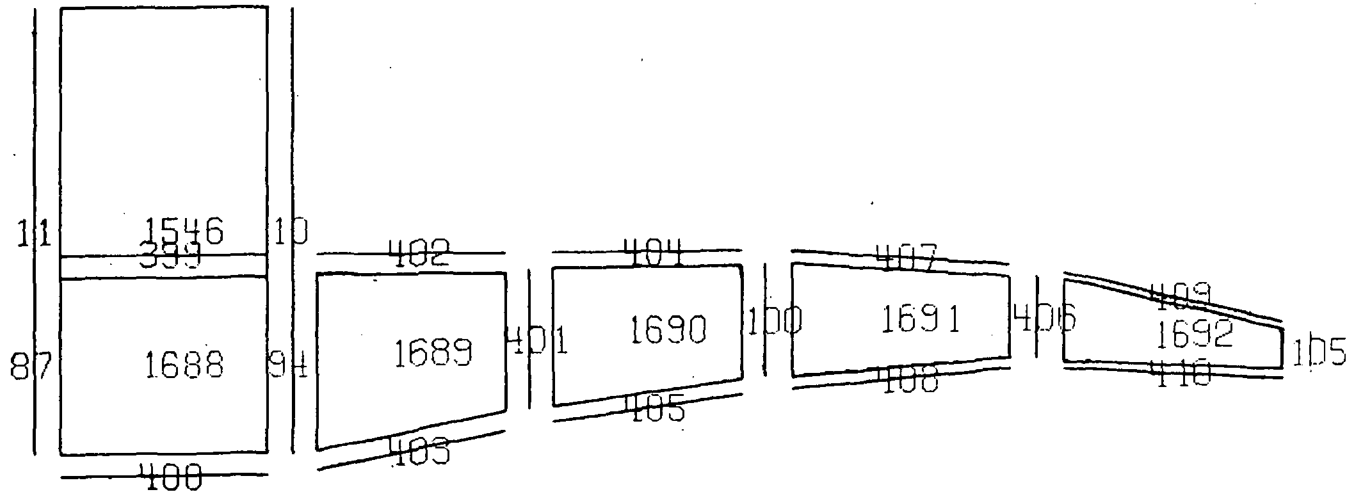


Figure 3.- Exploded plot of outboard portion of aircraft wing shown in figure 1.

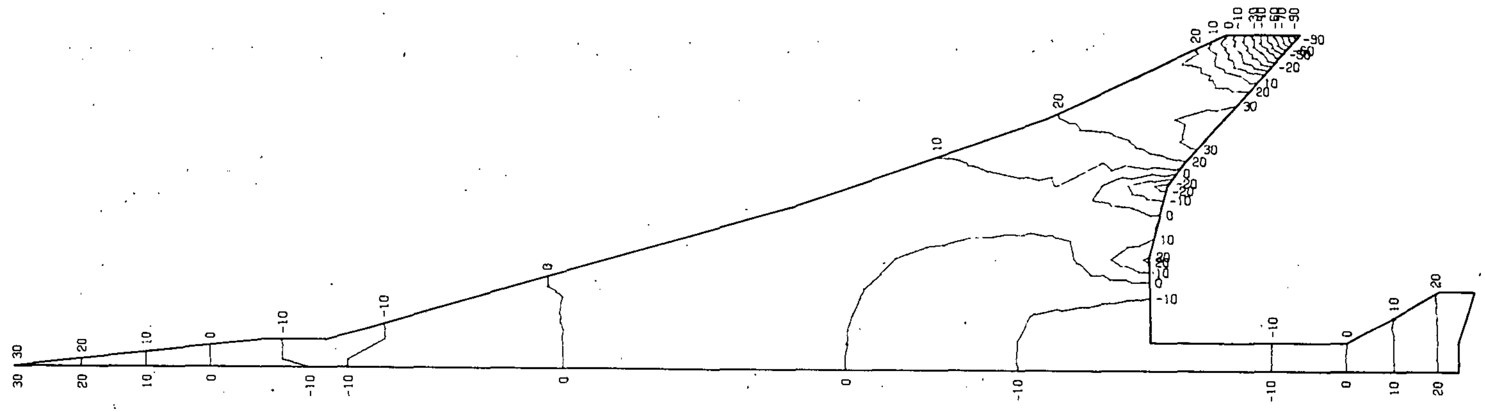


(a) Grid point numbers.

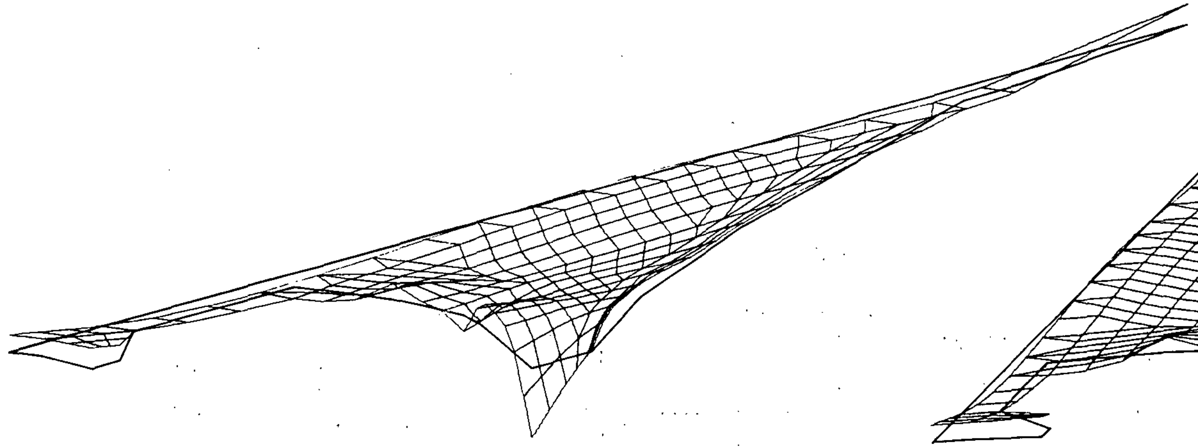


(b) Element numbers.

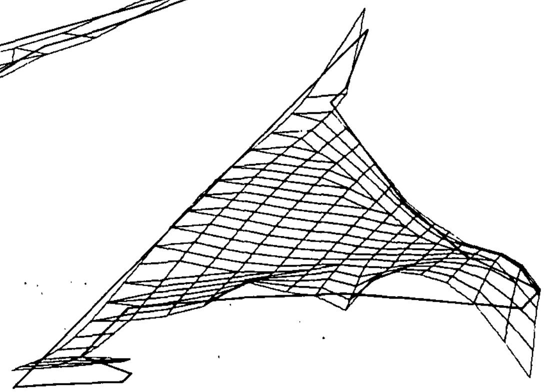
Figure 4.- Exploded plots of elements at a selected fuselage frame-wing spar location of figure 1.



(a) Contour plot.

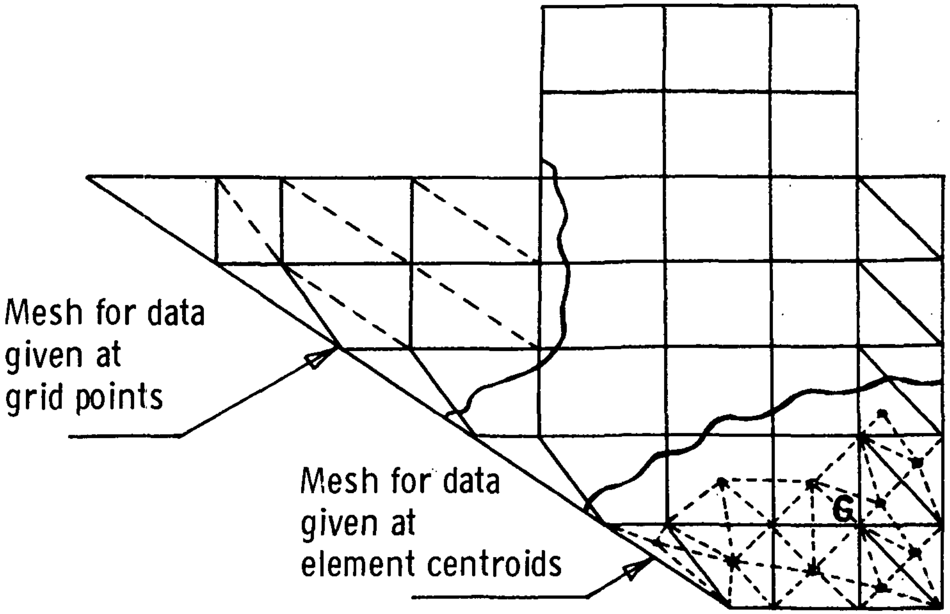


(b) View from side.

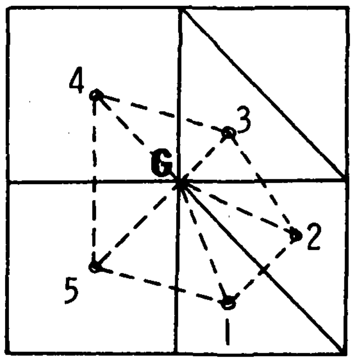


(c) View from rear.

Figure 5.- Contour plot and oblique orthographic projections of vibration mode shpaes of aircraft in figure 1.

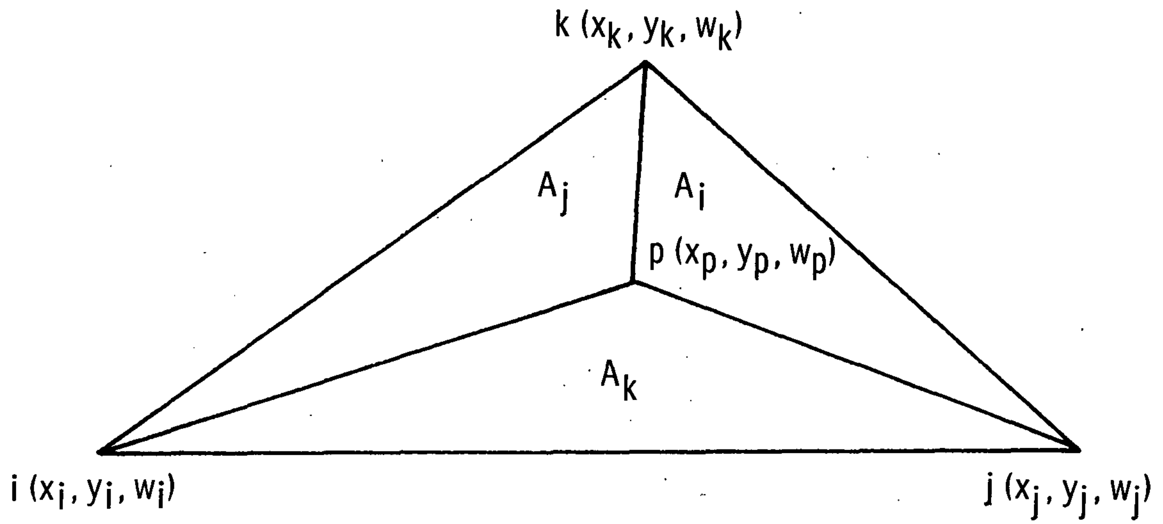


(a) Layout of typical model planform.



(b) Detail at grid point G.

Figure 6.- Triangular meshes on datum surface used in generating contour plots.



A_i = area of $\triangle$ jkp

A_j = area of $\triangle$ kip

A_k = area of $\triangle$ ijp

$\bar{A}$ = area of $\triangle$ ijk

Figure 7.- Diagram showing subareas of a triangle required to locate a generic point p by using area coordinates.

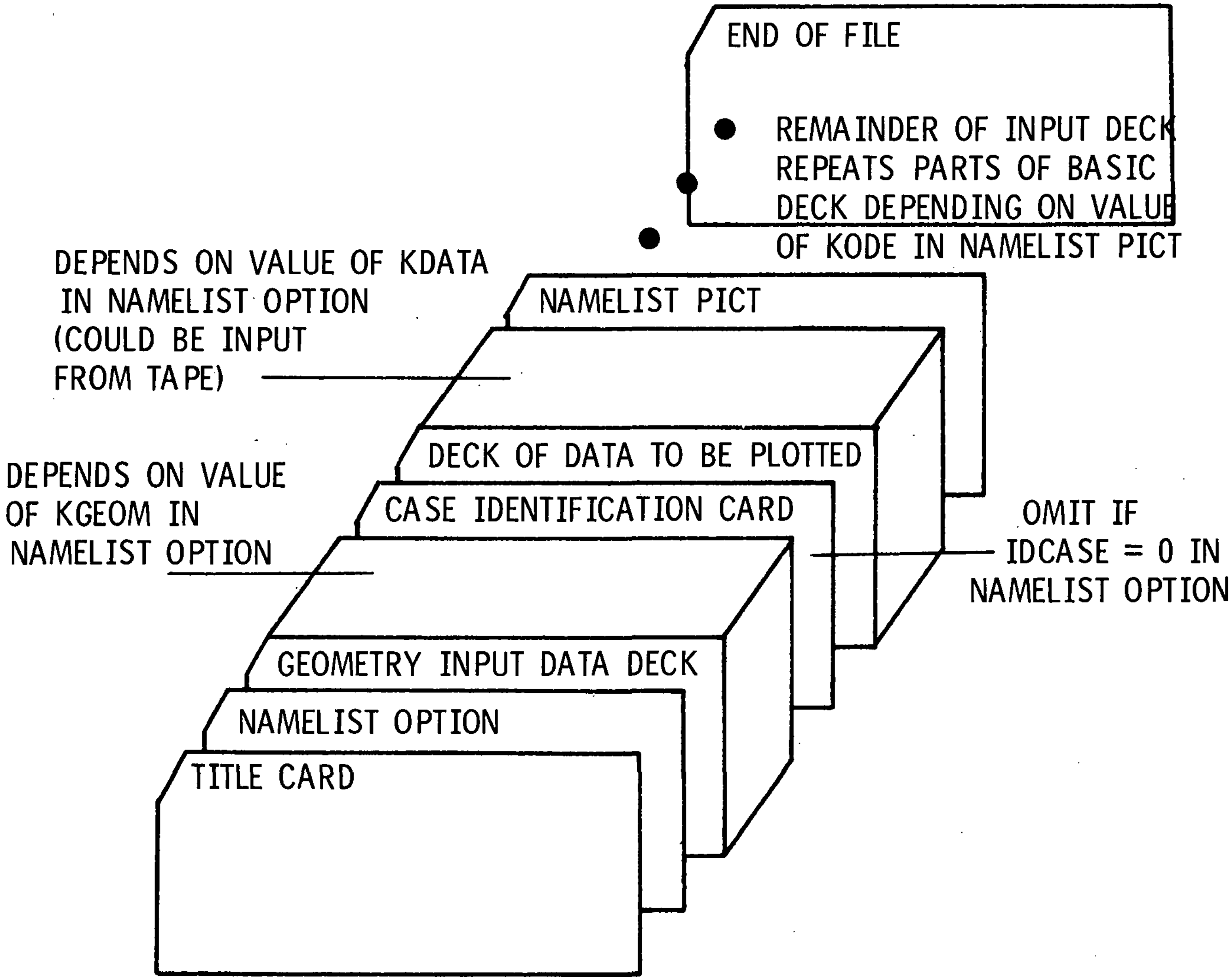
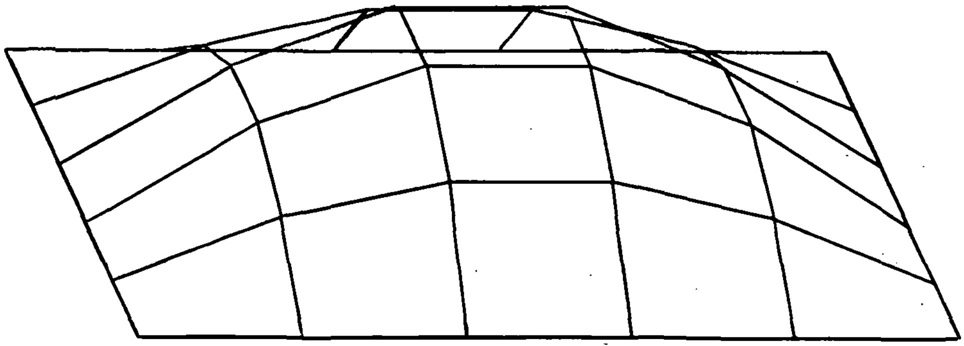
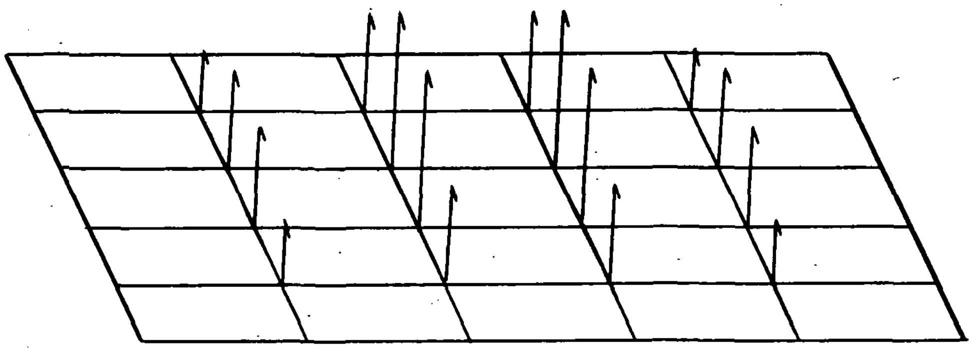


Figure 8.- A proper sequence for input cards.



(a) Deformed shape.



(b) Displacements represented by vectors.

Figure 9.- Oblique orthographic projections of a simply supported, square, flat plate under uniform pressure.

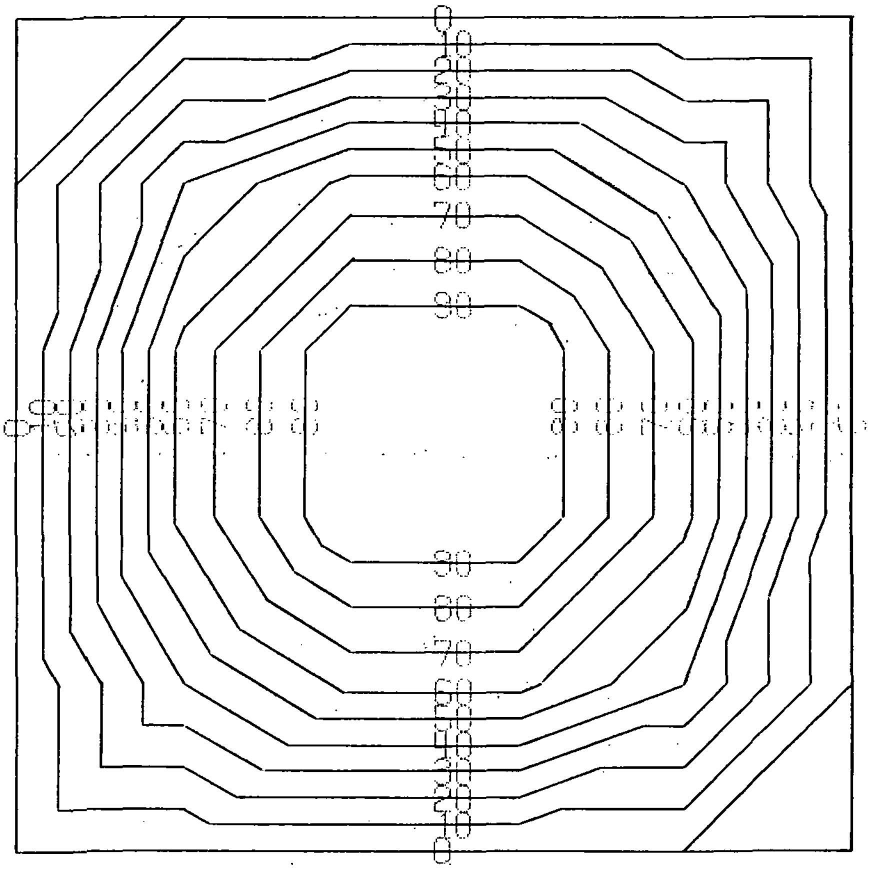
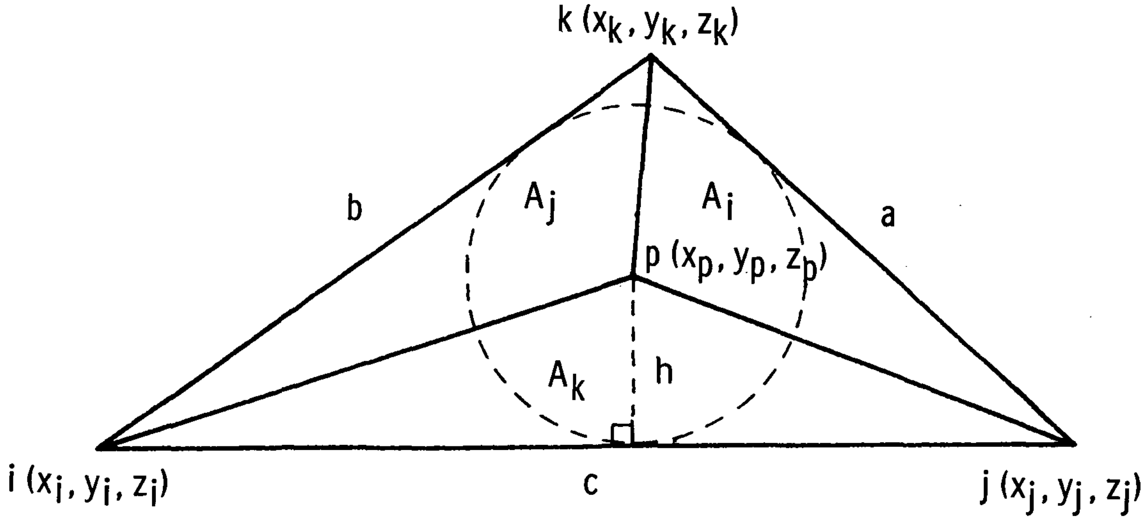


Figure 10.- Contour plot of a simply supported, square, flat plate under uniform pressure.



A_i = area of $\triangle$jkp

A_j = area of $\triangle$kip

A_k = area of $\triangle$ijp

$\bar{A}$ = area of $\triangle$ijk

Figure 11.- Diagram showing subareas of a triangle required to locate its incenter p by using area coordinates.

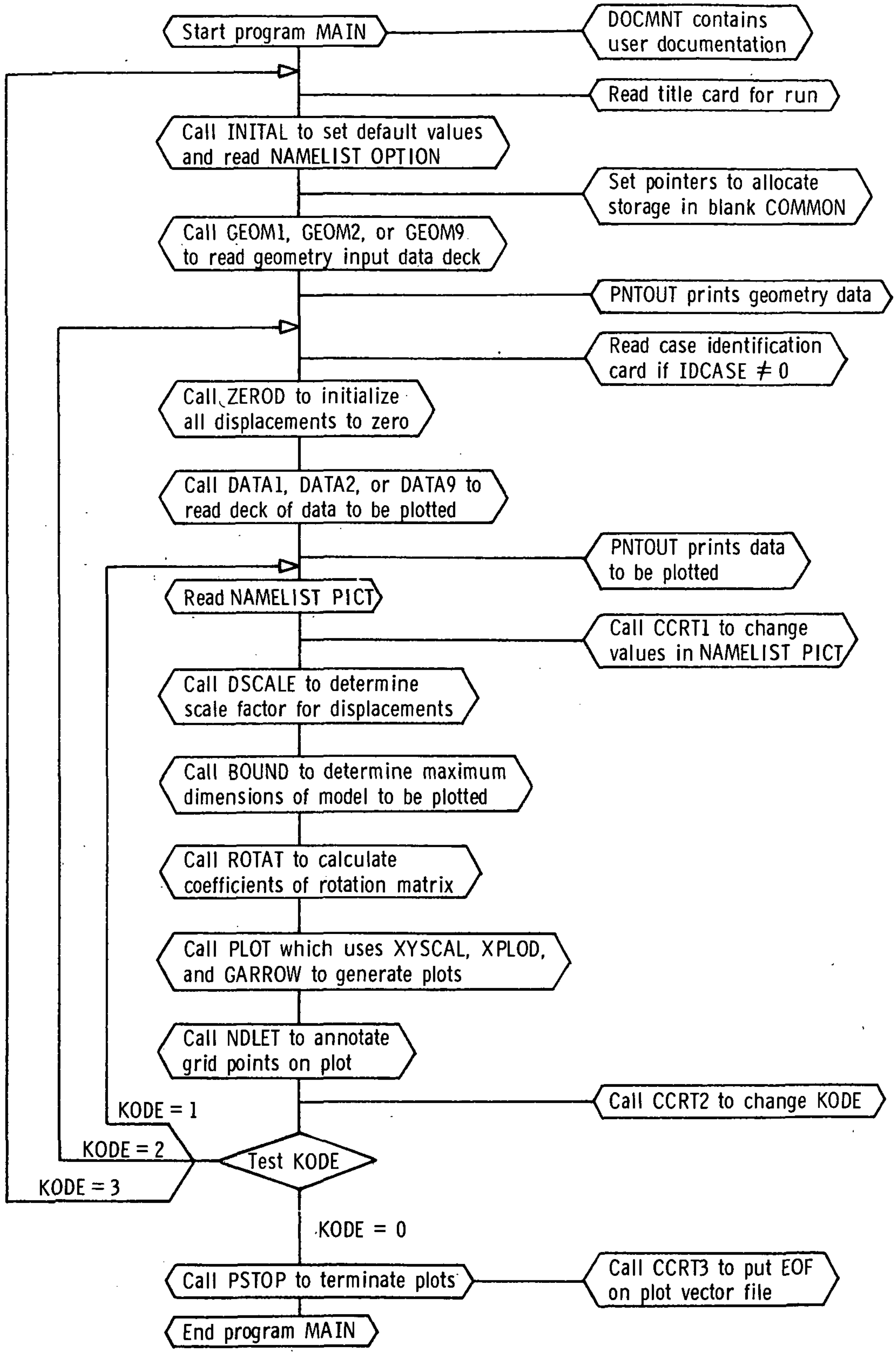


Figure 12.- Flow chart for program which generates oblique-orthographic-projection plots.

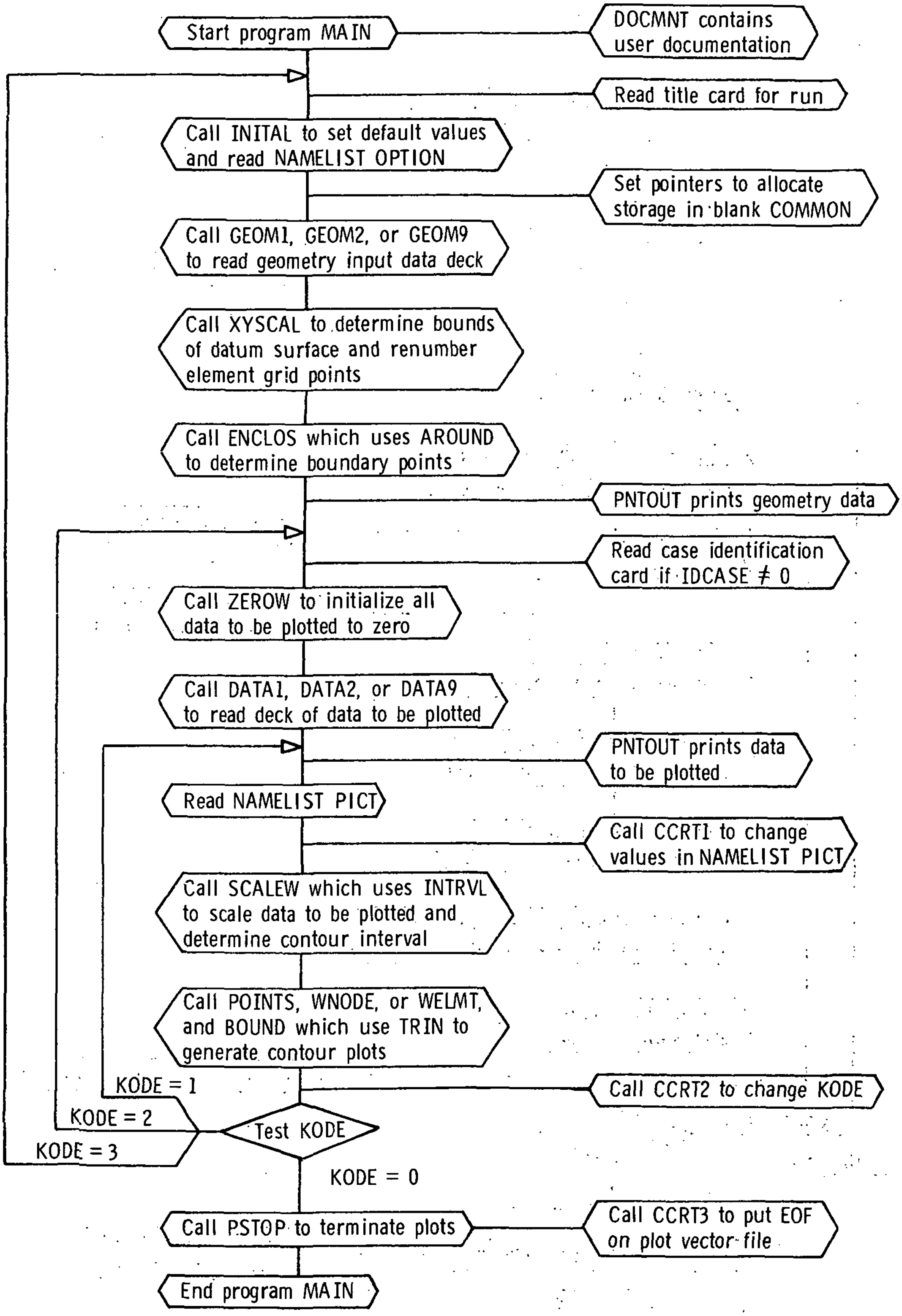


Figure 13.- Flow chart for program which generates contour plots.

NASA-Langley, 1975 L-9766

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

SPECIAL FOURTH-CLASS RATE
BOOK

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION
451

U.S.MAIL

059 001 C1 U 08 741213 S00120ES
PHILCO FORD CORP
AERONUTRONIC DIV
AEROSPACE & COMMUNICATIONS OPERATIONS
ATTN: TECHNICAL INFO SERVICES
JAMBOREE & FORD ROADS
NEWPORT BEACH CA 92663

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so 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 thereof."

—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:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546