N78-32463

NASA Contractor Report 157421

A COMPUTER GRAPHICS PROGRAM FOR GENERAL FINITE ELEMENT ANALYSES

Earl A. Thornton and Lynn M. Sawyer

OLD DOMINION UNIVERSITY RESEARCH FOUNDATION
Norfolk, VA 23508

NASA Grant NSG-1321
August 1978

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

A COMPUTER GRAPHICS PROGRAM FOR GENERAL FINITE ELEMENT ANALYSES

By
Earl A. Thornton
and
Lynn M. Sawyer

Technical Report

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Research Grant NSG 1321
Allan R. Wieting, Technical Monitor
Structures and Dynamics Division

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508

August 1978

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENT

The authors are pleased to note that the program described in this report is based upon an extensive revision and expansion of a program developed at the NASA Langley Research Center by Dr. Gary L. Giles.

A COMPUTER GRAPHICS PROGRAM FOR GENERAL FINITE ELEMENT ANALYSES

By

Earl A. Thornton[1] and Lynn M. Sawyer[2]

SUMMARY

Documentation for a computer graphics program for displays from general finite element analyses is presented. A general description of display options and detailed user instructions are given. Several plots made in structural, thermal and fluid finite element analyses are included to illustrate program options. Sample data files are given to illustrate use of the program.

INTRODUCTION

The widespread acceptance of the finite element method has led to a recognition of the importance of computer graphics in engineering analysis. Application of finite element computer programs to realistic structural, thermal, or fluid mechanics problems involves generation of large data arrays to specify the analytical model and its behavior. One of the advantages of the finite element method is that the analytical model and its calculated response can readily be displayed utilizing computer graphics. Several computer graphics computer programs, both passive and interactive (see refs. 1 and 2), have been developed particularly for applications to structural analysis. However, there still exists a need for well-documented, versatile, computer graphics programs which can be utilized for a variety of finite element analyses.

In a recent finite element thermal analysis study (ref. 3) a NASA-developed, passive, graphics program was utilized. The program (ref. 4), called GARYPLOT by users at NASA/LaRC, has a number of desirable features

---

[1] Associate Professor, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23508.

[2] Undergraduate Research Assistant, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23508.

such as several plotting options and a flexible format for accepting input data. GARYPLOT, however, has two significant limitations: (1) the program is restricted to lineal (rod) elements with two nodes and triangular or quadrilateral two-dimensional elements with three or four nodes; and (2) the program is limited to static response displacement plots. The element limitations are restrictive because finite element analyses frequently use higher order two-dimensional elements with more than four nodes and three-dimensional elements. The static response limitation is restrictive because of growing interest in finite element structural, thermal and fluid transient analyses.

The purpose of this report is to present a new, passive, computer graphics program (ELPLOT) which can be utilized for displays from general finite element analyses, including structural, thermal, and fluid applications. The program is based upon GARYPLOT, but extensive revisions and additions have been made to remove GARYPLOT's limitations and to increase ELPLOT's capabilities.

In the main body of the report, several features of ELPLOT are discussed: (1) display capabilities including view selection, cutting planes, exploded plots, symmetry options, etc., (2) alternative methods of data input, (3) one-, two-, and three-dimensional element types, and (4) static and dynamic response display capabilities. Plots generated by the program in structural, thermal, and fluid applications will be presented to illustrate the display features. User instructions and sample input data for the plots are presented in Appendixes A and B.

DISPLAY FEATURES

View Specification

Oblique orthographic projections are used to allow the analytical model to be viewed in any selected orientation. An example of oblique orthographic projection of a finite element structural model is shown in figure 1. The model was used in a NASA design study of the wing structure shown.

In the oblique orthographic projection approach, Euler angle transformations are used to specify orientation of the model relative to a projection (viewing) plane. The Euler transformation projects the coordinate system of the model onto a viewing plane. Figure 2 shows the coordinate systems and Euler angles utilized. The model coordinates are expressed in the x, y, z coordinate system, and the x_o, y_o, z_o coordinate system represents the fixed viewing planes. The model is rotated about the x, y, z axes to a selected orientation by the Euler angles (ψ, θ, ϕ). Model coordinates are then computed (transformation equations are given in ref. 4) by the program in a user-specified viewing plane such as the $x_o - y_o$ plane shown in figure 2. The order of the Euler angle rotations in the program is ψ, θ, and ϕ. The user as an option may select either $x_o - y_o$, $y_o - z_o$, or $y_o - z_o$ as the viewing plane.

Element Display

The program includes options to plot the complete undeformed model annotated with node or element numbers. Portions of the model can be isolated for closer examination by either: (1) sectioning (cutting) of planes parallel to the viewing planes ($x_o - y_o$, $y_o - z_o$, $y_o - z_o$), (2) specification of maximum and minimum node numbers to be included in a plot, or (3) inclusion of maximum and minimum element numbers in a plot. Also, exploded views can be generated which separate the elements in a model to aid in detecting the absence or presence of elements. For example, a rod element coincident with the edge of a quadrilateral element cannot easily be detected in an unexploded plot, but an exploded view will clearly display both elements. Another feature is the capability to plot a complete figure from a finite element model based on symmetry about one or more of the coordinate planes. For example, this is a useful option to aid in viewing the deflected shape of a symmetrical structure when only a portion of the structure is actually modeled in the analysis. Symmetry plotting can be performed about all three viewing planes.

Element Types

One-, two-, and three-dimensional finite element families with an arbitrary number of nodes may be plotted. Elements are drawn by consecutive straight lines between nodes, and element node numbers must follow patterns similar to those displayed in figure 3. Three-dimensional elements must have an even number of nodes. Alternative numbering schemes which arise in finite element analysis programs can easily be accomodated by renumbering the finite element nodes during input of the element data to ELPLOT (see "Finite Element Input Data Options" for further details).

Response Displays

Display options are available for plotting static and dynamic responses. In the static response option, nodal-dependent variables such as displacements or temperatures from a particular analysis can be superimposed on nodal coordinates of the model to display a deformed structure or spatial temperature distribution. Nodal-dependent variables can also be represented as vectors extending from the nodes. Spatial plots of nodal-dependent variables can be generated for an arbitrary number of displacement cases. For example, the deformed shape of a structure at selected instants of time or an arbitrary number of vibration mode shapes may be displayed.

In the dynamic response option, x - y plots are made of a nodal-dependent variable such as a displacement component, temperature versus time, or a timelike parameter. Dynamic response plots can be made at an arbitrary number of user-specified nodes.

PROGRAM EXECUTION

Control Commands

Three basic control commands are utilized to control program execution. These commands are standard FORTRAN NAMELIST statements. The control commands are

```
$OPTION ...  $
$PICT   ...  $
$HIST   ...  $
```

Each of these commands has a set of input parameters to provide program and plot control (see Appendix A). All of the parameters, however, have default values so that only a minimum number of control parameters actually must be specified.

NAMELIST OPTION is used to establish basic control values for a plot problem. The basic control values consist of parameters for allocating dynamic storage or selecting the output device (e.g. Calcomp or VARIAN at NASA/LaRC), paper size, the input mode for the finite element data, etc.

NAMELIST PICT is used to specify the desired plot display features. Options such as viewing planes, Euler angles, exploded views, cutting planes, and annotation can be selected and given desired values. NAMELIST PICT is used to control display features for undeformed, exploded, and static response plots.

NAMELIST HIST is used to specify display features for dynamic response plots. Options such as titles, scales, axes labels, and plot symbols can be selected and specified.

Execution Options

The basic execution options are shown in figure 4. Each new problem begins with a title card followed by a NAMELIST OPTION to specify the basic plot control parameters. Next, a NAMELIST PICT is used to specify display parameters for the first plot and set the value of the flow control parameter KODE. The first plot must be an undeformed plot of the finite element model.

The subsequent flow of the program is determined by the value specified for KODE on NAMELIST PICT. A value of KODE = 0 (the default value) on the first NAMELIST PICT causes the program to terminate after a single plot. A specified value of KODE = 1 will cause the program to read a $PICT for a second plot of the model. Alternatively, KODE = 2 or 3 will cause the program to read a $PICT or a $HIST for static or dynamic response plots, respectively. A value of KODE = 4 causes the program to read a title card for a new problem. A large number of plot options can be specified on a single ELPLOT execution since there is no limitation on looping within the program by using the optional values of KODE in NAMELIST PICT and HIST. A single NAMELIST PICT or HIST will generate several cases of static or dynamic response plots.

Finite Element Data Input Options

Three options are available to input the finite element model geometry and displacement data. These options permit ELPLOT to be compatible with a variety of finite element analysis programs. The options are (1) input data may be read from cards or files containing card images utilizing a user-specified format, (2) input data may be read from unformatted binary files according to ELPLOT specifications, and (3) user-supplied subroutines.

The first option is typically useful when a complete data deck describing a finite element model exists. Nodal coordinates and element connections can be read with user-specified formats, and the analytical model can be displayed by ELPLOT for data verification. Displacement data sets can also be input from cards or card images for response plots in a similar manner. The second option is useful when it is desired that ELPLOT serve as a postprocessor to a finite element analysis program. Typically, binary geometry and displacement data files are generated during execution of an analysis program, and ELPLOT subsequently reads these data as input. To use ELPLOT in this manner, the binary data files must be generated by unformatted write statements on tapes 8 and 20 in the analysis program. Geometry data is written on tape 8, and displacement data is written on tape 20. Sample subroutines illustrating these write statements are presented in Appendix A. The third option of user-supplied subroutines is facilitated by ELPLOT "dummy" subroutines in which a user may insert any desired FORTRAN statements to read input data.

The parameters KGEOM and KDATA specified in NAMELIST OPTION are used to select the finite element input data options. Proper sequences of NAMELISTS, geometry and displacement data are presented for option one in figure 5(a); proper sequences of NAMELISTS are presented for option two in figure 5(b).

APPLICATIONS

Computer plots obtained utilizing ELPLOT will be presented for five finite element models. Applications are presented for structural, thermal and fluid finite element analyses. The applications were selected to demonstrate several display features and the versatility of the program.

Wing Box Structural Optimization Model

A wing box utilized in a structural optimization study (ref. 5) is shown schematically in figure 6(a). The finite element model consists of 30 nodes connected by 68 rod elements and 8 triangular elements. The triangular elements have six nodes. The finite element data for the optimization program were prepared in card image form and were read into ELPLOT using finite element input data option one (KGEOM = 1). The input data file for the ELPLOT execution of the wing box is given in Appendix B.

Three plots of the finite element model are presented in figure 6 (b to d). Figure 6(b) shows an oblique view of the complete model annotated with node numbers. Figure 6(c) shows the top face of the model isolated by two cutting planes parallel to the x - y plane. An exploded view of the top face is shown in figure 6(d). The exploded view clearly shows the nodes to be connected by rods and triangles. The presence of the triangles is not obvious from the displays shown in figure 6(b and c) since rod elements alone could have generated the figures.

Bolted-Joint Specimen

A finite element structural model of a composite material, bolted-joint specimen is presented in figure 7. The bolted joint is symmetrical about the x - z plane, and only one-half of the specimen is modeled. The specimen is modeled with 15 three-dimensional finite elements. Each element has 16 nodes. Figure 7(a) shows the finite element model annotated with element numbers. The coordinate axes are offset from the model for clarity. Figure 7(b) shows an exploded view of the model with the symmetrical portion of the model plotted utilizing the symmetry option of ELPLOT. Figure 7(a and b) shows that two nodes connected to element 7 have erroneous coordinates.

Oceanographic Instrument Truss

The finite element model and static response (deformation) shapes of an oceanographic instrument truss are presented in figure 8(a to d). The model consists of 21 nodes connected by 51 rod elements. The structure was statically analyzed for two static load cases using the truss analysis program STAP (ref. 6). During the STAP execution, two binary data files were created and saved as plot-input files. Using input data option two (KGEOM = 2 and KDATA = 2), these data were subsequently read by ELPLOT. A separate file containing the plot control commands was used to execute ELPLOT; the control command file is given in Appendix B.

This example illustrates the use of ELPLOT as a postprocessor to a finite element analysis program. Most of the geometry data (nodal coordinates and elements) were generated internally by STAP. Since the geometry data and displacements were saved on permanent files, several executions of ELPLOT could be made without reexecution of STAP.

The plots shown illustrate the plotting of more than one displacement case with a single NAMELIST PICT. The static response option has also been executed twice to give two different views of the deformed truss. The displacements have been exaggerated by a magnification factor (DMAGS) for clarity. Coordinate axes are offset, and the y axis has been extended to serve as a vertical reference.

Scramjet Fuel-Injection Strut

Plots from a transient finite element thermal analysis of a scramjet fuel-injection strut are presented in figure 9(a to e). Figure 9(a) shows the engine schematically, and figure 9(b) shows the finite element model of the fuel-injection strut cross section. The finite element thermal model has 122 nodes and 118 elements. Element types include rods, triangles and quadrilateral elements with four and six nodes. Several plots of the finite element model shown in figure 9(b) have been made using ELPLOT as a postprocessor to the thermal analysis program. A typical section of the model is shown in figure 9(c); this plot was generated by specifying minimum and maximum node numbers. A transient temperature response of a typical node selected from figure 9(c) is shown in figure 9(d), which illustrates some of the available options for dynamic response plots such as plot scales, axes labels, and symbols. The plot control data file for the plots shown in figure 9(b to d) is given in Appendix B.

Fluid Flow About a Cylinder

Plots obtained from a finite element analysis of incompressible, inviscid flow about a cylinder are shown in figure 10(a to c). The flow was analyzed using the stream function program presented in reference 7. ELPLOT was executed as a postprocessor using finite element data files created by the fluid analysis program. The plot control data file used to generate the plots is given in Appendix B. Figure 10(a) presents the finite element model with the coordinate axes offset for clarity. Figure

10(b) shows an exploded view of the triangular elements. Figure 10(c) displays the finite element model with fluid velocity vectors plotted at the nodes. Vectors can be used in ELPLOT to represent any static response. The vectors can be scaled by an arbitrary magnification factor (DMAGS).

PROGRAM DESCRIPTION

Computer Requirements

The program ELPLOT is written in FORTRAN IV for the CDC 6600 series of computers. The program in its original form is operational on the NASA/LaRC Network Operating System. The program makes use of ten subroutines from the Langley graphics library. The program is also operational on the Old Dominion University DEC 10 computer system. Equivalent ODU graphics routines were substituted for the Langley subroutines. Where feasible, the program has been written to accomodate convenient transfer to other computer systems. For example, Hollerith fields are used in all FORMAT statements.

Dynamic storage allocation is used to accomodate problems of varying size. All large arrays are stacked in a blank COMMON designed ZZZ in the main program. The amount of blank COMMON required for model display and static response is determined by the total number of nodal points, the number of displacement components, and the maximum number of element nodes. The amount of blank COMMON required for dynamic response plots is determined by the number of x - y data points per plot and the number of plot cases. Dynamic storage is automatically allocated in the program. The storage requirements for model display and static response plots are included as part of the printed output.

Internal Documentation

The program is internally documented with pertinent comments. In addition, a subroutine DOCMNT is included to facilitate conversion to other systems. DOCMNT contains: (1) a complete description of all graphics subroutines, their purpose and definitions of parameters in the CALL statement, (2) a complete set of user instructions, (3) sample subroutines for creating binary finite element data files, and (4) a list of modifications required to transfer the program from the Langley computer system to the ODU system.

CONCLUDING REMARKS

A passive computer graphics program (ELPLOT) for displays from general finite element analyses has been presented. The program generates oblique orthographic projection plots of a finite model with several view and annotation options. General families of one-, two-, and three-dimensional finite elements are permitted. Display options are available for plotting static and dynamic responses. Three options for finite element data input are available to permit compatibility with a variety of finite element analysis programs. Several plots made in structural, thermal, and fluid finite element applications are presented to illustrate various program options. User instructions and sample input data are included in the Appendices.

APPENDIX A

PLOT CONTROL DATA

PLOT CONTROL DATA

Input Data

Input deck sequences for ELPLOT are shown schematically in figure 5. The data must be input in the order shown and is described in detail in this section. Some basic NAMELIST rules are summarized at the end of Appendix A.

Title Card.- This card contains any desired alphanumeric information in columns 1 through 80. The title will appear in the first plot frame and on the printed output.

NAMELIST OPTION.- This NAMELIST contains values to allocate storage in blank common and to specify control values needed by the program. The following values are included:

NNDEST = Estimated number of node points to be used. Value must be greater than or equal to the actual number of node points.

** Default = 200 **

MAXNDS = Maximum number of nodes on any element.

** Default = 8 **

NUDISP = 0 for no displacement data in x-direction.

= 1 for data including displacements in x-direction.

** Default = 0 **

NVDISP = 0 for no displacement data in y-direction.

= 1 for data including displacements in y-direction.

** Default = 0 **

NWDISP = 0 for no displacement data in z-direction.

= 1 for data including displacements in z-direction.

** Default = 0 **

KGEOM specifies subroutine and corresponding method of input for model geometry.

KGEOM = 1 node points and elements read from cards with user-specified format by subroutine GEOM1.

= 2 node points and elements read as binary records from TAPE 8 by subroutine GEOM2.

= 3 user-supplied subroutine - GEOM3.

** Default = 2 **

KDATA specifies subroutine and corresponding method of input for displacement data.

KDATA = 1 displacement data read from cards with user-specified format by subroutine DATA1.
= 2 displacement data read as binary records from TAPE 20 by subroutine DATA2.
= 3 user-supplied subroutine - DATA3.
** Default = 2 **

NCASES = the number of cases of displacements to be read from TAPE 20.
** Default = 1 **

IRESEQ = 0 for no resequencing of node point numbers.
= 1 to resequence node point numbers in same order as they are input.
** Default = 1 **

KPLOT specifies the type of output device to be used.

KPLOT = 1 for Calcomp.
= 2 for Calcomp with plotting speed reduced to use Leroy pens.
= 3 for VARIAN.
** Default = 3 **

XSPACE = space between plots in x-direction, in inches.
** Default = 10.0 **

PSIZE = paper size in y-direction in inches, used in scaling of plots to insure this dimension is not exceeded.
** Default = 13.0 **

NAXES = 1 for coordinate axes
= 0 for no coordinate axes
** Default = 1 **

<u>Finite element geometry.</u>- The finite element model geometry is input in 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 through 6 and a variable format corresponding to the format of the node point cards with left parenthesis starting in column 11 and up to 80 columns may be used.

(B) Deck of node point cards. Each card contains 4 values, node 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 ENDNODE in columns 1 through 7.

(D) A single card containing the word TYPE in columns 1 through 4 and the values for the word KTYPE and NEND in colums 11 through 20 and 21 through 30, respectively. KTYPE (integer) should have the value 1, 2, or 3 to indicate a one-, two-, or three-dimensional element. NEND (integer) is the number of element nodes.

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

(F) Deck of element cards. Each card contains the element number and the node connections. The format is specified in (E) above.

(G) A single card containing the word ENDGROUP in columns 1 through 7. (An arbitrary number of element groups may be used.)

(H) A single card containing the word ENDGEOM in columns 1 through 7.

KGEOM = 2

The finite element nodal and connection data are read from binary records on TAPE 8. Sample subroutines to generate these files are given in Appendix B.

Nodal displacement data.- Displacement data to be plotted is input in 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 through 6 and a variable format for the data cards with left parenthesis starting in column 11 and up to 80 columns may be used. If displacements are included for more than one node point per card, the number of node 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. A displacement set for each node point is defined to contain from 2 to 4 values, a node point number and displacements corresponding to NUDISP, NVDISP, or NWDISP equal to 1.

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

KDATA = 2

The nodal displacement data are read from binary records on TAPE 20. Sample subroutines to generate these files are given in Appendix B.

NAMELIST PICT.- This NAMELIST contains values to specify the plot desired and the information to be included on the plot.

The following values are included:

KHORZ = integer designating horizontal axis of viewing plane, where 1 = x, 2 = y, 3 = z.
** Default = 1 **

KVERT = integer designating vertical axis of view plane, where 1 = x, 2 = y, 3 = z.
** Default = 2 **

XOFF = real number designating origin of x-coordinate axis
** Default = 0.0 **

YOFF = real number designating origin of y-coordinate axis
** Default = 0.0 **

ZOFF = real number designating origin of z-coordinate axis
** Default = 0.0 **

XLNTH = length of x-axis
** Default = 15% of PSIZE **

YLNTH = length of y-axis
** Default = 15% of PSIZE **

ZLNTH = length of z-axis
** Default = 15% of PSIZE **

PHI = angular rotation of model about its x-axis, in degrees (must be taken third).

** Default = 0.0 **

THETA = angular rotation of model about its y-axis, in degrees (must be taken second).

** Default = 0.0 **

PSI = angular rotation of model about its z-axis, in degrees (must be taken first).

** Default = 0.0 **

NEWFR = 1 for frame change before plot is made. (A frame change resets the x-origin past previous plot by XSPACE and the y-origin at 0.0.)

NEWFR. NE. 1 for no frame change before plotting.

** Default = 1 **

ISCALE = 1 for internal origin location and scaling.

= 2 for user-specified origin and scaling.

** Default = 1 **

PLOTSZ = maximum dimension desired on completed plot. (Used for scaling if ISCALE = 1.)

** Default = 13.0 **

XORGN = x-location of plot origin (used if ISCALE = 2).

** Default = 0.0 **

YORGN = y-location of plot origin (used if ISCALE = 2).

** Default = 0.0 **

PSCALE = model size reduction factor, PSCALE = actual model size/desired plot size (used if ISCALE = 2).

** Default = 1.0 **

NOTAT = 0 for no numbering on plots.

= 1 for numbering of node points.

= 2 for numbering of elements.

** Default = 0 **

XLHT = height of integers specified by NOTAT, in inches.

** Default = 0.15 **

KDISP = 0 for undeformed plot.

= 1 for deformed plot.

= 2 for exploded plot.

= 3 for displacements represented by vectors.

** Default = 0 **

IDMAG = 1 for direct scaling of data by DMAGS.

= 2 for scaling of data to a maximum value of DMAGS.

** Default = 2 **

DMAGS = magnification of displacements (if KDISP = 1).

= reduction factor of elements (if KDISP = 2).

** Default = 1.0 **

KSYMXY = 1 for symmetry about x-y plane.

** Default = 0 **

KSYMXZ = 1 for symmetry about x-z plane.

** Default = 0 **

KSYMYZ = 1 for symmetry about y-z plane.

** Default = 0 **

XXMAX, YYMAX, ZZMAX, XXMIN, YYMIN, ZZMIN locate cutting planes parallel to principal (x-y, x-z, y-z) planes to limit plot.

** Default XXMAX = YYMAX = ZZMAX = 1.0E + 20 **

** Default XXMIN = YYMIN = ZZMIN = -1.0E + 20 **

NDMAX = maximum node point to be included in plot.

** Default = 9999999999 **

NDMIN = minimum node point to be included in plot.

** Default = 0 **

NELMAX = maximum element number to be included in plot.

** Default = 9999999999 **

NELMIN = minimum element number to be included in plot.

** Default = 0 **

KODE = 0 last plot, exit from program.

= 1 read another NAMELIST PICT.

= 2 read a NAMELIST PICT for deformed plot.

= 3 read a NAMELIST HIST for history plot.

** Default = 1 **

One undeformed plot must be made before plotting displacement data. On the NAMELIST $PICT which describes the first or any subsequent $PICT for undeformed plots a value of KODE equal to two or three will cause the program to read $PICT for a deformed plot or a history plot, respectively.

The last NAMELIST PICT or HIST must have KODE = 0 in order to exit from the program.

NAMELIST HIST.- This NAMELIST contains values to specify the nodes and displacement component for dynamic response plots and information to appear on the plots. The following values are included:

NODES = number of displacement history plots.

** Default = 1 **

IDISP = 1 x-displacement to be plotted.

= 2 y-displacement to be plotted.

= 3 z-displacement to be plotted.

** Default = 3 **

XPG = length of x-axis in inches

** Default = 9.0" **

YPG = length of y-axis in inches

** Default = 9.0" **

XTIC = distance in inches for x-axis major tic marks.

** Default = 1.0" **

YTIC = distance in inches for y-axis major tic marks.

** Default = 1.0" **

XDIV = number of minor tic mark divisions per inch for x-axis.

** Default = 10.0 **

YDIV = number of minor tic mark divisions per inch for y-axis.

** Default = 10.0 **

YLHT = height of letter in inches.

** Default = 0.15" **

NOCHAR = number of characters in axes labels.

** Default = 25 **

JLINE specifies how the data points are to be plotted.

JLINE = -1 symbol only.

= 0 line plot.

= +1 line and symbol plot.

** Default = 1 **

ISYM = integer number designating standard NASA symbol

** Default = 2 ** square symbol

ISIZE = integer number designating size of symbol.

= 1 small.

= 2 medium.

= 3 large.

** Default = 2 **

LSCALE = 1 scaling of plot by program.

= 2 user-specified scaling by input values for XMIN, XMAX, YMIN, YMAX.

XMIN = minimum value for x-axis

** Default = 0.0 **

YMIN = minimum value for y-axis

** Default = 0.0 **

XMAX = maximum value for x-axis

** Default = 1000.0 **

YMAX = maximum value for y-axis

** Default = 1000.0 **

KODE = 0 last plot, exit from program.

= 1 read another NAMELIST PICT.

= 2 read a NAMELIST PICT for deformed plot.

= 3 read a NAMELIST HIST for history plot.

** Default = 1 **

One undeformed plot must be made before plotting displacement data. On the NAMELIST $PICT which describes the first or any subsequent

$PICT for undeformed plots a value of KODE equal to two or three will cause the program to read a $PICT for a deformed plot or a history plot, respectively.

The last NAMELIST PICT or HIST must have KODE = 0 in order to exit from the program. If a NAMELIST HIST appears, the following two card sets must appear next.

Node list for dynamic response plots.- A card set containing the node numbers is required for dynamic response plots. The program reads a list of nodes utilizing a free-field format, i.e. a list-directed read statement. The list must contain a number of nodes equal to the parameter NODES specified on the NAMELIST HIST. The default value of NODES is one, so at least one node number must be specified.

Axis labels for dynamic response plots.- Two cards must appear in this set: (1) the first card contains the x-axis label, and (2) the second card contains the y-axis label. The number of alphanumeric characters appearing in the axes' labels is specified by the parameter NOCHAR in the NAMELIST HIST.

NAMELIST Rules

The following NAMELIST rules are standard on many FORTRAN compilers:

1. NAMELISTS begin in column 2 with a dollar sign.

   Example:
   ```
   12345678910.........................80
    $OPTION
   ```

2. NAMELIST parameters are separated by a comma after the last constant.

   Example: $OPTION NNDEST = 300, NAXES = 1,

3. NAMELISTS end with a blank and then a dollar sign, no final comma.

   Example: $OPTION NNDEST = 300, NAXES = 1, MAXNDS = 10 $

APPENDIX B

SAMPLE INPUT DATA

SAMPLE INPUT DATA

Input data files for four applications of ELPLOT are presented. The data demonstrate utilization of the NAMELISTS and specification of the plot control parameters (Appendix A) which were used to generate the plots in figures 6 through 10.

Plot Control Files

Wing box structural optimization model.- In this application all of the data were entered in card image form. The complete data file is given below. The plots generated by ELPLOT from this data file are presented in figure 6.

```
  FINITE ELEMENT MODEL OF A WINGBOX
 $OPTION KGEOM=1,KDATA=1,NNDEST=30,XSPACE=5.0,KPLOT=2,PSIZE=10.0 $
FORMAT     (I5,3F6.1)
    1  0.0    0.0    44.5
    2  40.0    0.0  44.5
    3  80.0  0.0   44.5
    4  80.0    0.0  0.0
    5 40.0  0.0    0.0
    6  0.0    0.0    0.0
    7  0.0   40.0  44.5
    8  40.0   40.0  44.5
    9  80.0   40.0  44.5
   10  80.0 40.0   0.0
   11  40.0   40.0  0.0
   12  0.0   40.0  0.0
   13  0.0    80.0  44.5
   14  40.0   80.0  44.5
   15  80.0   80.0  44.5
   16  80.0   80.0  0.0
   17  40.0   80.0  0.0
   18  0.0   80.0  0.0
   19  0.0   120.0  44.5
   20 40.0 120.0 44.5
   21  80.0 120.0 44.5
   22 80.0  120.0  0.0
   23  40.0 120.0  0.0
   24  0.0  120.0  0.0
   25  0.0  160.0  44.5
   26 40.0  160.0  44.5
   27  80.0   160.0 44.5
   28  80.0   160.0   0.0
   29  40.0   160.0   0.0
   30  0.0  160.0  0.0
ENDNODE
TYPE          1             2
FORMAT     (3I5)
    1    1    6
    2    2    5
    3    3    4
    4    2    6
    5    2    4
    6    6    5
    7    5    4
    8    2    1
    9    3    2
   10    7   12
   11    8   11
   12    9   10
   13    8   12
```

14	8	10
15	12	11
16	11	10
17	7	8
18	8	9
19	13	18
20	14	17
21	15	16
22	14	18
23	14	16
24	18	17
25	17	16
26	13	14
27	14	15
28	19	24
29	20	23
30	21	22
31	20	24
32	20	22
33	24	23
34	23	22
35	19	20
36	20	21
1	1	7
2	7	13
3	13	19
4	19	25
5	7	6
6	13	12
7	19	18
8	25	24
9	6	12
10	12	18
11	18	24
12	24	30
13	2	8
14	8	14
15	14	20
16	20	26
17	5	11
18	11	17
19	17	23
20	23	29
21	3	9
22	9	15
23	15	21
24	21	27
25	4	9
26	10	15

```
   27   16   21
   28   22   27
   29    4   10
   30   10   16
   31   16   22
   32   22   28
ENDGROUP
TYPE         2          6
FORMAT     (7I5)
    1   13   20   27   26   25   19
    2   13   14   15   21   27   20
    3    1    8   15   14   13    7
    4    1    2    3    9   15    8
    5   18   23   28   29   30   24
    6   18   17   16   22   28   23
    7    6   11   16   17   18   12
    8    6    5    4   10   16   11
ENDGROUP
ENDGEOM
 $PICT KVERT=3,KHORZ=2,NOTAT=1,XOFF=-10.,YOFF=-10.,ZOFF=-10.,
       XLNTH=5.,YLNTH=5.,ZLNTH=5.,THETA=-15.,PSI=15. $
 $PICT XOFF=-10.,ZOFF=43.5,ZZMIN=43.5,ZZMAX=45.5,KHORZ=1,
     KVERT=2,THETA=0.,PSI=0. $
 $PICT KDISP=2,DMAGS=.7,KODE=0 $
```

Oceanographic instrument truss.- The plot control data file for the truss plots presented in figure 8 is given below. The finite element data were read from binary files generated by the finite element analysis program STAP. Sample subroutines utilized in STAP to generate the binary files are presented in the next section of this appendix.

```
  TRUSS  PROBLEM SOLVED BY STAP
$OPTION NNDEST=30, NCASES=2, NUDISP=1,NVDISP=1,NWDISP=1 $
$PICT NOTAT=1,KODE=2,YLNTH=65.,XLNTH=12.,XOFF=-2.,YOFF=-2.,ZOFF=-2.,
      ZLNTH=12.,PHI=10. $
$PICT NOTAT=0,KDISP=1,DMAGS=7.,KODE=1 $
$PICT NOTAT=1,KDISP=0,DMAGS=1.0,KHORZ=3,KODE=2,PHI=0.,
     THETA=30.,PSI=10. $
$PICT KDISP=1,DMAGS=7.,NOTAT=0,KODE=0 $
```

Scramjet fuel-injection strut.- The plot control data file for the thermal analysis plots presented in figure 9 is given below. The finite element data were read from binary files generated by a thermal analysis program (TAP). Plot routines used in TAP are presented in the next section of this appendix.

```
  SCRAMJET FUEL INJECTION STRUT   AUGUST 16, 1978
$OPTION  NNODEST=122, NWDISP=1, KPLOT=3, NAXES=0, NCASES=51 $
$PICT KDISP=2, DMAGS=0.7,NDMIN=107,NDMAX=122,NOTAT=1,KODE=3 $
$HIST LSCALE=2,YMAX=2000.,XMAX=10.,NODES=2,KODE=0 $
  111   112
  TIME(SEC)
 TEMPERATURE(F)
```

Fluid flow about a cylinder.- The plot data file for the plots of the finite element flow analysis in figure 10 is given below. The finite element data were read from binary files generated in the fluid analysis program.

```
 THORNTON
$OPTION NUDISP=1,NVDISP=1,KGEOM=2,KDATA=2 $
$PICT KODE=1,NOTAT=1,XLHT=0.075,XOFF=-.5,YOFF=-.5,XLNTH=3.,YLNTH=3.
$PICT NOTAT=2,KDISP=2,DMAGS=.7,KODE=2 $
$PICT KDISP=3,NOTAT=0,KODE=0 $
```

Creation of Binary Finite Element Data Files

Node and element data.- The subroutines listed below illustrate the form of the input data read by ELPLOT subroutine GEOM2. These subroutines may be used in a finite element program to generate the binary records read in subroutine GEOM2. The nodal data should appear first on TAPE 8 in the form shown in sample subroutine NDPLOT. The element data should follow in the form shown by subroutine ELPLOT. An end of file (EOF) should be written on TAPE 8 after the element data.

Subroutine NDPLOT.-

```
C
      SUBROUTINE NDPLOT(NUMNP,X,Y,Z)
C
C       GENERATES PLOT FILE FOR SUBROUTINE GEOM2 IN PLOT PROGRAM
C       NODE LIST AND NODAL COORDINATES
C
      DIMENSION X(1),Y(1),Z(1)
      WRITE(8) NUMNP
      WRITE(8) (I,I=1,NUMNP)
      WRITE(8) (X(I),I=1,NUMNP)
      WRITE(8) (Y(I),I=1,NUMNP)
      WRITE(8) (Z(I),I=1,NUMNP)
C
      RETURN
      END
```

Subroutine ELPLOT.-

```
C
C
      SUBROUTINE ELPLOT(KTYPE,NUMEL,NEND,NODE)
C
C     CREATS PLOT FILE FOR SUBROUTINE GEOM2 IN PLOT PROGRAM
C
C       KTYPE---ELEMENT TYPE
C               .EQ.1  ONE DIMENSIONAL
C               .EQ.2  TWO
C               .EQ.3  THREE
C
C       NUMEL---ELEMENT NUMBER
C
C       NEND ---NUMBER OF ELEMENT NODES
C
C       NODE ---NODE NUMBERS LIST
C
      DIMENSION NODE(NEND)
C
      WRITE(8) KTYPE,NUMEL,NEND,(NODE(I),I=1,NEND)
C
      RETURN
      END
C
```

Nodal displacement data.- The subroutines given below illustrate two forms of displacement data that are read by ELPLOT subroutine DATA2. The subroutine TPLOT has been used in a finite element thermal analysis program. The temperature is to be plotted as the w displacement. Note that in this case NWDISP must be set equal to one in $OPTION. The subroutine DPLOT has been used in a finite element structural analysis program. The displacements are U, V, and W. Note that in this case NUDISP, NVDISP, and NWDISP must be set equal to one in $OPTION.

Subroutine TPLOT.-

```
C
C
      SUBROUTINE TPLOT(TIME,NUMNP,T)
C
C      GENERATES PLOT FILE FOR TEMPERATURE PLOTS • INPUT TO SUBROUTINE
C      DATA2 IN PLOT PROGRAM
C
      DIMENSION T(1)
      WRITE(20) TIME
      DO 15 I=1,NUMNP
      WRITE(20) T(I),I
   15 CONTINUE
C
      RETURN
      END
```

Subroutine WRITE.-

```
      SUBROUTINE WRITE (DISP,ID,NEQ,NUMNP)
C
C***********************************************************************
C*                                                                    *
C*    P R O G R A M                                                   *
C*         TO PRINT DISPLACEMENTS                                     *
C***********************************************************************
C
      COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT
      DIMENSION DISP(NEQ),ID(3,NUMNP)
      DIMENSION D(3)
C
C     CREATE DISPLACEMENT PLOT FILE-TAPE 20
C
      DUM=0.0
      WRITE(20) DUM
C
C     PRINT DISPLACEMENTS
C
      WRITE (IOUT,2000)
      IC=4
C
      DO 100 II=1,NUMNP
      IC=IC + 1
      IF (IC.LT.56) GO TO 105
      WRITE (IOUT,2000)
      IC=4
  105 DO 110 I=1,3
  110 D(I)=0.
C
      DO 120 I=1,3
      KK=ID(I,II)
      IL=I
  120 IF (KK.NE.0) D(IL)=DISP(KK)
C
      WRITE(20) II,D
  100 WRITE (IOUT,2010) II,D
C
C
      RETURN
C
 2000 FORMAT (///, 26H D I S P L A C E M E N T S  // 7H  NODE ,9X,
     114HX-DISPLACEMENT,4X,14HY-DISPLACEMENT,4X,14HZ-DISPLACEMENT)
 2010 FORMAT (1X,I3,8X,3E18.6)
C
      END
```

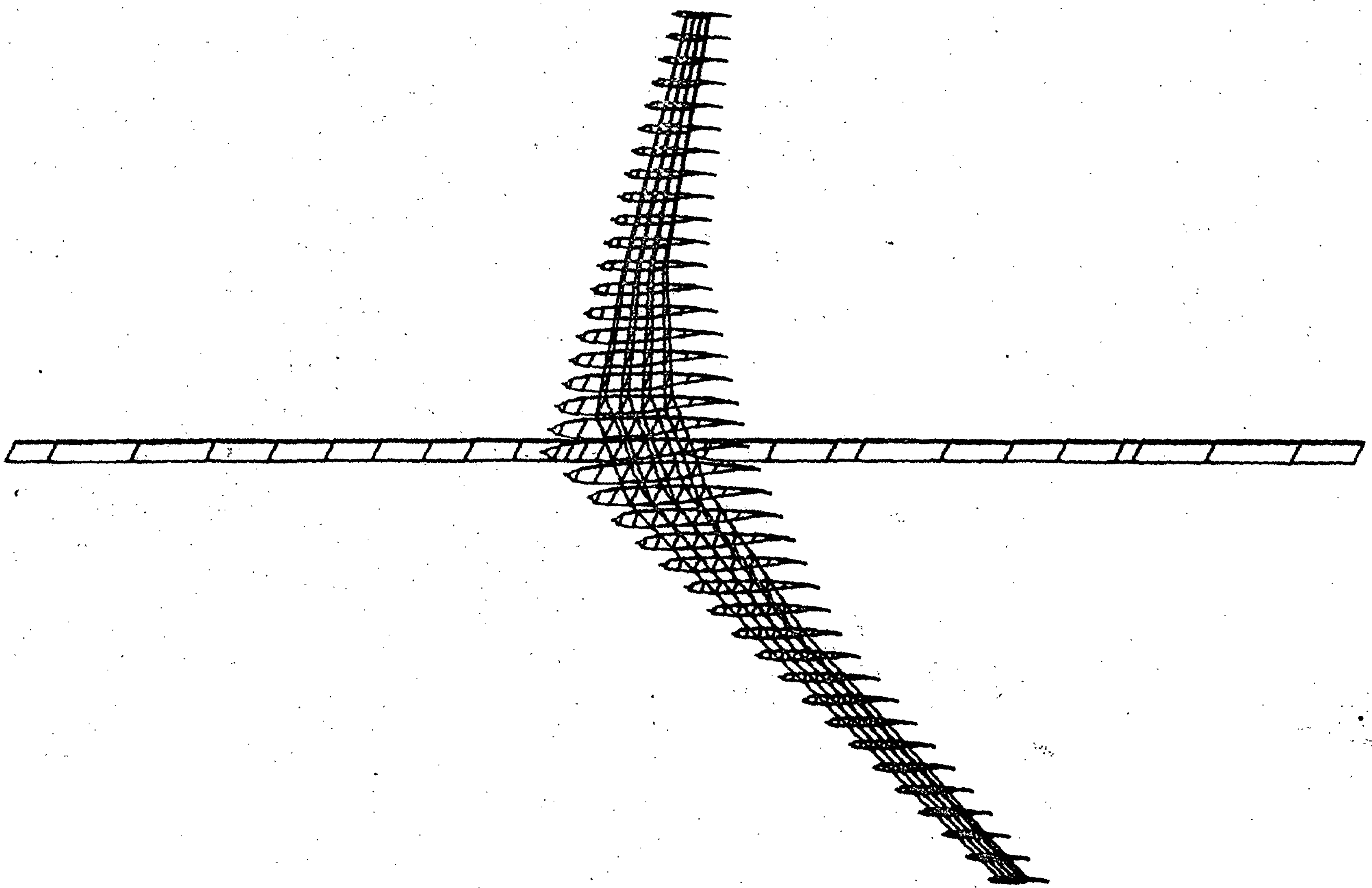


Figure 1. Oblique orthographic view of a finite element model of an aircraft wing structure.

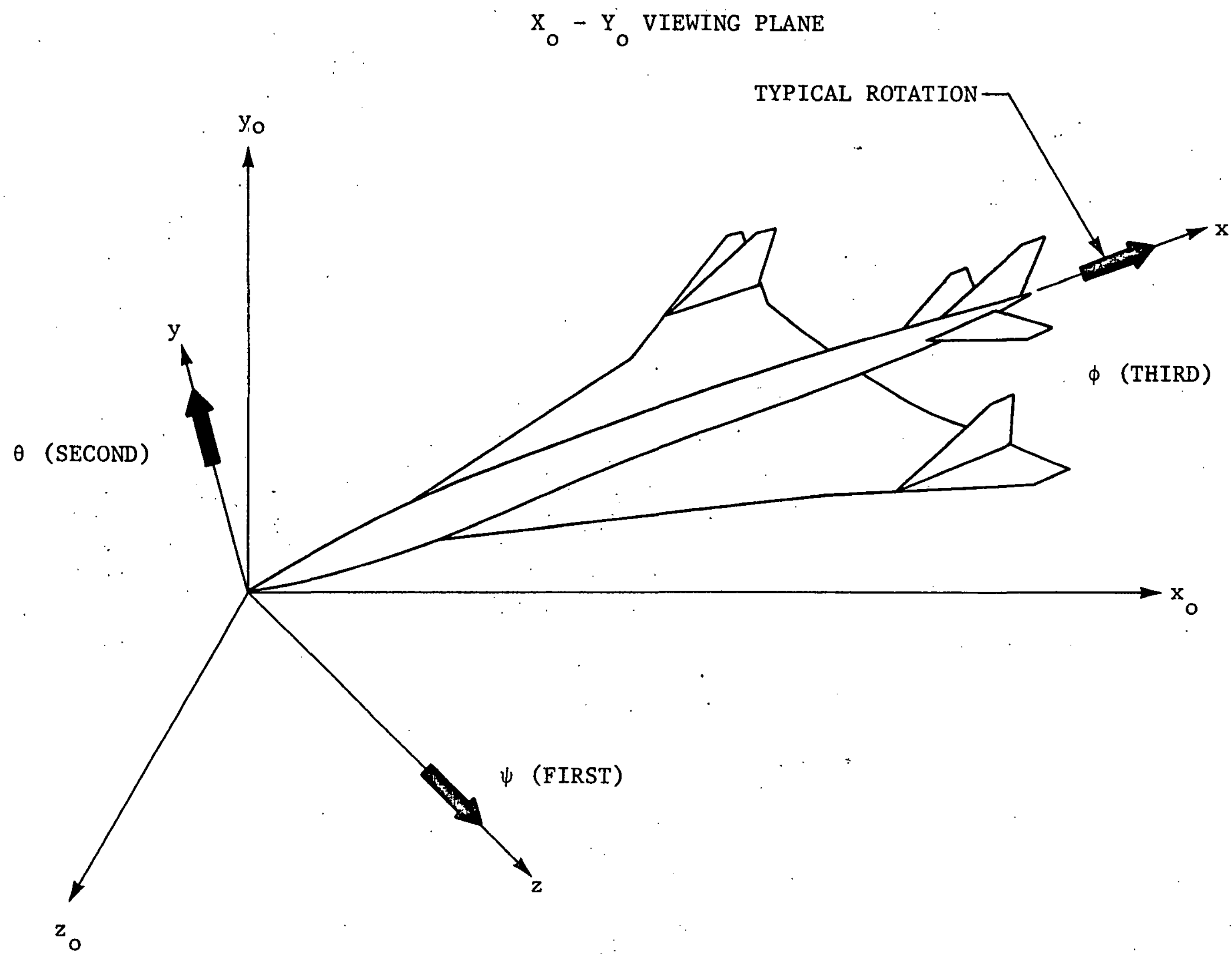
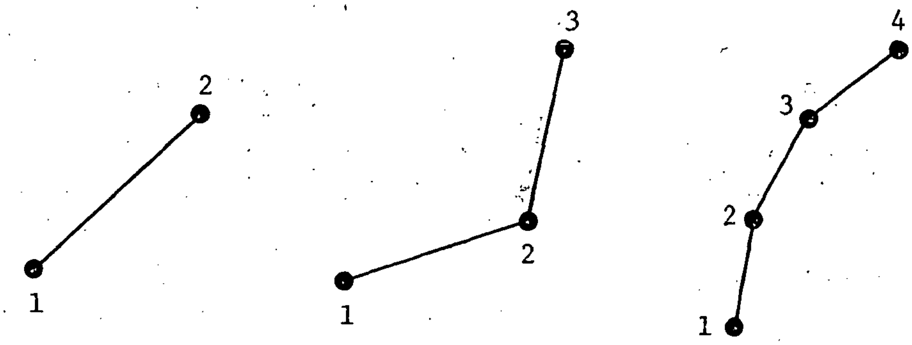
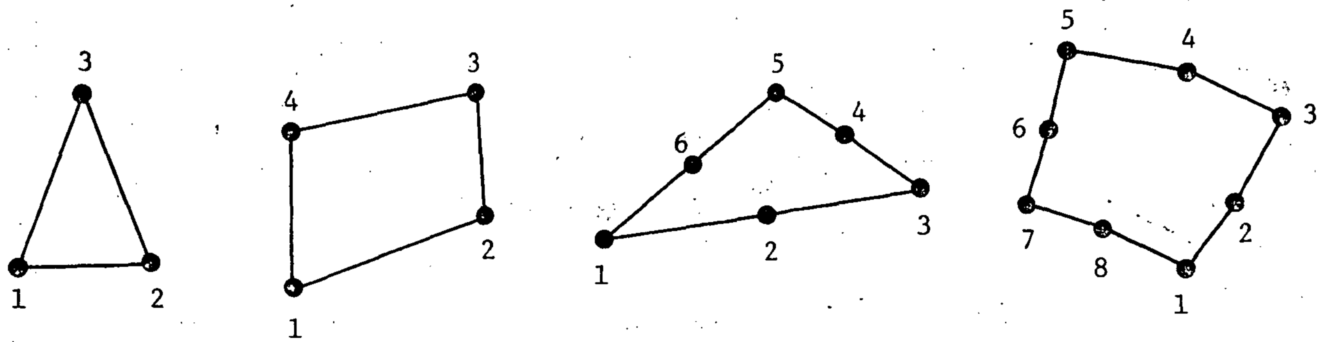


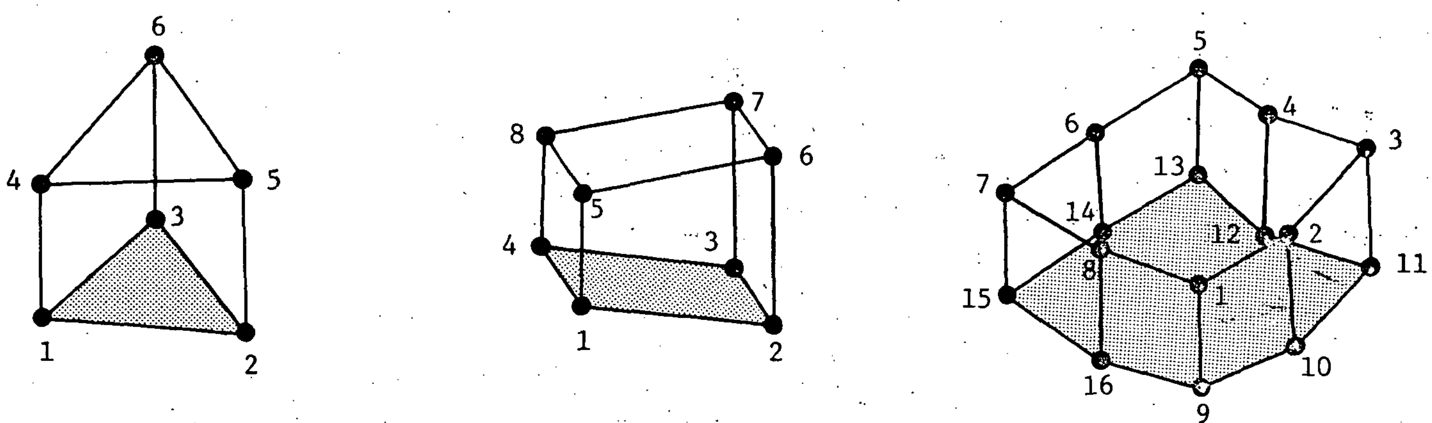
Figure 2. Coordinate systems and Euler angles for oblique orthographic projection.



(a) Family of one-dimensional elements.

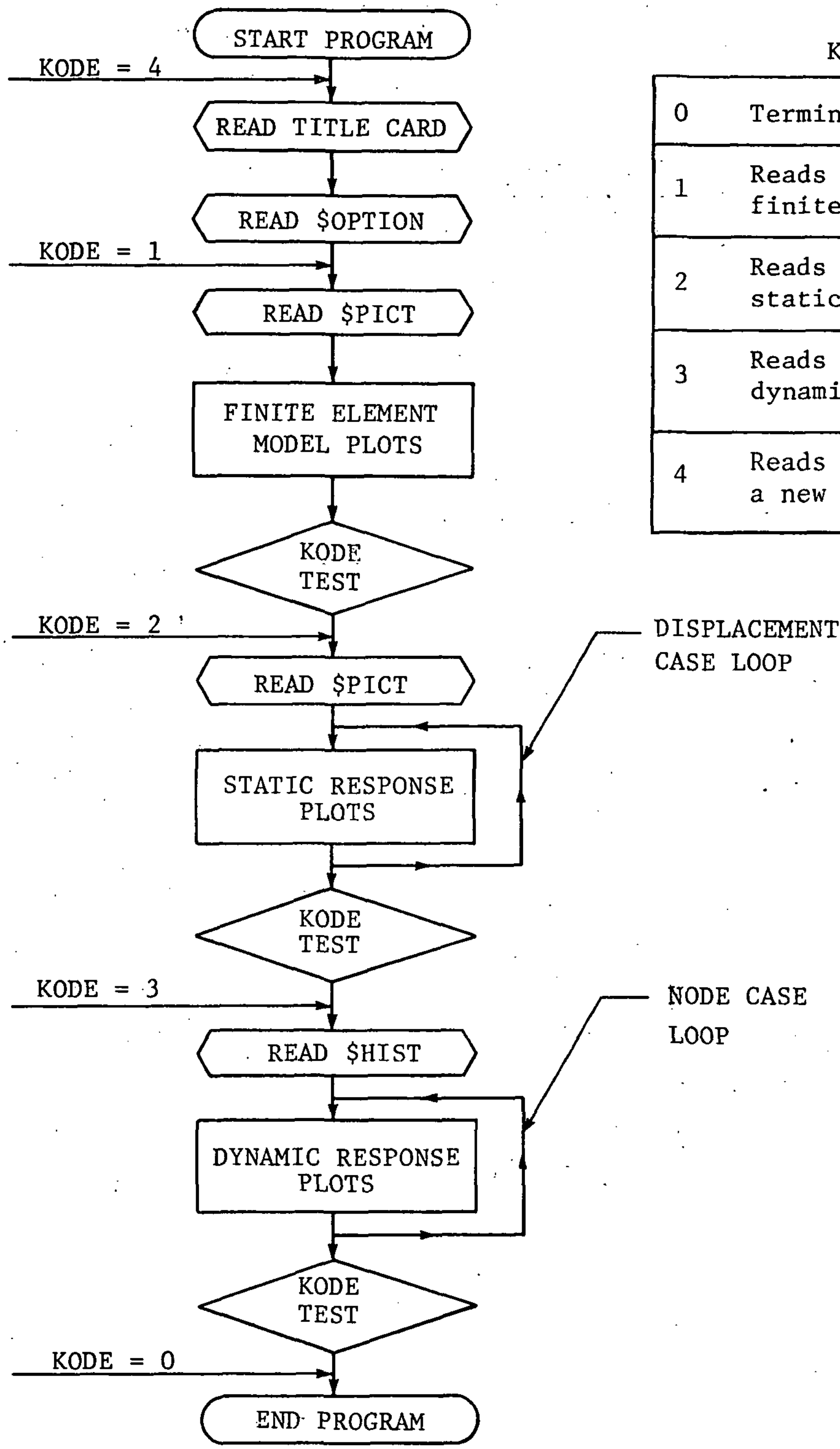


(b) Family of two-dimensional elements.



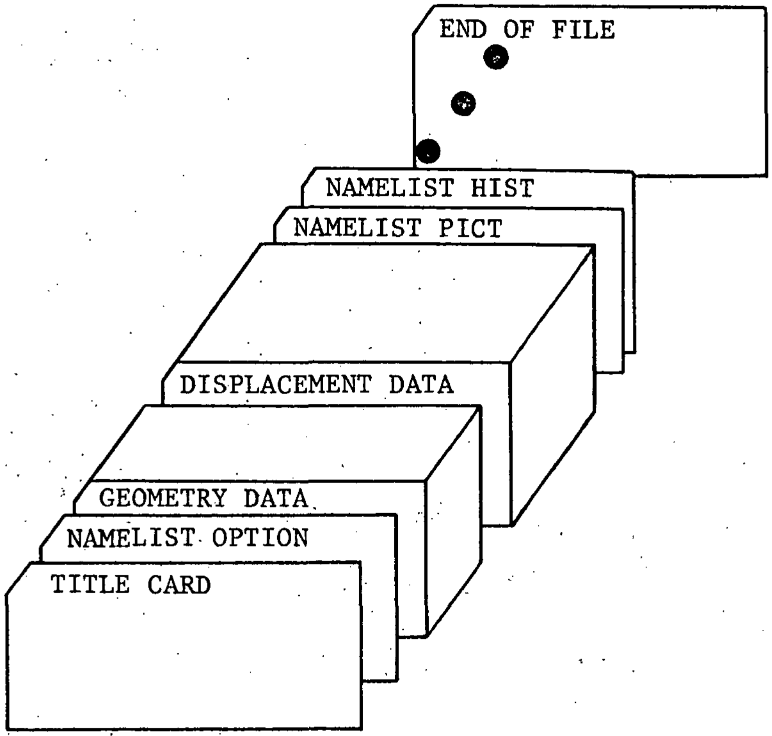
(c) Family of three-dimensional elements.

Figure 3. Finite element types.



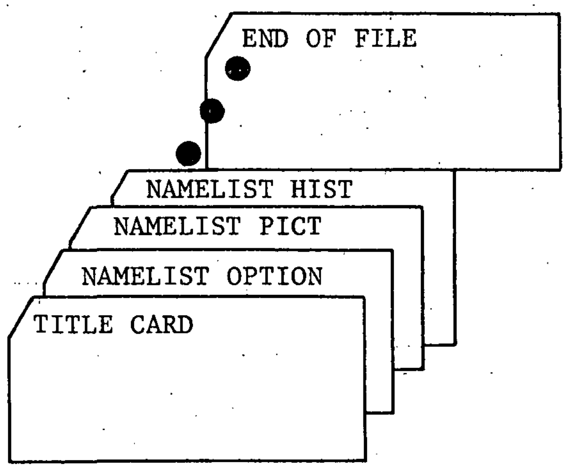
KODE	ACTION
0	Terminates problem
1	Reads a NAMELIST PICT for finite element model plots
2	Reads a NAMELIST PICT for static response plots
3	Reads a NAMELIST HIST for dynamic response plots
4	Reads a title card for a new problem

Figure 4. ELPLOT flow chart showing execution options specified by the parameter KODE.



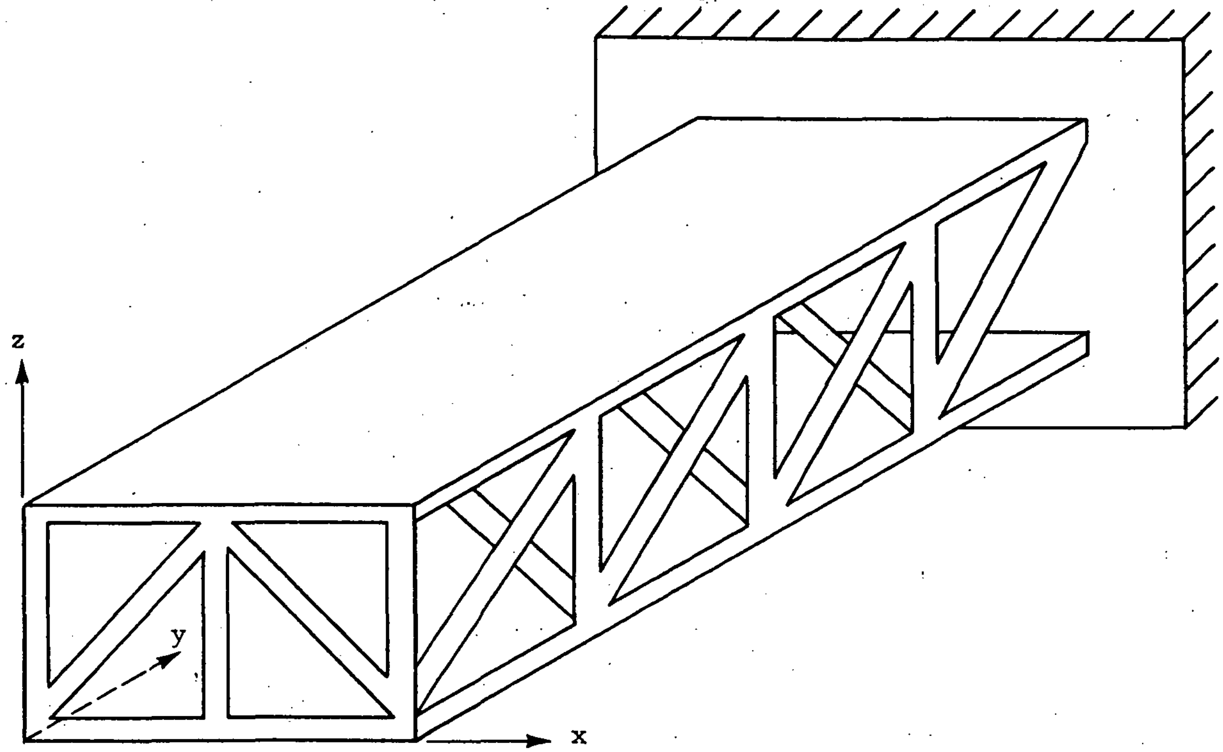
(a) Input data with user-specified formatted card images for KGEOM = 1, and KDATA = 1 in NAMELIST OPTION.

Geometry data to be read from file 8. Displacement data to be read from file 20.

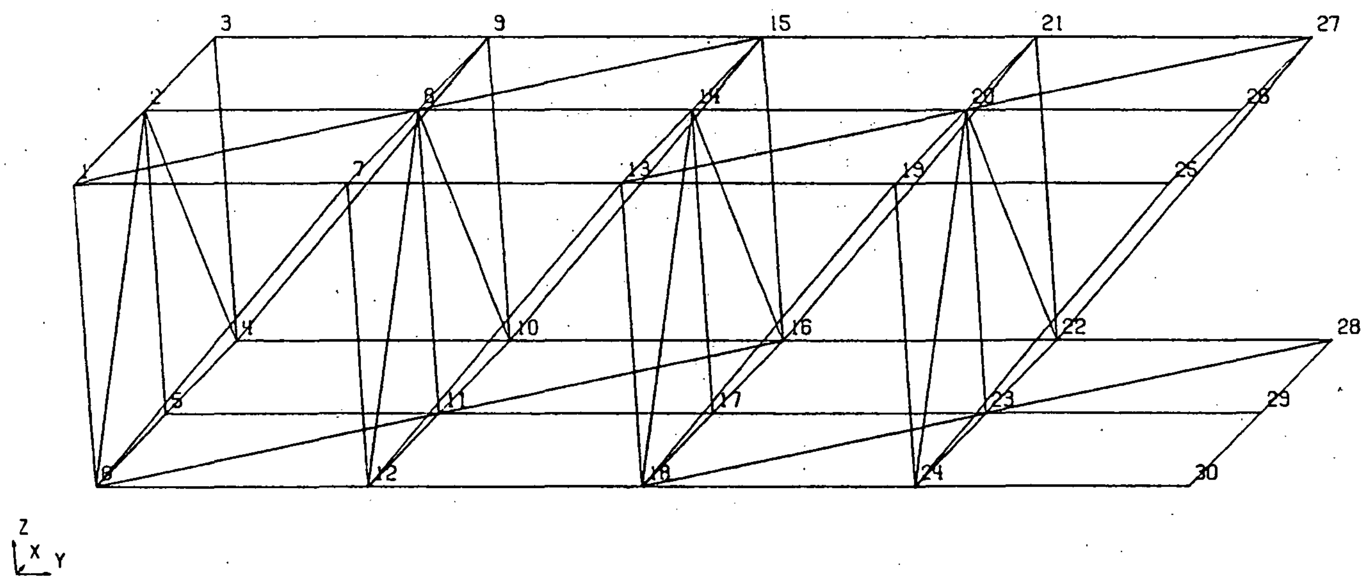


(b) Input data with unformatted card images for KGEOM = 2 and KDATA = 2 in NAMELIST OPTION.

Figure 5. Proper sequences of ELPLOT input data.

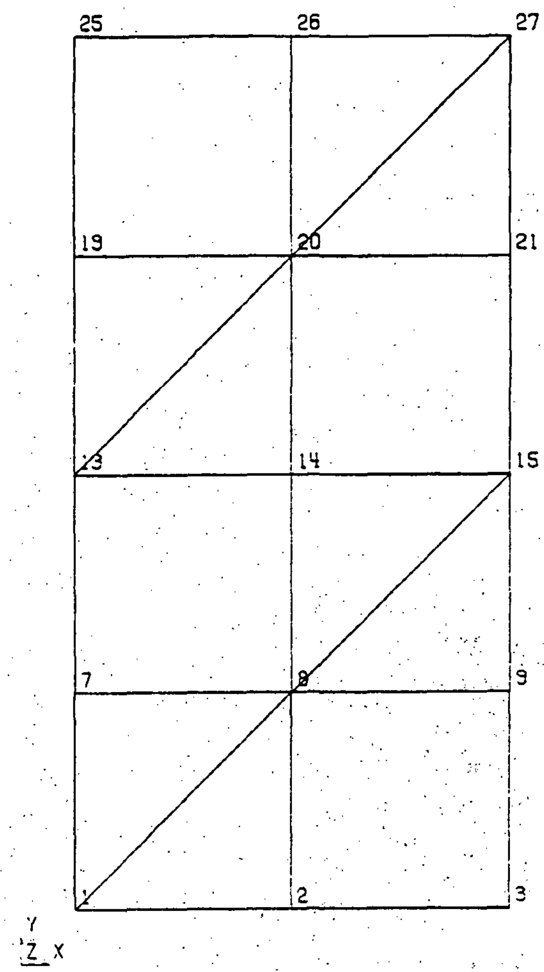


(a) Schematic of wing box.

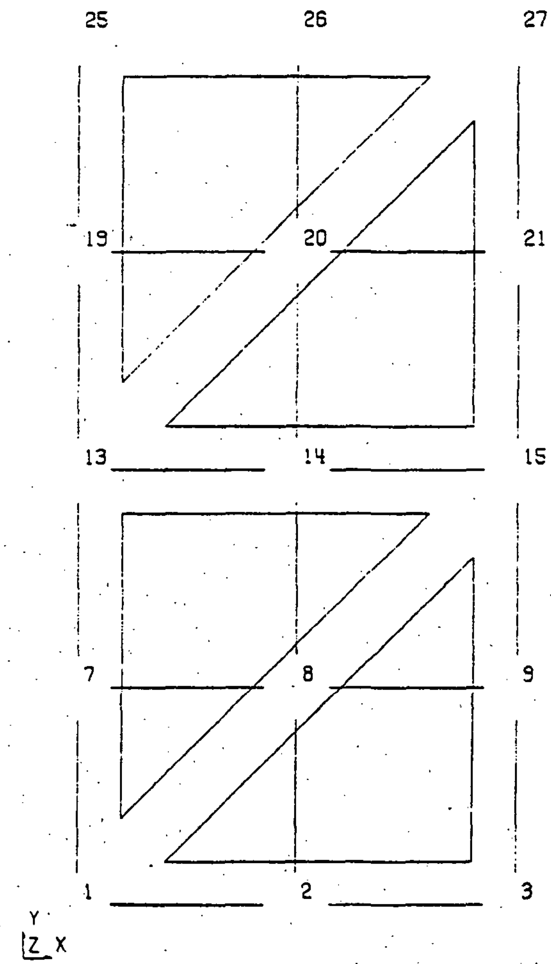


(b) Oblique view of wing box finite element model annotated with node numbers.

Figure 6. Wing box and plots of finite element model.

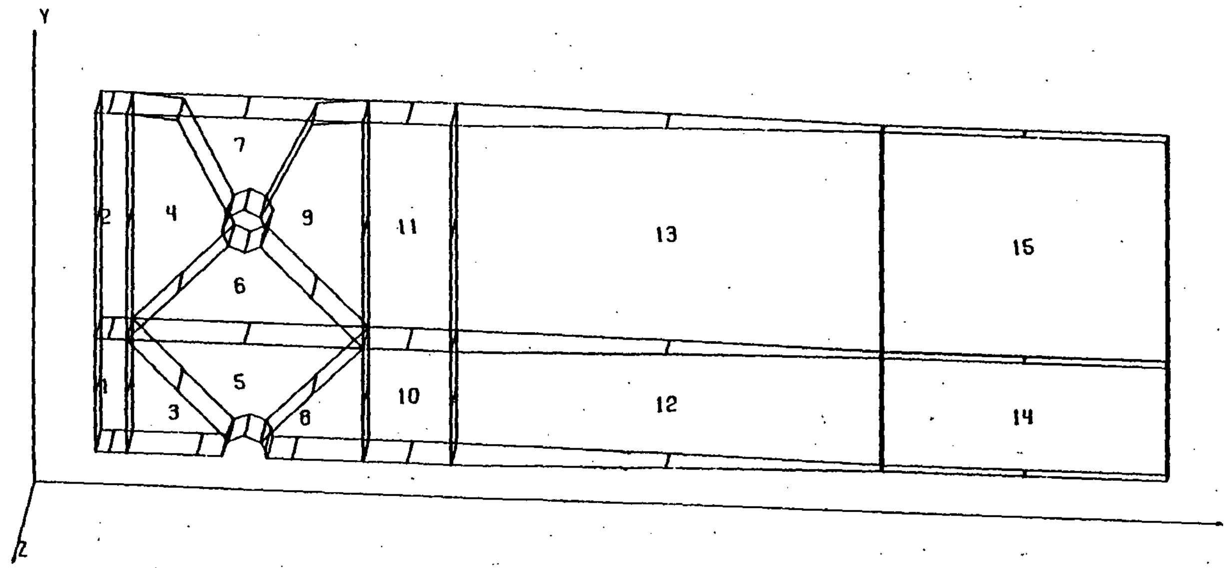


(c) Top face of wing box model isolated by cutting planes parallel to the x - y plane.



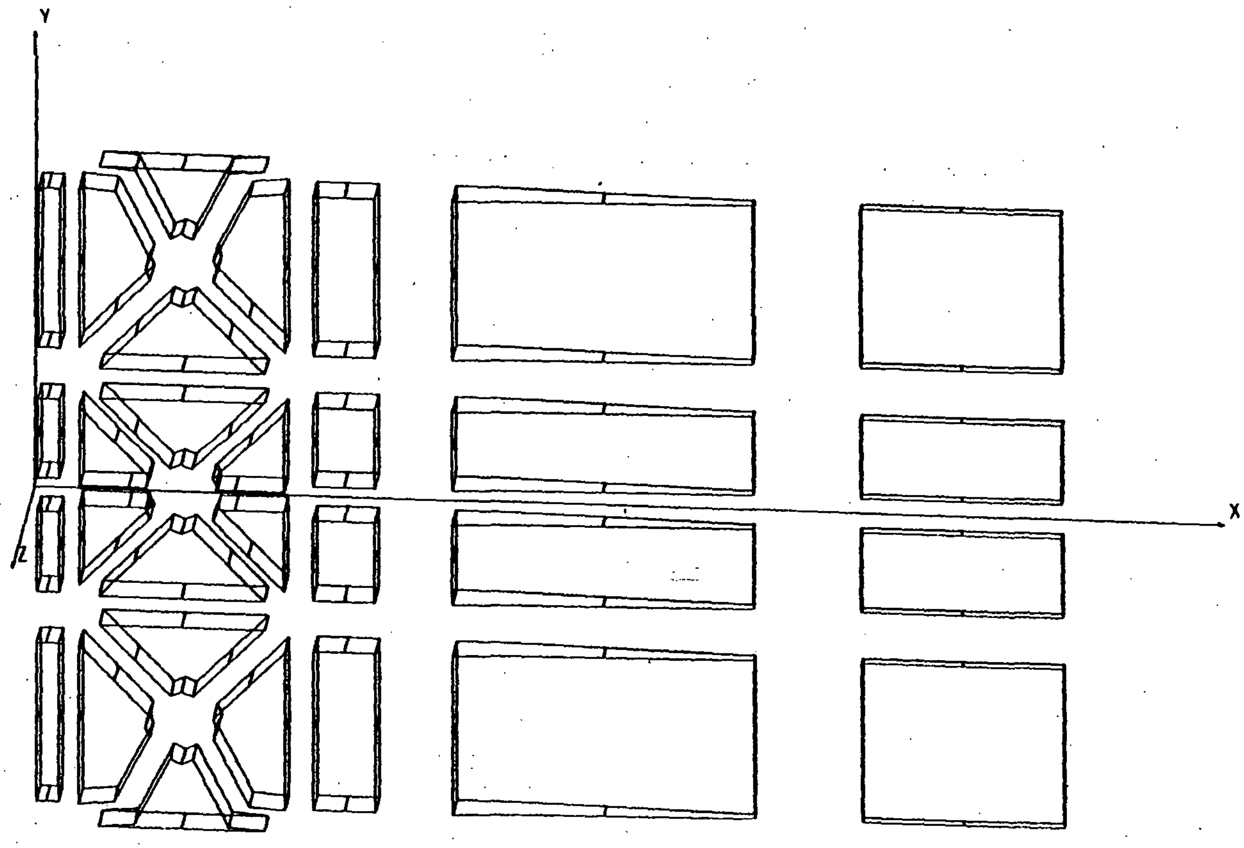
(d) Exploded view of isolated top face displaying rod and triangular elements.

Figure 6. (Concluded)



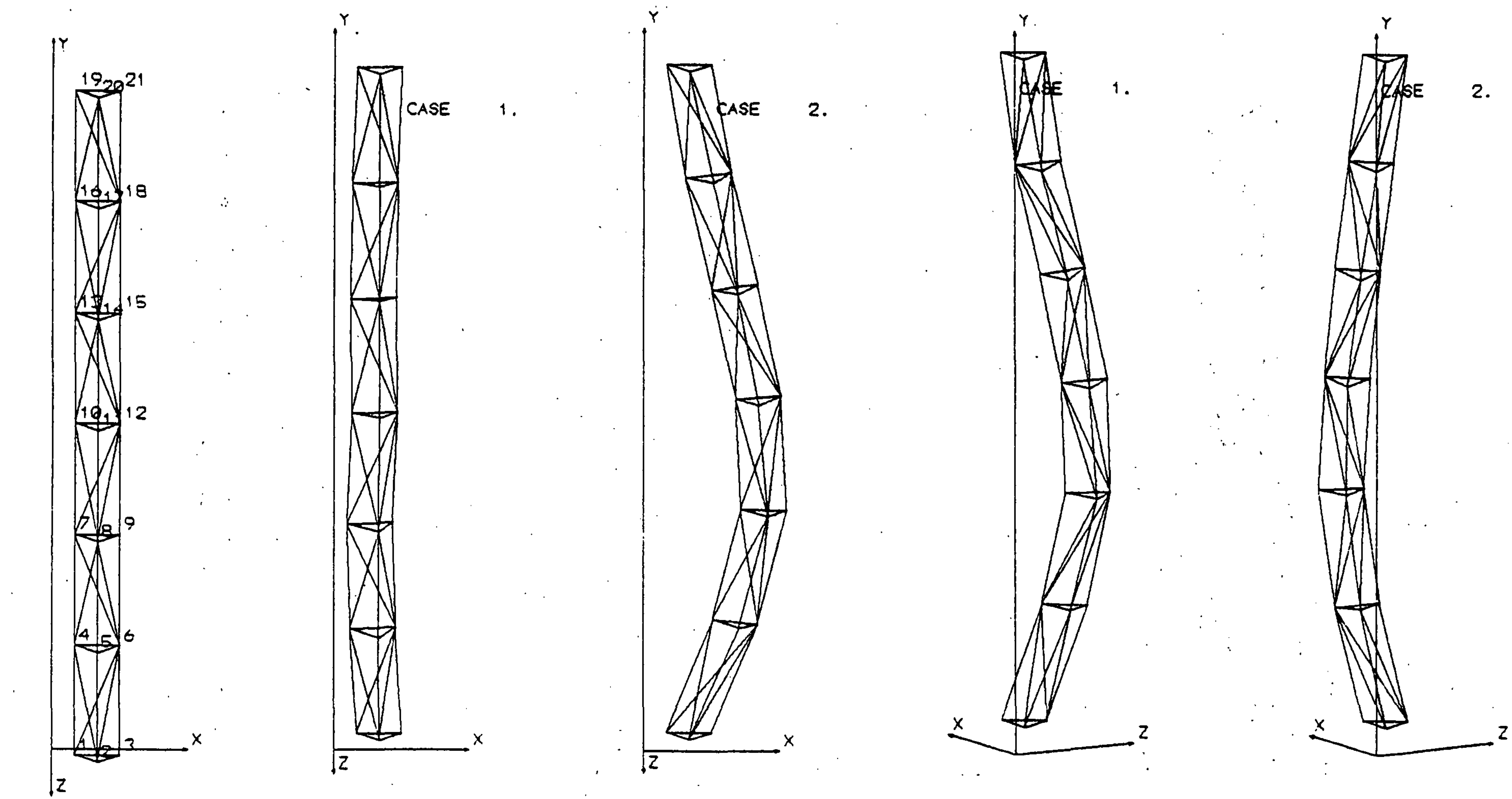
(a) Finite element model annotated with element numbers.

Figure 7. Plots of finite element model of a bolted-joint specimen.



(b) Exploded symmetrical plot.

Figure 7. (Concluded)



(a) Undeformed

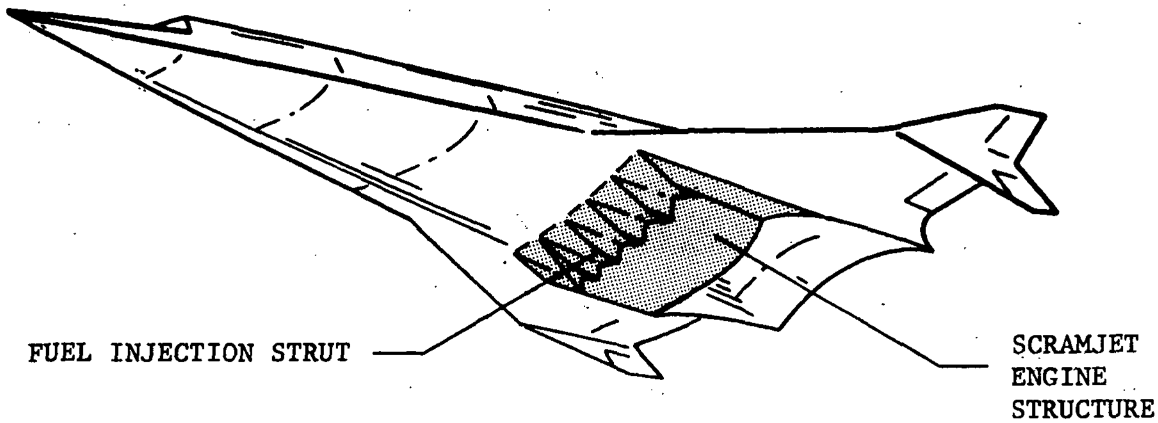
(b) Deformed, Load Case 1.

(c) Deformed, Load Case 2.

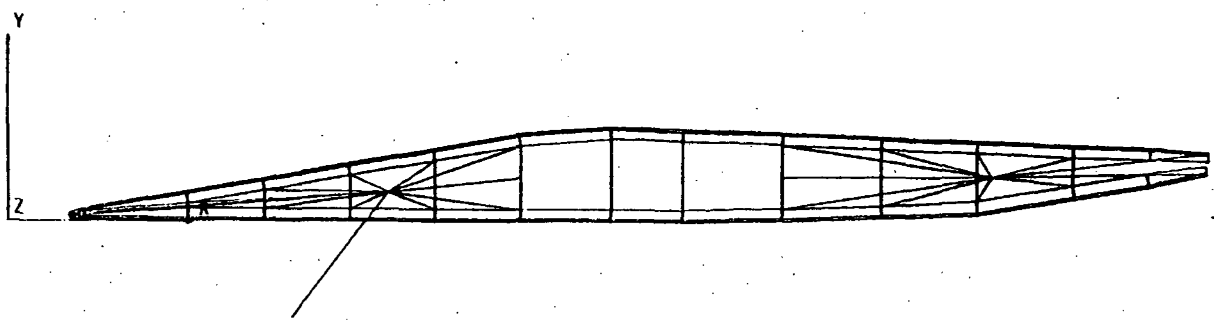
(d) Deformed, Load Case 1.

(e) Deformed, Load Case 2.

Figure 8. Plots of finite element model and static responses for an oceanographic instrument truss.

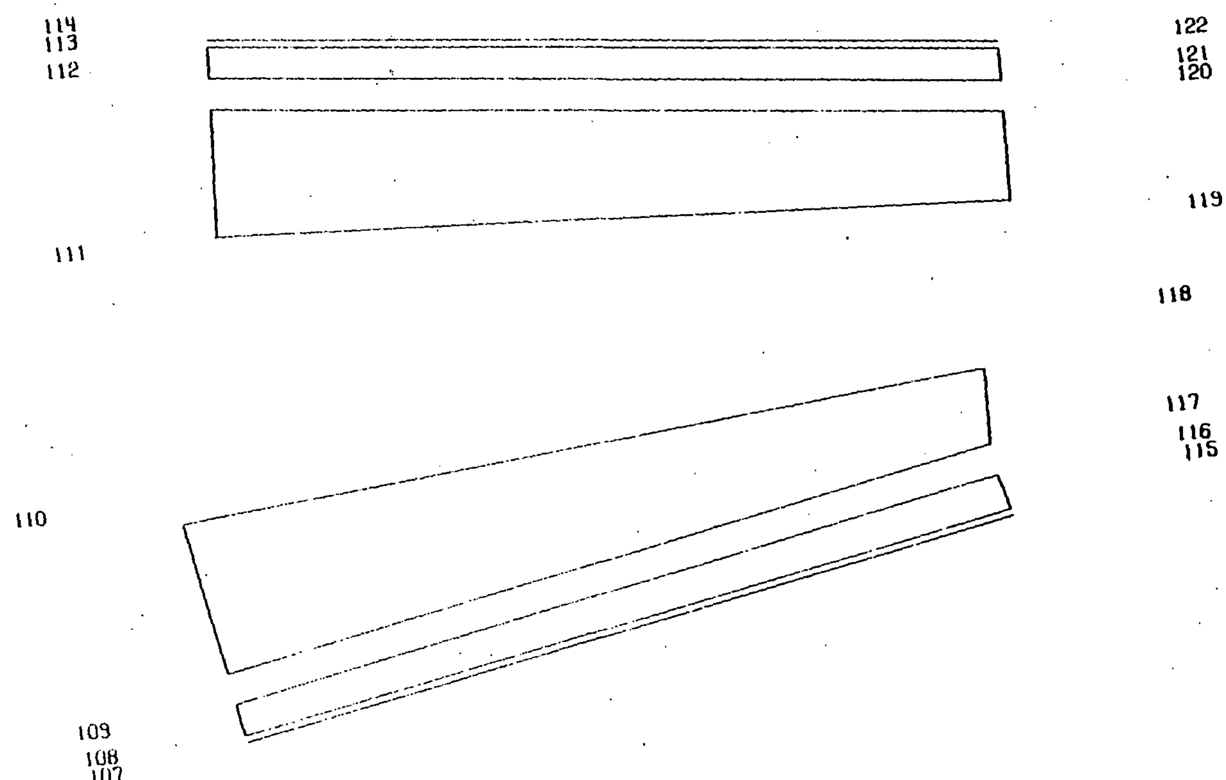


(a) Schematic of hypersonic aircraft scramjet engine.



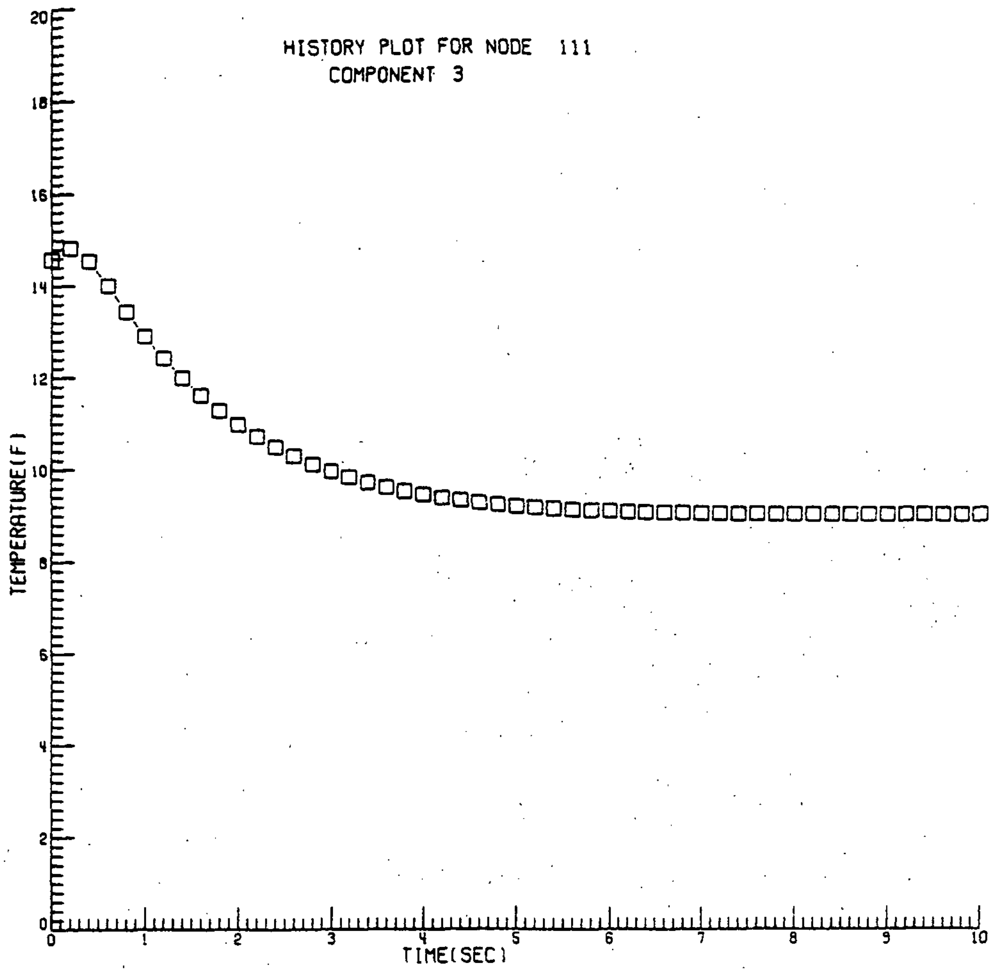
(b) Finite element thermal model of fuel-injection strut cross section.

Figure 9. Schematic of hypersonic aircraft scramjet engine and plots from finite element thermal analysis.



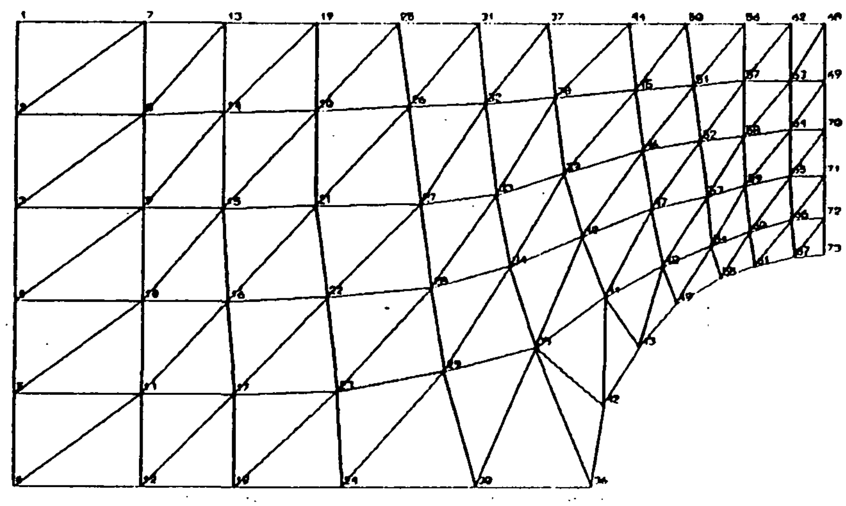
(c) Exploded section of finite element model annotated with node numbers.

Figure 9. (Continued).

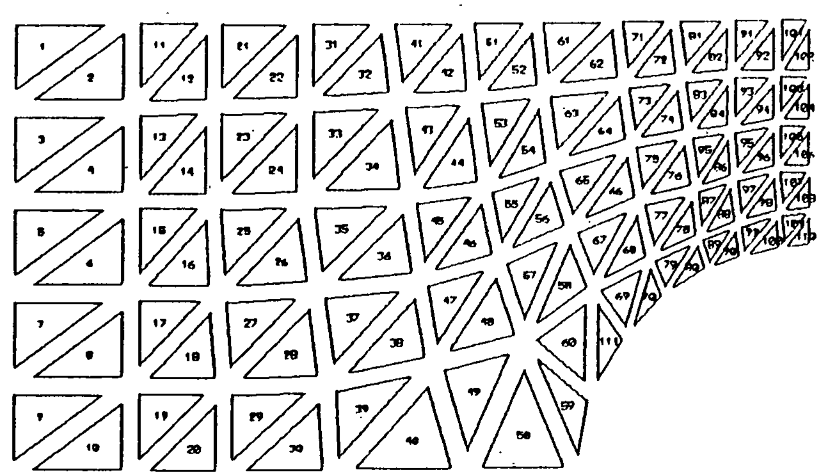


(d) Plot of transient temperature response.

Figure 9. (Concluded).



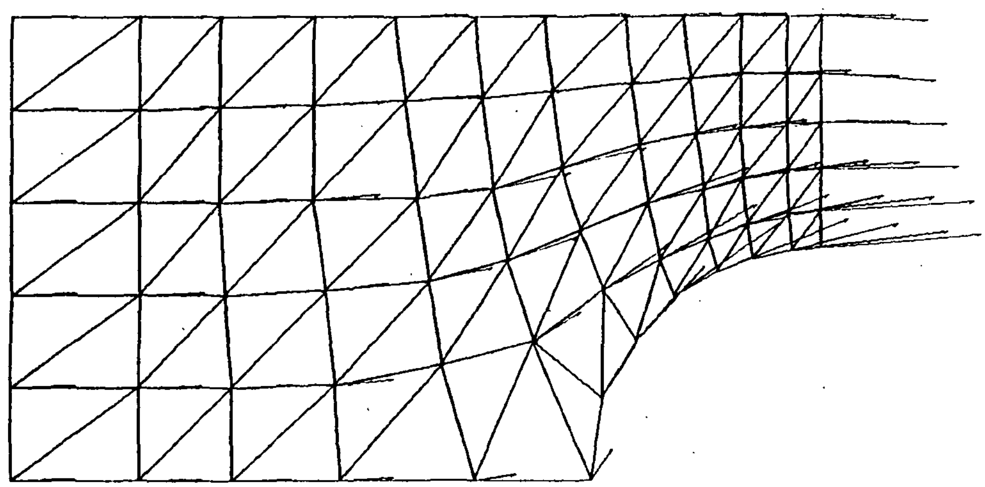
(a) Finite element model annotated with node numbers.



(b) Exploded finite element model annotated with element numbers.

Figure 10. Plots of a finite element analysis of fluid flow about a cylinder.

CASE 1.



(c) Vector plots of fluid velocities.

Figure 10. (Concluded).

REFERENCES

1. Pilkey, W.; Saczalski, K.; and Schaeffer, H., eds.: Structural Mechanics Computer Programs. Univ. of Va. Press (Charlottesville), 1974.

2. Interactive Computer Graphics in Engineering. Papers presented at the Winter Annual Meeting of the American Soc. Mech. Engrs., New York, New York, Dec. 5-10, 1976. ASME, 1977.

3. Thornton, Earl A.; and Wieting, Allan R.: Finite Element Methodology for Thermal Analysis of Convectively Cooled Structures. In: Heat Transfer, Leroy S. Fletcher, ed., Vol. 60. American Institute Aeronaut. and Astronaut. (New York), 1978.

4. Giles, Gary L.: Digital Computer Programs for Generating Oblique Orthographic Projections and Contour Plots. NASA TN D-7797, Jan. 1975.

5. Adelman, Howard M.; and Sawyer, Patricia L.: Inclusion of Explicit Thermal Requirements in Optimum Structural Design. NASA TM X-74017, Mar. 1977.

6. Bathe, Klau-Jürgen; and Wilson, Edward L.: Numerical Methods in Finite Element Analysis. Prentice Hall, Inc., 1976.

7. Chung, T.J.: Finite Element Analysis in Fluid Dynamics. McGraw-Hill, Inc., 1978.